

308917



UNIVERSIDAD PANAMERICANA

1
24

ESCUELA DE INGENIERIA

Con estudios incorporados a la Universidad

Nacional Autónoma de México

**" ANÁLISIS Y DISEÑO DE UN SISTEMA AUXILIAR
PARA LA ELABORACIÓN DE HORARIOS
Y ASIGNACIÓN DE AULAS "**

T E S I S

QUE PARA OBTENER EL TÍTULO DE :

INGENIERO MECÁNICO ELECTRICISTA

AREA INDUSTRIAL

PRESENTA :

ADRIAN ARMANDO ACUÑA LÓPEZ

DIRECTOR :

ING. FRANCISCO JAVIER CERVANTES CAMARENA

MEXICO, D.F

1996

**TESIS CON
FALLA DE ORIGEN**

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

**A mi madre,
que por su amor y comprensión
puedo salir siempre adelante.**

**A mi padre,
que por su cariño y apoyo
siempre puedo lograr lo que
me propongo.**

A Silvia,
por toda su comprensión, amor
y cariño que nos brindó siempre.

A Sergio,
por todo el apoyo y protección
que nos brindaste.

A mis hermanos,
Marco Antonio, Sergio, Silvia, Javier,
Alejandro y Miguel,
por todo el amor que existe
entre nosotros.

**A Maricarmen,
por todo el amor que recibo de ti,
por tu comprensión y apoyo.**

**A mi banda,
por todos los buenos momentos que
hemos pasado juntos y los que vienen.**

**A mis amigos,
por su apoyo y el compañerismo que
aprendí de ustedes.**

**Mi agradecimiento a mi director de tesis
Ing. Javier Cervantes
por su valiosa participación
para la realización de este proyecto.**

**A todos ellos,
mil gracias.**

INDICE

INTRODUCCIÓN

CAPÍTULO 1. Entorno organizacional.	1
1.1 ¿Qué es un sistema?	2
1.2 Sistemas de información organizacionales.	3
1.3 Sistemas para el soporte de decisiones.	4
1.4 Determinación de los requerimientos de un sistema.	5
1.5 Requerimientos del sistema.	6
1.6 Características importantes de los sistemas.	6
1.6.1 Productividad.	7
CAPÍTULO 2. Análisis del sistema.	10
2.1 Descripción del problema.	11
2.2 Descripción del proceso.	12
2.3 Metas y objetivos.	15
2.4 Responsabilidad de los usuarios.	16
2.5 Responsabilidades del analista.	17
2.6 Factibilidad del sistema.	19
2.7 Entrevista con el usuario.	19
2.8 Ciclo de vida del proyecto.	20
CAPÍTULO 3. Diseño del sistema.	22
3.1 Diseño.	24
3.2 Formatos de entrada y de salida.	25
3.3 Salidas del sistema.	27
3.4 Diagrama de flujo de datos.	32
3.5 Diccionario de datos.	41
3.6 Especificación de proceso.	44
3.7 Diagrama de Entidad - Relación.	61
3.6 Diagrama de Contexto.	63

CAPÍTULO 4. Pruebas del sistema.	64
4.1 Lenguajes de programación.	66
4.2 Programación.	67
4.3 Estrategias de prueba.	68
4.4 Conversión.	70
4.5 Instalación.	72
4.6 Capacitación.	73
CAPÍTULO 5. Manual de usuario.	75
5.1 Aceleradores.	76
5.2 Teclas de escape.	76
5.3 Menú principal.	77
5.3.1 Maestros.	79
5.3.2 Grupos.	82
5.3.3 Salones.	84
5.3.4 Materias.	87
5.3.5 Horario.	90
5.3.6 Reportes.	95
CONCLUSIONES.	104
BIBLIOGRAFÍA.	108
APÉNDICE A.- Código para Catálogos.	
APÉNDICE B.- Código para Operación.	
APÉNDICE C.- Código para Reportes.	

INTRODUCCION

La idea de realizar un sistema para la elaboración de horarios de clase y asignación de aulas, surge de observar la problemática de la tarea misma.

Siendo ésta una actividad que no se realiza muy a menudo, pero sin duda de gran importancia para la actividad cotidiana dentro de las labores del plantel.

Se observó que se realiza bajo un control un poco problemático y sin poder aprovechar los recursos computacionales, con los que cuenta la Universidad.

Este trabajo está dividido en cinco capítulos y conclusión.

El capítulo número uno comprende el entorno organizacional, en el cuál se describen aspectos generales de los sistemas y la importancia de éstos en el desarrollo de proyectos para diferentes empresas. Así como algunas características que deben considerarse.

El capítulo número dos contempla el análisis del sistema, recalcando algunos errores y propuestas en la elaboración de los horarios de clase, así como un estudio y motivos para reemplazar el método actual de elaboración de horarios.

El capítulo número tres habla del diseño que lleva el sistema, así como las presentaciones en pantalla y reportes del sistema, entendiéndose como reportes, todo aquello que el usuario espera como resultado del proceso de selección de horario de clase y asignación de aulas.

El capítulo número cuatro describe el tipo de pruebas que deben realizarse al sistema, así como características generales del tipo de lenguaje que se utilizó para el desarrollo del sistema de horas. Así como la adaptación de los requerimientos técnicos del programa, en esta sección se discuten y enuncian diferencias que requiere el usuario con el programador, a manera de poder realizar un trabajo más cómodo y más ágil.

En el capítulo cinco se enuncia el manual de usuario, el cuál describe la forma de operar el sistema, es importante recalcar que esta sección procura establecer un mayor vínculo con el usuario y a la vez enseñar al mismo a utilizar un programa con un lenguaje de cómputo desconocido, pero fácil de usar.

En las conclusiones se recalca la importancia de apoyarse en sistemas de cómputo, para la elaboración de tareas complejas, y de tanta repercusión en la Universidad.

1.- ENTORNO ORGANIZACIONAL

1.1.- ¿QUE ES UN SISTEMA?

En el sentido más amplio, un sistema es un conjunto de componentes que se interrelacionan entre sí para lograr un objetivo común. Nuestra sociedad está rodeada de sistemas. Por ejemplo, cualquier persona experimenta sensaciones físicas gracias a un complejo sistema nervioso formado por el cerebro, la médula espinal, los nervios y las células sensoriales especializadas que se encuentran debajo de la piel; estos elementos funcionan en conjunto para hacer que el sujeto experimente sensaciones de frío, calor, comezón, etc.

Las personas se comunican con el lenguaje, que es un sistema muy desarrollado formado por palabras y símbolos que tienen significado para el que habla y para quienes lo escuchan. Así mismo, las personas viven en un sistema económico en el que se intercambian bienes y servicios por otros de valor comparable y en el que, al menos en teoría, los participantes obtienen un beneficio en el intercambio.

Una organización es un sistema. Sus componentes tales como mercadotecnia, manufactura, ventas, investigación, embarques, contabilidad y personal, trabajan juntos para crear utilidades que benefician tanto a los empleados como a los accionistas de la compañía. Cada uno de estos componentes es a su vez un sistema. El departamento de contabilidad, por ejemplo, quizá esté formado por cuentas por pagar, cuentas por cobrar, facturación y auditoría entre otras.

Todo sistema organizacional depende , en mayor o menor medida, de una entidad abstracta denominada sistema de información. Este sistema es el medio por el cual los datos fluyen de una persona o departamento a otros y puede ser cualquier cosa, desde la comunicación interna entre los diferentes componentes de la organización y líneas telefónicas hasta sistemas de cómputo que generen reportes periódicos para varios usuarios. Los sistemas de información proporcionan servicio a todos los demás sistemas de una organización y enlazan todos sus componentes en forma tal que éstos trabajen con la eficiencia para alcanzar el mismo objetivo.

1.2.- SISTEMAS DE INFORMACIÓN ORGANIZACIONALES.

Las finalidades de los sistemas de información, como las de cualquier otro sistema dentro de una organización, son procesar entradas, mantener archivos de datos relacionados con la organización y producir información, reportes y otras salidas.

Los sistemas de información están formados por subsistemas que incluyen hardware, software, medios de almacenamiento de datos para archivos y bases de datos. Dado que los sistemas de información dan soporte a los demás

sistemas de información, primero se tiene que estudiar el sistema organizacional como un todo para entonces detallar sus sistemas de información.

1.3.- SISTEMAS PARA EL SOPORTE DE DECISIONES.

No todas las decisiones son de naturaleza recurrente. Algunas se presentan sólo una vez o escasamente. Los sistemas para el soporte de decisiones ayudan a los directivos que deben tomar decisiones no muy estructuradas. Una decisión se considera no estructurada si no existen procedimientos claros para tomarla y tampoco es posible identificar, con anticipación, todos los factores que deben considerarse en la decisión.

Un factor clave en el uso de estos sistemas es determinar la información necesaria. En situaciones bien estructuradas es posible identificar esta información con anticipación, pero en un ambiente no estructurado resulta difícil hacerlo.

1.4.- DETERMINACIÓN DE LOS REQUERIMIENTOS DE UN SISTEMA.

El aspecto fundamental del análisis de sistemas es comprender todas las facetas importantes de la parte de la empresa que se encuentra bajo estudio. Para lo cual se debe presentar una serie de preguntas como:

- ¿Qué es lo que se hace actualmente?
- ¿Quién lo hace?
- ¿Cómo se hace?
- ¿Con qué frecuencia se presenta?
- ¿Qué tan grande es el volumen de transacciones?
- ¿Cuál es el grado de eficiencia del trabajo realizado?
- Si existe un problema, ¿qué tan serio es? ¿Cuál es la causa que lo origina?

Conforme se contestan estas preguntas, se estudian los datos sobre requerimientos con la finalidad de identificar las características que debe tener el sistema.

1.5.- REQUERIMIENTOS DEL SISTEMA.

La determinación de requerimientos es el estudio de un sistema para conocer cómo se trabaja y dónde es necesario efectuar mejoras.

Un requerimiento es una característica que debe incluirse en un nuevo sistema y su investigación depende de las técnicas para encontrar datos. En muchos casos el objetivo más importante de este proceso es aprender cómo se manejan las excepciones, que siempre existen, dentro de la organización.

1.6.- CARACTERÍSTICAS IMPORTANTES DE LOS SISTEMAS.

La finalidad de un sistema es la razón de su existencia. Existe un sistema legislativo, por ejemplo, para estudiar los problemas que enfrentan los ciudadanos y aprobar la legislación que los resuelva. El sistema de encendido de un automóvil tiene el claro propósito de quemar el combustible para crear la energía que emplean los demás sistemas del automóvil.

Para alcanzar sus objetivos, los sistemas interactúan con su medio ambiente, el cual está formado por todos los objetos que se encuentran fuera de las fronteras de los sistemas. Los sistemas que interactúan con su medio ambiente (reciben entradas y producen salidas) se denominan sistemas abiertos.

1.6.1.- LA PRODUCTIVIDAD.

La meta principal de un sistema es lograr la productividad. La sociedad y los negocios modernos parecen exigir cada vez más: más sistemas, más complejidad y más rápido.

El problema de la productividad ha existido por muchos años en la profesión del desarrollo de sistemas, y muchas organizaciones están buscando de manera agresiva la forma de reducir radicalmente su retraso en las aplicaciones y disminuir el tiempo requerido para desarrollar un nuevo sistema.

Para lo anterior se han desarrollado algunas técnicas, tales como nuevos lenguajes de programación, especialización de analistas y programadores, mantenimiento, disciplinas de ingeniería de software.

Muchos de estos enfoques de productividad pueden usarse en conjunto, pues comprenden conceptos y técnicas complementarias. Individualmente, cada una de las técnicas antes mencionadas puede llevar a un 10 a 50 por ciento de mejoría; tomados en conjunto, fácilmente pueden doblar la productividad de la organización y, en casos especiales, tal vez incluso mejorar la productividad por un factor de 10.

El hecho es que no se puede elaborar un sistema que dé buen resultado, de alta calidad si no se sabe precisamente, y con suficiente detalle, que se supone que debe poder hacer, esto nos hace pensar que el análisis de un sistema constituye una parte esencial para el éxito del proyecto.

Pueden existir errores que disminuyen la confiabilidad del sistema. En muchos casos, nadie está muy seguro respecto a cuántos errores tiene un sistema, pues

- 1) Algunos errores jamás se encuentran antes de que caduque el sistema y,

2) El proceso de documentación y registro de errores es tan descuidado que la mitad de los errores encontrados no se reportan, aún dentro de la organización misma de desarrollo de sistemas.

2. ANÁLISIS DEL SISTEMA

2.1.- DESCRIPCIÓN DEL PROBLEMA.

Dentro de las organizaciones, el análisis y diseño de sistemas se refiere al proceso de examinar la situación de una empresa con el propósito de mejorarla con métodos y procedimientos más adecuados.

El análisis de sistemas, por consiguiente, es el proceso de clasificación e interpretación de hechos, diagnóstico de problemas y empleo de información para recomendar mejoras al sistema. En nuestro caso, se pretende atacar el problema, de la asignación de materias y aulas de trabajo, para el alumnado de la Universidad Panamericana con enfoque a la Escuela de Ingeniería.

Esto es debido al alto volumen de información que se maneja para lograr dicho objetivo. Este ha sido el problema al cual se enfrentan, cada inicio de semestre, los responsables en la elaboración de los horarios de clase.

Para enfrentar esta situación, existe un representante de cada Escuela que se encarga de organizar, planear y definir los horarios de clase.

2.2 DESCRIPCIÓN DEL PROCESO.

Para la elaboración del horario de clase, particularmente en la Escuela de Ingeniería, se lleva a cabo una serie de pasos, que consisten primeramente, en el establecer la disponibilidad de las aulas; determinando así, los salones que se pueden ocupar a lo largo de cierto período, ya sea por semestre o por año, así como en las horas en las que éstos pueden ser ocupados. Al mismo tiempo se discute la capacidad de las aulas requeridas.

Esta disposición queda sujeta a cambios y a "préstamos" de aulas específicas, ya sea parcial o permanentemente.

Se cuenta previamente con un programa de materias asignadas a cada semestre.

Estas se encuentran clasificadas en áreas. Los encargados de cada área son los responsables de recopilar la información de los distintos maestros y las horas que cuentan a la semana para impartir las diferentes materias.

Una vez recopilada esta información, cada representante de área informa al coordinador para comenzar a elaborar los horarios de clase.

El coordinador se auxilia de tres hojas de cálculo bajo lenguaje Excel. En la primera de ellas se cuenta con una lista de diferentes materias, separadas por áreas. Se determina el grupo y el profesor que impartirá la materia, tomando en cuenta un horario disponible y para ciertos casos un horario de preferencia para los profesores.

Se procede a la captura del horario a lo largo de la semana, auxiliado simultáneamente, por un archivo que le proporciona la información de salones desocupados y las horas en que se puede disponer de ellos.

Una vez tomada la decisión, se marca inmediatamente en ambos archivos el día, hora y salón a ocupar para cada grupo, para la materia en cuestión. Una tercera hoja de cálculo se habilita y se registra al igual la hora, salón y materia para cada profesor.

Existen algunos casos en donde se da cierta prioridad a asignar directamente salón, hora y día a determinados maestros. Esta se registra en primer término, para partir de esta selección para los otros maestros.

Este proceso se va complicando conforme continúa la captura. Hay que considerar que el flujo de información es muy grande y en ciertos casos complicada. El hecho de registrar simultáneamente en tres archivos distintos se presta a cometer errores de asignación y puede traer como consecuencia la repartición errónea de los salones de clase.

También existen problemas en cuanto a la disponibilidad de los profesores que puede cambiar de un momento a otro, y la modificación del horario se torna difícil de manejar. Hay ocasiones en donde se ve obligado el coordinador a pedir autorización, para ocupar un salón que no corresponde a la Escuela de Ingeniería; se lleva a cabo un "préstamo" de aulas para cubrir esos problemas, cuando esto es factible.

El coordinador también se encarga de asignar salones en forma parcial, ya sea por cierto día o períodos más largos. Así como para eventos que la Universidad organiza, en donde la demanda de salones resulta más crítica.

2.3.- METAS Y OBJETIVOS.

Lo que se busca, con la implementación de un sistema para la elaboración de los horarios de clase, es agilizar el proceso de selección de aulas, contando con una herramienta que pretende ser más flexible que el método tradicional y de igual forma con reportes de salida automáticos, y en el momento que se requieran, para evitar la elaboración manual de los mismos.

Se tendrá la posibilidad de disponer la información necesaria, para que en el momento que se requiera, se concedan salones fuera del período de asignación oficial de aulas, evitando hacer actualizaciones en otros archivos, para así, agilizar y optimizar el tiempo que lleva esta actividad.

Con lo anterior se resume como objetivo principal, proporcionar al usuario un medio que facilite, y a la vez mejore, la asignación de aulas y elaboración de horarios, realizando esto con el mínimo de tiempo y captura de datos.

2.4.- RESPONSABILIDADES DE LOS USUARIOS.

Existe un documento denominado especificaciones (del cual se detallará más adelante), por el cual el usuario podrá verificar que todas sus necesidades estén contempladas. Así mismo, podrá darse cuenta si el analista comprendió en forma correcta la información que le fue proporcionada.

El usuario podrá hacer una revisión y retomar el análisis para ver si hubo alguna omisión, ya sea por parte del analista o da él mismo.

También debe verificar que todas las validaciones a los campos estén incluidas y que las mismas estén correctas.

A través de las especificaciones podrá darse cuenta de que el sistema tenga el flujo lógico de operatividad de las transacciones cotidianas.

Las especificaciones son el primer contacto del usuario con el sistema, y por las cuales podrá familiarizarse con los términos utilizados en el ámbito informático y darse una idea de lo que será el sistema en el momento que sea liberado.

Fijar el compromiso del usuario ante el desarrollador de que la información

proporcionada es la correcta y está completa.

En resumen se tiene:

- Comprender el problema que va a ser abordado por la aplicación.
- Conocer los datos necesarios para abordar el problema.
- Saber cómo utilizar el software.
- Saber cómo utilizar la computadora.
- Adherirse a ciertos lineamientos o estándares.

2.6.- RESPONSABILIDADES DEL ANALISTA.

Una de las responsabilidades principales del analista o desarrollador es permitir las modificaciones pertinentes de acuerdo a las correcciones de la información proporcionadas por el usuario antes de empezar el desarrollo.

Las correcciones serán consecuencia de corroborar validaciones y conceptos en general. Vale hacer mención que las especificaciones servirán para detectar omisiones que deberán incluirse.

El desarrollador debe delimitar el alcance del sistema, que se determine hasta dónde llega su responsabilidad ante el usuario. Esto dará al desarrollador una idea más o menos precisa del tiempo que se necesitará para desarrollar el sistema y podrá comprometerse con el usuario en una fecha de liberación.

Las especificaciones servirán de antecedente para el desarrollo del manual del sistema (vea capítulo 5).

En resumen se tiene:

- Reconocer los beneficios de las aplicaciones desarrolladas por los usuarios.
- Traducir las necesidades generales de datos en especificaciones para éstos y en requerimientos de procesamiento.
- Proporcionar programas de entrenamiento.
- Proporcionar ayuda en la detección y corrección de errores.

2.6.- FACTIBILIDAD DEL SISTEMA.

En cuanto al Hardware que implica el sistema propuesto, se puede utilizar cualquier computadora personal.

El Software que se utiliza es un lenguaje llamado Object View. Es un lenguaje gráfico y flexible que nos permite manejar bases de datos y se considera apropiado para el objetivo que se ha planteado anteriormente.

2.7.- ENTREVISTA CON EL USUARIO.

El usuario es el participante más importante en un sistema. En este caso estamos hablando de un usuario operacional, ya que es éste el que hace funcionar el sistema y tiene una visión física del mismo.

Se utiliza la entrevista para reunir información proveniente, preferentemente del o los usuarios, para encontrar hechos. La entrevista es fundamental para el desarrollo de un proyecto, y cuando éste es complejo, se convierte en una actividad difícil y requiere involucrar a otros profesionales tales como: expertos en

la industria, psicólogos, negociadores del sindicato, etc. Para nuestro caso resulta una actividad informal, y no requiere de mucho tiempo.

Los aspectos que deberán ser tratados serán los siguientes:

- Reunir información acerca del sistema actual y enunciar los requerimientos del nuevo.
- Discutir las entradas y las salidas del sistema, así como la lógica de operación.

En muchos proyectos, se ha observado que los problemas más difíciles no involucran hardware ni software sino "genteware".

2.8.- CICLO DE VIDA DEL PROYECTO.

Como pudiera esperarse, las organizaciones pequeñas de proceso de datos tienden a ser relativamente informales: los proyectos de desarrollo de sistemas nacen de conversaciones entre el usuario y el administrador del proyecto, y el proyecto procede desde el análisis hasta el diseño e implantación sin mayor alboroto.

Recientemente, sin embargo, ha empezado a cambiar el enfoque que se le da al desarrollo de sistemas. Cada vez son más las organizaciones grandes y pequeñas que están adoptando un ciclo de vida uniforme y único para sus proyectos. Esto a veces se conoce como el plan del proyecto.

