

UNIVERSIDAD DON BOSCO



“SISTEMA DE APOYO PARA LA PROGRAMACION DE HORARIOS DE CLASES DE LA UNIVERSIDAD DON BOSCO”

Trabajo de Graduación preparado para la Facultad de Ingeniería.

Para Optar al Grado de
INGENIERO EN CIENCIAS DE LA COMPUTACIÓN

PRESENTADO POR:
Roberto Alfredo Burgos Lara
Oscar Orlando Magaña Alvarado
Fernando José Romero Rodríguez

MARZO DE 2005
SAN SALVADOR, EL SALVADOR, CENTRO AMERICA

UNIVERSIDAD DON BOSCO



RECTOR
ING. FEDERICO MIGUEL HUGUET RIVERA

SECRETARIO GENERAL
LIC. MARIO RAFAEL OLMOS

DECANO DE LA FACULTAD DE INGENIERÍA
ING. ERNESTO GODOFREDO GIRÓN

UNIVERSIDAD DON BOSCO
FACULTAD DE INGENIERIA



COMITÉ EVALUADOR DEL TRABAJO DE GRADUACIÓN

LIC. SANTIAGO ABARCA FUENTES

Asesor

RUDY ADALBERTO TORRES CORNEJO

Jurado

INGA. YESENIA XIOMARA MARTINEZ M.

Jurado

INGA. GLORIA HERNANDEZ PACHECHO

Jurado

Agradecimientos

Le agradezco a mi Señor Todopoderoso que siempre me dio fuerzas y animo para no rendirme, y a mi familia por todo su apoyo.

“Nadie te podrá derrotar en toda tu vida, y yo estaré contigo así como estuve con Moisés, sin dejarte ni abandonarte jamás”.

Josué 1:5.

Oscar Magaña.

Le agradezco a mi familia y mi novia por estar siempre brindando apoyo y consejo en este proceso de mi vida.

A mis compañeros de tesis por haber hecho posible este proyecto.

Y por ultimo, pero siempre primero, a Dios. Por darme vida y fuerza, sin Él nada de esto seria posible.

Roberto Burgos.

Le agradezco a mis padres por apoyarme y aconsejarme.

A mis compañeros de tesis, un equipo de trabajo eficiente y organizado.

Fernando Romero.

Índice

Introducción.....	I
Objetivo General.....	II
Objetivos Específicos.....	II
Justificación.....	III
Alcances.....	V
Limitaciones.....	VIII
Capítulo I: Marco Conceptual.....	1
1.1 Definición del Tema.....	2
1.2 Antecedentes.....	2
1.3 Situación Actual.....	4
1.3.1 Proceso Principal.....	5
1.3.2 Subprocesos.....	6
Capítulo II: Marco Teórico.....	11
2.1 Diagramas de Flujo de Datos.....	12
2.1.1 Consideraciones a tomar en la construcción de un DFD.....	14
2.2 Bases de Datos.....	15
2.2.1 Objetos de las Bases de datos.....	16
2.2.2 Las ventajas que brinda un sistema de bases de datos.....	17
2.2.3 Sistema Manejador de Bases de Datos (DBMS).....	17
2.3 Introducción a SQL.....	18
2.3.1 Como leer datos de la base de datos.....	19
2.3.2 Como agregar datos.....	19
2.3.3 Modificación de datos.....	20
2.3.4 Eliminación de Registros.....	20
2.4 Modelo Entidad Relación.....	20
2.4.1 Entidades y conjunto de entidades.....	21
2.4.2 Relaciones y conjunto de relaciones.....	21
2.4.3 Tipos de relaciones.....	22
2.4.4 Diagrama Entidad-Relación.....	23
2.5 Diccionario de Datos.....	24
2.6 Programación orientada a objetos.....	24
2.6.1 Conceptos importantes en la POO.....	25
2.7 Tecnología .NET de Microsoft.....	25
2.8 Arquitectura Cliente/Servidor.....	27
2.9 Aplicaciones Basadas en Windows.....	28
2.10 Aplicaciones Basadas en Web.....	29
2.11 Microsoft SQL Server.....	31
Capítulo III: Propuesta de Solución.....	35
3.1 Plan de Solución.....	36
3.2 Recursos a utilizar.....	39
3.2.1 Recursos técnicos.....	39
3.2.2 Recursos Económicos.....	40
Capítulo IV: Análisis y Diseño del Sistema.....	42
4.1 Descripción General del Sistema.....	43
4.1.1 Modulo Administrativo.....	43
4.1.2 Modulo de Asignación de Horarios.....	44
4.1.3 Modulo de Asignación de Salones.....	45

4.1.4 Modulo Web.....	45
4.2 Diagramas de Flujo de Datos.	46
4.2.1 Diagrama de Flujo de Datos Nivel 0 de la Aplicación.	46
4.2.2 Diagrama de Flujo de Datos Nivel 1 de la Aplicación.	47
4.2.3 Diagrama de Flujo de Datos Nivel 2 de la Aplicación (Modulo Administrativo).	49
4.2.4 Diagrama de Flujo de Datos Nivel 2 de la Aplicación (Modulo de Asignación de Horarios).....	50
4.2.5 Diagrama de Flujo de Datos Nivel 2 de la Aplicación (Modulo de Asignación de Salones).	51
4.2.6 Diagrama de Flujo de Datos Nivel 2 de la Aplicación (Modulo Web)...	52
4.2.7 Diagrama de Flujo de Datos Nivel 3 de la Aplicación (Proceso de Asignación de Horarios).....	53
4.2.8 Diagrama de Flujo de Datos Nivel 3 de la Aplicación (Proceso de Asignación de Salones).	54
4.2.9 Diagrama de Flujo de Datos Nivel 3 de la Aplicación (Proceso de Reserva de Salones).	55
4.3 Diagrama Entidad-Relación.....	56
4.3.1 Reglas para el nombramiento de Tablas.....	56
4.3.2 Reglas para el nombramiento de Campos	56
4.3.3 Diagrama Entidad-Relación	57
4.4 Diccionario de Datos.	58
4.5 Diagrama de Clases.	58
4.6 Esquema del funcionamiento básico del sistema.....	61
4.7 Diseño de la interfaz de usuario	63
4.8 Metodología de Programación.	65
Bibliografía.	67
Referencias Electrónicas.....	68
Glosario	69
Anexos	72

Introducción.

Durante años la programación de horarios de clase y la asignación de salones de clase de la Universidad Don Bosco se ha llevado a cabo de forma artesanal, es decir, sin ningún apoyo tecnológico. Esto ha causado muchas dificultades en el momento de realizar estas tareas, dentro de las cuales se pueden mencionar las largas y desgastantes reuniones para hacer encajar los horarios; y el poco conocimiento del estado de las variables que intervienen en el proceso de programación de horarios y asignación de salones.

Por lo explicado anteriormente se tomó la decisión de fabricar una herramienta informática que se encargará de sistematizar, gestionar y procesar la data relacionada a las actividades de la programación de horarios y asignación de salones; esta herramienta será capaz de contribuir al ordenamiento de la información, mejorará la comunicación interna del personal y producirá reportes de forma fácil y rápida. Todo esto mediante el aprovechamiento de las últimas y mejores utilidades que existen en el mercado de la informática.

En este documento se encuentran todos los estudios y análisis que fueron requeridos para lograr fabricar la herramienta que apoyará de forma adecuada al personal que se encarga de los procesos de programación de horarios y asignación de salones.

Objetivo General.

Diseñar y desarrollar un sistema informático que facilite el proceso de programación de horarios de clase y asignación de salones para la Universidad Don Bosco.

Objetivos Específicos.

1. Simplificar las tareas que intervienen en el proceso de creación de horarios y distribución de aulas a través de un sistema informático.
2. Coordinar de una manera eficiente todas las variables que intervienen en el proceso.
3. Permitir la consulta remota de horarios y salones de clase a través de una interfaz Web.
4. Aportar nuevas herramientas al proceso de programación de horarios de clases y asignación de salones.

Justificación.

En el año 1984 la Universidad Don Bosco inició sus labores con el objetivo de proporcionar educación superior a la población con difícil acceso a este rubro, debido a la distancia y cuotas de pago de centros de estudios similares; en esos años las carreras ofrecidas no eran tan variadas y la cantidad de estudiantes inscritos eran significativamente menor a la población estudiantil actual. En este momento la Universidad ha aumentado su calidad como institución educadora de profesionales destacándose como una de las mejores no solo en El Salvador sino también en Centro América, lo que le ha permitido aumentar considerablemente su población estudiantil tanto de estudiantes nacionales como internacionales; y además se ha acrecentado la variedad de carreras profesionales y técnicas que se imparten.

Con la situación descrita anteriormente, podemos inferir que una de las tareas administrativas que demanda tiempo y esfuerzo significativo es la programación de los horarios de clases debido a que se realiza de forma manual, y por cada ciclo que se impartirá deben considerarse las siguientes variables:

1. Las carreras de todas las Facultades.
2. Las materias de cada una de las carreras según el ciclo y sus respectivos laboratorios.
3. Los diferentes grupos por cada materia impartida.
4. La disponibilidad de aulas.
5. La disponibilidad de Profesores de Planta o Externos.

Se estima que esta actividad actualmente se realiza en un periodo de 2 semanas, con varias personas trabajando a tiempo completo, además se producen distintos problemas de colisión entre los diferentes horarios y disponibilidad de clases, debido a la falta de conocimiento de todas las variables que intervienen en la tarea. Otro punto importante es que un error

puede distorsionar todo el esquema, requiriendo una redefinición de toda la calendarización.

Por lo anterior, es de vital importancia desarrollar un software que facilite la programación de horarios y la administración de los mismos durante el ciclo de estudios, lo que redundará en una disminución significativa de tiempo invertido en el desarrollo de esta actividad, mejores controles sobre los cupos, horarios, disponibilidad de aulas y la centralización de la información, entre otros.

Alcances.

1. Creación de un sistema de apoyo al proceso de creación de horarios y asignación de espacio físico, el cual estará separado en los siguientes módulos:

a. Módulo Administrativo.

- i. Manejo de Edificios.
- ii. Manejo de Aulas.
- iii. Manejo de Escuelas.
- iv. Manejo de Carreras.
- v. Manejo de Materias.
- vi. Manejo de Pensums.
- vii. Manejo de Ciclos.
- viii. Manejo de Planes de Estudio.
- ix. Manejo de Usuarios.
- x. Reportes.

b. Módulo de Asignación de Horarios.

- i. Manejo de Docentes.
- ii. Proceso de Asignación de Horarios.
- iii. Reportes.

c. Módulo de Asignación de Salones.

- i. Manejo de Reserva de Salones.
- ii. Proceso de Asignación de Salones.
- iii. Reportes.

d. Módulo Web.

- i. Manejo de Tiempo Disponible de Docentes.
- ii. Consultas Web de Horarios.
- iii. Reporteador.

2. El proyecto será capaz de manejar los horarios de todas las Facultades y Escuelas de la Universidad Don Bosco.
3. El propósito del componente de Planes de Estudio, es ingresar y mantener los diferentes diagramas de precedencia¹.
4. La parte de manejo de docentes pretende formar una base de datos de los docentes que imparten cátedras en la universidad, así como también las horas que los mismos tienen libres para impartir sus materias.
5. El componente de manejo de cátedras tiene como finalidad la asignación de materias a docentes.
6. El componente de manejo de Edificios y Aulas mantendrá la información necesaria del espacio físico existente en el campus de la Universidad.
7. El componente manejo de usuarios restringirá el acceso, de acuerdo a políticas predefinidas en los grupos o perfiles.
8. El Proceso de Asignación de horarios es la parte más importante de todo el sistema, ya que es donde se crearán específicamente los horarios. Aquí se determinará el docente, el lugar, el día y la hora en que se estarán impartiendo las asignaturas.
9. El sistema será capaz de reasignar profesores en caso de que algún docente se retire por motivos de fuerza mayor.
10. El componente de reportes, comprende básicamente cuatro tipos de reportes los cuales son:
 - a. Reporte General por Plan: este reporte muestra todas las materias, lugares, tiempos y docentes responsables de la misma.
 - b. Reporte por Docente: este reporte mostrará el listado de las materias, lugares y tiempos relacionados a un docente.
 - c. Reporte por Materia: permite visualizar los docentes, lugares y tiempos relacionados a la materia.
 - d. Reporte General de Laboratorio: este despliega la información para materias que cuentan con prácticas de laboratorio.

¹ Estos diagramas detallan las materias que se imparten en cada ciclo.

e. Reporte de Salones de Clase

- i. Espacio Utilizado: imprimirá todos los salones que están siendo utilizados detallando día, hora, la asignatura y docente.
- ii. Espacio Disponible: imprimirá todas las aulas que no están siendo utilizadas detallando el día y hora en la que se encuentra libre.

- 11. El componente de disponibilidad de tiempo de docentes se utilizará para que los profesores ingresen las horas que tienen disponibles para impartir cátedras.
- 12. El componente de consultas Web permite a cualquier persona interesada en horarios de la Universidad, realizar consultas respecto a cátedras, docentes, lugares y tiempos.
- 13. El sistema notificará vía correo electrónico y notificaciones en línea su horario de clases a cada docente.

Limitaciones.

1. El software está diseñado para el uso del personal encargado de la programación de horarios y asignación de salones de la Universidad Don Bosco.
2. El sistema en ningún momento intenta imitar un comportamiento inteligente, por tanto, el sistema necesita ser manejado por un operador humano.
3. Para poder acceder al Módulo Web se necesitará que el cliente posea una conexión a Internet y un navegador Web capaz de ejecutar JavaScript.
4. Para poder utilizar la herramienta es necesario que los usuarios tengan conocimientos del proceso de programación de horarios y asignación de salones.

Capitulo I: Marco Conceptual

1.1 Definición del Tema

Para una institución educativa como es el caso de la Universidad Don Bosco, el proceso de creación de horarios al inicio de cada ciclo es de suma importancia, ya que es la forma en que pone sus recursos a disposición para los estudiantes.

Esta es una tarea compleja en la cual intervienen muchas variables. Dentro de ellas podemos mencionar:

1. Las asignaturas que van a ser impartidas.
2. El cupo de alumnos.
3. Cuantos grupos se abrirán para servir una materia.
4. Cuales de estas materias llevan laboratorio.
5. La disponibilidad de tiempo de los docentes.
6. Los espacios físicos disponibles que designaran.

Actualmente, el proceso está bajo la responsabilidad de una persona encargada por Escuela, la cual lo realiza de forma manual utilizando una plantilla de horarios. Dicha plantilla contiene los días y las horas que se disponen para impartir las cátedras, teniendo en cuenta todas las variables anteriormente mencionadas. Luego de haberse aprobado los tiempos de los horarios y designar los catedráticos que impartirán las asignaturas se procede a la organización de salones de clase.

Debido a que este proceso es realizado manualmente, es difícil detectar los choques entre materias del mismo ciclo, tiempos disponibles de los docentes para impartir cátedras, choque de horarios de asignaturas impartidas por un mismo catedrático y disponibilidad de las aulas de clase.

Con el interés de resolver estos inconvenientes y aportar nuevas opciones, se decide crear el Sistema de Apoyo para la Programación de Horarios de Clases de la Universidad Don Bosco. El cual será una herramienta que servirá para agilizar y centralizar el proceso de creación de horarios de clase.

1.2 Antecedentes

La Universidad Don Bosco se constituye jurídicamente el 27 de marzo de 1984 de acuerdo a las leyes de El Salvador. Inicia sus labores en enero de 1985 en la

Escuela Domingo Savio a través de dos facultades Ciencias y Humanidades e Ingeniería. El Registro Académico se encontraba ubicado en el edificio de dos plantas y las aulas estaban repartidas en el edificio de tres plantas.

La Universidad inicia con aproximadamente 400 alumnos en las facultades de Ingeniería y Humanidades. Ofreciendo la Facultad de Ingeniería las siguientes carreras: Ingeniería Mecánica, Ingeniería Electrónica, Ingeniería Eléctrica; la Facultad de Humanidades: Licenciatura en Ciencias de la Educación; y la Facultad de Teología: Profesorado en Teología.

El siguiente año en el ciclo 01, la Universidad ofrece nuevas carreras en la Facultad de Ingeniería como son: Ingeniería en Computación y Biomédica y la Facultad de Humanidades: Licenciatura en Ciencias de la Comunicación.

Por motivos del crecimiento de número de alumnos, en el ciclo 02 del mismo año, la Universidad afronta el problema de no contar con suficientes aulas para albergar a todos los alumnos, aunque la población universitaria a comparación de otras universidades, era pequeña. La Universidad opta por trasladarse, a la Parroquia Maria Auxiliadora (Don Rúa).

Debido al crecimiento de la población universitaria y con el objeto de satisfacer las necesidades de las mismas, se crea un Centro de Cómputo con el sistema IBM 9370; esta adquisición es una parte donada por la Congregación Salesiana de Nueva York y la otra un desembolso de la Universidad. Este equipo es adquirido en el año de 1988.

Para esta época Registro Académico comenzó a mecanizar el control de alumnos y el control de notas.

Debido al terremoto de 1986, el incremento de la cantidad de alumnos y la falta de espacio físico disponible en la Parroquia de Don Rúa para las facultades existentes, en 1989 la Facultad de Ingeniería junto con Registro Académico se trasladan al Instituto Técnico Ricaldone (ITR); quedándose la Facultad de Humanidades en la Parroquia Don Rúa.

Desde 1986 se tenía la necesidad de un campus para la Universidad y el Colegio Don Bosco, encontrando un nuevo hogar en la actual Ciudadela Don Bosco a la cual se mudaron en el año 1992.

En 1990, se implementa el Sistema Mecanizado de Inscripción de Materias.

En 1992, la Universidad Don Bosco, se traslada nuevamente, pero esta vez en forma definitiva a sus locales propios, ubicados en la Ciudadela Don Bosco, Soyapango.

En los años recientes la Universidad Don Bosco, a la vez que ha consolidado físicamente su presencia en el nuevo campus, también ha expandido sus servicios educativos a través de nuevas carreras, específicamente en las facultades de Ingeniería, Estudios Tecnológicos, Ciencias y Humanidades y a nivel de postgrados.

Desde sus inicios hasta ahora realiza la programación de horarios y asignación de aulas de forma manual.

1.3 Situación Actual

Actualmente la universidad cuenta con un centro de cómputo para los profesores de la Facultad de Ingeniería, equipado con 6 computadoras con Windows XP; además cada facultad cuenta con una cantidad de computadoras suficientes destinadas para el uso de los profesores, sus sistemas operativos varían entre Windows 98, ME, NT, 2000 y XP. Se debe de agregar que la Universidad cuenta con 3 Servidores Microsoft Windows 2003 que sirven para el Dominio, Servidor Web (IIS), y el Manejador de Base de Datos (SQL Server 2000).