Así, los tres objetivos principales que debe llevar un ciclo de vida del proyecto son:

1. Definir las actividades a llevarse a cabo en un proyecto de desarrollo de sistemas.
2. Lograr congruencia entre la multitud de proyectos de desarrollo de sistemas en una misma organización.
3. Proporcionar puntos de control y revisión administrativos de las decisiones sobre continuar o no con un proyecto.

3. DISEÑO DEL SISTEMA.

Dentro de cada labor, la actividad de diseño se dedica a la creación de una jerarquía apropiada de módulos de programas y de interfaces entre ellos para implantar la especificación creada en el análisis de diseño. Además, la actividad de diseño se ocupa de la transformación de modelos de datos de entidad-relación en un diseño de bases de datos.

Parte de esta actividad le interesará como analista: el desarrollo de algo conocido como el modelo de implantación del usuario. Este modelo describe los asuntos relacionados con la implantación que le importan al usuario al grado de que no se los quiere confiar a los diseñadores o programadores.

Los asuntos principales que suelen preocupar al usuario son aquellos relacionados con la especificación de la frontera humano-máquina y la especificación de la interfaz hombre-máquina. Esa frontera separa las partes del modelo esencial que llevará a cabo una persona, de las partes que se implantarán en una o más computadoras. De manera similar, la interfaz hombre-máquina es una descripción del formato y de la secuencia de entradas que los usuarios proporcionan a la computadora (por ejemplo el diseño de pantallas y el diálogo en línea entre el usuario y la computadora), además del formato y la secuencia de salidas -o productos- que la computadora proporciona al usuario.

3.1.- DISEÑO.

La actividad de diseño involucra el desarrollo de una serie de modelos, de forma similar a la que el analista desarrolla modelos durante la fase de análisis de un proyecto.

Aunque éstos modelos son propios de sistemas relativamente grandes, es conveniente mencionar cómo aplica para el sistema de horarios propuestos.

Para el sistema de horarios se determina que el modelo de procesador será para "computadora principal", ya que todas las actividades y trabajos que implica el sistema pueden desarrollarse en una sola computadora personal.

Para la etapa de diseño de un sistema, el desarrollador no debe pasar por alto los siguientes factores: Costo, eficiencia y seguridad.

En el rubro del costo, consideramos suficiente el equipo con el que se cuenta para el desarrollo del mismo. Se puede mencionar que no representa un problema para la elaboración del sistema.

En cuanto a la eficiencia, el programador debe de realizar un trabajo tal que, los tiempos de respuesta no sean muy grandes e innecesarios, por lo tanto se debe elegir un buen manejador de base de datos, para lograr este fin.

Para la seguridad se debe tomar en cuenta que el sistema operará bajo manejo de claves de usuario y contraseñas, para no permitir el acceso al sistema.

En el diseño se afina la calidad global del sistema.

3.2.- FORMATOS DE ENTRADA Y SALIDA.

Para poder estructurar una pantalla de entrada de datos, se deben considerar varios aspectos, tales como:

-El sistema debe pedir entradas y producir salidas en forma consistente.

-Pida información como una secuencia lógica.

-Hacer obvio al usuario el tipo de error que ha cometido, y dónde.

-Distinga entre edición de campos y ediciones de pantalla.

-Hacer la edición y la revisión de errores dependientes del usuario.

-Permita que el usuario pueda:

a) Cancelar parte de la transacción.

b) Cancelarla toda.

-Proporcione un mecanismo de "ayuda" conveniente.

-Si el sistema está realizando un proceso largo, despliegue un mensaje al usuario para que no crea que se detuvo.

El diseño de los formatos de entrada y de salida de un sistema tradicionalmente se conoce como el diseño de formas, pues la mayor parte de los sistemas de los años 60 y 70 requerían que el usuario codificara las entradas en forma de papel que luego se transcribían a tarjetas perforadas antes de ingresar al sistema de cómputo por lotes. Pero incluso en los sistemas en línea actuales se requiere un poco de diseño de formas.

Para el sistema de horarios propuesto, se tomó la política de implementar el mismo formato con el que se trabaja actualmente. Todo esto con el fin de no confundir al usuario, complementando así la nueva forma de operar con reportes anteriores.

Aunque hay muchos estilos diferentes para las formas, todas deben contener la siguiente información básica:

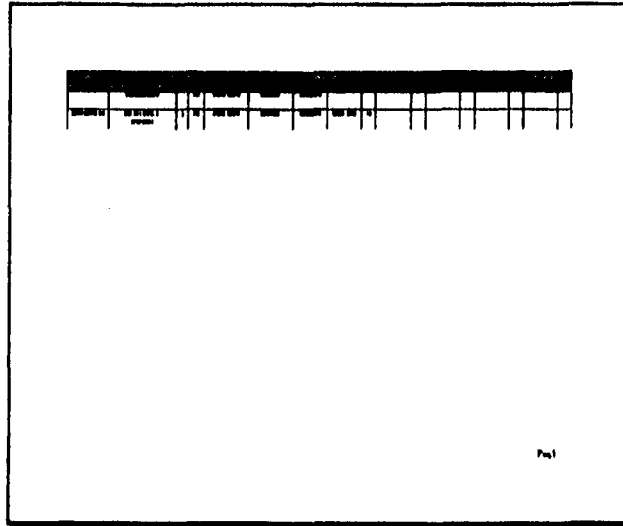
- Título
- Instrucciones
- Cuerpo

3.3.- SALIDAS DEL SISTEMA.

Para el diseño de salidas del sistema, se consideran todo tipo de requerimientos por parte del usuario, partiendo de esto, se diseña un formato de reportes que pretenden dar al usuario la información que necesita.

Se muestran a continuación los reportes de salida del sistema:

Reporte planta :



Reporte para horario de profesores :

Calle parte: [HORA] [LUGAR]

[Código de identificación]

INSTITUCIÓN
GRUPO A
COORDINADOR DEL PROGRAMA

LIBRE	MATEMÁTICA	FÍSICA	QUÍMICA	BIOLÓGICA
10:00 - 11:00				
11:00 - 12:00				
12:00 - 13:00				
13:00 - 14:00				
14:00 - 15:00				
15:00 - 16:00				
16:00 - 17:00				

3.4.- DÍAGRAMA DE FLUJO DE DATOS.

El diagrama de flujo de datos es una herramienta sencilla, pero poderosa para modelar las funciones de un sistema.

Se puede apreciar en el diagrama 3.1 el diagrama de flujo de datos general, describiendo en éste los procesos principales, así como las entidades que intervienen. Los diagramas siguientes son las descripciones de los procesos descritos en este diagrama.

DIAGRAMA 3.1.- DIAGRAMA DE FLUJO DE DATOS

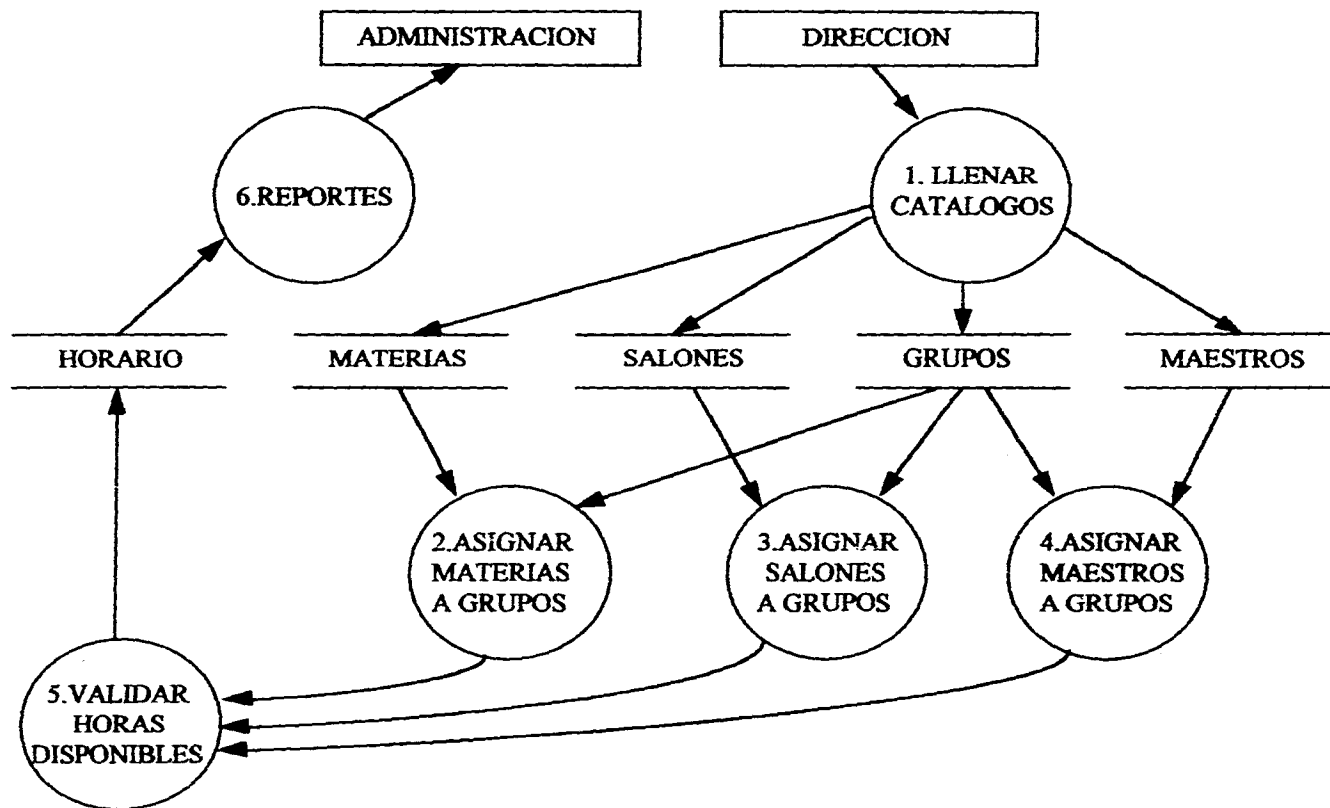


DIAGRAMA 3.2.- DIAGRAMA DE FLUJO DE DATOS 2

PROCESO 1. LLENAR CATÁLOGOS

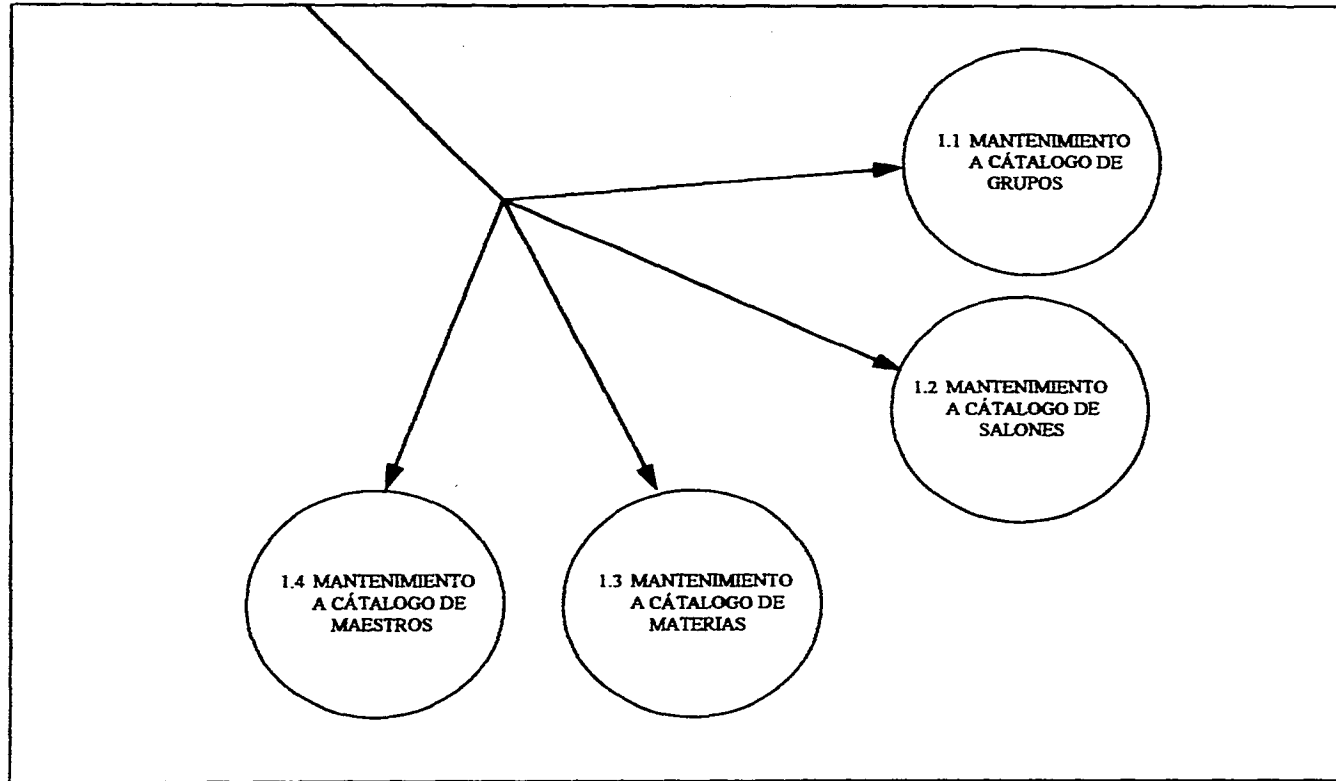


DIAGRAMA 3.3.- DIAGRAMA DE FLUJO DE DATOS 3
PROCESO 2. ASIGNAR MATERIAS A GRUPOS

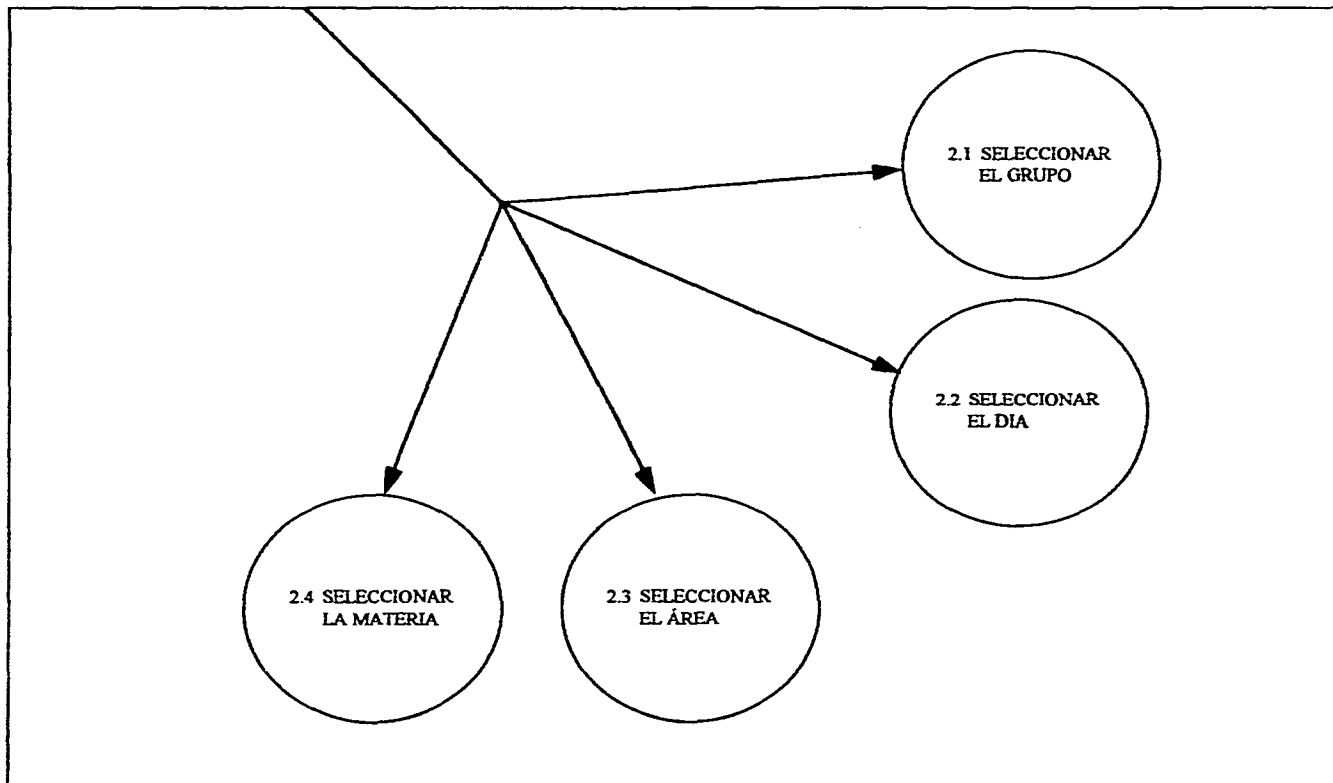


DIAGRAMA 3.4.- DIAGRAMA DE FLUJO DE DATOS 4

PROCESO 3. ASIGNAR SALONES A GRUPOS

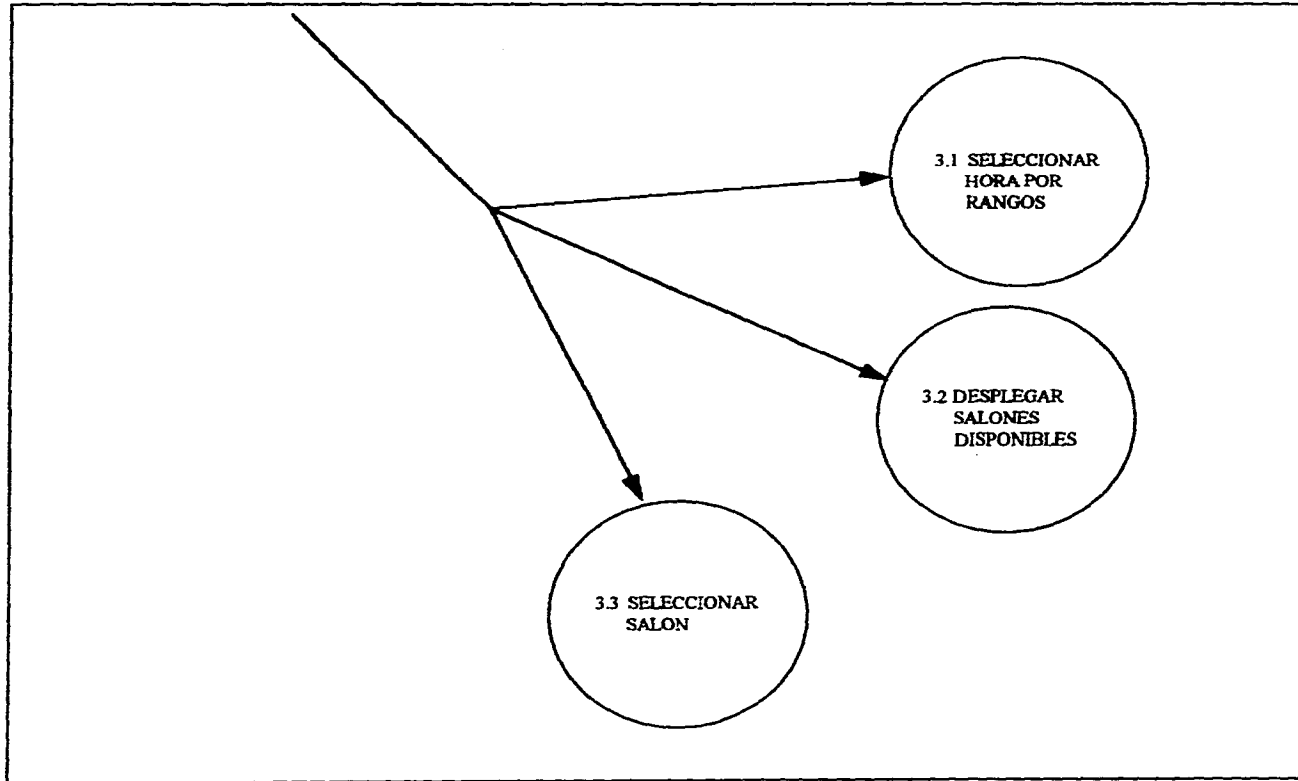


DIAGRAMA 3.4.- DIAGRAMA DE FLUJO DE DATOS 4
PROCESO 4. ASIGNAR MAESTROS A GRUPOS

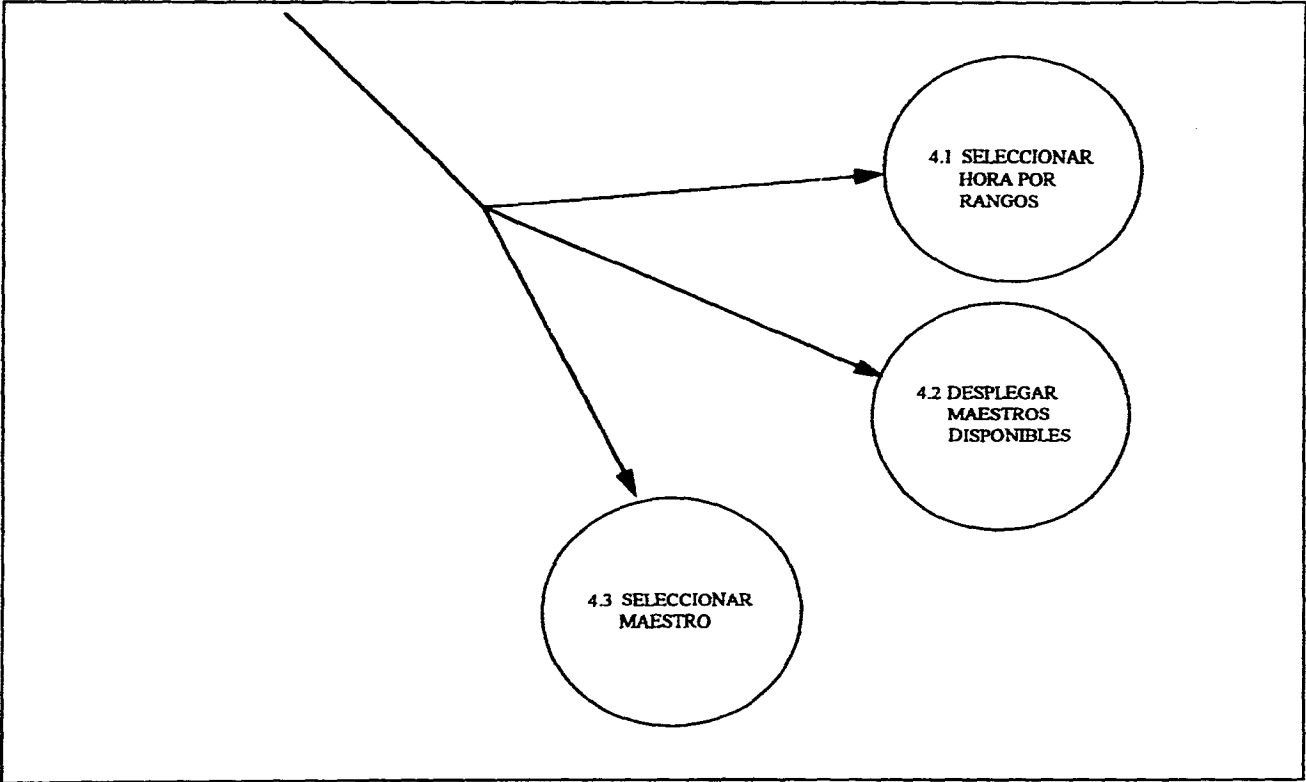


DIAGRAMA 3.5.- DIAGRAMA DE FLUJO DE DATOS 5

PROCESO 5. VALIDAR HORAS DISPONIBLES

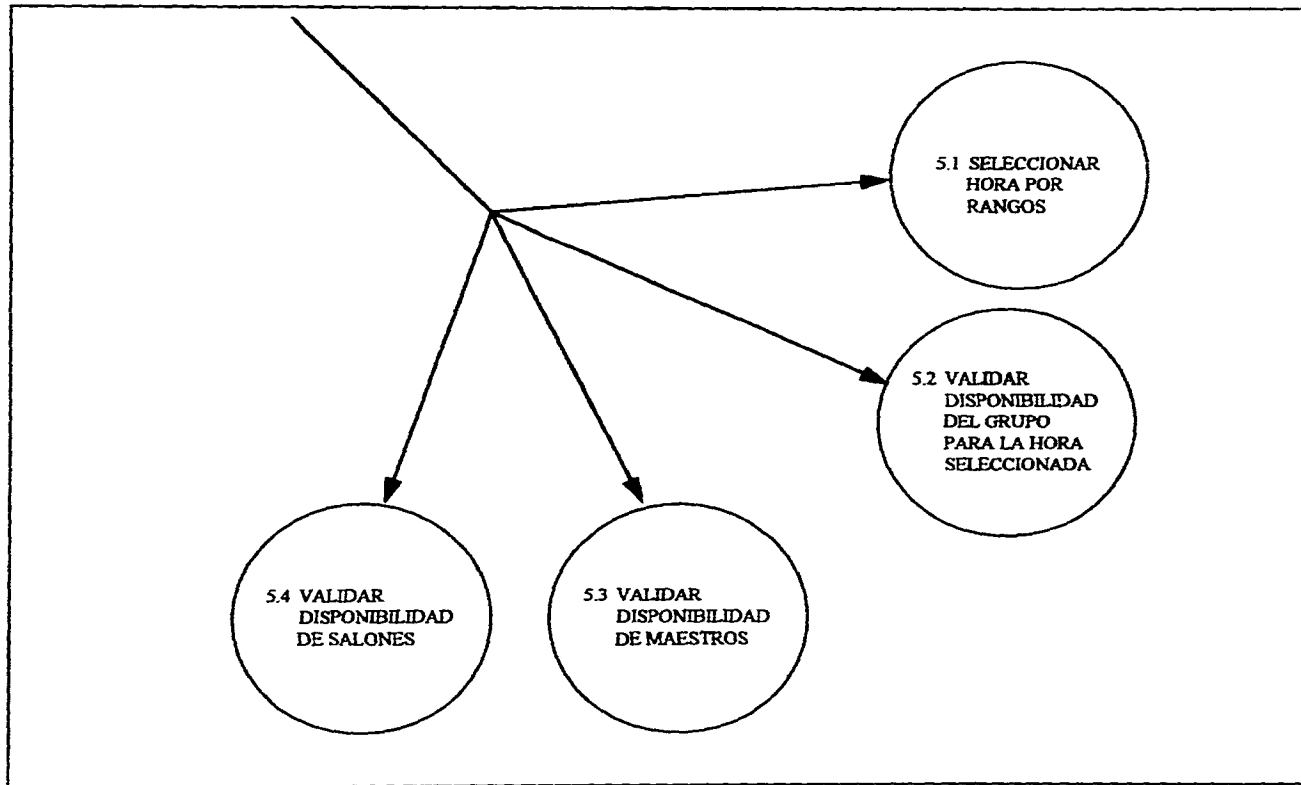
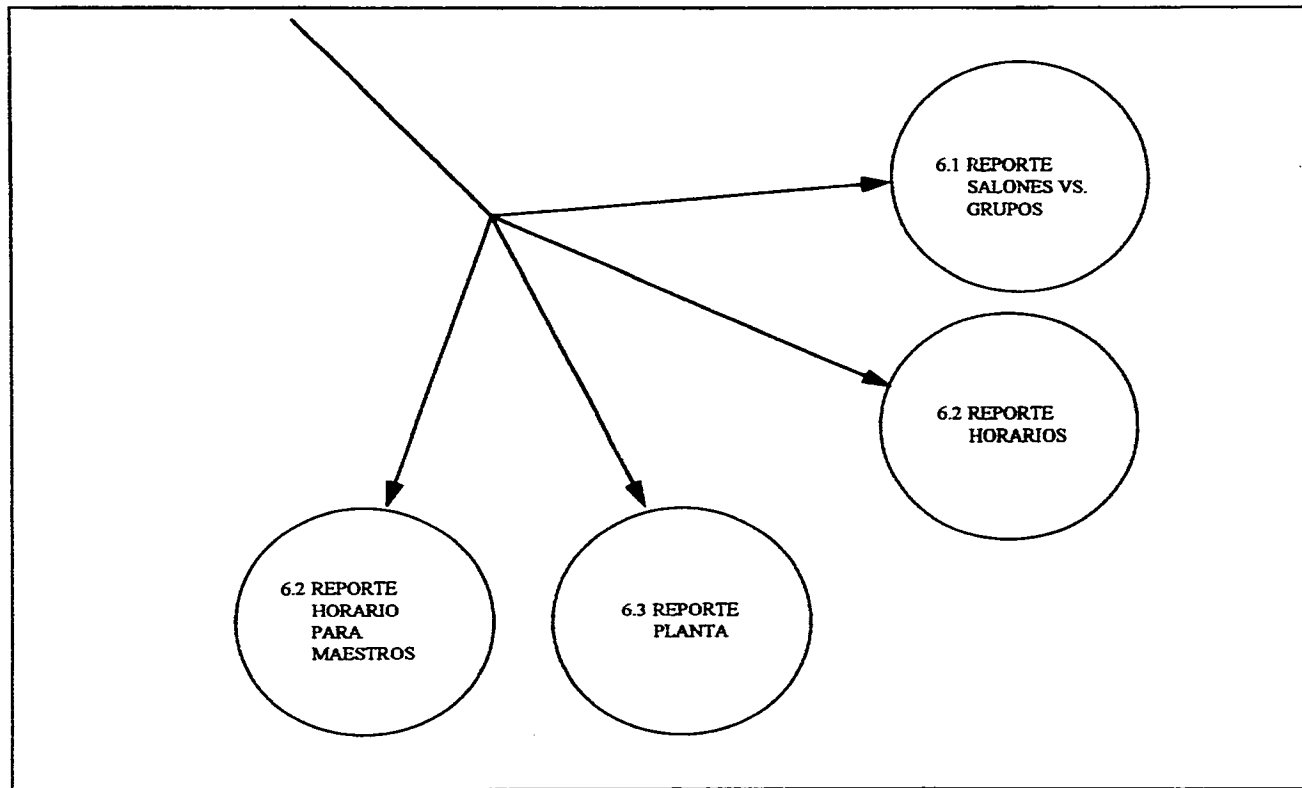


DIAGRAMA 3.6.- DIAGRAMA DE FLUJO DE DATOS 6

PROCESO 6. REPORTES



El **díagrama de flujo de datos (DFD)** es, como ya se mencionó, una herramienta gráfica, que sin explicación alguna, el usuario del sistema puede entenderla fácilmente. Muestra el flujo de información, del cual existen entradas y salidas. Realizadas a través de elementos llamados "procesos", y se representan gráficamente como una burbuja.

Las flechas representan los "flujos", que describen el desplazamiento de bloque de información, de un lugar del sistema a otro.

Existe otro elemento llamado "almacén", que representa una colección de paquetes de datos en reposo, se denota por líneas paralelas.

Por último, mencionamos el "terminador", gráficamente es un rectángulo, representa entidades externas con las cuáles el sistema interactúa, ya sea de manera constante o casual.

Haciendo referencia a nuestro DFD, podemos observar cuatro procesos que, básicamente representan la asignación de información, ya sea de maestros, grupos, materias, salones y horarios; todo esto bajo ciertos criterios de disponibilidad, selección, espacio y conteo.

Los cinco almacenes representan toda la información agrupada, para la elaboración de horarios de clase y dos terminadores, que representan la acción de la dirección, como medfador entre los recursos (maestros, grupos y materias) y el sistema, y el de administración, que representa la interacción entre el sistema y el resto de las entidades dentro de la universidad (en éste caso podría ser la rectoría, o la entidad encargada de asignar salones en forma global para la universidad).

Por último, como ya se mencionó, el DFD es una herramienta gráfica que no requiere mucha explicación, ya que por si sola muestra un panorama del sistema.

3.5.- DICCIONARIO DE DATOS.

El diccionario de datos es otra herramienta con la cual el usuario debe hallar relación y mayor significado a la estructura mostrada por el diagrama de flujo de datos. Esto es, describe el significado de los flujos y almacenes que se muestran en los diagramas de flujo de datos.

Para nuestro diagrama de flujo de datos se muestra a continuación el diccionario

de datos , correspondiente a cada almacén .

Salones

Nombre del campo	Tipo	Descripción	Ejemplos/Observaciones
salón	char(4)	Clave del salón	1D,2RN,5R
capacidad	number	Número de alumnos	1 - 100
día_disp	number	Día disponible	1 - 7
hora_disp	char(13)	Hora disponible	00:00 - 24:00

Materias

Nombre del campo	Tipo	Descripción	Ejemplos/Observaciones
área	char(25)	Nombre del área	Matemáticas, Térmica, etc
materia	char(30)	Nombre de la materia	Algebra Lineal, Dibujo ...
horas_sem	number	Horas a la semana	1 - 10
semestre	number	Semestre	1 - 10

Grupos

Nombre del campo	Tipo	Descripción	Ejemplos/Observaciones
grupo	char(4)	Clave del grupo	1A,2B,3C ...
numero	number	Número de alumnos	1 - 100
coordinador	number	Clave de coordinador	1 - 100

Maestros

Nombre del campo	Tipo	Descripción	Ejemplos/Observaciones
clave	number	Clave del maestro	1 - 100
nombre1	char(15)	Nombre de pila	
nombre2	char(15)	Segundo nombre	Opcional
apellido1	char(15)	Apellido paterno	
apellido2	char(15)	Apellido materno	Opcional
día_disp	number	Día disponible	1 - 7
hora_disp	char(13)	Hora disponible	00:00 - 24:00

Horarios

Nombre del campo	Tipo	Descripción	Ejemplos/Observaciones
grupo	char(4)	Clave del grupo	1A,2B,3C ...
día	number	Día de la semana	1 - 7
materia	char(30)	Nombre de la materia	Algebra Lineal, Dibujo ...
maestro	number	Clave del maestro	1 - 100
apellido2	char(15)	Apellido materno	Opcional
hora_disp	char(13)	Hora disponible	00:00 - 24:00
salón	char(4)	Clave del salón	1D,2RN,5R

3.6.- ESPECIFICACIÓN DEL PROCESO.

Las especificaciones de proceso son el documento escrito que se obtiene como resultado de las entrevistas con los usuarios, las cuales delimitan el alcance del sistema y fijan la problemática a resolver. Éstas deben contener los módulos con los que contará el sistema. De cada módulo se describen sus funciones, las variables involucradas y las validaciones de éstas.

En las validaciones se especifica la información de entrada y la información de salida y, se indicarán las longitudes de los campos.

Una vez considerado esto, la especificación del proceso es una descripción más detallada de lo que ocurre en una burbuja del DFD; básicamente es una explicación, bajo lenguaje clásico, de cómo transformar entradas de información en salidas.

- La especificación del proceso debe expresarse de una manera que puedan verificar tanto el usuario como el analista.

- El proceso debe especificarse en una forma que pueda ser comunicada efectivamente al público amplio que esté involucrado.

Como ya se mencionó, la especificación de proceso describe un orden lógico de operación para posteriormente, elaborar la programación del mismo. Se deberán utilizar palabras y verbos comunes, que sin ningún esfuerzo dé a entender su fin, tanto al programador como al usuario. Para referirse a la afectación de la base de datos, ésta se realiza de la siguiente manera: tabla/campo; Como se puede observar, primero se escribe el nombre de la tabla a utilizar seguida de una diagonal, para posteriormente escribir el cambio de dicha tabla, todo esto va subrayado. Dentro de las especificaciones de proceso se encuentran estructuras más comunes para los programadores, tales como ciclos tipo FOR, WHILE o bien, condicionales SI ENTONCES, DE OTRO MODO (En inglés IF THEN, ELSE). De modo que ésta parte del desarrollo del Sistema es más común para el analista o para el programador.

A continuación se describen las especificaciones de proceso para los diagramas de flujo de datos descritos anteriormente.

Proceso 1.1. Mantenimiento a catálogo de grupos.

a) Entrada

Desplegar ventana para grupos

Desplegar grupos/grupo existentes

Desplegar grupos/capacidad existentes

b) Alta

Desplegar ventana para detalle de grupos

Capturar grupos/grupo

Capturar grupos/capacidad

Capturar grupos/coordinador

Insertar en tabla grupos

Salir de la ventana para detalle de grupos

c) Baja

Seleccionar grupos/grupo que desea dar de baja

Desplegar ventana para detalle de grupos

Desplegar grupos/capacidad

Desplegar grupos/coordinador

Borra de la tabla grupos donde grupos/grupo = grupo seleccionado

Salir de la ventana para detalle de grupos

d) Cambio

Seleccionar grupos/grupo

Desplegar ventana para detalle de grupos

Desplegar grupos/capacidad

Desplegar grupos/coordinador

Reemplazar de la tabla grupos grupos/capacidad, grupos/coordinador con grupos/capacidad, grupos/coordinador donde grupos/grupo = grupo seleccionado

e) Salida

Cerrar ventana de grupos

Proceso 1.2. Mantenimiento a catálogo de salones.

a) Entrada

Desplegar ventana para salones

Desplegar salones/salón existentes

Desplegar salones/capacidad existentes

b) Alta

Desplegar ventana para detalle para salones

Capturar salones/salón

Capturar salones/capacidad

Capturar salones/día disponible

Capturar salones/hora disponible

Insertar en tabla salones

Salir de la ventana para detalle para salones

c) Baja

Seleccionar salones/salón que desea dar de baja

Desplegar ventana para detalle de salones

Desplegar salones/capacidad

Desplegar salones/día disponible

Desplegar salones/hora disponible

Buscar en la tabla horario donde horario/salón = salón seleccionado y

horario/día = salones/día disponible

Si se encuentra entonces desplegar el mensaje "El salón X tiene un grupo asignado"

De otro modo

Borra de la tabla salones donde salones/salón = salón seleccionado

Salir de la ventana para detalle de salones

d) Cambio

Seleccionar salones/salón

Desplegar ventana para detalle de salones

Desplegar salones/capacidad

Desplegar salones/día disp

Desplegar salones/hora disp

Reemplazar de la tabla salones salones/capacidad, salones/día disp,

salones/hora disp con salones salones/capacidad, salones/día disp,

salones/hora disp donde salones/salón = salón seleccionado

e) Salida

Cerrar ventana de salones

Proceso 1.3. Mantenimiento a catálogo de materias

a) Entrada

Desplegar ventana para materias

Desplegar materias/área existentes

b) Alta

Desplegar ventana para detalle para materias

Capturar materias/área

Capturar materias/materia

Capturar materias/horas sem

Capturar materias/semestre

Insertar en tabla materias

Salir de la ventana para detalle para materias

c) Baja

Seleccionar materias/área que desea dar de baja

Desplegar ventana para detalle de materias

Seleccionar materias/materia

Buscar en la tabla horario donde horario/materia = materia seleccionada

Si se encuentra entonces

desplegar el mensaje "La materia X está asignada en horario, no podrá borrarla."

De otro modo

Borra de la tabla materias donde materias/materia = materia seleccionada

Fin_si

Salir de la ventana para detalle de materias.

d) Cambio

Seleccionar materias/área

Desplegar ventana para detalle de materias

Seleccionar materias/materia que se desea cambiar

Reemplazar de la tabla materias materias/materia, materias/hora sem, materias/semestre con materias materias/materia, materias/hora sem, materias/semestre donde materias/materia = materia seleccionada

e) Salida

Cerrar ventana de materias

Proceso 1.4. Mantenimiento a catálogo de maestros.

a) Entrada

Desplegar ventana para maestros

Desplegar maestros/clave, maestros/apellido1, maestros/apellido2, maestros/nombre1 existentes

b) Alta

Desplegar ventana para detalle para maestros

Capturar maestros/nombre1

Capturar maestros/nombre2

Capturar maestros/apellido1

Capturar maestros/apellido2

Capturar maestros/día_disp

Capturar maestros/hora_disp

Insertar en tabla maestros

Salir de la ventana para detalle para maestros

c) Baja

Seleccionar maestros/clave que desea dar de baja

Desplegar ventana para detalle de maestros

Desplegar maestros/nombre1

Desplegar maestros/nombre2

Desplegar maestros/apellido1

Desplegar maestros/apellido2

Desplegar maestros/día_disp

Desplegar maestros/hora_disp

Buscar en la tabla horario donde horario/maestro = maestro seleccionado

Si se encuentra entonces

desplegar el mensaje "El maestro X tiene un grupo asignado"

De otro modo

Borra de la tabla maestros donde maestros/clave = maestro seleccionado

Fin_si

Salir de la ventana para detalle de maestros

d) Cambio

Seleccionar maestros/clave

Desplegar ventana para detalle de salones

Modificar maestros/nombre1

Modificar maestros/nombre2

Modificar maestros/apellido1

Modificar maestros/apellido2

Modificar maestros/día disp

Modificar maestros/hora disp

Reemplazar de la tabla maestros, maestros/nombre1, maestros/nombre2,
maestros/apellido1, maestros/apellido2, maestros/día disp, maestros/hora disp
con maestros/nombre1, maestros/nombre2, maestros/apellido1,
maestros/apellido2, maestros/día disp, maestros/hora disp donde
maestros/clave = maestro seleccionado

e) Salida

Cerrar ventana de maestros

Proceso 2. Asignar materias a grupos.

Desplegar grupos/grupo, grupos/capacidad existentes

Seleccionar grupos/grupo

Proceso 2.1. Seleccionar el día.

Desplegar los días de la semana (horario/día)

Seleccionar horario/día para la materia que se va a impartir

Proceso 2.2. Seleccionar el área.