El proceso de programación de horarios y asignación de aulas es un proceso que se realiza manualmente. Para comprender mejor este proceso a continuación presentamos los diagramas de proceso actuales:

1.3.1 Proceso Principal

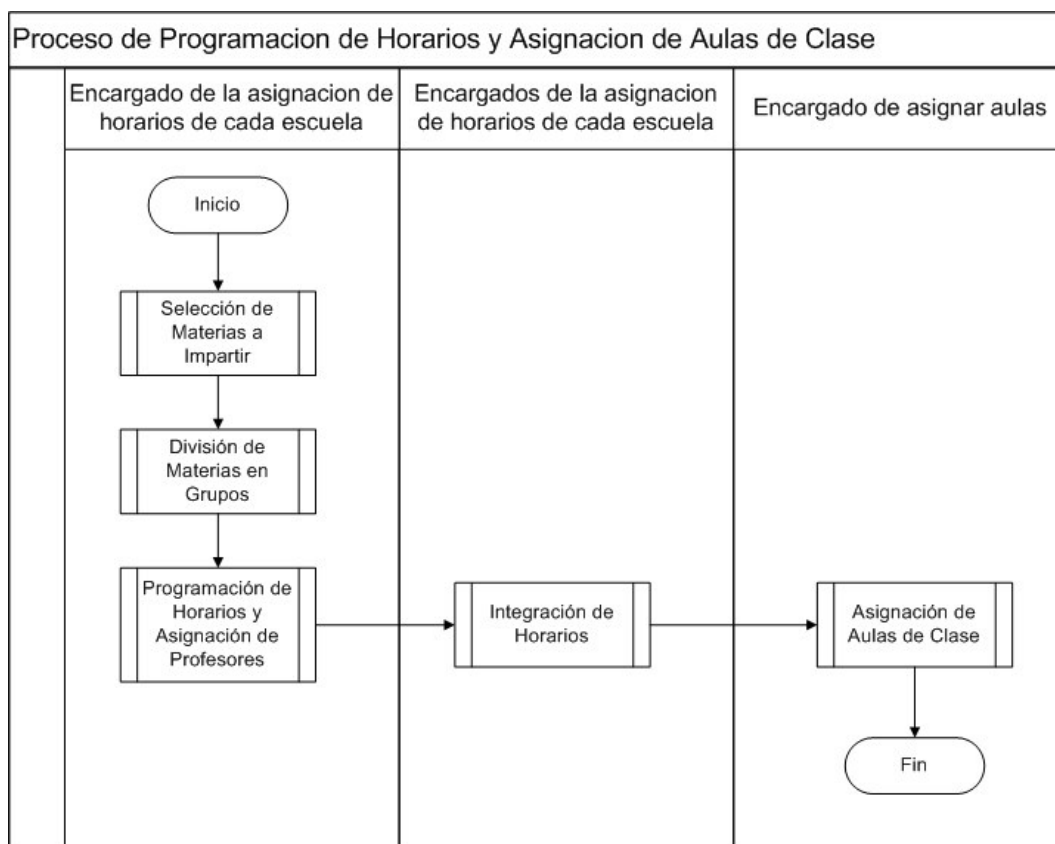


Figura 1.3.1.1: Proceso de programación de Horarios y Asignación de Aulas de Clase.

Este primer flujo presenta el proceso general de programación de horarios y asignación de aulas, siendo cada paso del proceso un subproceso distinto. Es importante destacar etapas principales en el proceso, los cuales son tres:

1. La programación de horarios, delegándose esta responsabilidad al encargado de cada escuela.
2. Reunión de coordinación para la programación de horarios, participando de esta todos los encargados de programación de horarios.
3. Asignación de Aulas, delegándose esta responsabilidad a la Directora de Administración Académica.

1.3.2 Subprocesos

El primer subproceso consiste en la selección de las materias que se impartirán, siendo responsables de esta tarea los encargados² de la programación de horarios de cada escuela.

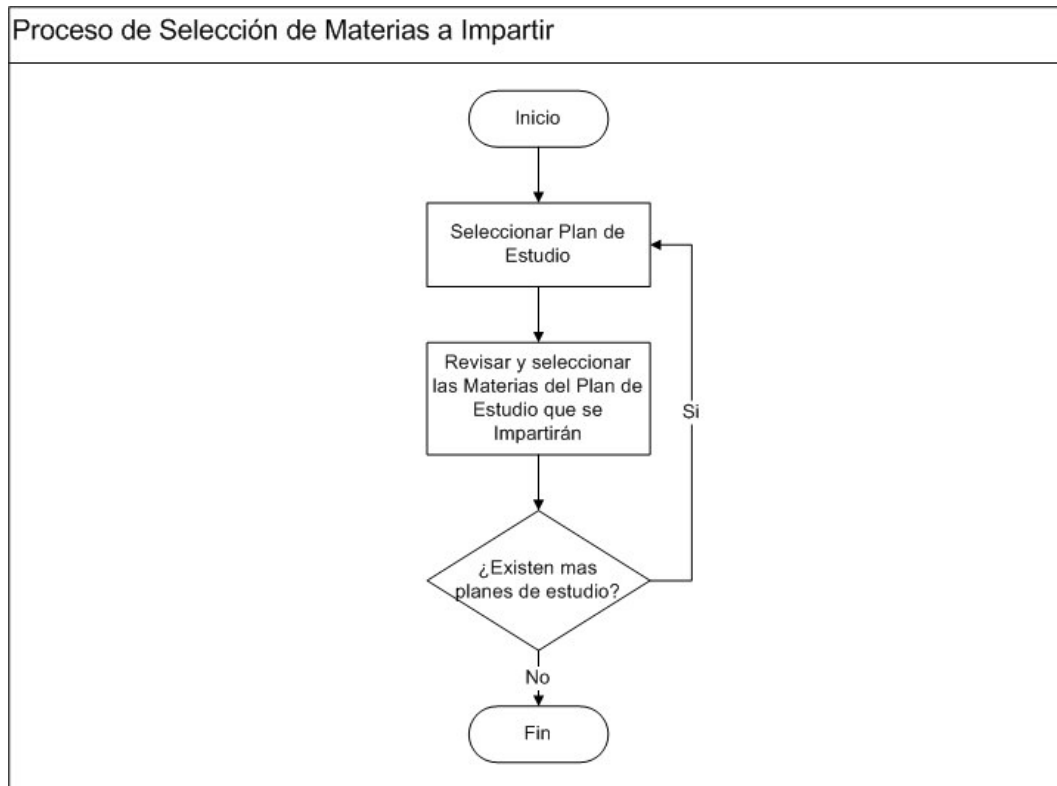


Figura 1.3.2.1: Proceso de selección de materias a impartir.

El segundo subproceso es el punto donde se decide cuantos grupos existirán para cada materia, esto depende del número estimado de alumnos que cursarán las materias en el ciclo.

² Es importante aclarar que los encargados de la programación de horarios intervienen en todos los subprocesos a excepción del último (asignación de salones de clase).

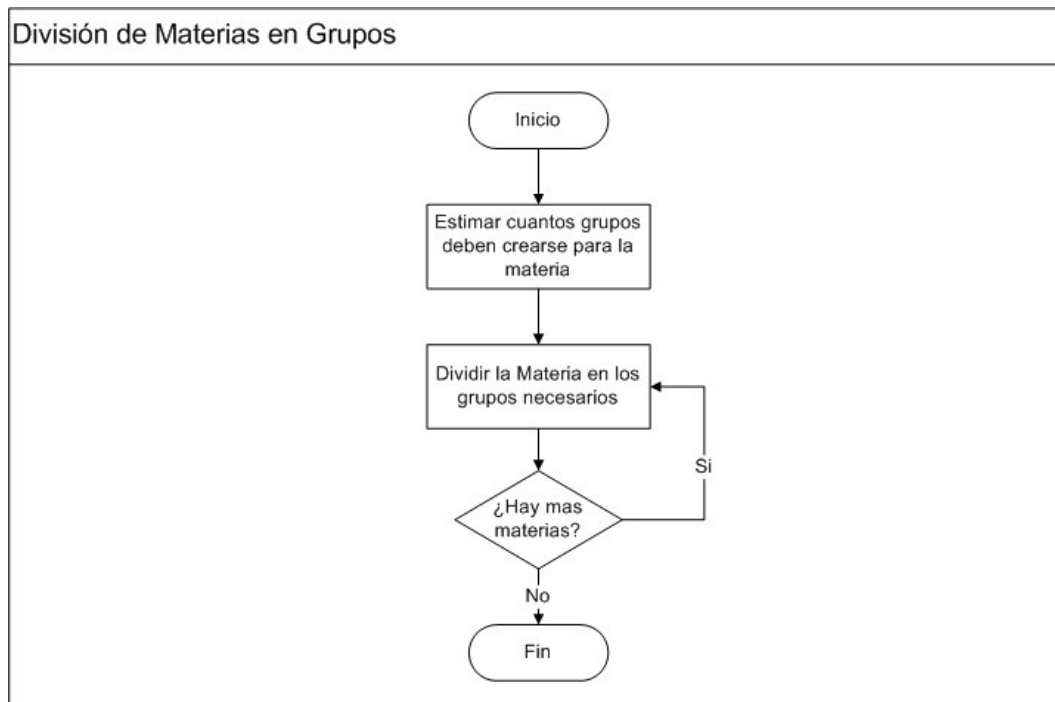
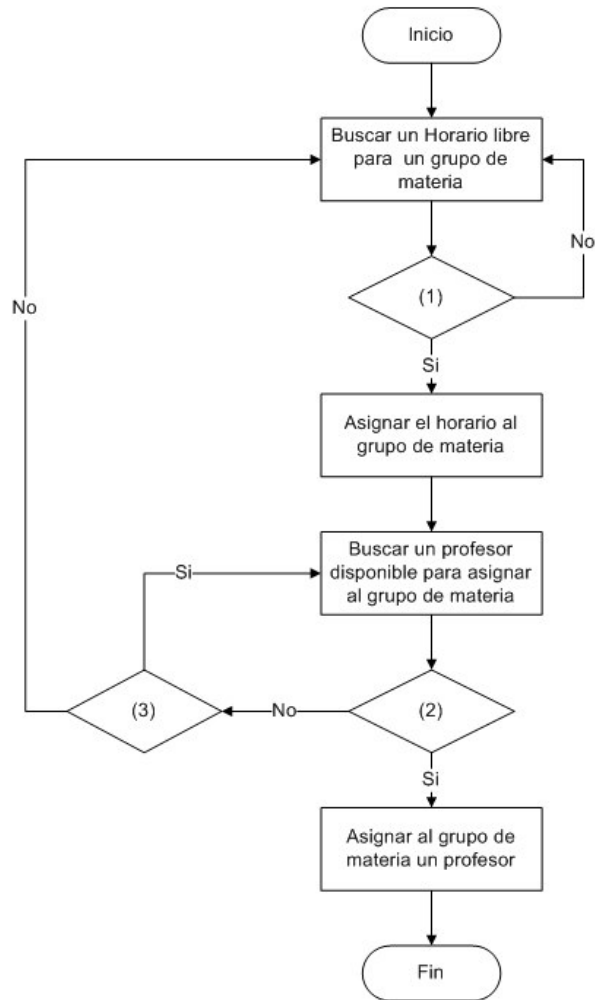


Figura 1.3.2.2: División de Materias en grupos.

El tercer subproceso es la programación de horarios y la asignación de profesores de cada grupo de materia, este paso pareciera que deberían de ser dos distintos pero en realidad son manejados como uno solo, ya que ambos van de la mano y depende de cómo lo maneje cada encargado. Este subproceso puede realizarse de varias maneras, pero a continuación presentamos la forma utilizada por la mayoría de encargados:

Programación de Horarios y Asignación de Profesores



- (1) ¿El Horario al que se asigno a ese grupo de materia esta libre?
(2) ¿El profesor esta libre en el horario que se impartirá el grupo de materia?
(3) ¿Existe otro profesor disponible?

Figura 1.3.2.3: Programación de horarios y asignación de profesores.

El cuarto subproceso es el más importante y difícil de llevar a cabo ya que no depende de un solo encargado, sino de varios reunidos a la vez, con el objetivo de hacer encajar todas las piezas del rompecabezas, de tal manera que los horarios no choquen y los recursos (hasta este punto catedráticos) sean distribuidos adecuadamente, a este le llamamos integración de horarios.

Este subproceso es el que más tiempo toma debido a que los encargados deben negociar que materias deben de moverse de horario, lo cual puede incluir posibles movimientos de profesores, llegando a ser demasiado complejo, no importando esta sección debe repetirse cuantas veces sea necesario.

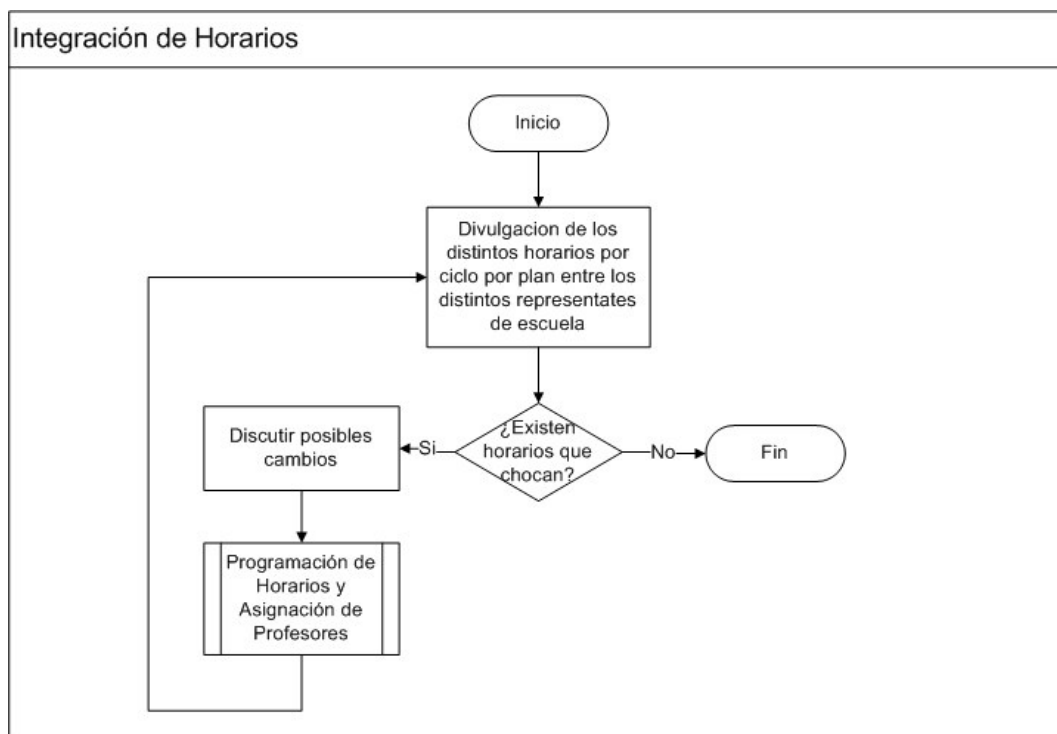


Figura 1.3.2.4: Integración de horarios

El último paso es el de asignar el espacio físico disponible para impartir clases. De este procedimiento se encarga Administración Académica, que debe tomar en cuenta la cantidad de alumnos que debe de tener cada grupo de materias y el cupo de alumnos por aula, además de llevar controlado los salones que ya están asignadas y los que están libres.

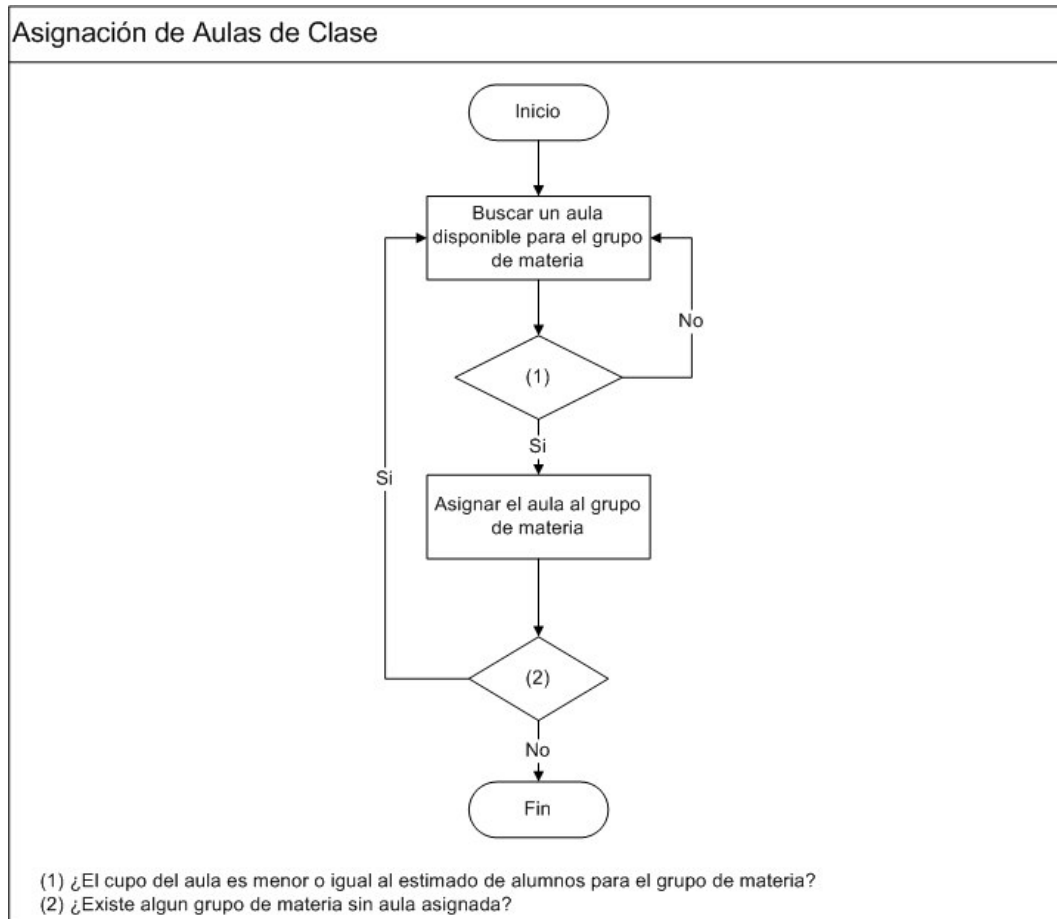


Figura 1.3.2.5: Asignación de aulas de clase.

Capítulo II: Marco Teórico

2.1 Diagramas de Flujo de Datos

Los Diagramas de Flujo de Datos, son herramientas básicas para el análisis de sistemas, ya que permiten representar gráficamente un sistema como una red de procesos y datos conectados entre sí.

Su importancia radica en el hecho de que todos los datos, entidades que intervienen, procesos de manipulación de datos, salidas y almacenes de información, quedan representados de manera gráfica, lo cual ayuda a una mejor y más rápida comprensión de los requerimientos del sistema.

Para la realización de los diagramas de flujos de datos se utilizan 4 símbolos básicos, los cuales se detallan a continuación:

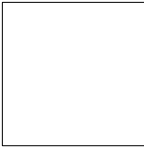
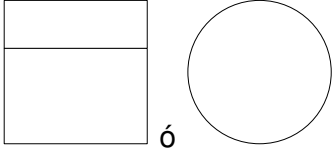


Símbolo	Significado
	Entidad
	Proceso
	Flujo
	Almacén de datos

Tabla 2.1.1: Símbolos utilizados para la creación de Diagramas de Flujo de Datos

Entidad: Se representa gráficamente como un rectángulo, y representan entidades externas con las cuales el sistema se comunica, puede representar a una persona o a un grupo, por ejemplo, una organización, una agencia, un

departamento o un sistema computacional con el cual el sistema que se está analizando tendrá comunicación.