Desplegar materias/área existentes

Seleccionar materias/área

Proceso 2.3. Seleccionar la materia

Desplegar materias/materia donde materias/área = área seleccionada

Seleccionar materias/materia

Proceso 3. Asignar salones a grupos.

Proceso 3.1. Seleccionar hora por rangos

Desplegar rangos de horas por intervalos de 30 minutos, a partir de las 07:00 hrs
a 23:00 hrs

Seleccionar el rango que se desea

Si se detectan dos o más rangos entonces

desplegar el mensaje "Debe capturar sólo un rango de horas"

De otro modo

horario/hora_disp = rango seleccionado

Fin_sl

Proceso 3.2. Desplegar salones disponibles

Seleccionar salones/salón, de la tabla de salones, horario, donde salones/día_disp = día seleccionado y salones/hora_disp >= hora seleccionada y salones/salón no se encuentre en horario/salón y horario/hora_disp sea diferente a la hora seleccionada (esto para no poder seleccionar un salón ocupado anteriormente).

Proceso 3.3. Seleccionar salón

Seleccionar salones/salón disponible para el grupo seleccionado.

Proceso 4. Asignar maestros a grupos

Proceso 4.1 Seleccionar hora por rangos

Desplegar rangos de horas por intervalos de 30 minutos, a partir de las 07:00 hrs
a 23:00 hrs

Seleccionar el rango que se desea

Si se detectan dos o más rangos entonces

desplegar el mensaje "Debe capturar sólo un rango de horas"

De otro modo

horario/hora_disp = rango seleccionado

Fin_sl

Proceso 4.2. Desplegar maestros disponibles

Seleccionar maestros/clave, maestros/apellido1, maestros/apellido2,
maestros/nombre1 de la tabla de maestros, horario donde maestros/día_disp =
día seleccionado y maestros/hora_disp >= hora seleccionada y maestros/clave
no se encuentre en horario/maestro y horario/hora_disp sea diferente a la hora
seleccionada (esto para no poder seleccionar un maestro ocupado).

Proceso 4.3. Seleccionar maestro

Seleccionar maestros/clave disponible para el grupo seleccionado.

Proceso 5. Validar horas disponibles

Proceso 5.1 Seleccionar hora por rangos

Desplegar rangos de horas por intervalos de 30 minutos, a partir de las 07:00 hrs a 23:00 hrs

Seleccionar el rango que se desea

Si se detectan dos o más rangos entonces

desplegar el mensaje "Debe capturar sólo un rango de horas"

De otro modo

horario/hora_disp = rango seleccionado

Fin_si

Proceso 5.2. Validar disponibilidad del grupo para la hora seleccionada

Seleccionar horario/grupo de la tabla de horario donde horario/grupo = grupo seleccionado y horario/día = día seleccionado y horario/hora_disp sea diferente a la hora seleccionada.

Si se encuentra entonces

Desplegar el mensaje "El grupo X ya tiene asignada una materia para el rango de horas Y"

Fin_si

Proceso 5.3 Validar disponibilidad de maestros

Seleccionar horario/maestro de la tabla de horario donde horario/maestro = maestro seleccionado y horario/día = día seleccionado y horario/hora_disp sea diferente a la hora seleccionada

Si se encuentra entonces

Desplegar el mensaje "El maestro X ya tiene asignado un grupo para el rango de horas Y"

Fin_si

Proceso 5.4 Validar disponibilidad de salones

Seleccionar horario/salón de la tabla de horario donde horario/salón = salón seleccionado y horario/día = día seleccionado y horario/hora_disp sea diferente a la hora seleccionada

Si se encuentra entonces

Desplegar el mensaje "El salón X ya tiene asignado un grupo para el rango de horas Y"

Fin_si

Proceso 6.1 Reporte salones vs. grupos

Seleccionar horario/salón de la tabla de horario donde horario/salón = salón seleccionado.

Desplegar rangos de horas por intervalos de 30 minutos, a partir de las 07:00 hrs
a 23:00 hrs

For día_x = 1 to 7 (días de la semana)

For y = 1 to 32 (número de intervalos de horas)

Intervalo = intervalo número y

Seleccionar horario/salón donde horario/día = día_x

y horario/día_disp = intervalo y horario/salón = salón seleccionado

End For

End For

Imprimir el resultado

Proceso 6.2 Reporte Horarios

Seleccionar horario/grupo, horario/capacidad de la tabla de horario donde
horario/grupo = grupo seleccionado.

For día_x = 1 to 7 (días de la semana)

Seleccionar horario/materia, horario/maestro, horario/hora_disp, horario/salón

donde horario/día = día_x y horario/grupo = grupo seleccionado

ordenado por horario/hora_disp

End For

Imprimir el resultado

Proceso 6.3 Reporte Planta

Seleccionar horario/grupo, horario/capacidad de la tabla de horario donde horario/grupo = grupo seleccionado.

Seleccionar materias/área, horario/materia, materias/hora sem, horario/grupo, maestros/apellido1, maestros/apellido2, maestros/nombre1, horario/hora disp, horario/salón donde horario/grupo = grupo seleccionado

Imprimir el resultado

Proceso 6.4 Reporte Horario para maestros.

Seleccionar maestros/clave, maestros/apellido1, maestros/apellido2, maestros/nombre1

For día_x = 1 to 7 (días de la semana)

Seleccionar horario/materia, horario/grupo, horario/hora disp, horario/salón

donde horario/día = día_x y horario/maestro = maestro seleccionado

ordenado por horario/hora disp

End For

Imprimir el resultado

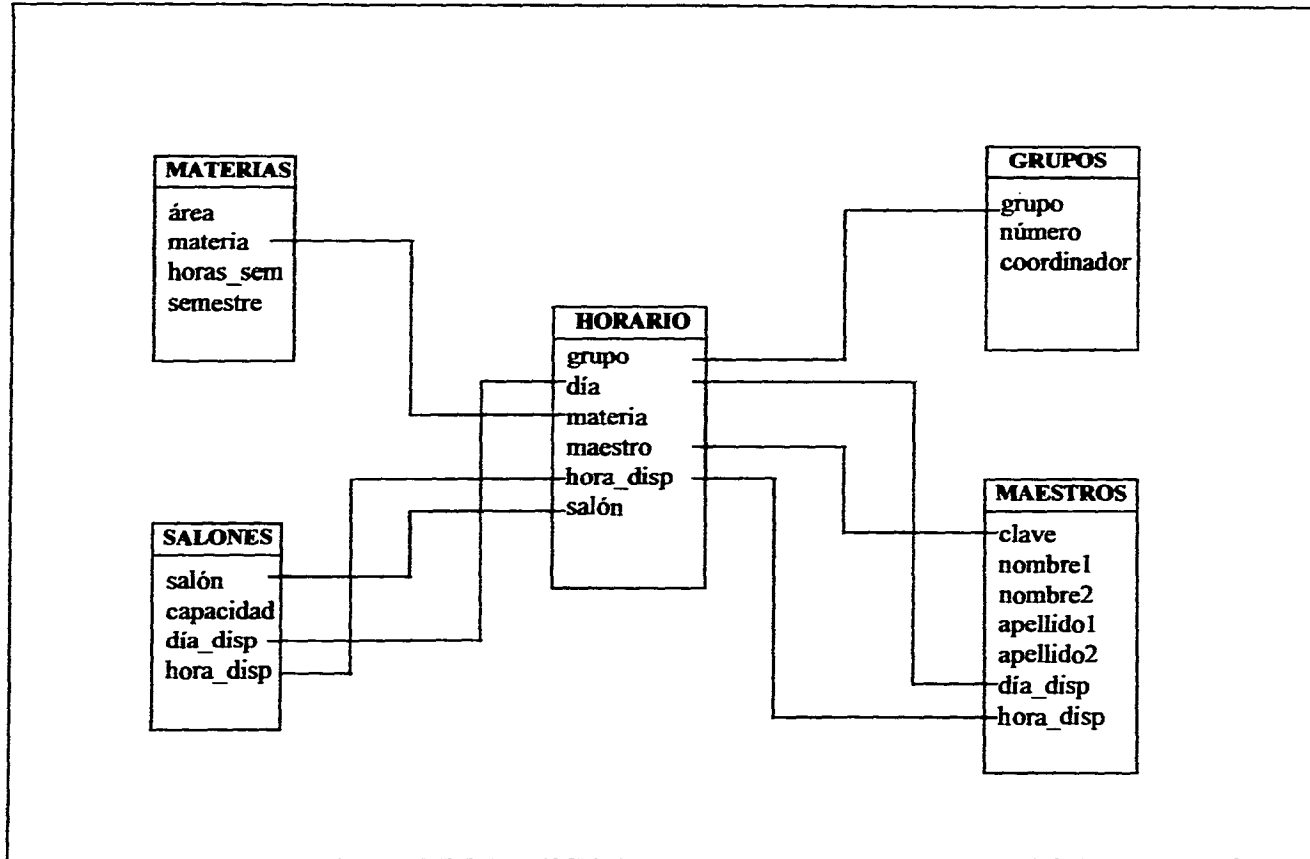
3.7.- DÍAGRAMA ENTIDAD - RELACIÓN.

El diagrama de entidad relación (DER) es aquél, en donde se muestra la distribución de datos que existen en el sistema. El diagrama de entidad relación es una herramienta efectiva de modelado para comunicarse con el grupo de administración de base de datos.

Basándose en la información presentada por el DER, el grupo de la administración de base de datos puede ver el tipo de claves o índices o apuntadores que se necesitarán para llegar de manera eficiente a los registros de la base de datos.

En el diagrama 3.8 se muestra el diagrama entidad - relación del sistema de horarios. Mostrando la relación que existe entre los datos de las entidades desarrolladas en el sistema.

DIAGRAMA 3.7.-DIAGRAMA ENTIDAD-RELACION



3.8.- DÍAGRAMA DE CONTEXTO.

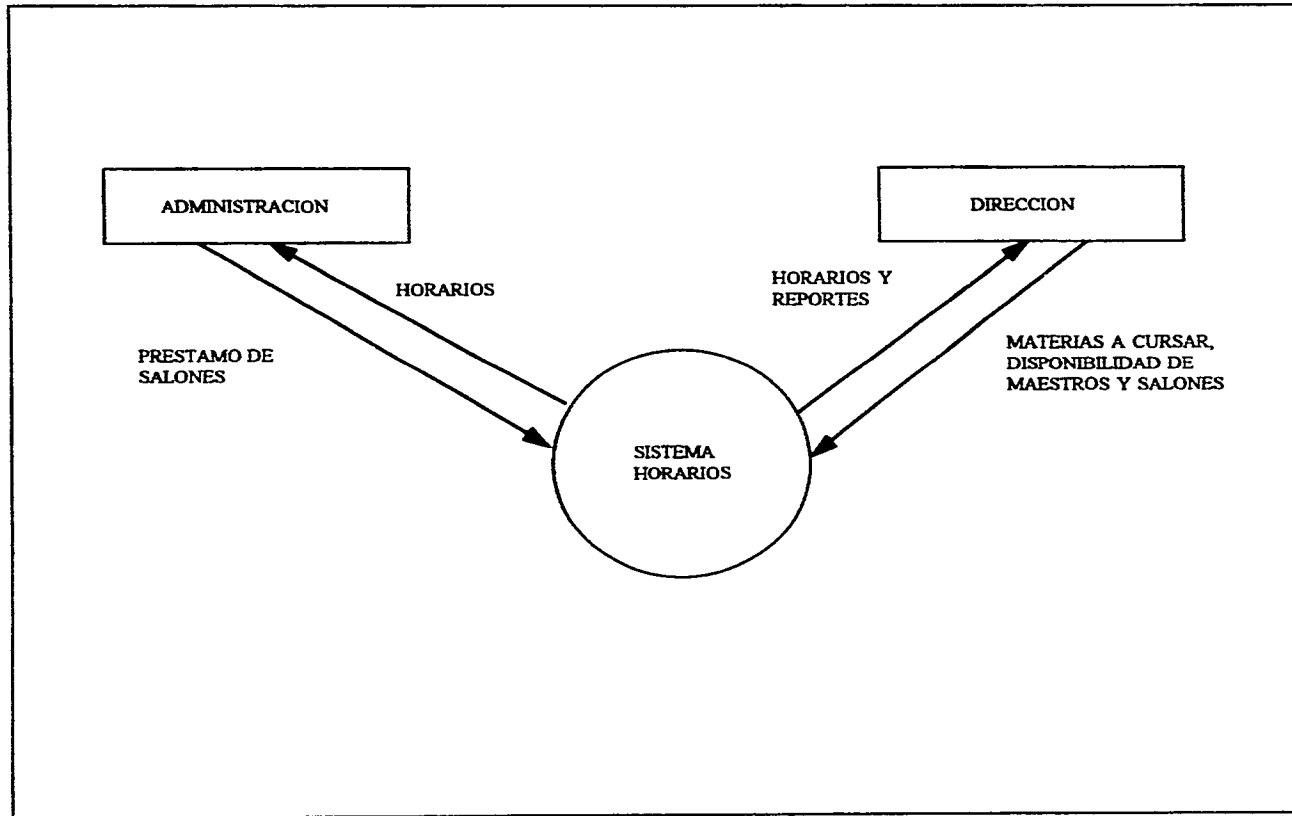
El diagrama de contexto es un caso especial del diagrama de flujo de datos, en donde se representa al sistema con una sola burbuja, poniendo así en relieve los factores externos que interactúan con el sistema.

Se mencionan a continuación algunas características del diagrama de contexto:

- Se determinan como terminadores todos aquellos elementos, personas, organizaciones y sistemas con los que se comunica al sistema.
- Los datos que el sistema recibe del mundo exterior y que deban tomarse en cuenta.
- Los datos que el sistema produce y que se envían al mundo exterior.
- Delimita la frontera entre el sistema y el resto del mundo.

En el diagrama 3.8 se muestra el diagrama de contexto propuesto para el sistema de horarios.

DIAGRAMA 3.8 DIAGRAMA DE CONTEXTO



4. PRUEBAS DEL SISTEMA

Por lo general, la programación y pruebas son la actividad que se realiza después del diseño. La fase de programación involucra una serie de instrucciones bajo una plataforma de lenguaje gráfico.

Es probable que el proceso de probar el sistema tome tanto como la mitad del tiempo programado para su desarrollo, dependiendo de qué tan cuidadosamente se hayan hecho las actividades iniciales de análisis, diseño y programación. Incluso si se hizo una labor perfecta de análisis, diseño y programación, se debe hacer algún esfuerzo para verificar que no haya errores. Si por otro lado, se hizo un trabajo imperfecto, entonces la prueba se vuelve iterativa: la primera tanda de pruebas muestra la presencia de errores, y las posteriores verifican si los programas corregidos funcionan correctamente.

La etapa de pruebas tiene la finalidad de ejercitar el sistema para asegurar que produzca las salidas apropiadas para una gama amplia de entradas.

4.1.- LENGUAJES DE PROGRAMACIÓN.

Se han estado escribiendo programas de computación desde que se desarrollaron las computadoras de propósito general, hace 40 años. Es conveniente agrupar a los distintos lenguajes en cuatro generaciones distintas

- Primera Generación: Fueron los lenguajes desarrollados en los años 50, utilizando unos y ceros binarios.

- Segunda Generación: Son los sucesores del lenguaje de máquina, se conocen como lenguajes de ensamble o ensamblador.

- Tercera Generación: Son los lenguajes como BASIC, COBOL, FORTRAN en donde una sola declaración usualmente representa cinco o diez declaraciones de lenguaje ensamblador.

- Cuarta generación: Los lenguajes de cuarta generación, o 4GL's son el desarrollo más importante en el campo de software en los últimos 20 años.

Como ejemplos se tiene a FOCUS, DBase-IV, INFORMIX. En donde, con una sola orden sencilla el programador puede especificar que la computadora debe capturar un tipo especificado de datos desde el teclado, validarlo, y volver a almacenarlo en un elemento de datos designado.

Como ya se mencionó, el lenguaje que se va a utilizar es un lenguaje gráfico de cuarta generación llamado Object View.

Los listados de los programas realizados para el sistema de horarios se encuentran en los apéndices A, B y C.

4.2.- PROGRAMACIÓN.

En la programación se afinan las especificaciones de proceso transformando éstas en código de programación. Cabe mencionar que las actividades de análisis, diseño y programación puedan desarrollarse paralelamente.

Este concepto se aplica también para el ramo de la construcción, ya que no es necesario llevar un orden riguroso de las actividades.

En el caso de desarrollo de un sistema de información esto significa que los productos de alto nivel del análisis de sistemas, es decir, los documentos que constituyen el marco de referencia, tales como diagramas de flujo de datos o diagramas de entidad relación pueden usarse como base para el diseño de alto nivel. Y éste puede usarse como fundamento para escribir código de alto nivel aun antes de haber terminado los detalles del análisis de sistemas.

Existe una gran flexibilidad en este enfoque; se puede terminar el 80 por ciento de la labor de análisis antes de comenzar con el diseño y la programación, o se puede terminar sólo el 10 por ciento.

Como la actividad de análisis, diseño y programación se realiza de manera concurrente, usualmente se tiene la oportunidad de disminuir el tiempo total del proyecto. Esto permite al desarrollador mostrar al usuario una versión esquelética del sistema antes de concluir la labor detallada del análisis.

El comenzar la tarea de la programación pronto, ayuda a disminuir los problemas que existen entre la forma de operar tradicional y la etapa de acoplamiento al nuevo sistema. Ya que se facilita la comunicación del programador y el usuario, disminuyendo así los errores de interpretación de tareas.

4.3.- ESTRATEGIAS DE PRUEBA.

Se pueden distinguir dos tipos distintos de pruebas: la prueba descendente y la prueba ascendente.

El enfoque ascendente empieza por probar módulos individuales pequeños separadamente; llamándole a esto la prueba de unidades, prueba de módulos o prueba de programas. Luego, los módulos individuales se combinan para

fornar unidades cada vez más grandes que se probarán en masa; llamándole a esto la prueba de subsistemas. Finalmente, todos los componentes del sistema se combinan para probarse ; esto se conoce como prueba del sistema, y puede estar seguido de las pruebas de aceptación, donde se permite al usuario usar sus propios casos de prueba para verificar que el sistema esté trabajando de manera correcta.

El enfoque de prueba descendente empieza con un esqueleto del sistema; es decir, la estrategia de prueba supone que se han desarrollado los módulos ejecutivos de alto nivel del sistema, pero que los de bajo nivel existen sólo como módulos vacíos. Dado que muchas de las funciones detalladas del sistema no se han implantado, las pruebas iniciales están muy limitadas; el propósito es simplemente comenzar a ejercitar las interfaces entre los subsistemas principales. Las pruebas siguientes abarcan y tratan aspectos cada vez más detallados del sistema. El enfoque descendente de prueba generalmente se considera preferible para muchos sistemas en la actualidad.

Existen otros tipos de prueba, como por ejemplo: pruebas de volumen de información, pruebas funcionales o pruebas exhaustivas.

En el proyecto ideal, se generarían casos de prueba para cubrir cada entrada posible y cada combinación posible de situaciones que el sistema pudiera enfrentar alguna vez; luego se probaría de manera exhaustiva para asegurar

que su comportamiento sea perfecto. Aunque esto no funciona en la realidad. El número de casos de prueba para un sistema grande y complejo típico es tan increíblemente grande, que aun si se pudiera realizar una prueba cada milésima de segundo tomaría mucho tiempo su elaboración.

4.4.- CONVERSION.

La Conversión es la tarea de traducir los archivos , formas y bases de datos actuales del usuario al formato que el nuevo sistema requiere. En algunos casos raros, esto puede ser una actividad no relevante, porque ya no hay datos. Sin embargo, si el usuario está reemplazando un sistema actual por uno nuevo, es probable que esto sea una tarea muy difícil. Se necesita desarrollar un plan de conversión, de preferencia en cuanto se complete el modelo de implantación del usuario, para cubrir los siguientes puntos:

- Si el usuario ya tiene datos existentes asociados con un sistema existente, probablemente querrá usarlos hasta el último momento posible antes de pasarse al nuevo sistema.

- Pudiera haber un volumen tan grande de datos existentes que sea impráctico considerar convertirlo todo a la vez. Los archivos y registros podrían tener que

convertirse en forma incremental. Esto obviamente requiere de una planeación y coordinación cuidadosa.

- La conversión debe llevarse a cabo de una manera automatizada; esto se puede hacer si los archivos y datos actuales existen en alguna forma automatizada. De ser así, debiera ser relativamente fácil escribir un programa para traducir los archivos actuales al formato requerido por el sistema nuevo.

Sin embargo, a veces resulta difícil convertir los datos en forma automatizada, sobre todo si los archivos existentes se tienen en distintas computadoras, en distintos formatos, etc. De hecho, desarrollar el software de conversión puede resultar ser por sí mismo un proyecto importante de desarrollo de sistemas.

- Los datos existentes pueden contener errores; de hecho, si se crearon y mantuvieron manualmente, puede estar casi seguro de que habrá errores. Por ello, parte del proceso de conversión es la detección y corrección de errores, que puede volver aun más difícil y tardado el proceso. Algunos archivos y registros existentes pueden resultar ilegibles o incomprensibles; en otros casos, puede ser obvio que los datos existentes están mal, pero podría no ser claro cuáles son los valores correctos.

- Además de convertir archivos existentes, puede ser necesario convertir programas y procedimientos existentes. En algunos casos, los programas y procedimientos existentes pueden usarse en su forma actual; en otros, se tendrán que desechar y reemplazar por completo.

4.5.- INSTALACIÓN.

La instalación del nuevo sistema puede ser un asunto instantáneo, pero a menudo es una tarea enorme. Usualmente se debe hacer lo siguiente:

- A la instalación del nuevo sistema debe precederle la preparación de la sede de la computadora, usualmente con varios meses de anticipación. Esto implica construir o rentar un local de cómputo con la corriente, espacio, iluminación y control ambiental (temperatura, polvo, humedad) apropiados. Esto muchas veces se hace en conjunto con el proveedor de hardware o el departamento de operaciones de cómputo, en este caso de la Universidad.

- La instalación de hardware, cuando el sistema requiere de su propia computadora, usualmente la efectúa el proveedor. En ocasiones se involucran varios proveedores, sobre todo para sistemas en línea. En el caso de un sistema sencillo como el que se propone, desarrollado para una computadora

personal, la instalación puede ser tan sencilla como sacar la computadora de su caja y conectarla.

- La instalación del software, que involucra cargar todos los programas que se escribieron para el nuevo sistema en la o las computadoras adecuadas , y prepararlos para su operación.

4.6. - CAPACITACIÓN.

La Capacitación es la tarea final: la capacitación de los usuarios, además de la preparación del personal de operaciones, los programadores de mantenimiento y varios niveles de administración. Se debe establecer un plan de capacitación, el cual debe considerar los siguientes puntos:

- Muchos proyectos de desarrollo de sistemas depende de manuales para el usuario y guías de referencia para proponer a los usuarios documentos escritos.

En caso extremo, la capacitación puede consistir en opciones de ayuda integradas al sistema mismo. Esto se está volviendo cada vez más popular con la proliferación de la computadoras personales, pero no es muy práctico para sistemas grandes con una comunidad grande y diversificada de usuarios.

- En algunos casos, los miembros del equipo de desarrollo de sistemas participan en el proceso, sobre todo dado que se supone que son expertos del sistema. Sin embargo se debe tener en mente que el mejor programador no siempre es el mejor maestro; de hecho quienes desarrollan el sistema deben comportarse de una manera muy defensiva si los usuarios empiezan a hacer preguntas que consideran hostiles.

- Obviamente se necesita capacitar a los usuarios antes de que usen el sistema; por otro lado, no resulta efectivo prepararlos seis meses antes de que puedan ver el nuevo sistema. Por ello, la capacitación debe hacerse en un tiempo relativamente corto; pero esto, a su vez, a menudo interferirá con el trabajo cotidiano normal que los usuarios tratan de hacer. Por tanto, se debe negociar un programa de actividades de capacitación.

Dado que el tamaño del sistema en propuesta no es muy grande, se trabajará bajo un manual de usuario, para así, llevar la capacitación el coordinador de horarios.

CAPITULO 5. MANUAL DE USUARIO.

El manual de usuario es el documento en el cual se describe la funcionalidad del sistema.

5.1.- Aceleradores.

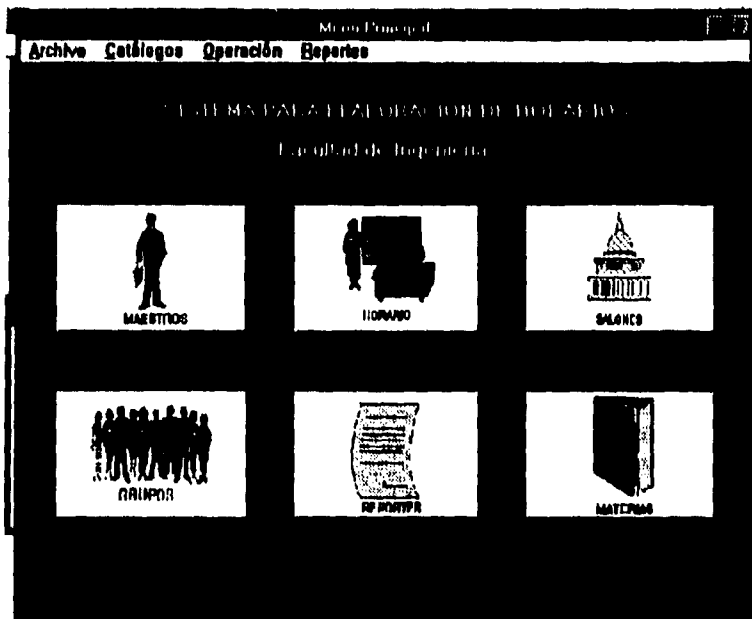
Los Aceleradores son las teclas que se encuentran subrayadas dentro de las opciones del menú, para poder accederlas más rápidamente.

5.2.- Teclas de Escape.

En todas las ventanas excepto la ventana principal, existe un botón para salir de la misma. Pero también pueden accionar la tecla de escape (Esc) para realizar la misma acción.

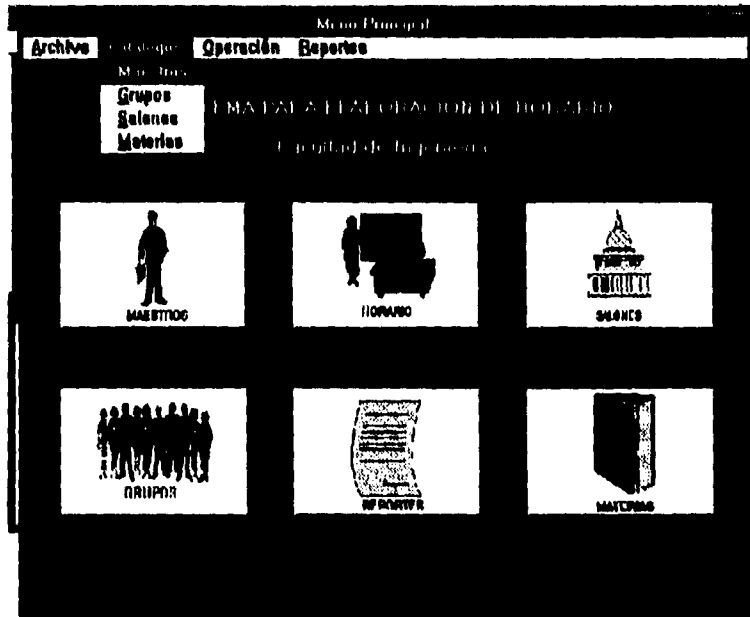
5.3.- Menú principal.

A continuación se muestra la pantalla inicial del sistema:



El menú principal contiene los conceptos Archivo, Catálogos, Operación y Reportes. La opción de Archivo contiene la subopción Salir, misma que al accionarse mediante un click del mouse, se podrá salir del sistema.

Oprima la opción Catálogos (mediante un click), dentro de la cual se tienen las siguientes subopciones:

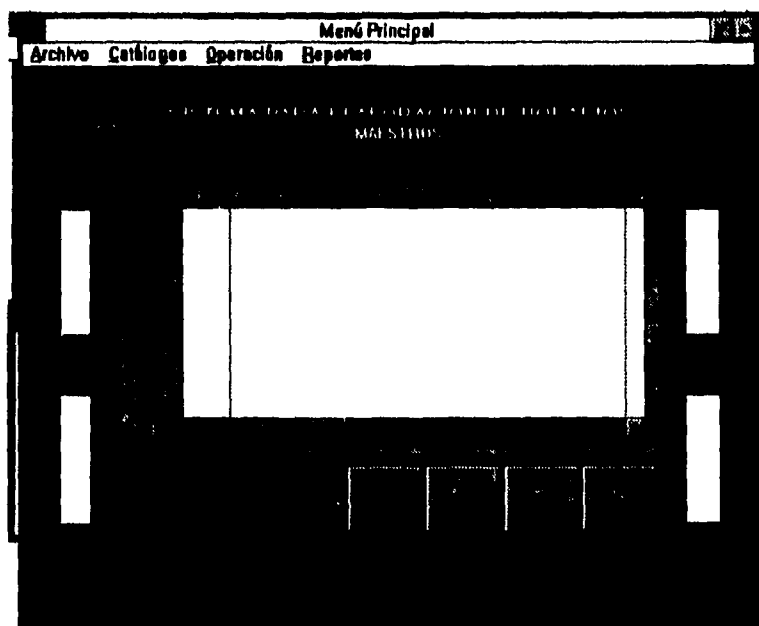


En esta opción se cuenta con las subopciones de Maestros, Grupos, Salones y Materias.

A continuación se muestra cada una de las opciones antes mencionadas, y la forma de utilizar el mantenimiento de éstas.

5.3.1.- Maestros.

Oprima la subopción Maestros, o bien, presione el botón con la leyenda MAESTROS; a continuación aparecerá la siguiente ventana:



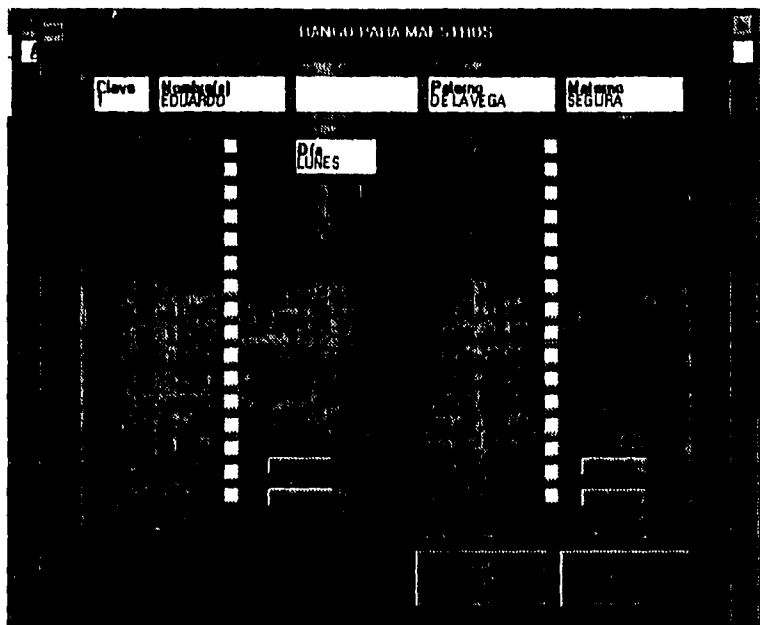
En la pantalla anterior se tienen botones para ALTA,BAJA,CAMBIO y SALIR.

Oprima la tecla ALTA, y la siguiente pantalla aparecerá:

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

The image shows a screenshot of a software application window. At the top, there is a title bar that reads "Menú Principal". Below the title bar is a menu bar with the following items: "Archivo", "Catálogos", "Operación", and "Reportes". The main content area of the window has a title "DETALLE PARA MAESTROS". The interface contains several input fields, some of which are empty. There is a dropdown menu that currently displays "1" and "LUNES". At the bottom of the form, there is a button labeled "Rangos".

En la pantalla anterior capture el (los) nombre(s), apellido(s), el día disponible y presione el botón con la leyenda Rangos, apareciendo a continuación:



Capture las horas en las que estará disponible para el día y el maestro seleccionado y presione ACEPTAR.

Para dar de baja algún maestro:

- Seleccione el maestro y presione el botón BAJA.
- Presione el botón ACEPTAR.

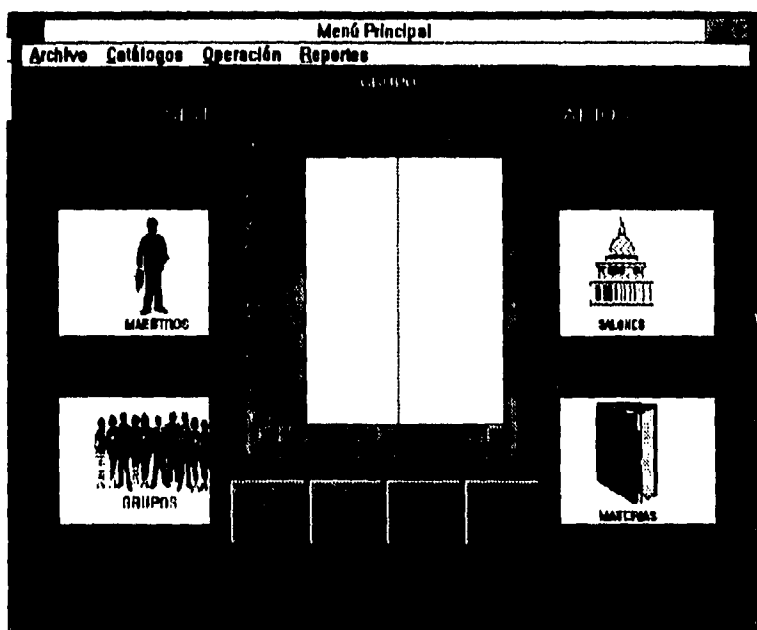
Para modificar los datos de un maestro:

- Seleccione el maestro y presione el botón CAMBIO.
- Modifique nombre(s), apellido(s) y presione el botón ACEPTAR.
- Si desea modificar las horas disponibles :
 - Seleccione el día a modificar y presione el botón Rangos.

- Modifique las horas y presione el botón ACEPTAR.

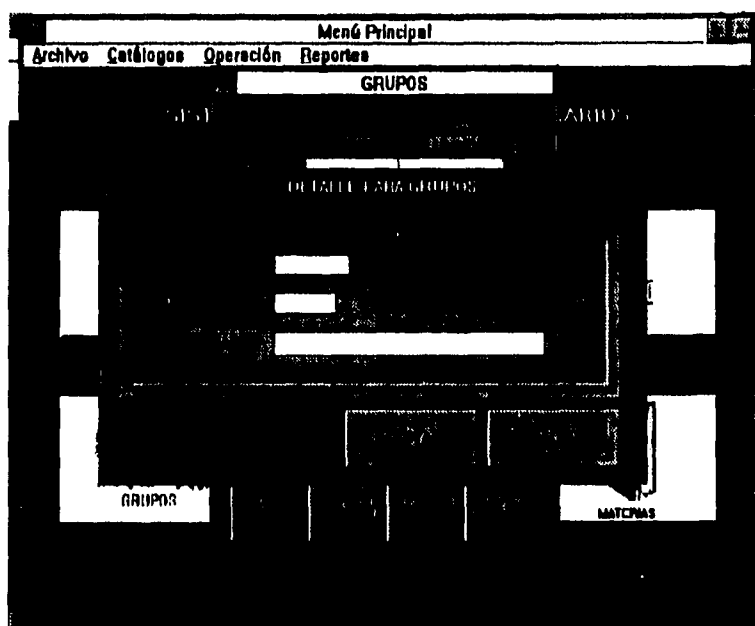
5.3.2.- Grupos.

Oprima la subopción Grupos o presione el botón con la leyenda GRUPOS; a continuación aparecerá la siguiente ventana:



En la pantalla anterior se tienen botones para ALTA,BAJA,CAMBIO y SALIR.

Oprima la tecla ALTA, y la siguiente pantalla aparecerá:



En la pantalla anterior capture el grupo, número de alumnos y seleccione al coordinador.

Para dar de baja algún grupo:

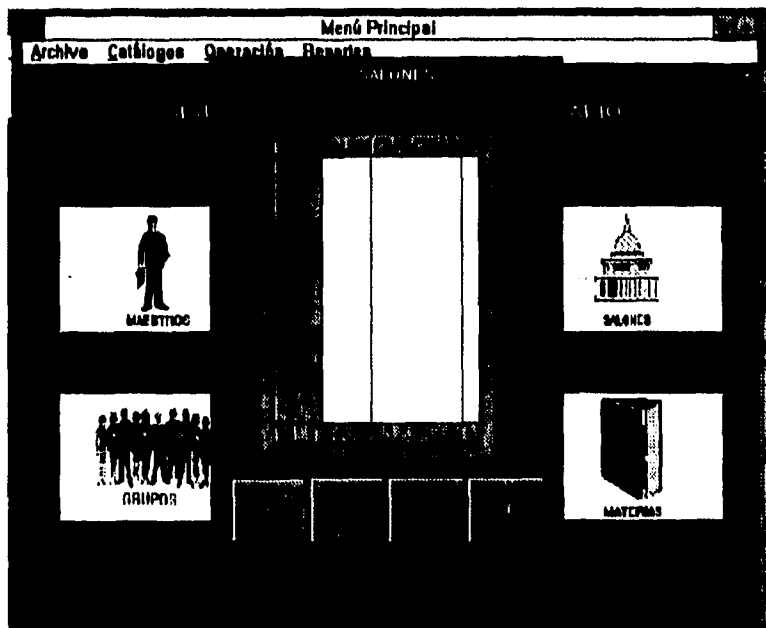
- Seleccione el grupo y presione el botón BAJA.
- Presione el botón ACEPTAR.

Para modificar los datos de un grupo:

- Seleccione el grupo y presione el botón CAMBIO.
- Modifique el número de alumnos del grupo y coordinador y presione el botón ACEPTAR.

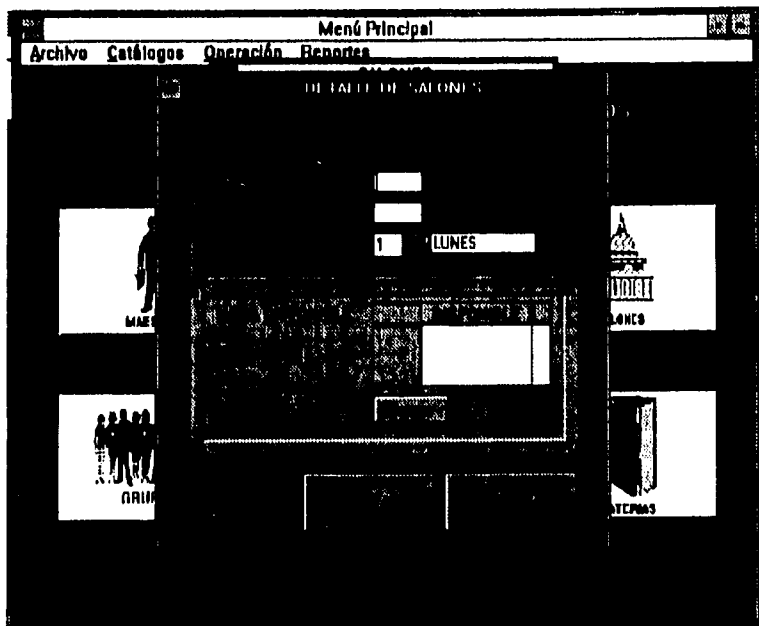
5.3.3.- Salones.

Oprima la subopción Salones, o bien, presione el botón con la leyenda SALONES; a continuación aparecerá la siguiente ventana:

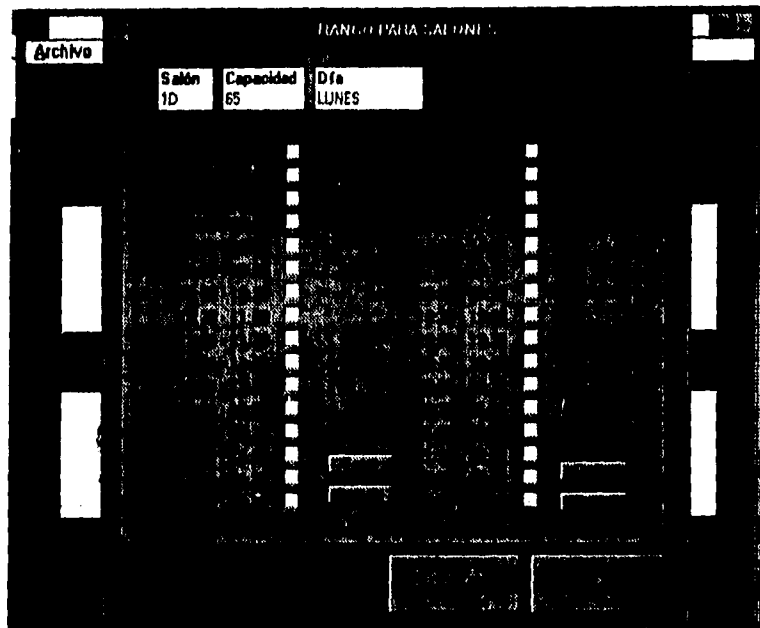


En la pantalla anterior se tienen botones para ALTA,BAJA,CAMBIO y SALIR.

Oprima la tecla ALTA, y la siguiente pantalla aparecerá:



En la pantalla anterior capture el salón, capacidad, día disponible y presione el botón con la leyenda Rangos, apareciendo a continuación:



Capture las horas en las que estará disponible para el día y el salón seleccionado y presione ACEPTAR.