Existen 3 conceptos importantes que se deben tomar en cuenta al trabajar con las entidades:

- Son externos al sistema que se está modelando.
- No se puede cambiar los contenidos de una entidad ni la manera en la que esta trabaja.
- Las relaciones que existan entre las entidades no se muestran en un DFD.

Proceso: Se le conoce también con los nombres de burbuja, función o transformación. Representa la parte del sistema que transforma las entradas en salidas, por lo tanto indican que se está realizando un trabajo dentro del sistema.

Gráficamente el proceso se representa por un círculo o un rectángulo con las esquinas redondeadas. Las diferencias entre las representaciones son puramente cosméticas.

Es importante notar que los procesos se escriben con una sola palabra, frase u oración sencilla.

Flujo: Son representados gráficamente por una flecha, y representan en movimiento de bloques o paquetes de información de un punto a otro dentro del sistema. Los datos que se mueven a través de un flujo viajarán ya sea a un proceso, una entidad o un almacén.

Los flujos muestran también la dirección: una flecha en cualquiera de los extremos nos muestra si los datos se están moviendo hacia el interior o hacia el exterior de los procesos. Los flujos con flechas en ambos extremos se denominan diálogos, ya que hay una pregunta y una respuesta en el mismo flujo. En el caso de un diálogo, los datos de cada extremo deben nombrarse.

Almacén: Denotado por dos líneas paralelas, representan a los datos del sistema en un estado de reposo. Para un analista de sistemas un almacén puede denotar muchas cosas, como una base de datos, un archivo, un medio óptico de almacenamiento, fichas de papel archivadas y muchas maneras más de almacenamiento de información, ya sean estas digitales o no.

Los almacenes pueden existir en uno o ambos de los siguientes contextos:

- Una Entidad o proceso conectado por un flujo hacia un almacén.
- Una Entidad o proceso conectado por un flujo desde un almacén.

Es importante mencionar que los almacenes son denotados por un nombre.

2.1.1 Consideraciones a tomar en la construcción de un DFD

Existen ciertas consideraciones que es necesario tomar en cuenta a la hora de diagramar un DFD para que este represente de la mejor manera posible el flujo de los datos en el sistema.

Estas consideraciones son:

- 1) **Los nombres de los procesos, flujos, almacenes y entidades deben ser significativos.** Los procesos dentro de un DFD representan que una función o transformación se esta llevando a cabo. La manera más comúnmente usada para nombrar un proceso, es usar un verbo activo y un objeto para formar una frase que lo describa. Tanto los nombre de los procesos, flujos, entidades y almacenes deben de provenir de vocabularios que tengan algún sentido para el usuario.
- 2) **Se deben numerar los procesos.** Es una practica muy conveniente para referirse al los procesos dentro de un DFD, es importante tener en cuenta que este sistema de numeración no indica el orden en el cual los procesos son realizados, ya que un DFD es una red de procesos asíncronos que están intercomunicados.
- 3) **Evitar que el DFD se vuelva muy complejos.** Un DFD debe representar de manera sencilla y precisa los procesos que un sistema lleva a cabo y las relaciones entre ellos. Un detalle a tener en cuenta es que no se debe crear un DFD con demasiados procesos, almacenes, flujos o entidades.
- 4) **Asegurarse que el DFD sea lógicamente consistente.** Para evitar problemas de inconsistencia en los DFD tenga en mente las siguientes reglas:
 - a. Evite los procesos que no tienen salidas.
 - b. Evite los procesos que tiene salidas pero no tienen entradas.
 - c. No deje ningún flujo, proceso o entidad sin nombre.

2.2 Bases de Datos

Desde que los equipos de cómputo aparecieron, se descubrió que estos eran adecuados para trabajar con datos, en especial, cuando el volumen de los mismos era considerable. Hoy en día, aun con los avances de los sistemas de cómputo, el procesamiento de datos sigue siendo la parte fundamental de cualquier sistema de cómputo.

Podemos definir las bases de datos como un conjunto de información la cual es almacenada en dispositivos de almacenamiento secundario, y un conjunto de programas (llamado Manejadores de Bases de Datos) los cuales permiten el acceso a dichos datos. Los tres niveles distintos desde los que se pueden estudiar las bases de datos son:

1. **Nivel Físico:** Se refiere al nivel real en donde los datos están siendo almacenados (ya sea en registros, archivos, etc.). Este nivel lleva asociado una representación de los datos que se denomina, nivel físico.
2. **Nivel Conceptual:** Corresponde a una visión de la Base de Datos desde el punto de vista del mundo real. Se representan a los datos almacenados como entidades, sin importar como se encuentran almacenados físicamente. Este nivel lleva asociado una representación denominada, esquema conceptual.
3. **Nivel de Visión:** Se trata de partes del sistema conceptual, con la diferencia que en el nivel conceptual se muestra toda la Base de Datos, en el nivel de visión solo se tiene acceso a ciertas partes de la Base de Datos. El esquema asociado a este nivel es el esquema de Visión.

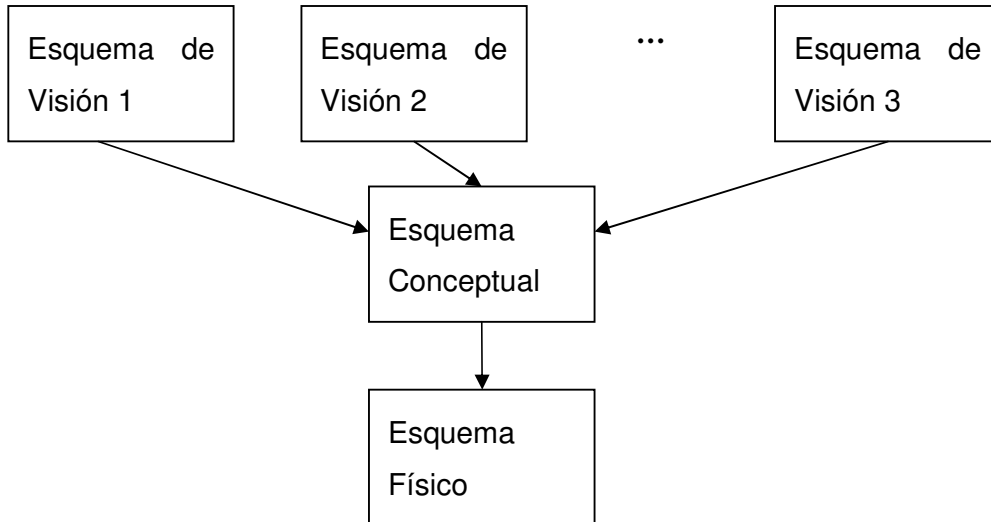


Figura 2.2.1: Los 3 diferentes niveles de una Base de Datos

Para las personas involucradas en el diseño y desarrollo de Bases de Datos, El esquema conceptual es el más utilizado, ya que les permite ver la Base de Datos como un conjunto de tablas que agrupan datos acerca del mismo tema y que contienen columnas y filas de información. Estas tablas pueden estar relacionadas entre sí (a esto ultimo se le conoce como una base de datos relacional).

2.2.1 Objetos de las Bases de datos

En una Base de Datos esta compuesta por un gran conjunto de objetos, entre esta gran cantidad de objetos, los más comunes son:

1. **Tablas:** Son los objetos que contienen los tipo de datos y los datos en sí.
2. **Columnas:** Son las partes de una tabla que contienen los datos, a las columnas se le s debe asignar un tipo de dato y un nombre único.
3. **Tipos de Datos:** el tipo de información que se estará almacenando en una columna especifica, estos pueden ser caracteres, números enteros, fechas, entre otros.
4. **Claves primarias:** Se utilizan esencialmente en las Bases de Datos relacionales, y se utilizan para identificar de manera única cada elemento que se desea almacenar.

5. **Claves externas:** Al igual que las claves primarias, estas son utilizadas en las Bases de Datos relacionales, e indican las columnas que hacen referencia a claves primarias.
6. **Valores predeterminados:** valores que se le asignan a las columnas de modo que, si no se registran datos en una instrucción de inserción en esta columna, se utilizaran dichos valores para llenarla.
7. **Índices:** Ayudan a organizar los datos a efecto de que las consultas se ejecuten con mayor rapidez.

2.2.2 Las ventajas que brinda un sistema de bases de datos.

Las ventajas que una Base de Datos brindan son:

1. **Reducir la redundancia y la inconsistencia de los datos:** La redundancia e inconsistencia de datos se refiere a que existe duplicidad de ciertos datos, y cada uno de esos datos no concuerdan entre sí. Aunque esto depende en gran medida del diseño de la Base de Datos.
2. **Acceso a los datos:** Facilita el manejo de los datos para los usuarios y los pone a su disposición para que sean fáciles de consultar.
3. **Acceso concurrente a los datos:** Muchos sistemas de Bases de Datos permiten que múltiples usuarios consulten y actualicen los datos simultáneamente.
4. **Seguridad:** Permiten crear reglas de acceso a los datos, lo que garantiza la autenticación y protección de los datos.

2.2.3 Sistema Manejador de Bases de Datos (DBMS)

Los DBMS son conjuntos de rutinas y programas interrelacionados, que se encargan de guardar, modificar y extraer información de una Base de Datos.

Funciones de un DBMS:

1. Crear y organizar las Bases de Datos
2. Establecer y mantener los accesos a la Base de Datos.
3. Registrar el uso de la Base de Datos.
4. Respaldar y restaurar la información almacenada.

5. Controlar los accesos concurrentes.

6. Implementar los mecanismos que manejan la seguridad e integridad.

Podríamos decir que un DBMS es el centro de la base de datos, ya que es el encargado de llevar un control de todos los posibles aspectos que la pueden afectar.

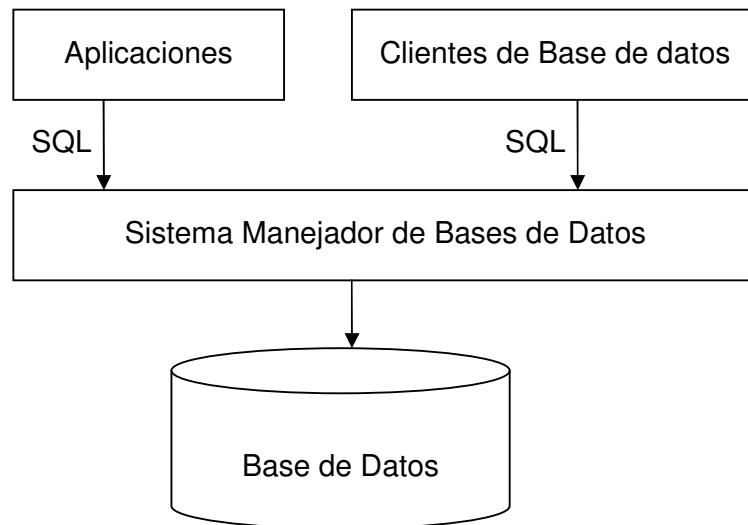


Figura 2.2.3.1: Esquema General del Funcionamiento de un DBMS

2.3 Introducción a SQL

Una vez que se encuentran los datos almacenados en una Base de Datos, es necesario manipularlos desde algún tipo de programa. Para ello se desarrollo SQL (lenguaje de consultas estructurado), un lenguaje para acceder a Bases de Datos que contienen comandos para manejar los datos. Es utilizado por prácticamente todos los DBMS existentes en el mercado.

Las sentencias de SQL están divididas en 2 grandes categorías:

Lenguaje de Definición de Datos (DDL): sentencias utilizadas para crear y modificar las estructuras de las tablas y otros objetos en la base de datos.

Lenguaje de Manipulación de Datos (DML): sentencias que son utilizadas para trabajar con los datos en las tablas.

2.3.1 Como leer datos de la base de datos

La instrucción SELECT de SQL es la encargada de obtener datos de la base de datos, la sintaxis básica de esta instrucción es la siguiente:

```
SELECT <campos> FROM <tablas>
```

Esto regresaría un conjunto de campos y filas de la tabla especificada, Si se quisieran obtener todos los campos de la tabla se usa el carácter * en lugar de escribir la lista completa de campos.

Si desea especificar el orden de los registros obtenidos, se debe agregar a la instrucción SELECT la cláusula ORDER BY, la cual se basa en una lista de campos que se utilizaran para realizar el ordenamiento. De forma predeterminada, los campos especificados después del ORDER BY se ordenan de manera ascendente, aunque también se le puede indicar la manera de realizar el ordenamiento agregando las cláusulas ASC (ascendente) o DESC (descendente) para que realice el ordenamiento de la forma especificada.

```
SELECT <campos> FROM <tablas> ORDER BY <campos> <ASC : DESC>
```

Es posible especificar también a la sentencia SELECT criterios para la obtención de datos. Esto se lleva a cabo con la utilización de la cláusula WHERE. Con esta cláusula se pueden especificar n numero de criterios o condiciones para restringir la obtención de los datos. Para combinar varios criterios se pueden utilizar los operadores AND y OR.

```
SELECT <campos> FROM <tablas> WHERE <condición1> <AND : OR>  
<condición 2> ....
```

2.3.2 Como agregar datos

EL comando SQL que se utiliza para agregar datos a la Base de Datos se conoce como INSERT y tiene la siguiente sintaxis.

```
INSERT INTO <tabla> (campo1,campo2,...) VALUES ('valor1',' valor2',...)
```

Es importante mencionar que las comillas simples que acompañan a los valores se utilizan para diferenciar entre el valor y el SQL, estas comillas no son necesarias para valores que no sean texto.

2.3.3 Modificación de datos

La instrucción UPDATE de SQL ofrece la posibilidad de realizar modificaciones a los datos que ya se encuentran almacenados en la base de datos. La sintaxis básica de la instrucción es:

```
UPDATE <tabla> SET <campo> = <nuevo_valor>
```

UPDATE al igual que SELECT puede utilizar la cláusula WHERE, ya que si no se especifica un WHERE en el UPDATE, se modificarán todos los registros de la tabla.

```
UPDATE <tabla> SET <campo> = <nuevo_valor> WHERE <condiciones>
```

2.3.4 Eliminación de Registros

SQL ofrece la instrucción DELETE, la cual le permite eliminar datos de una tabla en la Base de Datos. Así como UPDATE y SELECT también se puede utilizar la cláusula WHERE, la cual es optativa pero importante, ya que si no se especifica, borrará todos los datos de la tabla indicada.

```
DELETE <tabla> WHERE <condiciones>
```

2.4 Modelo Entidad Relación

El modelo E-R (entidad relación) se basa en una percepción del mundo real, la cual está formada por objetos básicos llamados entidades y las relaciones entre estos objetos así como las características de estos objetos llamados atributos.

2.4.1 Entidades y conjunto de entidades

Una entidad es un objeto que existe y se distingue de otros objetos de acuerdo a sus características llamadas atributos. Las entidades pueden ser de dos tipos, fuertes y débiles. Una entidad débil es una entidad cuya existencia depende de otra entidad, Una entidad fuerte es la que puede existir por si misma sin depender de otra entidad.

Una entidad se caracteriza y distingue de otra por los atributos, en ocasiones llamadas propiedades, que representan las características de una entidad. Los atributos de una entidad pueden tomar un conjunto de valores permitidos al que se le conoce como dominio del atributo. Así cada entidad se describe por medio de un conjunto de parejas formadas por el atributo y el valor de dato.

Las entidades son representadas gráficamente por rectángulos y el nombre de la misma aparece al interior de estos. Un nombre de entidad solo puede aparecer una vez en el esquema conceptual.

2.4.2 Relaciones y conjunto de relaciones.

Una relación es la asociación que existe entre dos a más entidades. Cada relación tiene un nombre que describe su función, se representan gráficamente mediante rombos y su nombre aparece en el interior.

La cantidad de entidades en una relación determina el grado de la relación, por ejemplo la relación ALUMNO-MATERIA es de grado 2, ya que intervienen la entidad ALUMNO y la entidad MATERIA, la relación PADRES, puede ser de grado 3, ya que involucra las entidades PADRE, MADRE e HIJO.

Aunque el modelo entidad-relación permite relaciones de cualquier grado, la mayoría de las aplicaciones del modelo sólo consideran relaciones del grado 2. Cuando son de tal tipo, se denominan relaciones binarias.

Existen 4 tipos de relaciones que pueden establecerse entre entidades, las cuales establecen con cuantas entidades de tipo B se pueden relacionar una entidad de tipo A:

2.4.3 Tipos de relaciones.

Relación uno a uno: Se presenta cuando existe una relación como su nombre lo indica uno a uno, denominado también relación de matrimonio. Una ocurrencia de la entidad A solo se puede relacionarse con una ocurrencia de la entidad B, y viceversa;

Es representado gráficamente de la siguiente manera:

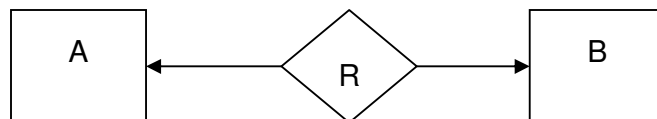


Figura 2.4.3.1: Relación uno a uno

Relación uno a muchos: Significa que una ocurrencia de la entidad A puede relacionarse con cualquier cantidad de ocurrencias de la entidad B, y una ocurrencia de la entidad B solo puede estar relacionada con una ocurrencia de la entidad A.

Su representación gráfica es la siguiente:

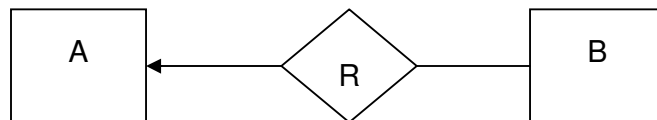


Figura 2.4.3.2: Relación uno a muchos

Muchos a uno: Indica que una ocurrencia de la entidad B puede relacionarse con cualquier cantidad de ocurrencias de la A, mientras que cada ocurrencia de la entidad A solo puede relacionarse con solo una ocurrencia de la entidad B.

Su representación gráfica es:

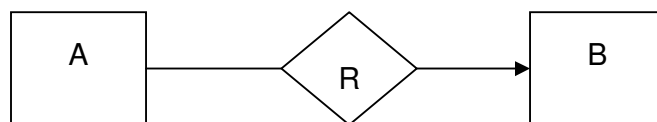


Figura 2.3.4.3: Relación muchos a uno

Muchos a muchos: Establece que cualquier cantidad de ocurrencias de la entidad A pueden estar relacionados con cualquier cantidad de ocurrencias de la entidad B.

Gráficamente está representada por:

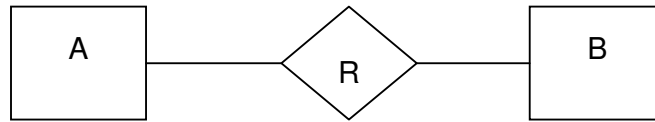


Figura 2.3.4.4: Relación muchos a muchos

A los tipos de relaciones antes descritos, también se le conoce como cardinalidad.