Para dar de baja algún salón:

- Seleccione el salón y presione el botón BAJA.
- Presione el botón ACEPTAR.

Para modificar los datos de un salón:

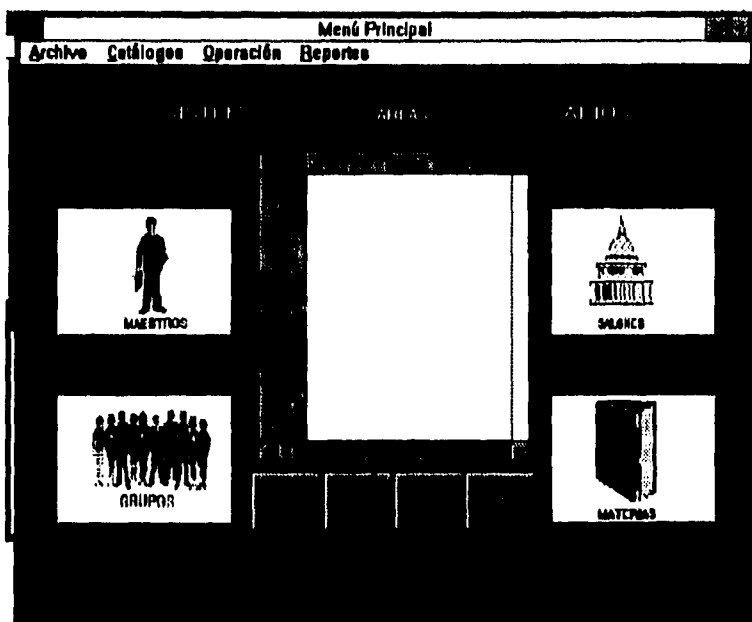
- Seleccione el salón y presione el botón CAMBIO.
- Modifique la capacidad del salón y presione el botón ACEPTAR.

- Si desea modificar las horas disponibles :

- Seleccione el día a modificar y presione el botón Rangos.
- Modifique las horas y presione el botón ACEPTAR.

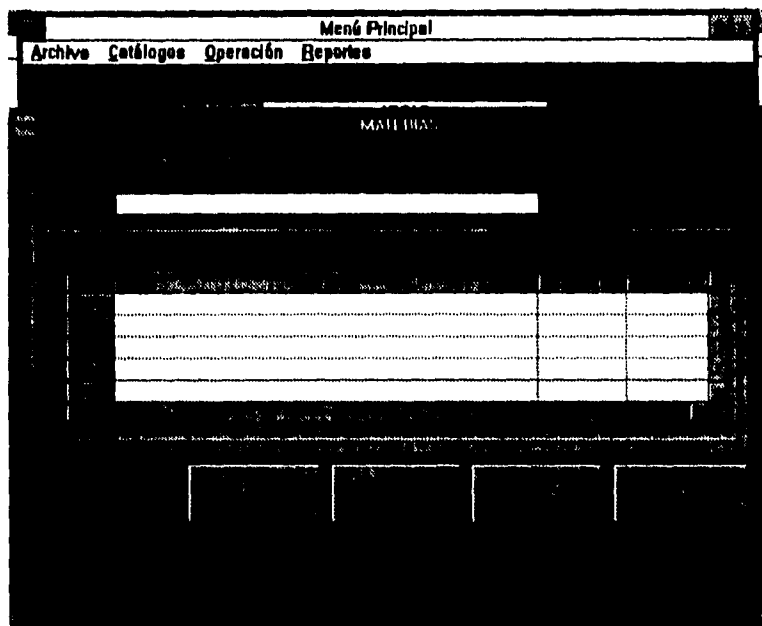
5.3.4.- Materias.

Oprima la subopción Materias o presione el botón con la leyenda MATERIAS;
a continuación aparecerá la siguiente ventana:



En la pantalla anterior se tienen botones para ALTA,BAJA,CAMBIO y SALIR.

Oprima la tecla ALTA, y la siguiente pantalla aparecerá:



En la pantalla anterior capture el área y presione el botón con la leyenda NUEVA, apareciendo a continuación:

MENU PRINCIPAL

Archivo Catálogos Operación Reportes

DETALLE PARA MATERIAS

AREA: MATEMATICAS

MATERIA:

HORAS / SEMANA:

SEMESTRE: 3

ACEPTAR SALIR

Capture la materia, horas a la semana y semestre para el cual se impartirá la materia y presione ACEPTAR.

Para dar de baja alguna materia:

- Seleccione el área correspondiente y presione el botón CAMBIO.
- Seleccione la materia y presione el botón BAJA.
- Presione el botón ACEPTAR en la pantalla DETALLE PARA MATERIAS.

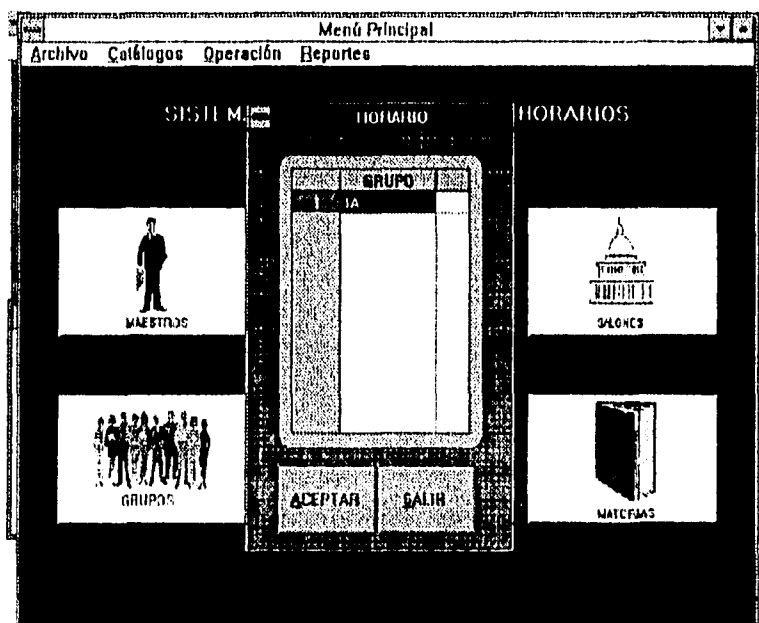
Para modificar los datos de una materia:

- Seleccione el área correspondiente y presione el botón CAMBIO.
- Seleccione la materia y presione el botón MODIFICAR
- Modifique la materia, horas a la semana y semestre y presione

el botón ACEPTAR.

5.3.5- Horario.

Oprima la subopción Horario o presione el botón con la leyenda HORARIO; a continuación aparecerá la siguiente ventana:



En la pantalla anterior se tienen botones para ACEPTAR y SALIR.

Seleccione el grupo y oprima la tecla ACEPTAR y la siguiente pantalla aparecerá:

Menú Principal

Archivo Catálogos Operación Reportes

DETALLE PARA HORARIOS

1A LUNES

MATEMATICAS CALCULO DIF E INTEGRAL

MAYOR	MENOR	HORA

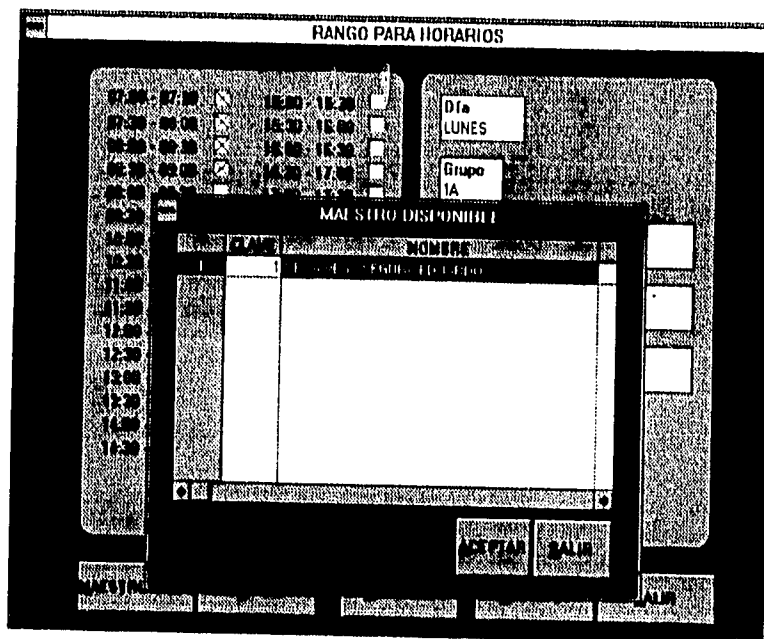
ALTA BORRAR CAMBIO ACEPTAR SALIR

En la pantalla anterior capture el día, área y materia y presione el botón con la leyenda ALTA, apareciendo a continuación:

RANGO PARA HORARIOS

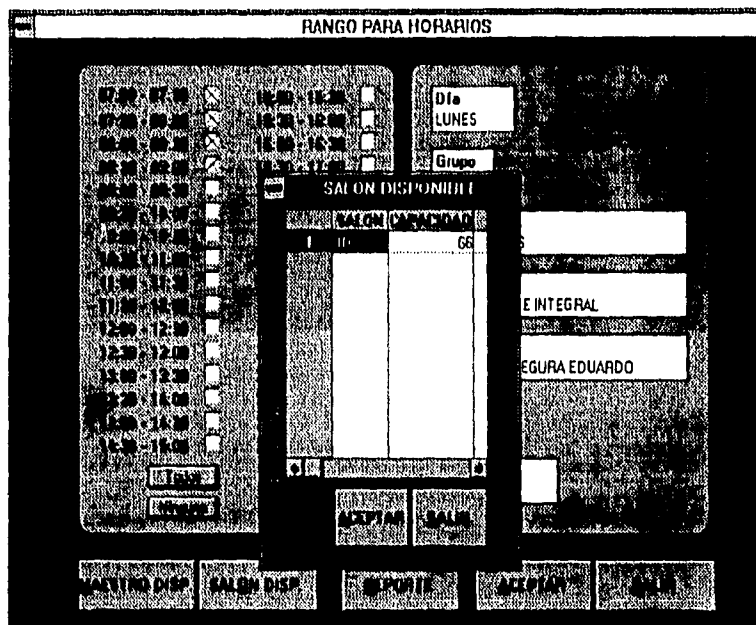
07:00 - 07:30	14:30 - 15:00		Día
07:30 - 08:00	15:00 - 15:30	<input type="checkbox"/>	LUNES
08:00 - 08:30	15:30 - 16:00	<input type="checkbox"/>	Grupo
08:30 - 09:00	16:00 - 16:30	<input type="checkbox"/>	1A
09:00 - 09:30	16:30 - 17:00	<input type="checkbox"/>	Area
09:30 - 10:00	17:00 - 17:30	<input type="checkbox"/>	MATEMATICAS
10:00 - 10:30	17:30 - 18:00	<input type="checkbox"/>	Materia
10:30 - 11:00	18:00 - 18:30	<input type="checkbox"/>	CALCULO DIF. E INTEGRAL
11:00 - 11:30	18:30 - 19:00	<input type="checkbox"/>	Maestro
11:30 - 12:00	19:00 - 19:30	<input type="checkbox"/>	
12:00 - 12:30	19:30 - 20:00	<input type="checkbox"/>	Salón
12:30 - 13:00	20:00 - 21:00	<input type="checkbox"/>	
13:00 - 13:30	21:00 - 21:30	<input type="checkbox"/>	Hora
13:30 - 14:00	21:30 - 22:00	<input type="checkbox"/>	
14:00 - 14:30	22:00 - 22:30	<input type="checkbox"/>	
14:30 - 15:00	22:30 - 23:00	<input type="checkbox"/>	

Seleccione el rango de horas deseado, a continuación presione la tecla MAESTRO DISP. ACEPTAR y aparecerá la siguiente pantalla, con los maestros disponibles para el día y rango de horas seleccionado:



Seleccione un maestro y presione la tecla ACEPTAR.

Presione el botón con la leyenda SALON DISP. y aparecerá la siguiente pantalla:



Seleccione un salón y presione el botón ACEPTAR.

A continuación presione el botón ACEPTAR de la pantalla RANGO PARA HORARIOS.

Para dar de baja alguna materia en horario:

- Seleccione el área correspondiente, materia y día, señale el detalle de horario que desea borrar y presione el botón BORRAR.

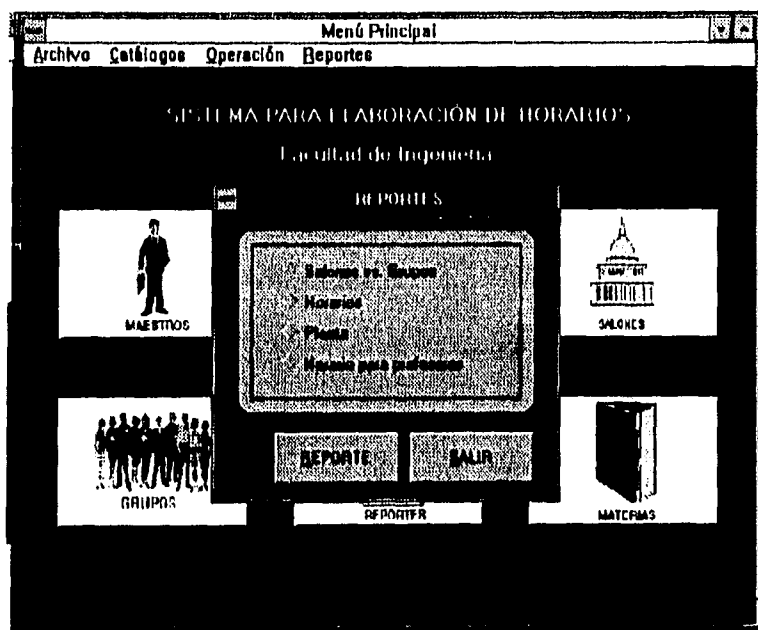
Para modificar los datos de una materia en horario:

- Seleccione el área correspondiente, materia y día, señale el detalle de horario que desea modificar y presione el botón CAMBIO.

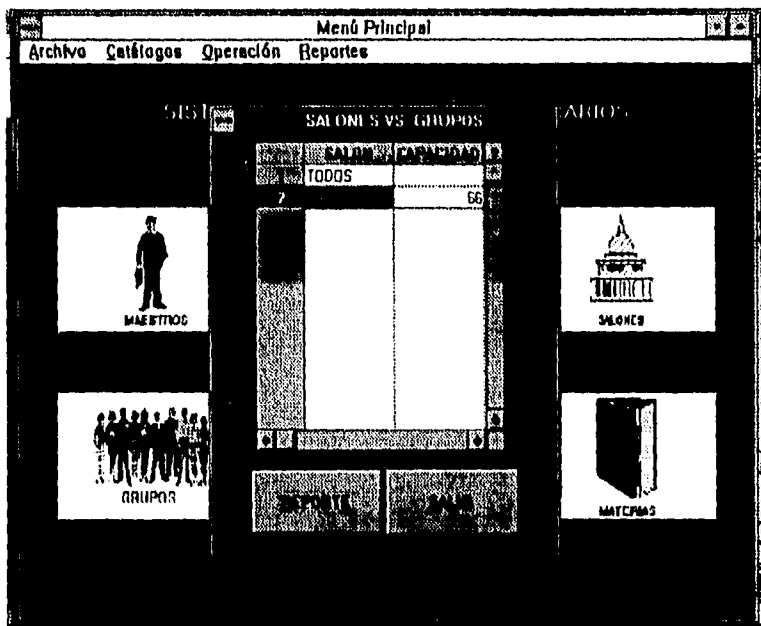
- Modifique los rangos de horas, así como el maestro y el salón disponibles y presione el botón ACEPTAR.
- Oprime el botón REPORTE para ver una presentación preliminar del horario que ha construido hasta ahora, para el grupo seleccionado.

5.3.6.- Reportes.

Oprima la subopción Reportes o presione el botón con la leyenda REPORTES; a continuación aparecerá la siguiente ventana:



En la pantalla anterior se tienen botones para REPORTE y SALIR. Seleccione el tipo de reporte Salones vs. Grupos y oprima la tecla REPORTE y la siguiente pantalla aparecerá:



En la pantalla anterior capture el salón, o bien seleccione la opción TODOS, y presione el botón con la leyenda REPORTE, apareciendo a continuación:

REPORTE DE SALONES VS GRUPOS

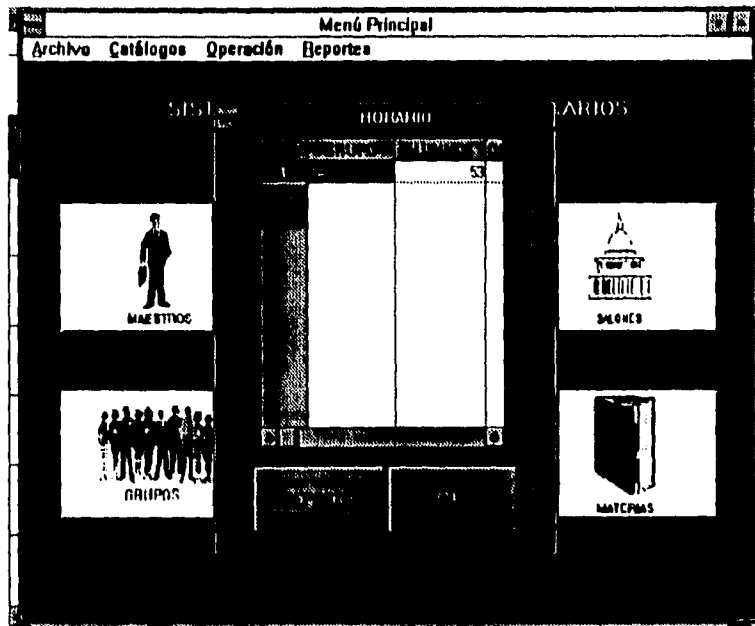
Archivo

10 66

07:00 - 07:30 1A			
07:30 - 08:00 1A			
08:00 - 08:30 1A			
08:30 - 09:00 1A			
09:00 - 09:30			
09:30 - 10:00			
10:00 - 10:30			
10:30 - 11:00			
11:00 - 11:30			
11:30 - 12:00			
12:00 - 12:30			
12:30 - 13:00			
13:00 - 13:30			
13:30 - 14:00			
14:00 - 14:30			
14:30 - 15:00			
15:00 - 15:30			
15:30 - 16:00			

Oprima el botón **IMPRIMIR**, a continuación se tendrá una presentación preliminar del reporte, para posteriormente imprimir el reporte.

Seleccione el tipo de reporte **Horario** y oprima la tecla **REPORTE** y la siguiente pantalla aparecerá:

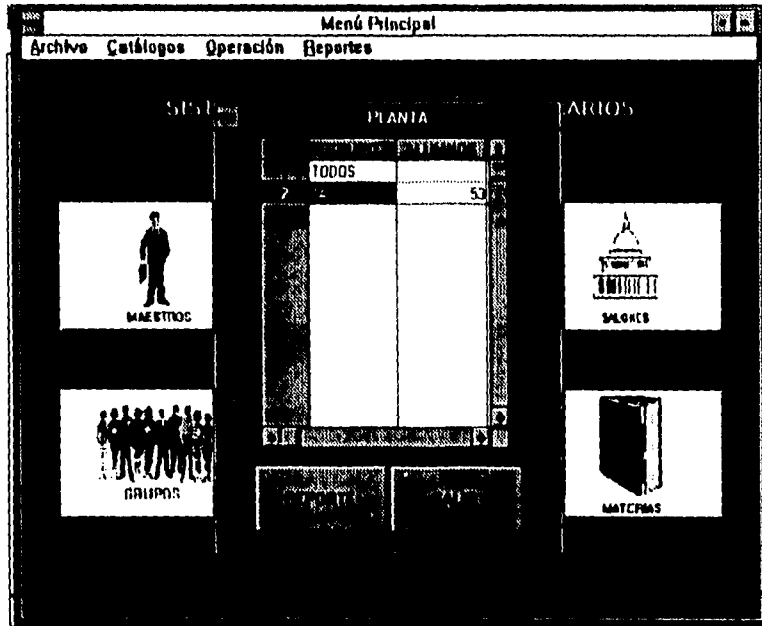


En la pantalla anterior seleccione el grupo y presione el botón con la leyenda **REPORTE**, apareciendo a continuación:

REPORTE DE HORARIOS				
1A	53	DE LA VEGA SEGURA EDUARDO		
CALCULO DIF E INTEG DE LA VEGA SEGURA 07.00 - 09.00 1D				

Oprima el botón IMPRIMIR, a continuación se tendrá una presentación preliminar del reporte, para posteriormente imprimir el reporte

Seleccione el tipo de reporte Planta y oprima la tecla REPORTE y la siguiente pantalla aparecerá:



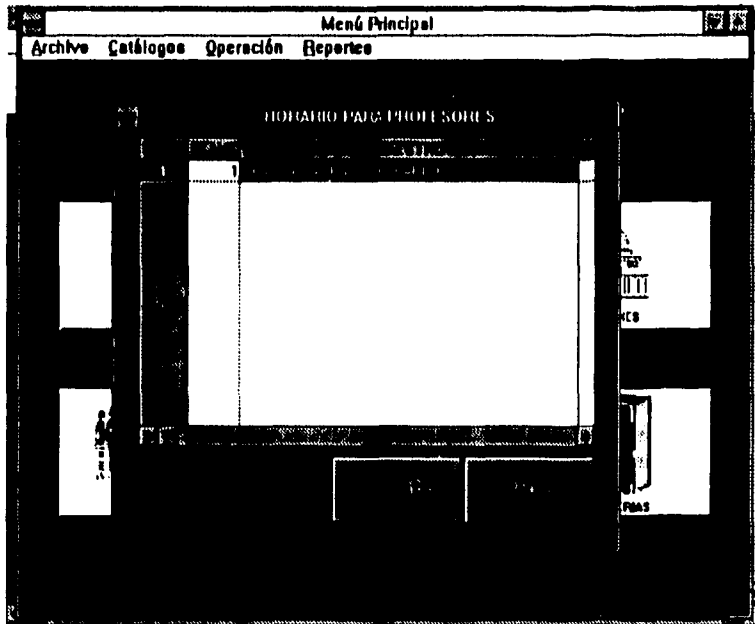
En la pantalla anterior seleccione el grupo y presione el botón con la leyenda REPORTE, apareciendo a continuación:

The screenshot shows a window titled "REPORTE PUNTA" with a table. The table has five columns: "COORDINADOR", "IA", "DE LA VEGA", "SEGURA", and "EDUARDO". The first row contains "MATEMATICAS" and "CALCULO DIF E INTEGRAL". The rest of the table is empty.