La cardinalidad nos especifica los tipos de relaciones que existen entre las entidades. Cabe mencionar que la cardinalidad para cada conjunto de entidades depende del punto de vista que se le de al modelo en estudio.

2.4.4 Diagrama Entidad-Relación

Denominado por sus siglas como E-R, describe con un alto nivel de abstracción la distribución de datos almacenados en un sistema.

Un diagrama E-R, esta representado a través de una colección de tablas. Para cada una de las entidades y relaciones existe una tabla única a la que se le asigna como nombre el nombre de la entidad, cada tabla tiene un número de columnas que son definidas por la cantidad de atributos y las cuales tienen el nombre del atributo.

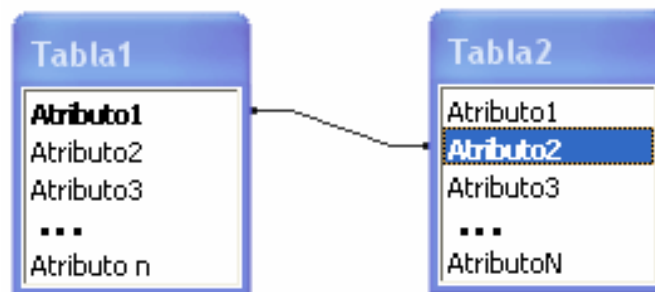


Figura 2.4.4.1: Ejemplo de un diagrama E-R sencillo

2.5 Diccionario de Datos

El diccionario de datos es un listado organizado que contiene las características lógicas de todos los datos que se utilizarán para el desarrollo de un sistema con definiciones precisas y rigurosas para que tanto el usuario como el analista tengan un entendimiento común de todas las entradas, salidas, datos almacenados y cálculos. Además es una forma de documentación para el diseñador de bases de datos.

Las razones por las cuales se deben utilizar los diccionarios de datos son:

1. Manejar detalles en los sistemas. Los sistemas al sufrir cambios continuos, es muy difícil manejar todos los detalles. Por esto el diccionario de datos ayuda a registrar la información.
2. Asignarle un solo significado a cada uno de los elementos y actividades del sistema. Proporcionan asistencia para asegurar significados comunes para los elementos y actividades del sistema y llevando registro de detalles adicionales relacionadas con el flujo de datos en el sistema.
3. Documentar las características del sistema, incluyendo partes o componentes así como los aspectos que los distinguen. También es necesario saber bajo que circunstancias se lleva a cabo cada proceso y con que frecuencia ocurren.
4. Facilitar el análisis de los detalles con la finalidad de evaluar las características y determinar donde efectuar cambios en el sistema.
5. Localizar errores y omisiones en el sistema.

2.6 Programación orientada a objetos

La programación orientada (POO) a objetos es una forma de diseñar y generar aplicaciones. Su parte central radica en el concepto de objeto, que es un elemento de programación que se utiliza para reunir información y funcionalidad.

Los objetos por lo general se utilizan para describir entidades, estas entidades cuentan con ciertos atributos que se utilizan para describirlas, así, como también cuentan con ciertas acciones que pueden ejecutarse en ellos, en la POO las entidades son los objetos, los atributos se denominan propiedades y las acciones se conocen como métodos.

2.6.1 Conceptos importantes en la POO

Los conceptos de la POO han sido definidos de manera que sean universales en todos los lenguajes de programación, a continuación se explican algunos de estos conceptos:

1. **Clases, Objetos e Instancias:** Estos tres objetos son la base fundamental del trabajo con la POO. Una Clase es una plantilla de un objeto, representa la estructura básica o los planos de construcción de un objeto. Cada objeto generado de una clase se conoce como una instancia de ella, y cuentan con su propio espacio de memoria asociado.
2. **Propiedades:** Una propiedad es un valor que forma parte de un objeto, y que puede ser establecido o leído mediante el objeto. Ya que como lo mencionamos anteriormente, cada instancia tiene su propio espacio de memoria, los valores de las propiedades se almacenan de manera independiente para cada instancia.
3. **Métodos:** Las clases también pueden tener cierta funcionalidad, a este conjunto de funcionalidades se le conoce como métodos. Los métodos permiten incluir en una clase operaciones. La ejecución de los métodos puede ser realizada desde cualquier instancia de la clase. Pero, al igual que las propiedades, se ejecutan para una instancia específica.
4. **Herencia:** Este es uno de los conceptos primordiales de la POO, se refiere a la idea de basar una clase en otra, con la finalidad de modificar o agregar las características o métodos de la clase original.

2.7 Tecnología .NET de Microsoft

Para comenzar a definir que es .NET y el Framework.NET realizaremos una analogía con un sistema operativo.

Cuando se ejecuta una aplicación en una computadora el sistema operativo brinda una capa base que provee servicios a la aplicación. En el caso de Windows esta capa base se llama API Win32. El API de Win32 provee servicios como leer archivos del disco, cambiar resolución de la pantalla, leer de memoria, etc. La aplicación final solo tiene que ejecutar las funciones respectivas del API Win32.

En el nuevo ambiente .NET, el Framework.NET es muy similar al API Win32. Las aplicaciones desarrolladas con .NET utilizan el Framework.NET y sus clases como base para realizar tareas. Por ejemplo, si se desea realizar gráficos en dos dimensiones en una aplicación .NET se puede llamar a las clases de GDI+ contenidas en el Framework.NET. Básicamente el Framework.NET posee clases que la aplicación final utiliza para realizar una diversidad de tareas.

En la realidad el Framework.NET es otra capa de abstracción entre la aplicación final y los dispositivos de Hardware. El Framework.NET no sustituye al API del sistema operativo ya que al final los comandos para los dispositivos de Hardware tienen que pasar a través del Sistema Operativo.

A continuación se muestra un diagrama de los niveles de abstracción para una aplicación .NET:

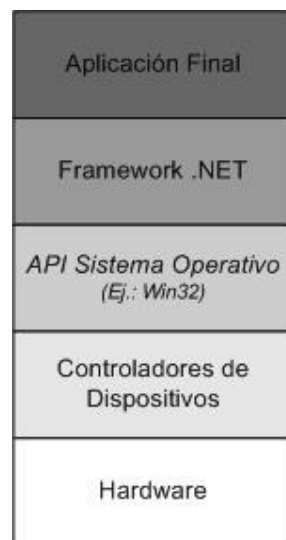


Figura 2.7.1: Niveles de abstracción de una aplicación en .NET

Además de esta nueva arquitectura el .NET ofrece grandes ventajas sobre las tecnologías anteriores de desarrollo de Microsoft. El .NET tiene todas las características de los nuevos ambientes de desarrollo. A continuación se listan las principales características:

1. **Programación Totalmente Orientada a Objetos:** La utilización de objetos (clases) permite lograr reutilización y encapsulación de código.

2. **Multiplataforma:** Las aplicaciones creadas en .NET pueden ser ejecutadas en cualquier sistema operativo, siempre y cuando exista una versión del Framework.NET compatible.
3. **Variedad de Controles:** NET ofrece mayor variedad de controles en su Framework.NET, que la mayoría de entornos de desarrollo, lo que permitirá crear una interfaz de usuario mas amigable para el modulo administrativo.
4. **Multi-lenguaje:** El .NET permite a los desarrolladores programar en Visual Basic, C#, C++ o Java. Es posible crear proyectos que combinen la utilización de varios lenguajes.

El ambiente de desarrollo .NET incluye la habilidad de desarrollar aplicaciones basadas en Windows y aplicaciones basadas en Web.

La tecnología de .NET para el desarrollo Web es el ASP.NET. El ASP.NET es considerado uno de los mejores entornos de desarrollo para aplicaciones Web.

2.8 Arquitectura Cliente/Servidor

La arquitectura cliente/servidor se basada en la red para su funcionamiento. En dicha arquitectura todas las computadoras de la red son un cliente o un servidor.

Los servidores son los encargados de proveer servicios a los clientes de la red. Un servidor puede proveer a los clientes:

1. Archivos
2. Una base de datos
3. Aplicaciones
4. Servicios de Impresión entre otros.

Por definición común en las arquitecturas cliente/servidor los servidores realizan la mayor parte de procesos y cálculos complejos. Es por ello que es común crear en el servidor de base de datos procedimientos almacenados y subrutinas que se ejecutan en el servidor y no en el cliente. De igual manera las aplicaciones basadas en Web de ejecutan en su mayor parte en el servidor y el cliente solo procesa el HTML puro y pequeñas rutinas JavaScript de validación de errores.

La ejecución de la mayoría de procesos en el servidor esta cambiando en la actualidad. Esto se debe a que las computadoras cliente son ahora muy potentes

y tienen una capacidad elevada debido a los bajos costos del Hardware. Los servidores son ahora más utilizados para la centralización de los datos para todos los clientes, pero no como fuentes de procesamiento de datos.

.NET apoya esta nueva tendencia de utilizar la capacidad de procesamiento de los clientes, en lo que respecta la conectividad y transacciones con la base de datos del servidor. El .NET ofrece el nuevo ADO.NET para la conectividad con las fuentes de datos. El ADO.NET promueve la utilización de trabajo en modo desconectado con la base de datos.

El modo desconectado del ADO.NET consiste en mantener una conexión en la base de datos durante el menor tiempo posible, con el fin de reducir los recursos utilizados en el servidor. Para lograrlo la conexión extrae una copia de los datos útiles para el cliente de la base de datos, esto incluye los registros, las tablas y sus relaciones. Una vez que el cliente tiene la copia de los datos se trabaja en ellos; se pueden modificar eliminar y agregar datos. Cuando el trabajo en los datos finaliza se realiza nuevamente la conexión a la base de datos y esta se actualiza.

En cuanto a los sistemas basados en Web siempre se ejecutan totalmente en el servidor (con la excepción de cuando se incluyen pequeños controles que requieren procesamiento en el cliente).

2.9 Aplicaciones Basadas en Windows

Las aplicaciones basadas en Windows se refieren al tipo de arquitectura utilizada para la interfaz de usuario. Esta arquitectura esta basada en las ventanas de Windows. Una ventana es el área de la pantalla utilizada por una aplicación para mostrar información. Una aplicación puede tener varias ventanas pero cada ventana solo puede pertenecer a una aplicación. La ventana en conjunto con los elementos secundarios como cajas de texto, menús, cuadros de lista constituyen una interfaz de usuario. Estos elementos secundarios se les llaman controles.

La finalidad de los controles es mostrar información a los usuarios y permitir que los usuarios introduzcan información.

El .NET permite la creación de la interfaz de usuario como aplicaciones de Windows utilizando su tecnología Windows Forms. Dicha metodología de

desarrollo es muy similar a la del Visual Basic 6 con la diferencia que sus controles y todos sus objetos están basados en el Framework.NET.

Las aplicaciones basadas en Windows tienen dos ventajas sobre las aplicaciones Web:

1. Pueden utilizar cualquier tipo de control para la interfaz de usuario (aun controles personalizados), y todos sus componentes se instalan con la aplicación. El disponer de mayor variedad de controles permite crear una interfaz de usuario más amigable y funcional.
2. En las aplicaciones Web cada vez que el usuario interactúa con el sistema es necesario esperar que la página se refresque ya que todos los procesos se realizan en el servidor Web. En las aplicaciones Windows en el servidor solo se realizan las consultas a la base de datos.

Una desventaja de las aplicaciones basadas en Windows es que requieren su instalación en cada equipo donde se ejecutara. En versiones anteriores de Visual Basic esta instalación (implementación del sistema) podría causar problemas inesperados por falta de DLLs en la computadora cliente. En .NET este problema desaparece ya que para la implementación/instalación de una aplicación solo es necesario realizar los siguientes pasos:

Instalación del Framework.NET en la maquina cliente: Este proceso es sencillo ya que el Framework.NET se encuentra gratuitamente en Internet. Además sistemas operativos nuevos como XP/2003 ya poseen la opción de instalación del Framework.NET.

Copiar Archivos a computadora cliente: Con las aplicaciones .NET basta copiar todos los archivos ejecutables y librerías DLL del sistema en una carpeta para ejecutarlos.

2.10 Aplicaciones Basadas en Web

La arquitectura de aplicaciones basadas en Web se refiere a la utilización de una página Web y el explorador de Internet como interfaz de usuario. Las aplicaciones Web son rutinas que se ejecutan en el servidor y regresan como resultado texto en formato HTML el cual puede ser desplegado por el explorador de Internet.

Aunque al explorador cliente solo se le envía el resultado HTML es posible enviar código dentro del HTML. Esto se realiza generalmente mediante el envío de rutinas JavaScript. El JavaScript es la mejor opción ya que es un lenguaje común que la gran mayoría de exploradores pueden leer. La utilización más común del código en el servidor es la validación de formularios.

Los formularios son campos HTML utilizados para capturar información del usuario. Dicha información es enviada a la cabecera HTTP.

Las aplicaciones basadas en Web eliminan el requerimiento de instalar la aplicación a todos los clientes que la utilizaran, ya que el único requerimiento para ejecutarlas en un explorador de Internet con capacidad de ejecutar JavaScript. El prescindir de la instalación nos lleva a una implementación más sencilla. Además para realizar actualizaciones a la aplicación basta con realizar cambios en el servidor Web.

Aun cuando las aplicaciones basadas en Web tienen la ventaja de eliminar el requerimiento de instalación tiene varias desventajas:

1. Para cada operación una aplicación basada en Web debe de realizar un viaje redondo de red. Esto se refiere a que se debe enviar al servidor una solicitud, el servidor debe procesarla y enviar una respuesta. Esto se debe a que el HTML recibido es totalmente estático. Esta desventaja causa que el funcionamiento de una aplicación Web sea más lento y utilice más recursos de red que una aplicación basada en Windows.
2. Una aplicación Web está limitada a la cantidad de controles que provee el HTML, la utilización de controles propietarios no es aceptable ya que estos requieren la instalación en cada cliente (lo cual es contradictorio con el fin principal de las aplicaciones Web, el cual es evitar las instalaciones en el cliente)

Las aplicaciones Web trabajan generalmente utilizando tres capas:

1. **Un Explorador de Internet:** Esto es lo que el cliente requiere y tiene que tener instalado. La ventaja es que en la actualidad todos los sistemas operativos poseen un explorador de Internet. El explorador de Internet recibe del servidor archivos en formato HTML. Esta información es

procesada lo que resulta en la interfaz de usuario que se le muestra al cliente.

2. **Herramientas para la Creación de Web Dinámicas:** Las herramientas para la creación de Web dinámicas se encargan de procesar código el cual resulta en el archivo formato HTML que se le envía al explorador de Internet. Dichas herramientas tienen varias funcionalidades útiles para el desarrollador del sistema, entre ellas esta la accesibilidad a Bases de Datos. La ejecución de los programas creados con estas herramientas se realiza totalmente en el servidor, cuando la ejecución termina el resultado HTML es enviado vía RED (generalmente Internet) al cliente. Estas herramientas pueden recibir datos del cliente a través de las cabeceras HTTP. Algunas herramientas para la creación de Web dinámicas son: ASP, ASP.NET, PHP, Java Servlets.
3. **Base de Datos:** Las herramientas para la creación de Web dinámicas se conectan generalmente a una base de datos que contiene la información con la interactuara el cliente. La información puede ser manipulada.

La tecnología .NET utiliza el ASP.NET para la creación de aplicaciones Web. Dicha tecnología aprovecha todas las características mencionadas para el .NET.

2.11 Microsoft SQL Server

SQL Server es el servidor de bases de datos relacionales de Microsoft. Es un servidor capacitado para altos niveles transaccionales, dicha base de datos tiene costos económicos muy competitivos en relación a la competencia.

SQL Server es basado en el Sybase SQL Server y fue inicialmente desarrollado para sistemas operativos OS/2.

SQL Server esta específicamente desarrollado para aplicaciones cliente/servidor. En dicha arquitectura el SQL Server es parte del servidor.

En la actualidad hay varios tipos de SQL Server los cuales son:

- Edición Estándar: Es la mas utilizada de SQL Server y presenta todas las funcionalidades al usuario. Como limitantes principales esta versión solo soporta servidores con cuatro procesadores o menos.

- Edición Empresarial: Tiene todas las características de la edición Estándar pero soporta hasta 32 procesadores y permite Clustering. El Clustering consiste en la creación de SQL Servers espejos para reducir el tráfico y evitar fallas.
- Edición para negocios pequeños: Esta edición posee varias limitantes para reducir su costo. Una base de datos no puede ser mayor a 10Gb, solo se permiten 50 usuarios simultáneos de acceso.

Microsoft SQL Server es un sistema robusto de bases de datos transaccionales para soportar la carga de trabajo de cualquier sistema transaccional de alta escala. También posee ventajas como procedimientos almacenados, triggers e integridad referencial los cuales permiten la creación de sistemas más robustos y seguros.

Además del soporte para grandes sistemas transaccionales, SQL Server también brinda herramientas para el análisis de datos (OLAP). Dichas herramientas permiten el análisis de DataWarehouse.

Como se menciona el SQL Server permite la creación de triggers y procedimientos almacenados. Estos son pequeñas subrutinas que se ejecutan en el servidor de bases de datos. La programación de dichas subrutinas se crean utilizando el lenguaje Transact-SQL, este lenguaje es utilizado en Microsoft SQL Server y Sybase SQL Server esta basado en el Estándar SQL pero con características adicionales para crear subrutinas más robustas y funcionales.

El Microsoft SQL Server es la mejor opción para utilizar con las herramientas de desarrollo .NET. Algunas ventajas de utilizar el Microsoft SQL Server combinado con el Framework .NET son:

- El ambiente gráfico de desarrollo de .NET permite acceder a la administración del SQL Server de forma similar que el mismo Enterprise Manager.
- El Framework .NET posee un proveedor de conexión a bases de datos especialmente diseñado para Microsoft SQL Server. Es decir que el .NET no tiene que utilizar OleDb u ODBC para realizar las conexiones con las bases de datos de Microsoft SQL Server. Este proveedor de conexión

propio para SQL Server posee métodos especializados para las conexiones con SQL Server que ningún otro proveedor ofrece, además de mejorar el rendimiento de la conexión.

En la siguiente tabla se listan las principales características del SQL Server:

Característica	MS SQL Server
Tamaño máximo de la base de datos	1,048,516 TB
Numero de usuarios concurrentes	Limitado por la memoria del servidor.
Numero de campos por tabla	1,024
Tamaño máximo de una tabla	Limitado por el espacio disponible en el Disco Duro.
Numero de tablas en una consulta.	256
Capacidad de trabajar con procedimientos almacenados	Si
Capacidad de trabajar con Triggers	Si
Soporte para vistas	Si
Integración con la seguridad de Windows NT/2000/XP	Soportado
Respaldos automáticos programados	Soportado
Compatible con Crytal Reports	Si
Compatible con Centura Report Builder	Si

Capitulo III: Propuesta de Solución

3.1 Plan de Solución

Con el fin de agilizar el proceso de creación de horarios de clases de la Universidad Don Bosco, surge la siguiente propuesta como una alternativa viable.

El Sistema de apoyo para la Programación de Horarios de Clase de la Universidad Don Bosco consistirá en cuatro módulos bases, uno de estos será una aplicación Web y el resto aplicaciones cliente-servidor, estos se detallan a continuación:

1. Módulo Administrativo.
2. Módulo de Asignación de Horarios.
3. Módulo de Asignación de Salones.
4. Módulo Web.

El Módulo Administrativo se encargara de reunir y manejar todas las variables que intervienen en los procesos de asignación de horarios y salones. Este módulo contiene los siguientes componentes:

1. **Manejo de Edificios.** Componente de mantenimiento de los edificios existentes en la Universidad.
2. **Manejo de Aulas.** Componente de mantenimiento de las aulas hábiles de cada Edificio en la Universidad.
3. **Manejo de Escuelas.** Componente de mantenimiento de las Escuelas de la Universidad.
4. **Manejo de Carreras.** Componente de mantenimiento de carreras que se imparten en la Universidad.
5. **Manejo de Materias.** Componente de mantenimiento de todas las materias que se imparten en la universidad.
6. **Manejo de Pensums.** Componente de mantenimiento de los distintos pensums que se manejan en la Universidad.
7. **Manejo de Ciclos.** Componente que se encarga de definir el ciclo actual, es decir el que esta en proceso de programación.

8. **Manejo de Planes de Estudios.** Componente de mantenimiento de los Planes de Estudio de la Universidad.
9. **Manejo de Usuarios.** Componente de mantenimiento de los usuarios que accederán a los distintos módulos pertenecientes al SAPH.
10. **Reportes.** Componente que se encarga de generar reportes. Este componente se comunica con el Módulo Web, generando reportes en un navegador Web, el listado de reportes disponibles se encuentra especificado en los Alcances, punto No. 10.

El Módulo de Asignación de Horarios es la parte medular del software, en éste se organizarán las variables que intervienen en el proceso de programación de horarios de clase que han sido recopiladas con la ayuda del Módulo Administrativo:

1. **Manejo de Docentes.** Administrará un listado de catedráticos que estarán disponibles para impartir las distintas asignaturas.
2. **Proceso de Asignación de Horarios.** Este proceso se encarga de armar los horarios propiamente, aquí se determinan el número de grupos para cada materia, quien impartirá cada grupo, y que horarios le serán asignados.
3. **Reportes.** Componente que se encarga de generar reportes. Este componente se comunica con el Módulo Web, generando reportes en un navegador Web, el listado de reportes disponibles se encuentra especificado en los Alcances, punto No. 10.

El Módulo de Asignación de Salones en éste se organizarán las variables que intervienen en el proceso de asignación de salones de clase que han sido recopiladas con la ayuda del Módulo Administrativo:

1. **Reservación de Espacios Físicos.** Este es el encargado de reservar un aula o salón para usos varios, que no se encuentran contemplados dentro de los horarios regulares de la Universidad.
2. **Proceso de Asignación de Salones.** Este proceso se encarga de establecer los salones disponibles a los distintos horarios de los grupos de materia que se impartirán durante el ciclo activo.

3. **Reportes.** Componente que se encarga de generar reportes. Este componente se comunica con el Módulo Web, generando reportes en un navegador Web, el listado de reportes disponibles se encuentra especificado en los Alcances, punto No. 10.

También, la herramienta aportará nuevas opciones que serán de gran utilidad, las cuales se encuentran contenidas en el **Módulo Web:**

1. **Manejo de disponibilidad de tiempo de docentes.** Una herramienta Web que permita a los docentes especificar las horas de las cuales ellos disponen para impartir sus cátedras. Dicha herramienta será de gran utilidad para aquellos docentes que son externos a la universidad. Además este componente ayudará a liberarle carga al responsable de la creación de horarios, logrando que cada catedrático se encargue de establecer sus horas disponibles.
2. **Consultas de horarios Web.** Permitirá a personas interesadas en los horarios, visualizar la información relacionada de distintas formas. Como su nombre lo indica esta herramienta únicamente permite consultar la información.
3. **Reporteador.** Es el componente que se encarga de recibir peticiones de reportes por parte de todos los módulos que componen el SAPH y darles salida en el navegador Web, el listado de reportes disponibles se encuentra especificado en los Alcances, punto No. 10.

Para el desarrollo de este proyecto se utilizará la nueva tecnología de Microsoft .NET. Para la programación de la aplicación cliente-servidor se utilizará Visual Basic .NET y para la aplicación Web se utilizará ASP.NET.

Para el almacenamiento de la información de horarios, docentes, asignaturas y todo lo relacionado con el sistema se utilizará Microsoft SQL Server 2000 como gestor de base de datos.

3.2 Recursos a utilizar

3.2.1 Recursos técnicos

El equipo que se recomienda para lograr un desarrollo satisfactorio del Sistema de apoyo para la Programación de Horarios de Clase de la Universidad Don Bosco se describe a continuación:

3 Computadoras con las siguientes características
Procesador de 1.0 a 1.5 GHz.
256 a 512 MB de RAM.
Disco Duro de 40 GB.
Floppy Drive.
CD-ROM 52 X
Case ATX.
Teclado.
Mouse Óptico.
Monitor de 15"
Tarjeta de Red 10/100

Para comunicaciones
Un Switch de 8 puertos

Otros equipos recomendados
Una impresora a colores.

Software Necesario
3 Microsoft Windows XP Professional.
1 Microsoft Visual Studio .NET
1 Microsoft SQL Server 2000 Developer.

3.2.2 Recursos Económicos

A continuación se muestra el valor económico presupuestado para el proyecto, estos se dividen en dos partes, el presupuesto real de desarrollo, que se ajusta al caso del desarrollo del software por parte de una entidad externa a la Universidad Don Bosco. La otra parte, refleja el presupuesto de desarrollo si el Sistema se desarrollara dentro de la Universidad.

Presupuesto de desarrollo por parte de una entidad externa

Recursos de Personal				
Cantidad	Responsabilidad	Sueldo Mensual	Sueldo total Mensual	Sueldo 8 Meses
3	Analista / Programador	\$500	\$1,500	\$12,000
			Total	\$12,000

Recursos de Software			
Cantidad	Concepto	Precio U	Precio Total
1	Visual Studio .NET Profesional	\$1,100	\$1,100
1	SQL Server 2000 Developer	\$50	\$50
		Total	\$1,150

Recursos de Equipo			
Cantidad	Concepto	Precio U	Precio Total
3	Equipos de desarrollo con Windows XP	\$700	\$2,100
		Total	\$2,100

Gastos Generales		
Gasto	Mensual	Por 8 Meses
Servicio de agua	\$10	\$80
Servicio de teléfono	\$15	\$120
Servicio de electricidad	\$25	\$200
Servicio de Internet	\$60	\$480
Alquiler de local	\$120	\$960
Suministros	---	\$300
	Total	\$2140

Capitulo IV: Análisis y Diseño del Sistema

4.1 Descripción General del Sistema.

Como ya se menciona anteriormente el sistema se divide en 4 módulos principales:

1. Módulo Administrativo.
2. Módulo de Asignación de Horarios.
3. Módulo de Asignación de Salones.
4. Módulo Web.

Estos módulos son los encargados de interactuar con las diferentes entidades que utilizaran el sistema, las cuales son:

1. Administración Académica de la Universidad Don Bosco.
2. Los Encargados de la programación de horarios de cada escuela.
3. Los Encargados de la asignación de salones de los distintos edificios.
4. Los docentes de la Universidad Don Bosco.
5. Los Alumnos de la Universidad Don Bosco.

Estas entidades se relacionan específicamente con ciertos módulos específicos del sistema, así, Administración Académica se relaciona directamente con el Modulo Administrativo; los encargados de la realización de horarios se relacionan directamente con el Modulo de Asignación de Horarios; los encargados de asignación de salones con el Módulo de Asignación de Salones; y los docentes, alumnos y otros usuarios externos a la Universidad, se relacionan con el modulo Web.

A continuación explicaremos con más detalle cada una de estas relaciones.

4.1.1 Modulo Administrativo.

Este modulo esta diseñado para ser utilizado por el personal de Administración Académica de la Universidad Don Bosco. Este cuenta con un proceso de validación de usuario y contraseña, una vez que se comprueba que el usuario es válido, se presentan una serie de opciones de trabajo, las cuales son:

1. Manejo de Edificios: un catálogo que permite ingresar, modificar o eliminar los edificios disponibles dentro del Campus.

2. Manejo de Aulas: un catalogo que permite ingresar, modificar o eliminar las aulas disponibles dentro del Campus.
3. Manejo de Escuelas: un catalogo que permite ingresar, modificar o eliminar las escuelas existentes en la Universidad.
4. Manejo de Carreras: un catalogo que permite ingresar, modificar o eliminar las carreras que se imparten en la Universidad.
5. Manejo de Pensums: un proceso permite manejar la estructura de los diagramas de precedencia de las diferentes carreras impartidas en la Universidad.
6. Manejo de Ciclos: un catalogo que permite ingresar, modificar, activar y desactivar los ciclos en los que se desea trabajar la programación de horarios y asignación de salones.
7. Manejo de Planes de Estudio: un catalogo que permite ingresar, modificar y eliminar los años de planes de estudio disponibles.
8. Manejo de usuarios: Aquí se crean las cuentas de los usuarios y permisos de los usuarios que accederán a los distintos módulos del SAPH.
9. Reportes: En este se maneja el listado de reportes que se pueden crear desde el módulo.

4.1.2 Modulo de Asignación de Horarios.

Como ya se menciona este modulo del sistema se relaciona directamente con el personal encargado de la realización de horarios de clase de la Universidad Don Bosco.

El módulo también cuenta con un proceso de validación el cual necesita que sean digitados un usuario y contraseña validos para la utilización de la aplicación, una vez que estos han sido validados, se cuentan con las siguientes opciones de trabajo:

1. Manejo de docentes: este permite agregar, modificar o eliminar los datos de un catedrático. Esto con la finalidad de contar con un catalogo de docentes disponibles para impartir clases.

2. Proceso de Asignación de Horarios: este es uno de los dos componentes medulares de todo el SAPH, en este se crean la cantidad de grupos necesarios para cada materia, a cada grupo se le asigna un docente y su respectivo horario.
3. Reportes: en este se maneja el listado de reportes que se pueden crear desde el módulo.

4.1.3 Modulo de Asignación de Salones.

Este modulo del sistema se relaciona con el personal encargado de la asignación de espacio libre de la Universidad Don Bosco.

El módulo al igual que los dos módulos pasados cuenta con un proceso de validación de usuario y contraseña para la utilización de la aplicación, una vez que estos han sido validados, se cuentan con las siguientes opciones de trabajo:

1. Manejo de Reserva de Salones: este permite reservar un salón determinado en una fecha y hora específica. Algo muy importante que hay que resaltar de este componente es que no permite reservar salones que ya han sido reservados en la fecha y hora específica o que están siendo utilizados por una materia de acuerdo a los horarios del ciclo en proceso.
2. Proceso de Asignación de Salones: este es el segundo componente medular del SAPH, en este se asignan los salones a los horarios de cada grupo de materia. Este componente muestra solo las aulas disponibles de cada edificio para el horario específico que se desea resolver.
3. Reportes: en este se maneja el listado de reportes que se pueden crear desde el módulo.

4.1.4 Modulo Web.

Este módulo esta relacionado íntimamente con dos entidades del sistema, las cuales son los docentes, el alumnado y los usuarios externos a la Universidad. Las partes que componen este modulo son los siguientes:

1. Manejo de Tiempo Disponible de Docentes utilizado por los catedráticos, mediante este el catedrático elige el horario que tiene disponible para dar clases, este horario es almacenado en la base de datos para luego ser utilizados en el Modulo de horarios.

2. Consultas Web diseñado para el alumnado y los usuarios externos, este componente provee de distinta información referente a la programación de horarios y asignación de salones mediante una interfaz Web a partir de ciertos datos como lo son la carrera y el ciclo.
3. Reporteador, este componente no es visible a ningún usuario, sin embargo es el componente utilizado por todos los módulos del SAPH para proveer de los reportes necesarios.

4.2 Diagramas de Flujo de Datos.

4.2.1 Diagrama de Flujo de Datos Nivel 0 de la Aplicación.

En la figura 4.2.1.1 se puede observar de manera general como fluirá la información en el sistema; a continuación se presenta que datos son los que fluyen a través del SAPH:

- 1) Los usuarios administradores del sistema se encargan de dar mantenimiento a las variables principales del sistema, como pueden ser edificios, aulas, materias, pensums, escuelas, usuarios entre otros.
- 2) Los usuarios encargados de asignar horarios son los encomendados de armar todas las variables para la programación de horarios.
- 3) Los usuarios encargados de asignar salones son los encomendados de distribuir adecuadamente el espacio disponible de la universidad entre los horarios programados.
- 4) Los usuarios docentes son encomendados de colocar su disponibilidad de tiempo para impartir clases.
- 5) Los usuarios externos hacen peticiones de información de horarios al sistema, esta es proporcionada por los puntos 6 y 7.
- 6) El sistema se encarga de proveer de los horarios finales de acuerdo a la información procesada.
- 7) El sistema se encarga de proveer de distintas consultas referente a horarios de clase de la universidad, incluyendo espacio libre y utilizado.

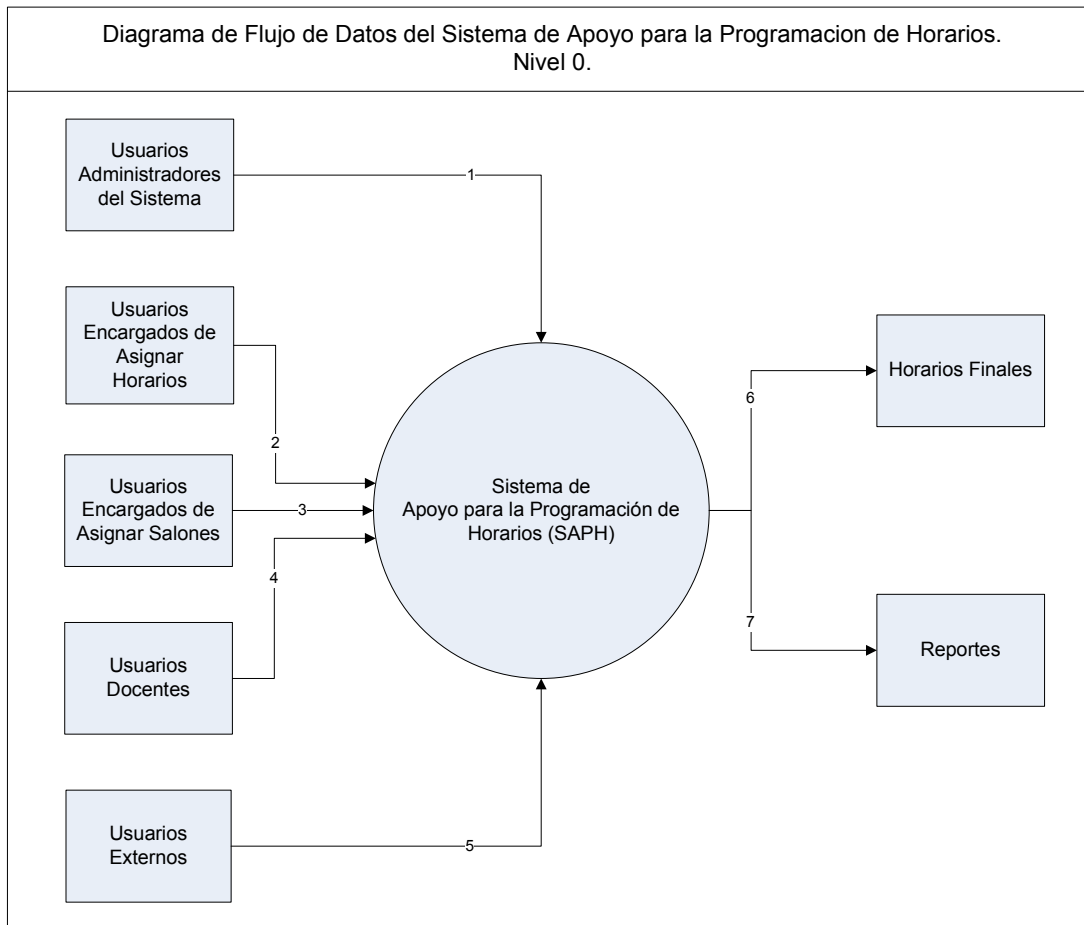


Figura 4.2.1.1: Diagrama DFD de Nivel 0.

4.2.2 Diagrama de Flujo de Datos Nivel 1 de la Aplicación.

En la figura 4.2.2.1 se observa como el sistema se fragmenta en 4 módulos básico que realizan todas las tareas específicas del sistema de apoyo a la programación de horarios. A continuación se detalla usuario-modulo:

- 1) El módulo administrativo es el que recibe datos por parte de los usuarios administrativos, entre los datos de entrada encontramos: edificios, aulas, materias, usuarios, planes, pensum y carreras.
- 2) El módulo de asignación de horarios es el que recibe datos por parte de los usuarios de la programación de horarios, este modulo se encarga de dar mantenimiento a los docentes y de la programación de horarios, este ultimo será detallado mas adelante en el punto 4.2.4.

- 3) El modulo de asignación de salones es el que recibe datos por parte de los usuarios de la asignación de salones, este modulo es encargado de la asignación de espacio libre y reserva de salones, estos procesos serán detallados mas adelante en el punto 4.2.5.
- 4) El modulo Web esta relacionado con los usuarios docentes, usuarios externos, y con todos los módulos que conforman el SAPH, debido a que este es el encargado de proveer de reportes tanto en pantalla como impresos para todos los usuarios. Este modulo solo guarda data de horarios disponibles para impartir clases de los docentes.