	COORDINADOR	IA	DE LA VEGA	SEGURA	EDUARDO
MATEMATICAS	CALCULO DIF E INTEGRAL				

Oprima el botón IMPRIMIR, a continuación se tendrá una presentación preliminar del reporte, para posteriormente imprimir el reporte.

Seleccione el tipo de reporte Horario para profesores y oprima la tecla REPORTE y la siguiente pantalla aparecerá:



En la pantalla anterior seleccione el maestro y presione el botón con la leyenda REPORTE, apareciendo a continuación:

REPORTE HORARIO MAESTROS				
DE LA VEGA SEGURA EDUARDO				
CALCULO DIF. E INTEG				
1A				
07.00 - 09.00				
1D				

Oprima el botón IMPRIMIR, a continuación se tendrá una presentación preliminar del reporte, para posteriormente imprimir el reporte.

CONCLUSIÓN

Una de las tareas principales de un Ingeniero Industrial es la de aprovechar al máximo los recursos de una empresa para el bien de la misma. Por lo tanto, lo que se pretende con este proyecto es de aprovechar un recurso, como lo son las computadoras, para facilitar una tarea importante dentro de la Universidad como lo es la elaboración de horarios y asignación de aulas.

Dado el volumen de información que se maneja y la complejidad que semestre a semestre se enfrentan los encargados de la elaboración de horarios, se planteó la posibilidad de modificar la forma de realizar esta labor.

Por lo anterior, se diseñó un sistema capaz de manejar sin complejidad alguna el manejo de información que implica la elaboración de horarios en la Universidad.

Esto se logró gracias a una herramienta gráfica que trabaja bajo ambiente Windows.

Es claro comentar que apoyarse en herramientas computacionales, es una labor que prevalece dentro de cualquier empresa, y de cualquier ramo.

De esta forma, en la Universidad se cuenta con el equipo necesario para desarrollar una tecnología que nos permita agilizar trámites administrativos, de cobranza o de cualquier operación dentro de la Institución.

Para el desarrollo del Sistema de Horarios, se tomaron en cuenta las etapas clásicas de desarrollo de un sistema: Análisis, Diseño, Pruebas, Liberación y Capacitación. Aunque ciertas etapas como la de liberación y de capacitación son tareas que, gracias al tamaño de sistema, se convierte en una actividad informal y requieren de poco tiempo.

En el desarrollo del proyecto se tienen toda clase de tecnicismos que, pudieran confundir al usuario, pero basta con una revisión al manual de usuario, para comprender de manera correcta el funcionamiento total del sistema, así como todas las ventajas que éste representa.

Aunque el usuario pudiera encontrar problemas en el tiempo de respuesta al incluir un número considerado de datos, éste podrá encontrar asesoría y apoyo por parte del desarrollador. Existe la vieja teoría de que no existen sistemas perfectos y tarde o temprano, se presentará alguna irregularidad en el sistema.

El sistema resulta fácil de manejar, y aún más para un usuario que tiene experiencia en manejo de windows.

Dentro de la etapa de implantación de cualquier sistema, existe una fase de acoplamiento que involucra un esfuerzo extra, ya que el usuario realiza en "paralelo" la actividad de costumbre y por otro lado aplica el sistema en propuesta. De esta manera, se pone en relieve las ventajas que tiene el sistema y de igual forma se aprecia la disminución en el tiempo de elaboración de esta actividad.

BIBLIOGRAFÍA

- YOURDON, Edward; Análisis Estructurado Moderno, México; De. Prantice Hall ; 1993;
- JAMES, Martin; Structured Techniques for Computing; Englewood Clift, N.J.; Ed. Prantice Hall; 1986; (3°).
- INMON, William; Classics in Software Engineering, Nueva York, Ed. Prantice Hall; 1992; (4°).
- DAVIS, William, Análisis y Diseño de Sistemas; Nueva York; Ed. Prentice Hall; 1983; (3°).
- KELLER, John; Herramientas de Modelado, Nueva York; Ed. McGraw Hill; 1985; (2°).
- JACKSON, Harry; Database Processing; Nueva York, Wiley; 1980;

APÉNDICE A

PROGRAMA : MAESTROS
OBJETIVO: PANTALLA PRINCIPAL CATALOGO MAESTROS

```
$include : "EVENTS.INC"  
DECLARE SUB open (Event, p1$, p2$)  
DECLARE SUB s_alta ()  
DECLARE SUB s_baja ()  
DECLARE SUB s_cambio ()  
DECLARE SUB s_llena ()  
SUB open (Event, p1$, p2$)  
  
select case (Event)  
case EVN_OPEN  
SQLSetTitles(0)  
ServerName$ = "sqlbase_10"  
UserName$ = ""  
Password$ = ""  
DatabaseName$ = "DEMO"  
ConnectNbr = GetVal("gws_connect")  
s_llena()  
exit sub  
  
case EVN_CLOSE  
  
case else  
  
end select  
END SUB  
  
SUB s_alta()  
SetString("gwc_opera$", "A")  
MODALPOPUP ("ALTAMAE")  
s_llena()  
END SUB  
  
SUB s_baja()  
IF GETCURREW("TB_MAESTROS1") > 0 THEN  
SetString("gwc_opera$", "B")  
MODALPOPUP ("ALTAMAE")  
s_llena()  
ELSE  
MESSAGE("Error", "Debe seleccionar un maestro para ser borrado.")  
EXIT SUB  
END IF  
END SUB  
  
SUB s_cambio()  
IF GETCURREW("TB_MAESTROS1") > 0 THEN  
SetString("gwc_opera$", "C")  
MODALPOPUP ("ALTAMAE")  
s_llena()  
ELSE  
MESSAGE("Error", "Debe seleccionar un maestro para ser modificado.")  
EXIT SUB  
END IF  
END SUB  
  
SUB s_llena()
```

```
CLEAR("EB_QRY")
CLEAR("TB_MAESTROS")
CLEAR("TB_MAESTROS1")
ADDLINE("EB_QRY","SELECT DISTINCT
CLAVE,APELLIDO1,APELLIDO2,NOMBRE1,NOMBRE2 "+
" FROM MAESTROS ORDER BY
APELLIDO1,APELLIDO2,NOMBRE1,NOMBRE2")
sqlquery(ConnectNbr,"TB_MAESTROS","EB_QRY")
FOR I = 1 TO GETNBROW("TB_MAESTROS")
  LSClave$ = GETTEXT$("TB_MAESTROS",I,1)
  LSApellido1$ = GETTEXT$("TB_MAESTROS",I,2)
  LSApellido2$ = GETTEXT$("TB_MAESTROS",I,3)
  LSNombre1$ = GETTEXT$("TB_MAESTROS",I,4)
  LSNombre2$ = GETTEXT$("TB_MAESTROS",I,5)
  ADDLINE("TB_MAESTROS1",LSClave$+" "+LSApellido1$+" "+LSApellido2$+"
"+LSNombre1$+" "+LSNombre2$)
NEXT I

END SUB
```

PROGRAMA : ALTA MAESTROS
OBJETIVO : MANTENIMIENTO DETALLE PARA MAESTROS

```
$include : "EVENTS.INC"  
DECLARE SUB open (Event, p1$, p2$)  
DECLARE SUB s_alta ()  
DECLARE SUB s_rangos ()  
DECLARE SUB s_dia ()  
DECLARE SUB s_salon ()  
SUB open (Event, p1$, p2$)  
  
select case (Event)  
case EVN_OPEN  
SQL.SetTitles(0)  
Connect = GetVal("gws_connect")  
LSOpera$ = GetString$("gwc_operas")  
CLEAR("df_diadisp")  
DISABLEINPUT("df_diadisp")  
IF LSOpera$ = "A" THEN  
SETFOCUS("df_nombre1")  
ELSE  
IF LSOpera$ = "B" OR LSOpera$ = "C" THEN  
hWnd = WinGetHandle()  
hParentWnd = WinGetParent(hWnd)  
hOldWnd = SwitchContext(hParentWnd)  
LLCurrow = getcurrow("TB_MAESTROS1")  
LSClave$ = gettext$("TB_MAESTROS",LLCurrow,1)  
LSApellido1$ = gettext$("TB_MAESTROS",LLCurrow,2)  
LSApellido2$ = gettext$("TB_MAESTROS",LLCurrow,3)  
LSNombre1$ = gettext$("TB_MAESTROS",LLCurrow,4)  
LSNombre2$ = gettext$("TB_MAESTROS",LLCurrow,5)  
SwitchContext(hOldWnd)  
Settext("df_clave",LSClave$)  
Settext("df_nombre1",LSNombre1$)  
Settext("df_nombre2",LSNombre2$)  
Settext("df_apellido1",LSApellido1$)  
Settext("df_apellido2",LSApellido2$)  
LSClave$ = gettext$("df_clave")  
LSNombreo1$ = gettext$("df_nombre1")  
LSNombreo2$ = gettext$("df_nombre2")  
LSApellidoo1$ = gettext$("df_apellido1")  
LSApellidoo2$ = gettext$("df_apellido2")  
IF LSOpera$ = "B" THEN  
DISABLEINPUT("df_nombre1")  
DISABLEINPUT("df_nombre2")  
DISABLEINPUT("df_apellido1")  
DISABLEINPUT("bu_apellido2")  
DISABLEINPUT("bu_rangos")  
ELSE  
SETFOCUS("df_nombre1")  
END IF  
END IF  
END IF  
SetCurPos("cb_dia",1)  
s_dia ()  
exit sub  
case EVN_CLOSE  
  
case else
```

```
end select  
END SUB
```

```
SUB s_aña()  
  LSClave$ = GETTEXT$("df_clave")  
  LSNombre1$ = Ltrim$(Rtrim$(gettext$("df_nombre1"))) )  
  LSNombre2$ = Ltrim$(Rtrim$(gettext$("df_nombre2"))) )  
  LSApellido1$ = Ltrim$(Rtrim$(gettext$("df_apellido1"))) )  
  LSApellido2$ = Ltrim$(Rtrim$(gettext$("df_apellido2"))) )  
  LSDiaDisp$ = getCurtext$("cb_dia")  
  IF LSNombre1$ = "" THEN  
    MESSAGE("Error", "Debe capturar el primer nombre por lo menos.")  
    SETFOCUS("df_nombre1")  
    EXIT SUB  
  END IF  
  IF LSApellido1$ = "" THEN  
    MESSAGE("Error", "Debe capturar el apellido paterno por lo menos.")  
    SETFOCUS("df_apellido1")  
    EXIT SUB  
  END IF  
  
  IF LSDiaDisp$ = "" THEN  
    MESSAGE("Error", "Debe capturar el dia disponible.")  
    SETFOCUS("cb_dia")  
    EXIT SUB  
  END IF  
  IF LSOpera$ = "A" THEN  
    CLEAR("EB_QRY")  
    CLEAR("df_qry")  
  ELSE  
    IF LSOpera$ = "B" THEN  
      CLEAR("EB_QRY")  
      CLEAR("df_qry")  
      ADDLINE("EB_QRY", "SELECT maestro FROM horario WHERE maestro = "  
        + LSClave$ )  
      sqlquery(Connect, "df_qry", "EB_QRY")  
      IF GETTEXT$("df_qry") <> "" THEN  
        MESSAGE("Error", "El maestro "+LSClave$+ " tiene un grupo asignado.")  
        EXIT SUB  
      END IF  
      CLEAR("EB_QRY")  
      ADDLINE("EB_QRY", "DELETE FROM maestros WHERE clave = " + LSClave$ )  
      sqlexec(Connect, "", "EB_QRY")  
    ELSE  
      SQLBegin(Connect)  
      CLEAR("EB_QRY")  
      ADDLINE("EB_QRY", "UPDATE maestros SET nombre1 = " + LSNombre1$ + ",")  
      ADDLINE("EB_QRY", " nombre2 = " + LSNombre2$ + ",")  
      ADDLINE("EB_QRY", " apellido1 = " + LSApellido1$ + ",")  
      ADDLINE("EB_QRY", " apellido2 = " + LSApellido2$ + ",")  
      ADDLINE("EB_QRY", " WHERE clave = " + LSClave$ )  
      sqlexec(Connect, "", "EB_QRY")  
      SQLCommit(Connect)  
    END IF  
  END IF  
  close()  
END SUB
```

```

SUB s_rangos ()
  LSClave$ = gettext$("df_clave")
  LSNombre1$ = gettext$("df_nombre1")
  LSNombre2$ = gettext$("df_nombre2")
  LSApellido1$ = gettext$("df_apellido1")
  LSApellido2$ = gettext$("df_apellido2")
  LSDiaDisp$ = getCurtext$("cb_dia")
  LSDia$ = GETTEXT$("df_diadisp")
  IF LSClave$ = "" THEN
    SETSTRING("gws_opc","A")
  ELSE
    SETSTRING("gws_opc","C")
  END IF
  IF LSNombre1$ = "" THEN
    MESSAGE("Error","Debe capturar el primer nombre por lo menos.")
    SETFOCUS("df_nombre1")
    EXIT SUB
  END IF
  IF LSApellido1$ = "" THEN
    MESSAGE("Error","Debe capturar el apellido paterno por lo menos.")
    SETFOCUS("df_apellido1")
    EXIT SUB
  END IF
  IF LSDiaDisp$ = "" THEN
    MESSAGE("Error","Debe capturar el día disponible.")
    SETFOCUS("cb_dia")
    EXIT SUB
  END IF
  SETSTRING("gws_clave",LSClave$)
  SETSTRING("gwc_nombre1",LSNombre1$)
  SETSTRING("gwc_nombre2",LSNombre2$)
  SETSTRING("gwc_apellido1",LSApellido1$)
  SETSTRING("gwc_apellido2",LSApellido2$)
  SETSTRING("gwc_dia",LSDia$)
  SETSTRING("gwc_diadisp",LSDiadisp$)
  IF GETTEXT$("tb_rangos",1,1) <> "" THEN
    LSRangos$ = "1"
  ELSE
    LSRangos$ = "0"
  END IF
  SETSTRING("gwc_rangos",LSRangos$)
  MODALPOPUP("RANGOSM")
  LSClave$ = GETSTRING$("gws_clave")
  SETTEXT ("df_clave",LSClave$)
END SUB

```

```

SUB s_dia()
  CLEAR("tb_rangos")
  LSDia$ = GetCurText$("cb_dia")
  LSClave$ = Gettext$("df_clave")
  IF LSDia$ = "1" THEN
    SetText("df_diadisp","LUNES")
  END IF
  IF LSDia$ = "2" THEN
    SetText("df_diadisp","MARTES")
  END IF
  IF LSDia$ = "3" THEN
    SetText("df_diadisp","MIERCOLES")
  END IF
  IF LSDia$ = "4" THEN

```



```
    SetText("df_diadisp","JUEVES")
END IF
IF LSDia$ = "5" THEN
    SetText("df_diadisp","VIERNES")
END IF
IF LSDia$ = "6" THEN
    SetText("df_diadisp","SABADO")
END IF
IF LSDia$ = "7" THEN
    SetText("df_diadisp","DOMINGO")
END IF
CLEAR("EB_QRY")
ADDLINE("EB_QRY","SELECT hora_disp FROM maestros ")
ADDLINE("EB_QRY"," WHERE clave = " + "" + LSClave$ + """)
ADDLINE("EB_QRY"," AND dia_disp = " + "" + LSDia$ + """)
SQLQUERY(Connect,"tb_rangos","EB_QRY")
END SUB
```