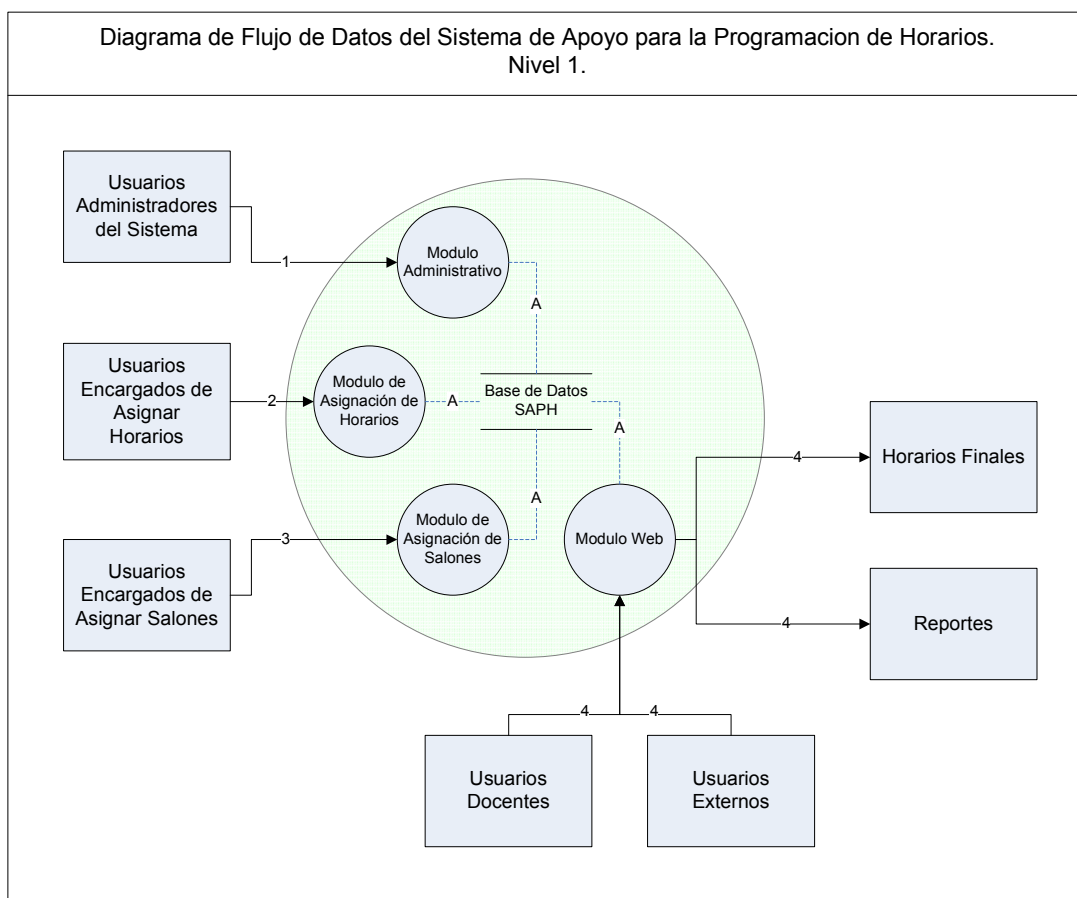


Figura 4.2.2.1: Diagrama DFD de Nivel 1.

- A) Todos los módulos realizan consultas de lectura y escritura a la base de datos que utiliza el SAPH, para más detalles de la base de datos refiérase al punto 4.3.

4.2.3 Diagrama de Flujo de Datos Nivel 2 de la Aplicación (Modulo Administrativo).

En la figura 4.2.3.1 se puede observar todos los componentes que comprenden al Modulo Administrativo y su relación con el Modulo Web (4).

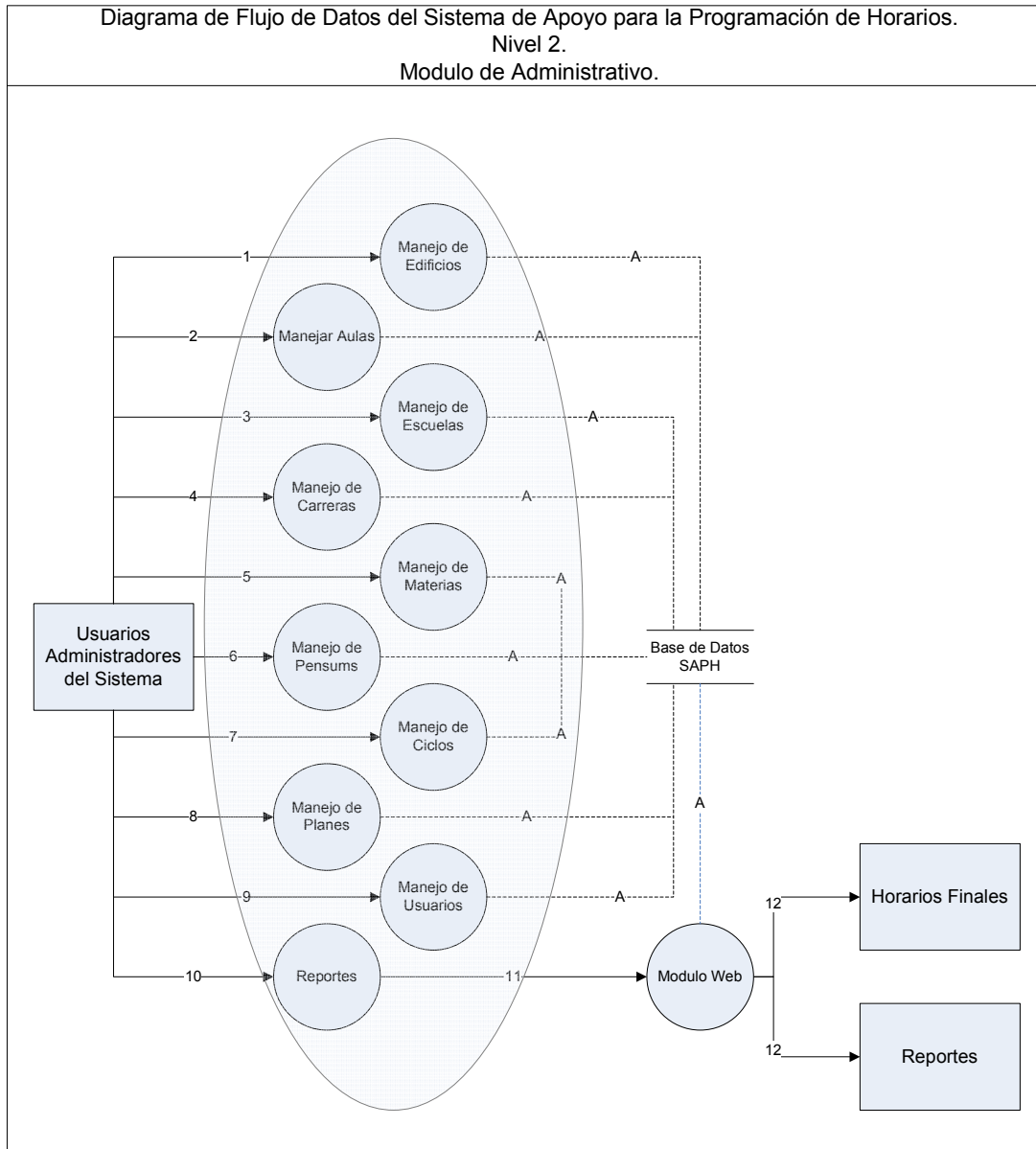


Figura 4.2.3.1: Diagrama DFD de Nivel 2 (Modulo Administrativo).

Cada uno de los componentes que conforman este modulo son básicamente mantenimientos de catálogos, el único que puede ser considerado un proceso es el manejo de Pensums (6), ya que es el encargado de manejar la jerarquía de materias por ciclo de un pensum específico.

4.2.4 Diagrama de Flujo de Datos Nivel 2 de la Aplicación (Modulo de Asignación de Horarios).

En la figura 4.2.4.1 se muestra los componentes del Modulo de Asignación de Horarios y su relación con el Modulo Web (4).

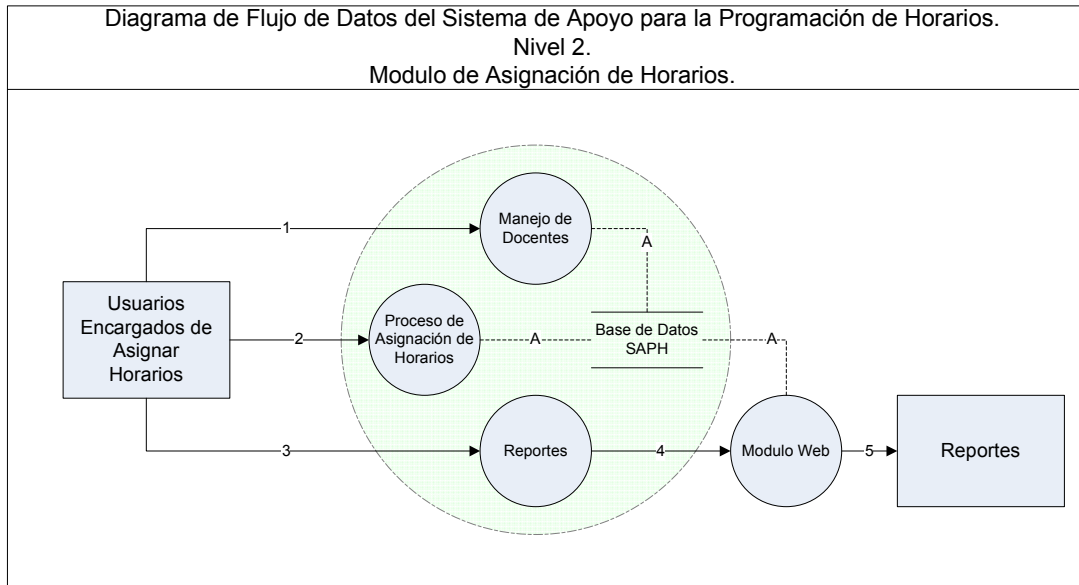


Figura 4.2.4.1: Diagrama DFD de Nivel 2 (Modulo de Asignación de Horarios).

- 1) El Manejo de Docentes, es el encomendado a dar mantenimiento al catalogo de docentes.
- 2) El proceso de Asignación de Horarios, es el que se encarga de programar los horarios, por medio de este se crean los grupos a cada materia, se asignan los docentes y horarios a cada grupo.
- 3) Reportes, se comunica con el Modulo Web para proveer de los reportes necesarios al usuario.

4.2.5 Diagrama de Flujo de Datos Nivel 2 de la Aplicación (Modulo de Asignación de Salones).

En la figura 4.2.5.1 se pueden observar los componentes que conforman al Módulo de Asignación de Salones y su relación con el Modulo Web

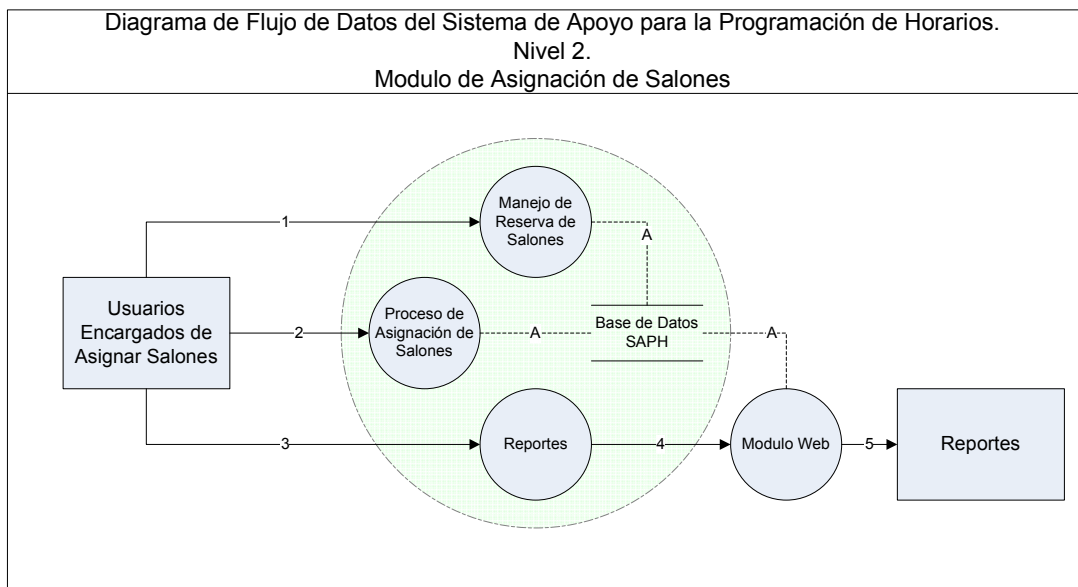


Figura 4.2.5.1: Diagrama DFD de Nivel 2 (Modulo de Asignación de Salones).

- 1) Manejo de Reserva de Salones, encomendado de reservar aulas tomando en cuenta si ya están siendo utilizadas por alguna materia en el ciclo en proceso o si ya ha sido reservada.
- 2) Proceso de Asignación de Salones, encomendado de distribuir el espacio disponible de la Universidad a los horarios de cada grupo de materia.
- 3) Reportes, se comunica con el Modulo Web para proveer de los reportes necesarios al usuario.

4.2.6 Diagrama de Flujo de Datos Nivel 2 de la Aplicación (Modulo Web).

En la figura 4.2.6.1 se muestran los elementos por los que esta formado el Modulo Web. A continuación se describen brevemente cada uno de estos elementos:

- 1) Manejo de Tiempo Disponible del Docente, encargado de capturar el horario para el cual el docente esta dispuesto para impartir clases.
- 2) Reporteador, encargado de recibir peticiones por parte de los usuarios externos y los pertenecientes al resto de módulos del SAPH para la creación de reportes en pantalla y físicos.

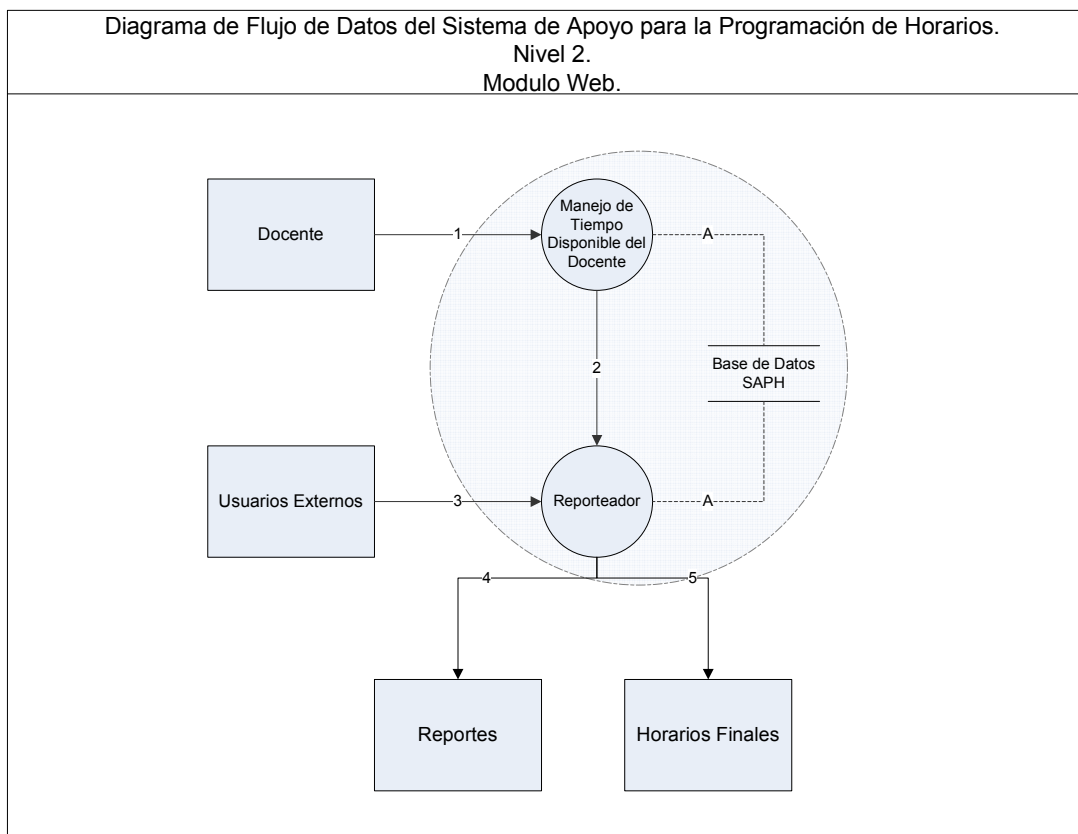


Figura 4.2.6.1: Diagrama DFD de Nivel 2 (Modulo Web).

4.2.7 Diagrama de Flujo de Datos Nivel 3 de la Aplicación (Proceso de Asignación de Horarios).

En la figura 4.2.7.1 se muestran los elementos por los que esta formado el Proceso de Asignación de Horarios.

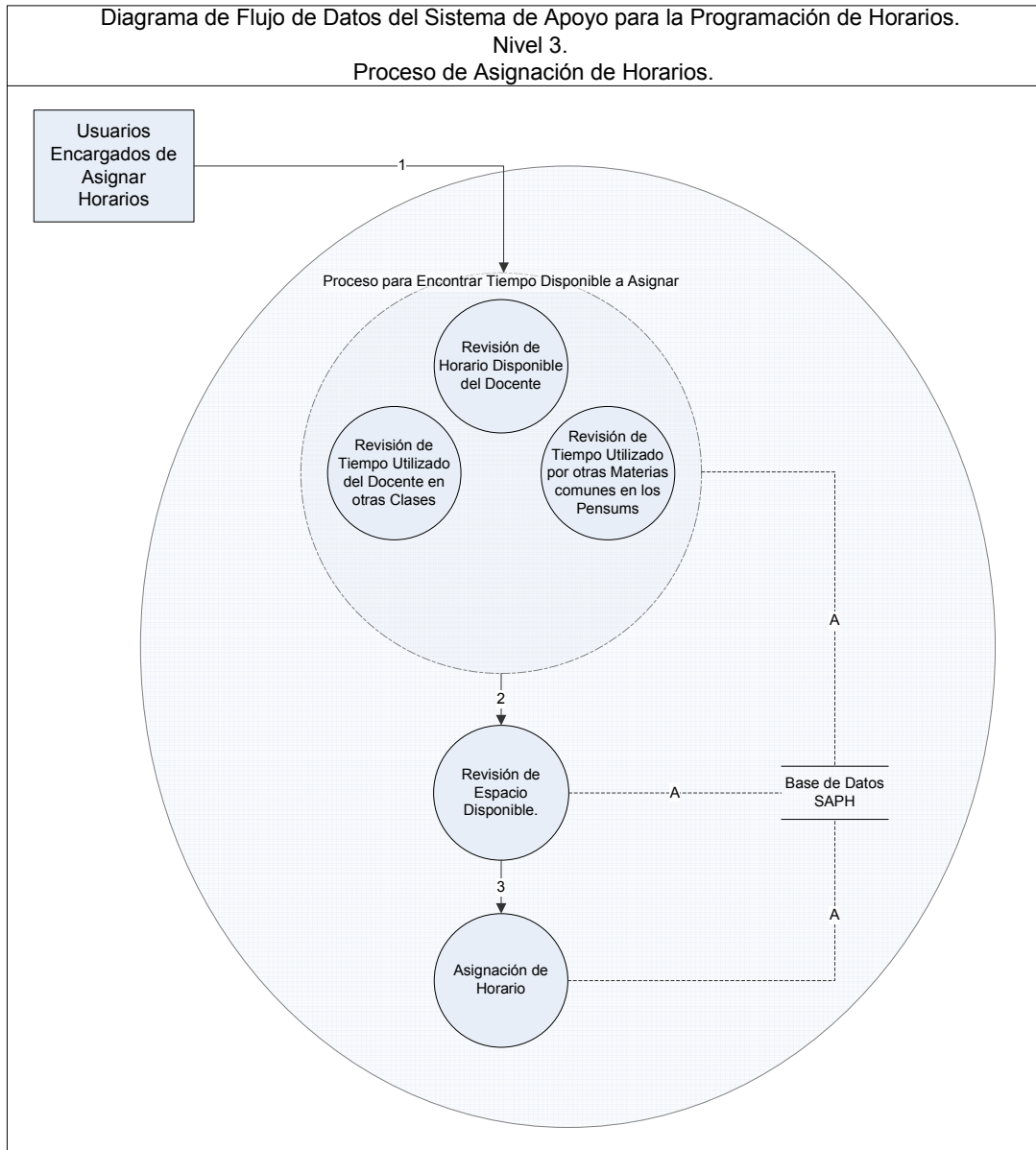


Figura 4.2.7.1: Diagrama DFD de Nivel 3 (Proceso de Asignación de Horarios).

A continuación se describen brevemente los pasos a tomar para que el proceso de asignación de horarios se lleve a cabo:

- 1) El proceso para encontrar tiempo disponible a asignar, esta compuesto por tres subprocesos, los cuales revisan lo siguiente:
 - i. El tiempo disponible del docente.
 - ii. El tiempo utilizado del docente en otros grupos de materias.
 - iii. El tiempo utilizado por otros grupos de materias comunes a los distintos pensums que pertenece la materia.
- 2) Una vez se ha encontrado el tiempo disponible que se puede asignar, se revisa el cupo de espacio libre disponible, si la función da un resultado negativo, no se asigna el horario, en caso contrario pasa al siguiente proceso.
- 3) En este último paso se guarda la información del horario asignado.

4.2.8 Diagrama de Flujo de Datos Nivel 3 de la Aplicación (Proceso de Asignación de Salones).

En la figura 4.2.8.1 se muestran los elementos por los que esta formado el Proceso de Asignación de Salones.

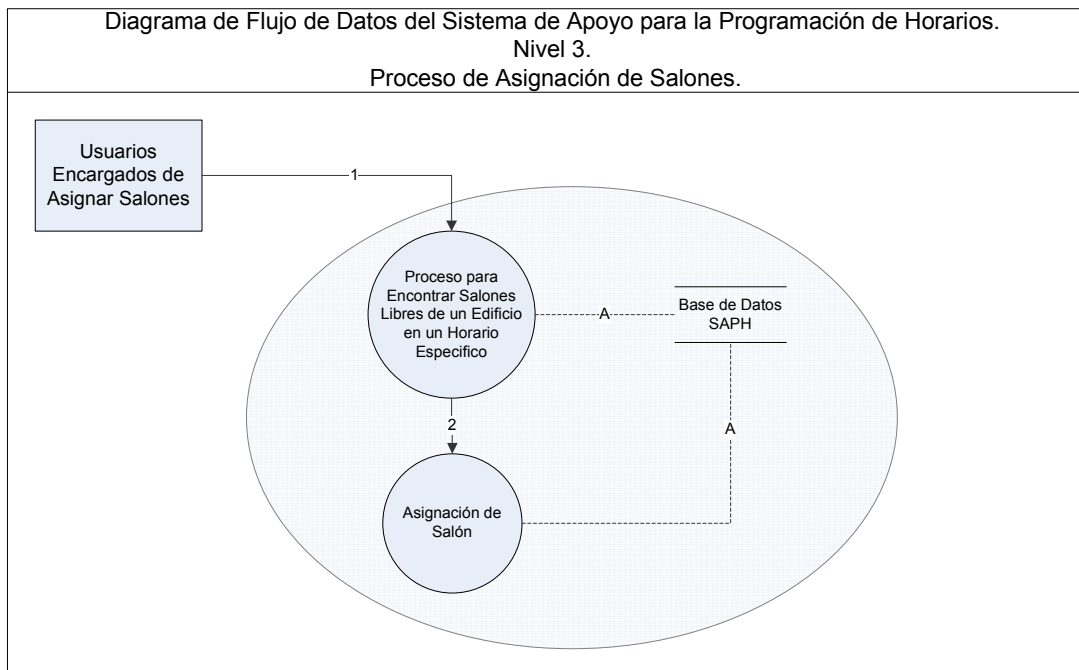


Figura 4.2.8.1: Diagrama DFD de Nivel 3 (Proceso de Asignación de Salones).

A continuación se describen brevemente los pasos necesarios para el proceso de asignación de salones:

- 1) El primer paso es encontrar que salones se encuentran libres en un edificio en un horario específico.
- 2) Luego se asigna el salón específico al horario deseado.

4.2.9 Diagrama de Flujo de Datos Nivel 3 de la Aplicación (Proceso de Reserva de Salones).

En la figura 4.2.9.1 se muestran los elementos por los que esta formado el Proceso de Reserva de Salones.

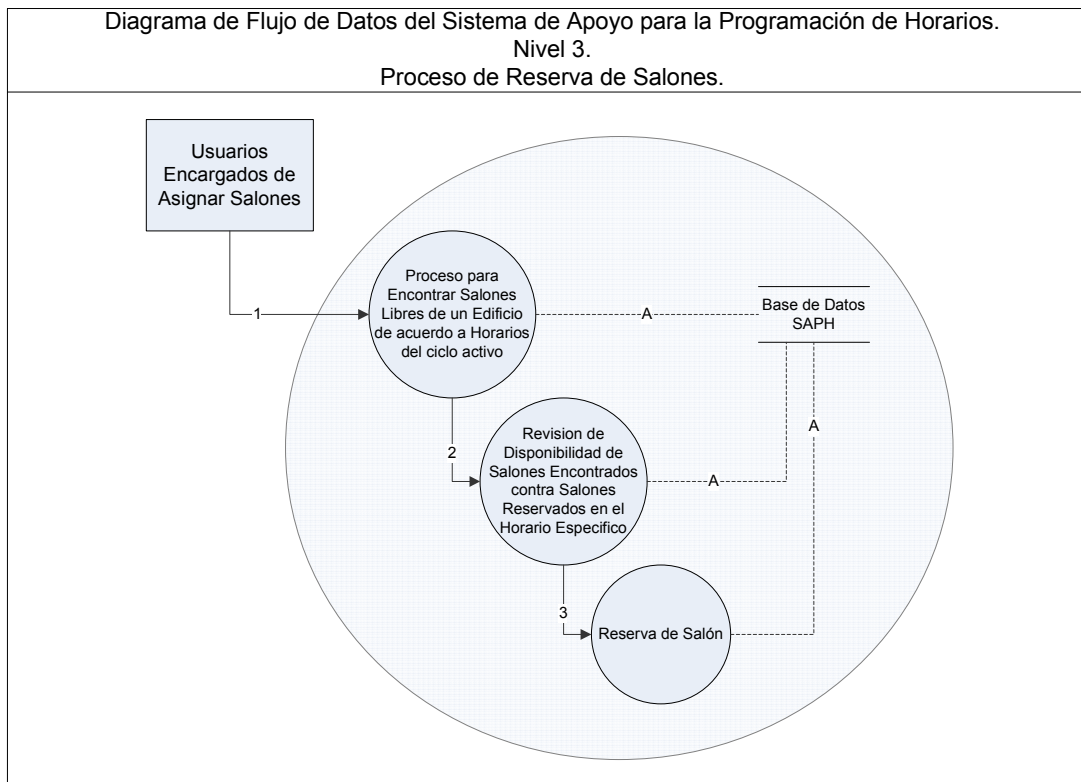


Figura 4.2.9.1: Diagrama DFD de Nivel 3 (Proceso de Reserva de Salones).

A continuación se describen brevemente el proceso de reserva:

- 1) Primero se revisa cuales son los salones libres de un edificio, de acuerdo a los horarios del ciclo en proceso.
- 2) Segundo se revisa cuales son los salones que han sido reservados para la fecha y hora deseada.

- 3) Por ultimo, se muestran solo las aulas libres del edificio, en el horario deseado, se escoge el aula libre que se desee, y se guarda en la base de datos.

4.3 Diagrama Entidad-Relación

En el diagrama entidad-relación se muestran las tablas involucradas en el Sistema de Apoyo para la Programación de Horarios de Clases de la Universidad Don Bosco.

Con el objetivo de facilitar el mantenimiento futuro del sistema, asegurar que este funcione correctamente y lograr encontrar posibles errores relacionados a la base de datos es recomendable usar una estandarización en el nombre de las tablas y campos de la base; por lo que se utilizaron las siguientes reglas para cumplir este punto:

4.3.1 Reglas para el nombramiento de Tablas

Los nombres de las tablas están compuestos por tres caracteres iniciales identificadores de la tabla seguida del nombre de la tabla, separando cada palabra entre por un guión bajo (_); los primeros tres caracteres son seleccionados de la siguiente manera:

Si el nombre de la tabla es de una palabra, los 3 primeros caracteres de esa palabra.

Si el nombre de la tabla es de dos palabras, los 2 primeros caracteres de la primera palabra, más el primer carácter de la segunda palabra.

Si el nombre de la tabla es de tres o mas palabras, se juntaran el primer carácter de las primeras tres palabras.

4.3.2 Reglas para el nombramiento de Campos

Los nombres de los campos están compuestos por tres caracteres iniciales identificadores de la tabla a la que pertenece el campo seguida del nombre del campo.

4.3.3 Diagrama Entidad-Relación

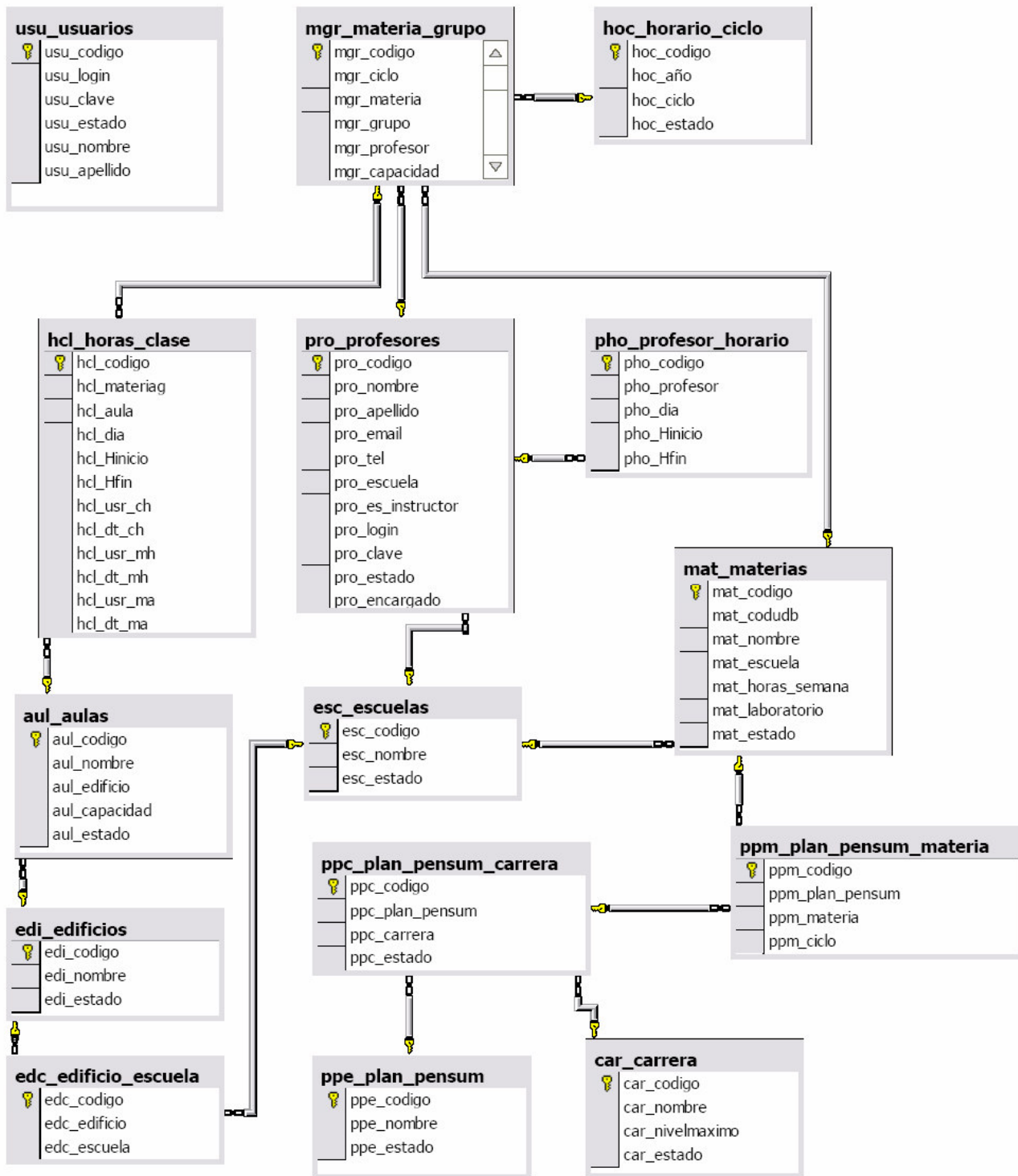


Figura 4.2.3: Diagrama Entidad-Relación de la Base de Datos del Sistema

4.4 Diccionario de Datos.

El diccionario de Datos se presenta en el anexo 1.

4.5 Diagrama de Clases.

En la figura 4.5.1 se muestra el diagrama de clases con la relación que existe entre las diferentes clases que componen el sistema. Las clases son las encargadas de trabajar directamente con la base de datos y realizar los procesos internos del sistema.

A continuación se explicaran de forma sencilla cada una de las clases:

1. Escuelas: Es la clase encargada de llevar toda la información de las escuelas de la universidad y los procesos internos del sistema que estos utilizan.
2. Usuarios: es la encargada de manejar la información de los usuarios administrativos de la Universidad, como un ejemplo de los métodos especiales de esta clase se encuentra el método validar.
3. Materia: La clase materias controla los datos relacionados con las materias de la Universidad y contiene los métodos que le permiten acceder directamente a la base de datos.
4. Profesor: Esta clase se encarga del manejo de profesores y de los procesos relacionados directamente con los docentes de la Universidad. Así, tiene acceso directo a la base y tiene procesos que le permiten relacionarse con otras clases y sacar la información de ellas, como lo son los procesos gruposQueImparte y tiempoDisponible.
5. ProfesorHorario: Es la clase encargada de controlar los horarios disponibles de los profesores, y se utiliza para realizar acciones directamente con la base de datos.
6. PlaPensumMateria: Se encarga de los manejos de pensum, y posee los métodos necesarios para armar los diferentes diagramas de precedencia de las carreras, los cuales son AgregarMateriasPensum y RemoverMateriasPensum.

7. MateriasGrupo: Se encarga de dividir las materias existentes en grupos de clases y asignarlas a los diferentes horarios y aulas disponibles en la Universidad.
8. Aulas y Edificios: Son las clases que se encargan de llevar la información de los espacios físicos de la Universidad. Un ejemplo de los métodos que utilizan estas clases es el proceso DisponibilidadTiempo de la clase Aulas, que regresa todas las horas disponibles de un aula específica.
9. Carrera: Es una clase que se encarga de manejar los datos de las diferentes carreras que se imparten en la Universidad Don Bosco.
10. ColeccionHorasProfesores y ColeccionHorasClase: Son clases intermedias que se utilizan para poder relacionar otras clases

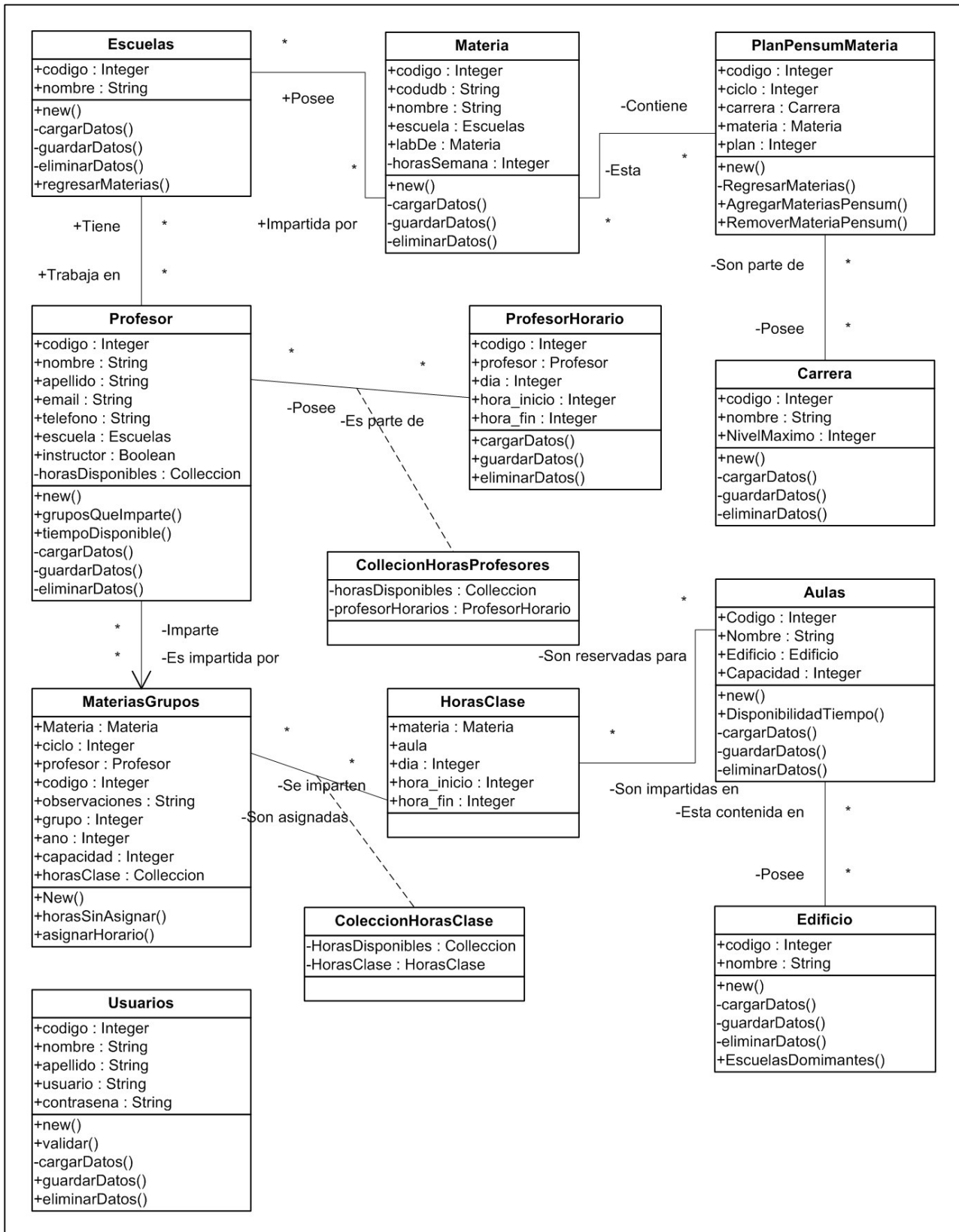


Figura 4.5.1: Diagrama de Clases.

4.6 Esquema del funcionamiento básico del sistema.

En la figura 4.6.1 se muestra la interacción básico de los componentes físicos del sistema, como se observa, los usuarios del modulo de horarios y el modulo administrativo utilizaran una aplicación la cual se encuentra instalada en los distintos edificios de la Universidad Don Bosco, estas aplicaciones se comunicarán directamente con el Servidor de Base de datos; en cambio los usuarios externos y los docentes que utilicen el modulo Web se comunicaran a través del Internet con el sistema, teniendo que pasar por el cortafuegos de la Universidad, comunicándose luego con el Servidor Web, que funcionaria como el servidor de aplicación, y este ultimo consultando al servidor de base de datos.

A continuación se muestra un esquema básico de cómo será el funcionamiento del sistema.

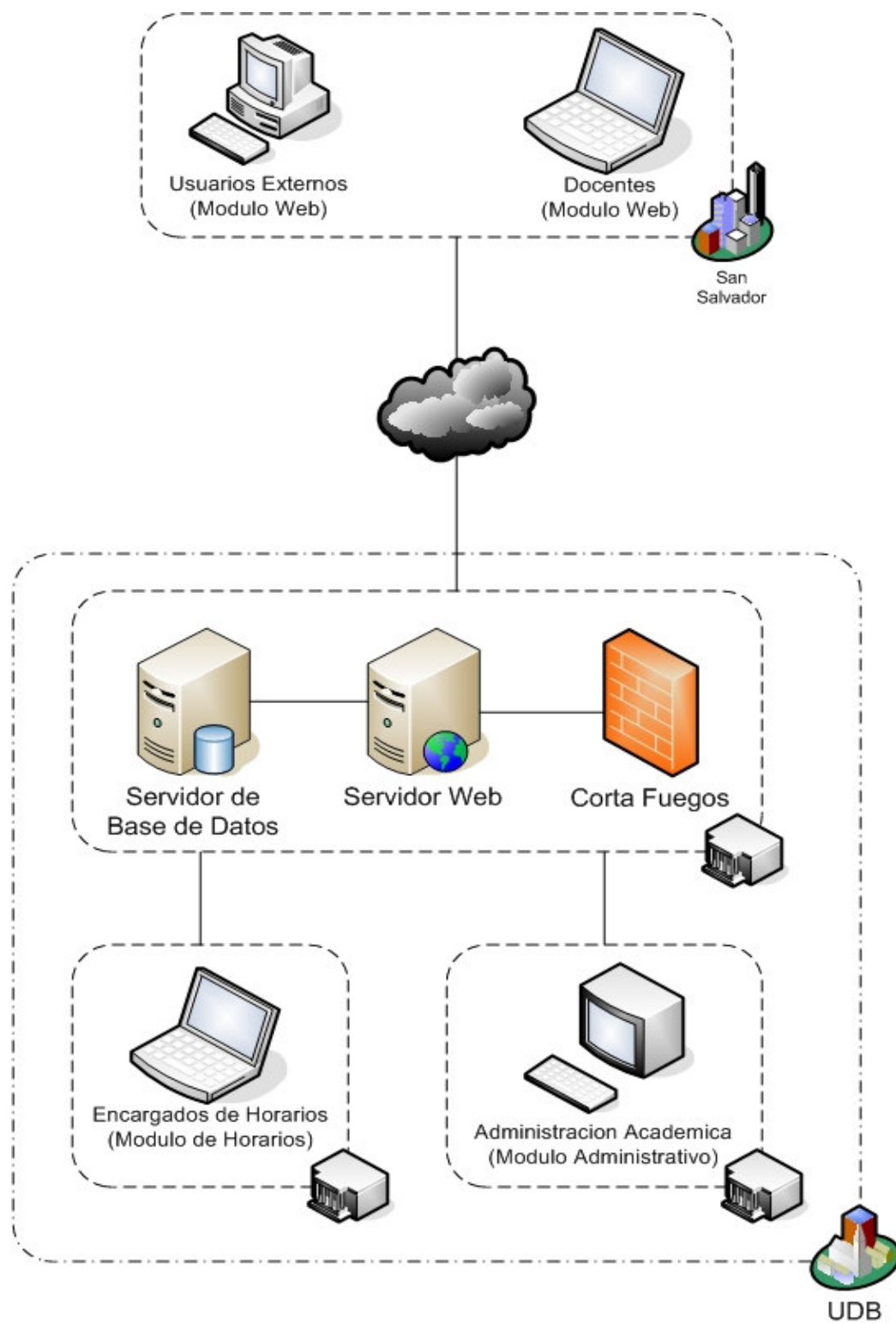


Figura 4.6.1: Diagrama Arquitectónico del Funcionamiento del Sistema

4.7 Diseño de la interfaz de usuario

El diseño de interfaz de usuario es un paso importante en el desarrollo de sistemas, ya que es esta, la que estará en contacto directo con el usuario.

La interfaz del sistema se ha creado con el objetivo de llenar las siguientes características:

- 1) Interfaz Limpia, es decir que sea agradable a la vista.
- 2) Interfaz Amigable, es decir, de fácil entendimiento y uso.

Las interfaces siguen un estándar, para lograr una simple comprensión del sistema y homogeneidad del mismo.

A continuación presentamos algunas de las pantallas diseñadas para el Sistema de Apoyo para la Programación de Horarios de Clases de la Universidad Don Bosco.

Como primer paso en la utilización del sistema, tenemos que autenticar nuestro usuario. Para llevar a cabo este proceso se utiliza la ventana de autenticación de usuario, la cual nos pide un usuario y una contraseña para ingresar al sistema.

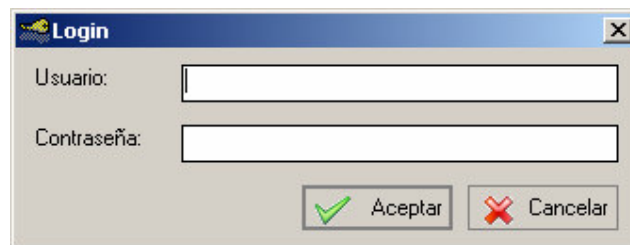


Figura 4.7.1: Pantalla de autenticación de usuario

Una vez se ha autenticado el usuario, se pueden realizar diferentes procesos y mantenimientos.

En la imagen 4.7.2 se muestra la pantalla principal que nos brinda el acceso al modulo administrativo del sistema; desde esta, podemos llamar las ventanas que nos permiten la manipulación de los datos. Esta y todas las pantallas principales constan de 2 controles primarios: una barra de menú y una barra de herramientas. La barra de menú nos da acceso a todas las demás partes del sistema, mientras que la barra de herramientas nos da accesos rápidos a ciertas opciones predefinidas del sistema.

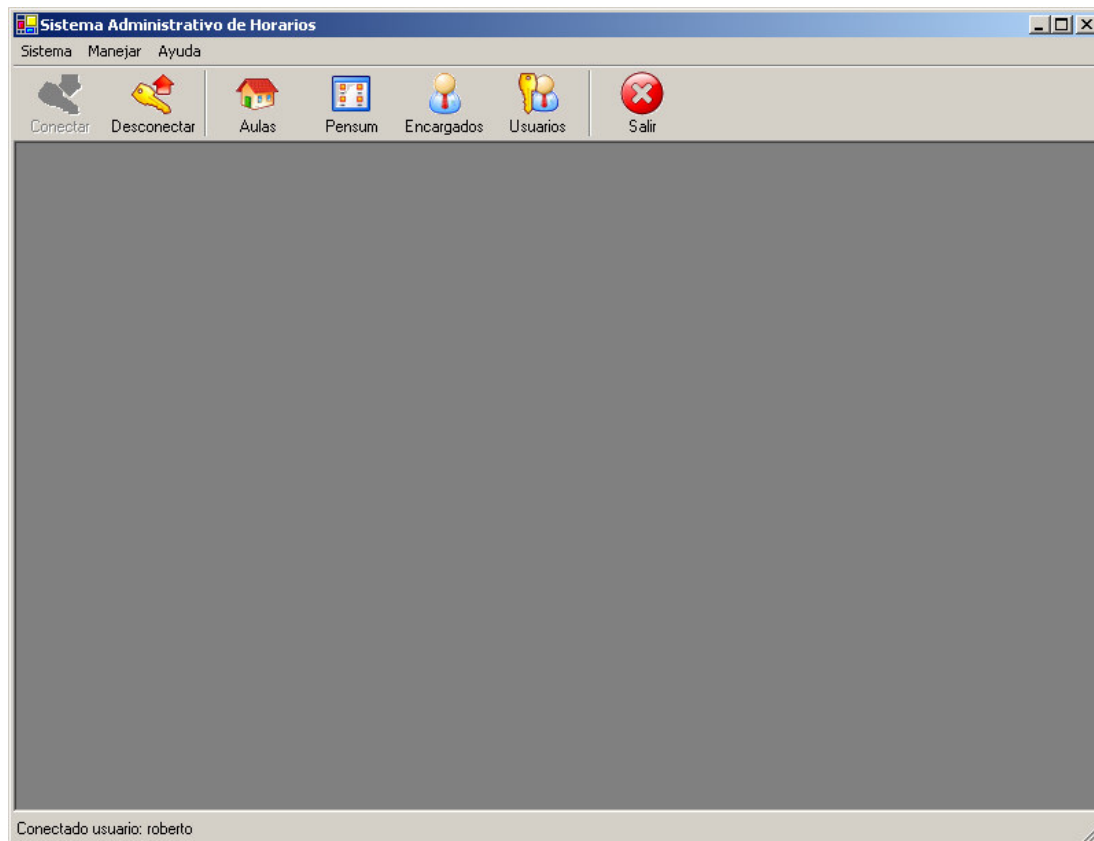


Figura 4.7.2: Pantalla de Inicio del modulo administrativo

Es posible tener acceso a los mantenimientos desde el menú manejar. Entre los diferentes mantenimientos del modulo administrativo tenemos: Aulas, Edificios, Escuelas, Materias, Pensum, Planes, Usuarios.

Todos los mantenimientos son similares a la figura 4.7.3.

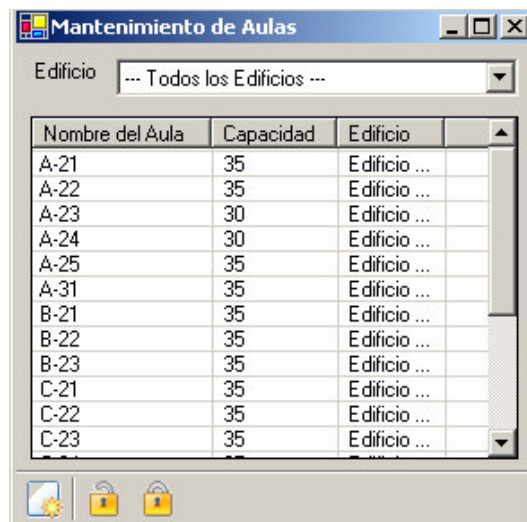


Figura 4.7.3: Diseño básico de las pantallas de mantenimientos.

Los mantenimientos de Aulas, Edificios, Escuelas, Carreras y Planes, siguen el mismo esquema básico de pantallas. Este esquema básico permite la adición, modificación o eliminación de registros.

Los comandos de adición y eliminación de registros se encuentran en la parte final de la ventana de la figura 4.7.3.

Para llevar a cabo la modificación de un registro, damos doble clic sobre el registro a modificar y se nos abrirá una ventana con los datos cargados listos para la modificación como se puede observar en la figura 4.7.4.

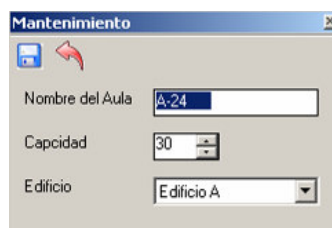


Figura 4.7.4: Pantalla de modificación y adición de registros.

Estas ventanas, al igual que las anteriores, son similares entre la mayoría de mantenimientos del sistema, y cuentan con 2 funciones básicas, guardar datos y restaurar los datos originales, la función de guardado de datos esta representada por un icono de un disco y la función de restaurar datos esta representada por una flecha rojo que apunta hacia atrás, ambas ubicadas en la parte superior de la pantalla (figura 4.7.4).

4.8 Metodología de Programación.

Para la codificación del Sistema de Apoyo para la Programación de Horarios de Clase de la Universidad Don Bosco, se seguirá la metodología descrita a continuación:

El programa será desarrollado siguiendo una metodología basada en objetos, la cual nos permite mayor flexibilidad en la programación, y permite separar de manera total la interfaz grafica del almacenamiento de los datos.

Las tareas de programación se dividen en 3 tareas:

1. Diseño y desarrollo de la Base de Datos: Esta es la parte del sistema que almacena la información que se necesita para llevar a cabo todas las

labores que el sistema será capaz de realizar, en esta parte se detallan de que tipo y de que longitud serán los datos almacenados, que valor tendrán por defecto y si estos aceptan tener un valor nulo.

2. Diseño y desarrollo de clases: Como se menciono anteriormente, el sistema esta basado totalmente en el concepto de programación orientada a objetos, esto permite un código más comprensible. Además, permite la separación de la interfaz de usuario del sistema de Base de Datos, lo cual es muy conveniente, ya que con solo modificar las clases, se pueden agregar o quitar funcionalidades del sistema, sin la necesidad de rediseñar o reprogramar la interfaz de usuario.
3. Diseño y desarrollo de la interfaz de usuario: En esta parte, se diseñan y codifican las pantallas con las cuales el usuario tendrá interacción, aquí también se llevan a cabo todas las validaciones de las entradas.

Bibliografía.

Título: Aprendiendo Visual Basic .NET en 21 lecciones avanzadas

Autor(es): Duncan Mackenzie, Kent Sharkey

Editorial: Sams – Prentice Hall

ISBN: 970-26-0379-X

Año: 2003

Título: Programing Microsoft Visual Basic .NET core reference

Autor: Francesco Balena

Editorial: Wintellect

ISBN: 0-7356-1375-3

Año: 2002

Título: Aprendiendo Microsoft SQL server 7.0 en 21 días

Autor(es): Richard Waymire, Rick Sawtell

Editorial: Sams

ISBN: 970-17-0246-8

Año: 2000

Título: Manual para Elaborar Trabajos de Investigación Documental.

Autor(es): Dra. Guillermina Baena

Editorial: Editores Mexicanos Unidos, S. A.

ISBN: 968-15-0441-0

Año: 1981

Título: Instrumentos de Investigación. Manual para elaborar trabajos de investigación y tesis profesionales.

Autor(es): Dra. Guillermina Baena

Editorial: Editores Mexicanos Unidos, S. A.

ISBN: 968-15-0371-6

Año: 1982

Referencias Electrónicas.

<http://www.itlp.edu.mx/publica/tutoriales/analisis/41.htm>

<http://www.monografias.com/trabajos11/basda/basda.shtml>

http://www.webopedia.com/TERM/D/database_management_system_DBMS.html

<http://www.tomjewett.com/dbdesign/dbdesign.php?page=ddlddl.php&imgsize=medium>

<http://www.udeci.edu.mx/INF/apuntes/analisis/default.asp>

<http://www3.uji.es/~mmarques/f47/apun/node83.html>

<http://www.monografias.com/trabajos5/inso/inso2.shtml>

Glosario

1. **ActiveX:** Lenguaje elaborado por Microsoft con el objeto de crear aplicaciones exportables a la red, con la capacidad de trabajar en cualquier plataforma utilizando los navegadores Web.
2. **ADO:** Abreviación de ActiveX Data Object, un estándar de Microsoft para la manipulación de datos.
3. **API:** Abreviación de Application Program Interface, consiste en un conjunto de funciones y protocolos útiles para construir aplicaciones.
4. **ASP:** Abreviación de Active Server Pages, una especificación de Microsoft para la creación dinámica de paginas Web.
5. **Cardinalidad:** especifica los tipos de relaciones que existen entre las entidades.
6. **Clave Primaria:** Se utilizan esencialmente en las Bases de Datos relacionales, y se utilizan para identificar de manera única cada elemento que se desea almacenar.
7. **Clave Externa:** Al igual que las claves primarias, estas son utilizadas en las Bases de Datos relacionales, e indican las columnas que hacen referencia a claves primarias.
8. **Cliente/Servidor:** Arquitectura de Software que se basa en la idea de una computadora servidor, que brinda sus servicios a las demás computadoras, que son los clientes.
9. **Columnas:** Son las partes de una tabla que contienen los datos, a las columnas se le s debe asignar un tipo de dato y un nombre único.
10. **Clases:** Una Clase es una platilla de un objeto, representa la estructura básica o los planos de construcción de un objeto.
11. **Corta fuegos (Firewall):** Un sistema diseñado para prevenir acceso no autorizado a las redes privadas de computadoras. Puede ser implementado en hardware o software.

12. **DFD:** Diagramas de Flujo de Datos. Herramientas básicas para el análisis de sistemas, ya que permiten representar gráficamente un sistema como una red de procesos y datos conectados entre sí.
13. **DDL (Lenguaje de Definición de Datos):** sentencias utilizadas para crear y modificar las estructuras de las tablas y otros objetos en la base de datos.
14. **DML (Lenguaje de Manipulación de Datos):** sentencias que son utilizadas para trabajar con los datos en las tablas.
15. **DBMS (Database Management System):** conjuntos de rutinas y programas interrelacionados, que se encargan de guardar, modificar y extraer información de una Base de Datos.
16. **DLL: (Dynamic Link Library):** Librería de Enlace Dinámico. Una librería de funciones ejecutables que pueden ser usadas por los programas que se ejecutan en Windows.
17. **DataWarehouse:** Una colección de datos diseñada para ser utilizada en la toma de decisiones administrativas.
18. **E-R (Entidad Relación):** sdjfkfjskldf
19. **Framework:** Una infraestructura de programación utilizada para desarrollar, desplegar y ejecutar aplicaciones.
20. **GUI:** Abreviatura de graphical user interface, se refiere a toda aquella interfaz grafica por medio de la cual un usuario utiliza los recursos de una computadora.
21. **HTML:** Abreviatura de HyperText Markup Language, o Lenguaje de Marcado de Hipertexto, es el lenguaje que se utiliza para la creación de páginas Web.
22. **HTTP:** Abreviatura de HyperText Transfer Protocol, o Protocolo de Transferencia de Hipertexto, es el protocolo que se utiliza para la transferencia de mensajes en el Web. http define como los mensajes son formateados y transmitidos, así como también las acciones que los servidores y navegadores Web deben tomar como respuesta a diferentes comandos.

23. **JavaScript:** Lenguaje propietario de Netscape, que tiene una sintaxis parecida a Java, pero es el único en que son semejantes. JavaScript se escribe directamente dentro del HTML.
24. **ODBC:** Open DataBase Connectivity, un estándar de acceso de base de datos. El objetivo del ODBC es hacer posible el acceso a datos desde cualquier aplicación, independientemente del sistema DBMS que utilice.
25. **OLAP:** Online Analytical Processing, una categoría de herramientas de software que provee análisis a los datos almacenados en bases de datos. OLAP es utilizada en minería de datos.
26. **OS/2:** Un sistema operativo desarrollado por Microsoft e IBM, es compatible con DOS y Windows.
27. **POO:** Programación Orientada a Objetos. Un tipo de programación en la cual los programadores definen el tipo de datos de una estructura de datos, sino que también las operaciones que se pueden aplicar a dicha estructura.
28. **PHP:** Hypertext PreProcessor, un lenguaje de creación de scripts de código fuente abierto que se ejecuta en el servidor, utilizado para crear paginas Web dinámicas.
29. **SQL:** Structured Query Language. Un lenguaje para acceder a Bases de Datos que contiene comandos para manejar los datos
30. **Triggers:** Un procedimiento escrito en SQL que inicia su acción una vez que un evento ocurre, estos son guardados y manejados por el DBMS.

Anexos