PROGRAMA : RANGOS PARA MAESTROS
OBJETIVO: RANGO DISPONIBLE DE HORAS PARA MAESTROS

```
$include : "EVENTS.INC"  
DECLARE SUB open (Event, p1$, p2$)  
DECLARE SUB s_alta ()  
DECLARE SUB s_todos ()  
DECLARE SUB s_ninguno ()  
DECLARE SUB s_todos1 ()  
DECLARE SUB s_ninguno1 ()  
DECLARE SUB s_todos2 ()  
DECLARE SUB s_lista ()  
DECLARE SUB s_llena_rangos ()  
DECLARE SUB s_conteo ()  
SUB open (Event, p1$, p2$)  
  
select case (Event)  
case EVN_OPEN  
    Connect = GetVal("gws_connect")  
    LSOpc$ = GETSTRING$("gws_opc")  
    LSClave$ = GETSTRING$("gws_clave")  
    LSNombre1$ = GETSTRING$("gwc_nombre1")  
    LSNombre2$ = GETSTRING$("gwc_nombre2")  
    LSApellido1$ = GETSTRING$("gwc_apellido1")  
    LSApellido2$ = GETSTRING$("gwc_apellido2")  
    LSDia$ = GETSTRING$("gwc_dia")  
    LSDiadisp$ = GETSTRING$("gwc_diadisp")  
    LSCapacidad$ = GETSTRING$("gwc_capacidad")  
    SETTEXT("df_clave",LSClave$)  
    SET TEXT("df_nombre1",LSNombre1$)  
    SETTEXT("df_nombre2",LSNombre2$)  
    SETTEXT("df_apellido1",LSApellido1$)  
    SETTEXT("df_apellido2",LSApellido2$)  
    SETTEXT("df_dia",LSDia$)  
    SETTEXT("df_diadisp",LSDiadisp$)  
    SETTEXT("df_capacidad",LSCapacidad$)  
    LSRangos$ = GETSTRING$("gwc_rangos$")  
    hWnd = WinGetHandle ()  
    hParentWnd = WinGetParent (hWnd)  
    ObjCopyDataW(hParentWnd, "tb_rangos", hWnd, "tb_rangos")  
    ObjCopyDataW(hParentWnd, "tb_rangos", hWnd, "tb_rangos1")  
    IF LSRangos$ = "1" THEN  
        s_llena_rangos ()  
    END IF  
  
case EVN_CLOSE  
    IF LLBan = 1 THEN  
        FOR i = 1 TO GetNBROW("tb_rangos1")  
            LSHoraDisp$ = GETTEXT$("tb_rangos1",i,1)  
            CLEAR("EB_QRY")  
            ADDLINE("EB_QRY","INSERT INTO maestros VALUES (" + LSClave$  
                + ",'" + LSNombre1$ + "','" + LSNombre2$ + "','" +  
                "' + LSApellido1$ + "','" +  
                LSApellido2$ + "','" + LSDiaDisp$ + "','" + LSHoraDisp$ + "')")  
            sqlexec(Connect,"","EB_QRY")  
        NEXT i  
        hWndSrc = WinGetHandle ()  
        hWndDest = WinGetParent (hWndSrc)  
        ObjCopyDataW(hWndSrc, "tb_rangos1", hWndDest, "tb_rangos")
```

```
END IF
case else

end select
END SUB
```

```
SUB s_alta()
IF LSOps$ = "A" THEN
CLEAR("EB_QRY")
CLEAR("df_qry")
ADDLINE("EB_QRY","SELECT MAX (clave) FROM maestros ")
sqlquery(Connect,"df_qry","EB_QRY")
LSClave$ = GETTEXT$("df_qry")
LLClave = VAL(LSClave$) + 1
SETTEXT("df_clave",str$(LLClave))
SETSTRING("gws_clave",str$(LLClave))
END IF
LSClave$ = gettext$("df_clave")
LSNombre1$ = Ltrim$(Rtrim$(Gettext$("df_nombre1")))
LSNombre2$ = Ltrim$(Rtrim$(gettext$("df_nombre2")))
LSApellido1$ = Ltrim$(Rtrim$(gettext$("df_apellido1")))
LSApellido2$ = Ltrim$(Rtrim$(gettext$("df_apellido2")))
CLEAR("tb_horas")
IF ButtonCheck("rb_1") = 1 THEN
LSHora$ = gettext$("r1")
Addline("tb_horas",LSHora$)
END IF
IF ButtonCheck("rb_2") = 1 THEN
LSHora$ = gettext$("r2")
Addline("tb_horas",LSHora$)
END IF
IF ButtonCheck("rb_3") = 1 THEN
LSHora$ = gettext$("r3")
Addline("tb_horas",LSHora$)
END IF
IF ButtonCheck("rb_4") = 1 THEN
LSHora$ = gettext$("r4")
Addline("tb_horas",LSHora$)
END IF
IF ButtonCheck("rb_5") = 1 THEN
LSHora$ = gettext$("r5")
Addline("tb_horas",LSHora$)
END IF
IF ButtonCheck("rb_6") = 1 THEN
LSHora$ = gettext$("r6")
Addline("tb_horas",LSHora$)
END IF
IF ButtonCheck("rb_7") = 1 THEN
LSHora$ = gettext$("r7")
Addline("tb_horas",LSHora$)
END IF
IF ButtonCheck("rb_8") = 1 THEN
LSHora$ = gettext$("r8")
Addline("tb_horas",LSHora$)
END IF
IF ButtonCheck("rb_9") = 1 THEN
LSHora$ = gettext$("r9")
Addline("tb_horas",LSHora$)
END IF
```

```
IF ButtonCheck("rb_10") = 1 THEN
  LSHora$ = gettext$("r10")
  Addline("lb_horas",LSHora$)
END IF
IF ButtonCheck("rb_11") = 1 THEN
  LSHora$ = gettext$("r11")
  Addline("lb_horas",LSHora$)
END IF
IF ButtonCheck("rb_12") = 1 THEN
  LSHora$ = gettext$("r12")
  Addline("lb_horas",LSHora$)
END IF
IF ButtonCheck("rb_13") = 1 THEN
  LSHora$ = gettext$("r13")
  Addline("lb_horas",LSHora$)
END IF
IF ButtonCheck("rb_14") = 1 THEN
  LSHora$ = gettext$("r14")
  Addline("lb_horas",LSHora$)
END IF
IF ButtonCheck("rb_15") = 1 THEN
  LSHora$ = gettext$("r15")
  Addline("lb_horas",LSHora$)
END IF
IF ButtonCheck("rb_16") = 1 THEN
  LSHora$ = gettext$("r16")
  Addline("lb_horas",LSHora$)
END IF
IF ButtonCheck("rb_17") = 1 THEN
  LSHora$ = gettext$("r17")
  Addline("lb_horas",LSHora$)
END IF
IF ButtonCheck("rb_18") = 1 THEN
  LSHora$ = gettext$("r18")
  Addline("lb_horas",LSHora$)
END IF
IF ButtonCheck("rb_19") = 1 THEN
  LSHora$ = gettext$("r19")
  Addline("lb_horas",LSHora$)
END IF
IF ButtonCheck("rb_20") = 1 THEN
  LSHora$ = gettext$("r20")
  Addline("lb_horas",LSHora$)
END IF
IF ButtonCheck("rb_21") = 1 THEN
  LSHora$ = gettext$("r21")
  Addline("lb_horas",LSHora$)
END IF
IF ButtonCheck("rb_22") = 1 THEN
  LSHora$ = gettext$("r22")
  Addline("lb_horas",LSHora$)
END IF
IF ButtonCheck("rb_23") = 1 THEN
  LSHora$ = gettext$("r23")
  Addline("lb_horas",LSHora$)
END IF
IF ButtonCheck("rb_24") = 1 THEN
  LSHora$ = gettext$("r24")
  Addline("lb_horas",LSHora$)
END IF
```

```

IF ButtonCheck("rb_25") = 1 THEN
  LSHora$ = gettext$("r25")
  Addline("tb_horas",LSHora$)
END IF
IF ButtonCheck("rb_26") = 1 THEN
  LSHora$ = gettext$("r26")
  Addline("tb_horas",LSHora$)
END IF
IF ButtonCheck("rb_27") = 1 THEN
  LSHora$ = gettext$("r27")
  Addline("tb_horas",LSHora$)
END IF
IF ButtonCheck("rb_28") = 1 THEN
  LSHora$ = gettext$("r28")
  Addline("tb_horas",LSHora$)
END IF
IF ButtonCheck("rb_29") = 1 THEN
  LSHora$ = gettext$("r29")
  Addline("tb_horas",LSHora$)
END IF
IF ButtonCheck("rb_30") = 1 THEN
  LSHora$ = gettext$("r30")
  Addline("tb_horas",LSHora$)
END IF
IF ButtonCheck("rb31") = 1 THEN
  LSHora$ = gettext$("r31")
  Addline("tb_horas",LSHora$)
END IF
IF ButtonCheck("rb_32") = 1 THEN
  LSHora$ = gettext$("r32")
  Addline("tb_horas",LSHora$)
END IF
s_conteo ()
LSDia$ = gettext$("df_dia")
LSDiaDisp$ = gettext$("df_diadisp")
CLEAR("EB_QRY")
SQLBegtran(Connect)

```

```

ADDLINE("EB_QRY","DELETE FROM maestros ")
ADDLINE("EB_QRY"," WHERE clave = " + LSClave$ )
ADDLINE("EB_QRY"," AND dia_disp = "+LSDiaDisp$)
sqlexec(Connect,"","EB_QRY")
SQLCommit(Connect)
LLBan = 0
CLEAR("EB_QRY")
CLEAR("tb_horas_ocu")
ADDLINE("EB_QRY","SELECT hora_disp FROM horario ")
ADDLINE("EB_QRY"," WHERE dia = "+LSDiaDisp$)
ADDLINE("EB_QRY"," AND maestro = "+LSClave$+"")
sqlquery(Connect,"tb_horas_ocu","EB_QRY")
FOR k = 1 TO GETNBROW("tb_horas_ocu")
  LSHI$ = MID$(GETTEXT$("tb_horas_ocu",k,1),1,5)
  LSHF$ = MID$(GETTEXT$("tb_horas_ocu",k,1),9,5)
  LLBan = 1
  FOR i = 1 TO GetNBROW("tb_rangos")
    LSHoraDisp$ = GETTEXT$("tb_rangos",i,1)
    LSHIa$ = MID$(LSHoraDisp$,1,5) 'hora inicial asignada
    LSHFa$ = MID$(LSHoraDisp$,9,5) 'Hora final asignada
    IF (LSHI$ >= LSHIa$) AND (LSHF$ <= LSHFa$) THEN

```

```

        LLBan = 0
    END IF
    NEXT i
    IF LLban = 1 THEN
        MESSAGE("Error","Existen grupos con el horario anterior ya asignados, no podrá
hacer el cambio.")
    EXIT SUB
    END IF
    NEXT k

```

```

FOR I = 1 TO GetNBROW("tb_rangos")
    CLEAR("EB_QRY")
    LSHoraDisp$ = GETTEXT$("tb_rangos",I,1)
    SQLBegtran(Connect)
    ADDLINE("EB_QRY","INSERT INTO maestros VALUES (" + LSClave$
        + "," + LSNombre1$ + "," + LSNombre2$ + "," +
        "" + LSApellido1$ + "," + LSApellido2$ + "," + LSDiaDisp$ + "," + "" + LSHoraDisp$ + ")")
    sqlxec(Connect,"","EB_QRY")
    CLEAR("EB_QRY")
    LSQry1$ = "UPDATE maestros SET nombre1 = " + "" + LSNombre1$ + "," +
        " nombre2 = " + "" + LSNombre2$ + "," +
    LSQry2$ = " apellido1 = " + "" + LSApellido1$ + "," +
        " apellido2 = " + "" + LSApellido2$ + "" +
        " where clave = " + LSClave$
    ADDLINE("EB_QRY",LSQry1$+LSQry2$)
    sqlxec(Connect,"","EB_QRY")
    SQLCommit(Connect)
    NEXT i

CLOSE ()

```

END SUB

```

SUB s_todos ()
    setcheck("rb_1","1")
    setcheck("rb_2","1")
    setcheck("rb_3","1")
    setcheck("rb_4","1")
    setcheck("rb_5","1")
    setcheck("rb_6","1")
    setcheck("rb_7","1")
    setcheck("rb_8","1")
    setcheck("rb_9","1")
    setcheck("rb_10","1")
    setcheck("rb_11","1")
    setcheck("rb_12","1")
    setcheck("rb_13","1")
    setcheck("rb_14","1")
    setcheck("rb_15","1")
    setcheck("rb_16","1")
END SUB

```

```

SUB s_ninguno ()
    uncheck("rb_1")
    uncheck("rb_2")
    uncheck("rb_3")
    uncheck("rb_4")

```

```
uncheck("rb_5")
uncheck("rb_6")
uncheck("rb_7")
uncheck("rb_8")
uncheck("rb_9")
uncheck("rb_10")
uncheck("rb_11")
uncheck("rb_12")
uncheck("rb_13")
uncheck("rb_14")
uncheck("rb_15")
uncheck("rb_16")
END SUB
```

```
SUB s_ninguno1 ()
uncheck("rb_17")
uncheck("rb_18")
uncheck("rb_19")
uncheck("rb_20")
uncheck("rb_21")
uncheck("rb_22")
uncheck("rb_23")
uncheck("rb_24")
uncheck("rb_25")
uncheck("rb_26")
uncheck("rb_27")
uncheck("rb_28")
uncheck("rb_29")
uncheck("rb_30")
uncheck("rb31")
uncheck("rb_32")
END SUB
```

```
SUB s_todos1 ()
setcheck("rb_17","1")
setcheck("rb_18","1")
setcheck("rb_19","1")
setcheck("rb_20","1")
setcheck("rb_21","1")
setcheck("rb_22","1")
setcheck("rb_23","1")
setcheck("rb_24","1")
setcheck("rb_25","1")
setcheck("rb_26","1")
setcheck("rb_27","1")
setcheck("rb_28","1")
setcheck("rb_29","1")
setcheck("rb_30","1")
setcheck("rb31","1")
setcheck("rb_32","1")
END SUB
```

```
SUB s_conteo ()
CLEAR("lb_rangos")
LSHorai$ = left$(gettext$("lb_horas",1,1),5)
LSHoralf$ = Mid$(gettext$("lb_horas",1,1),9,5)
FOR i = 2 TO GetNrow("lb_horas")
    LSHora$ = left$(gettext$("lb_horas",i,1),5)
    IF LSHora$ = LSHoralf$ THEN
```

```

    LSHoraii$ = Mid$(gettext$("tb_horas",i,1),9,5)
ELSE
    LSRango$ = LSHoraii$ + " - " + LSHoraii$
    ADDLINE("tb_rangos",LSRango$)
    LSHoraii$ = LSHora$
    LSHoraii$ = Mid$(gettext$("tb_horas",i,1),9,5)
END IF
NEXT i
LSRango$ = LSHoraii$ + " - " + LSHoraii$
ADDLINE("tb_rangos",LSRango$)
hWndSrc = WinGetHandle ()
hWndDest = WinGetParent (hWndSrc)
ObjCopyDataW(hWndSrc, "tb_rangos", hWndDest, "tb_rangos")
END SUB

SUB s_llena_rangos ()
FOR i = 1 TO GETNBROW("tb_rangos")
    LSRango$ = LEFT$(GETTEXT$("tb_rangos",i,1),5)
    LSRango1$ = MID$(GETTEXT$("tb_rangos",i,1),9,5)
    FOR j = 1 TO 32
        LSTexto$ = LEFT$(GETTEXT$("r"+str$(j)),5)
        LSTexto1$ = MID$(GETTEXT$("r"+str$(j)),9,5)
        IF (LSRango$ <= LSTexto$ AND LSRango1$ >= LSTexto1$) THEN
            setcheck("rb_"+str$(j),"1")
        END IF
    NEXT j
    LSTexto$ = LEFT$(GETTEXT$("r31"),5)
    LSTexto1$ = MID$(GETTEXT$("r31"),9,5)
    IF (LSRango$ <= LSTexto$ AND LSRango1$ >= LSTexto1$) THEN
        setcheck("rb31","1")
    END IF
NEXT i
END SUB

```


PROGRAMA : GRUPOS
OBJETIVO: PANTALLA PRINCIPAL CATALOGO GRUPOS

```
$include : "EVENTS.INC"  
DECLARE SUB open (Event, p1$, p2$)  
DECLARE SUB s_alta ()  
DECLARE SUB s_baja ()  
DECLARE SUB s_cambio ()  
DECLARE SUB s_llena ()  
SUB open (Event, p1$, p2$)  
  
select case (Event)  
case EVN_OPEN  
    SQLSetTitles(0)  
    ServerName$ = "sqlbase_10"  
    UserName$ = ""  
    Password$ = ""  
    DatabaseName$ = "DEMO"  
    ConnectNbr = GetVal("gws_connect")  
    s_llena()  
exit sub  
  
case EVN_CLOSE  
  
case else  
  
end select  
END SUB  
  
SUB s_alta()  
    SetString("gwc_operas","A")  
    MODALPOPUP ("ALTAGRU")  
    s_llena()  
END SUB  
  
SUB s_baja()  
    IF GETCURROW("TB_GRUPOS") > 0 THEN  
        SetString("gwc_operas","B")  
        MODALPOPUP ("ALTAGRU")  
        s_llena()  
    ELSE  
        MESSAGE("Error","Debe seleccionar un grupo para ser borrado.")  
    EXIT SUB  
    END IF  
END SUB  
  
SUB s_cambio()  
    IF GETCURROW("TB_GRUPOS") > 0 THEN  
        SetString("gwc_operas","C")  
        MODALPOPUP ("ALTAGRU")  
        s_llena()  
    ELSE  
        MESSAGE("Error","Debe seleccionar un grupo para ser modificado.")  
    EXIT SUB  
    END IF  
END SUB
```

END SUB

SUB s_llena()

CLEAR("EB_QRY")

ADDLINE("EB_QRY","SELECT DISTINCT GRUPO,NUMERO,COORDINADOR "+
" FROM GRUPOS")

sqlquery(ConnectNbr,"TB_GRUPOS","EB_QRY")

END SUB

PROGRAMA : RANGOS
OBJETIVO: VALIDACION PARA RANGOS DE HORAS

```
$include : "EVENTS.INC"  
DECLARE SUB open (Event, p1$, p2$)  
DECLARE SUB s_alfa ()  
DECLARE SUB s_todos ()  
DECLARE SUB s_ninguno ()  
DECLARE SUB s_todos1 ()  
DECLARE SUB s_ninguno1 ()  
DECLARE SUB s_todos2 ()  
DECLARE SUB s_lista ()  
DECLARE SUB s_llena_rangos ()  
DECLARE SUB s_conteo ()  
SUB open (Event, p1$, p2$)  
  
select case (Event)  
case EVN_OPEN  
    Connect = GetVal("gws_connect")  
    LSSalon$ = GETSTRING$("gwc_salon")  
    LSDia$ = GETSTRING$("gwc_dia")  
    LSDiadisp$ = GETSTRING$("gwc_diadisp")  
    LSCapacidad$ = GETSTRING$("gwc_capacidad")  
    SETTEXT("df_salon",LSSalon$)  
    SETTEXT("df_dia",LSDia$)  
    SETTEXT("df_diadisp",LSDiadisp$)  
    SETTEXT("df_capacidad",LSCapacidad$)  
    LSRangos$ = GETSTRING$("gwc_rangos")  
    hWnd = WinGetHandle ()  
    hParentWnd = WinGetParent (hWnd)  
    ObjCopyDataW(hParentWnd, "tb_rangos", hWnd, "tb_rangos")  
    ObjCopyDataW(hParentWnd, "tb_rangos", hWnd, "tb_rangos1")  
    IF LSRangos$ = "1" THEN  
        s_llena_rangos ()  
    END IF  
  
case EVN_CLOSE  
    IF LLBan = 1 THEN  
        FOR i = 1 TO GetNBROW("tb_rangos1")  
            LSHoraDisp$ = GETTEXT$("tb_rangos1",i,1)  
            CLEAR("EB_QRY")  
            ADDLINE("EB_QRY","INSERT INTO salones VALUES  
(*+LSSalon$+*"+LSCapacidad$+*"+*"+LSDiaDisp$+*"+*"+LSHoraDisp$+*")  
            sqlexec(Connect,"EB_QRY")  
        NEXT i  
        hWndSrc = WinGetHandle ()  
        hWndDest = WinGetParent (hWndSrc)  
        ObjCopyDataW(hWndSrc, "tb_rangos1", hWndDest, "tb_rangos")  
    END IF  
case else  
  
end select  
END SUB  
  
SUB s_alfa()  
    CLEAR("tb_horas")  
    IF ButtonCheck("rb_1") = 1 THEN
```

```
LSHora$ = gettext$("r1")
Addline("tb_horas",LSHora$)
END IF
IF ButtonCheck("rb_2") = 1 THEN
LSHora$ = gettext$("r2")
Addline("tb_horas",LSHora$)
END IF
IF ButtonCheck("rb_3") = 1 THEN
LSHora$ = gettext$("r3")
Addline("tb_horas",LSHora$)
END IF
IF ButtonCheck("rb_4") = 1 THEN
LSHora$ = gettext$("r4")
Addline("tb_horas",LSHora$)
END IF
IF ButtonCheck("rb_5") = 1 THEN
LSHora$ = gettext$("r5")
Addline("tb_horas",LSHora$)
END IF
IF ButtonCheck("rb_6") = 1 THEN
LSHora$ = gettext$("r6")
Addline("tb_horas",LSHora$)
END IF
IF ButtonCheck("rb_7") = 1 THEN
LSHora$ = gettext$("r7")
Addline("tb_horas",LSHora$)
END IF
IF ButtonCheck("rb_8") = 1 THEN
LSHora$ = gettext$("r8")
Addline("tb_horas",LSHora$)
END IF
IF ButtonCheck("rb_9") = 1 THEN
LSHora$ = gettext$("r9")
Addline("tb_horas",LSHora$)
END IF
IF ButtonCheck("rb_10") = 1 THEN
LSHora$ = gettext$("r10")
Addline("tb_horas",LSHora$)
END IF
IF ButtonCheck("rb_11") = 1 THEN
LSHora$ = gettext$("r11")
Addline("tb_horas",LSHora$)
END IF
IF ButtonCheck("rb_12") = 1 THEN
LSHora$ = gettext$("r12")
Addline("tb_horas",LSHora$)
END IF
IF ButtonCheck("rb_13") = 1 THEN
LSHora$ = gettext$("r13")
Addline("tb_horas",LSHora$)
END IF
IF ButtonCheck("rb_14") = 1 THEN
LSHora$ = gettext$("r14")
Addline("tb_horas",LSHora$)
END IF
IF ButtonCheck("rb_15") = 1 THEN
LSHora$ = gettext$("r15")
Addline("tb_horas",LSHora$)
END IF
IF ButtonCheck("rb_16") = 1 THEN
```

```
LSHora$ = gettext("r16")
Addline("lb_horas",LSHora$)
END IF
IF ButtonCheck("rb_17") = 1 THEN
LSHora$ = gettext("r17")
Addline("lb_horas",LSHora$)
END IF
IF ButtonCheck("rb_18") = 1 THEN
LSHora$ = gettext("r18")
Addline("lb_horas",LSHora$)
END IF
IF ButtonCheck("rb_19") = 1 THEN
LSHora$ = gettext("r19")
Addline("lb_horas",LSHora$)
END IF
IF ButtonCheck("rb_20") = 1 THEN
LSHora$ = gettext("r20")
Addline("lb_horas",LSHora$)
END IF
IF ButtonCheck("rb_21") = 1 THEN
LSHora$ = gettext("r21")
Addline("lb_horas",LSHora$)
END IF
IF ButtonCheck("rb_22") = 1 THEN
LSHora$ = gettext("r22")
Addline("lb_horas",LSHora$)
END IF
IF ButtonCheck("rb_23") = 1 THEN
LSHora$ = gettext("r23")
Addline("lb_horas",LSHora$)
END IF
IF ButtonCheck("rb_24") = 1 THEN
LSHora$ = gettext("r24")
Addline("lb_horas",LSHora$)
END IF
IF ButtonCheck("rb_25") = 1 THEN
LSHora$ = gettext("r25")
Addline("lb_horas",LSHora$)
END IF
IF ButtonCheck("rb_26") = 1 THEN
LSHora$ = gettext("r26")
Addline("lb_horas",LSHora$)
END IF
IF ButtonCheck("rb_27") = 1 THEN
LSHora$ = gettext("r27")
Addline("lb_horas",LSHora$)
END IF
IF ButtonCheck("rb_28") = 1 THEN
LSHora$ = gettext("r28")
Addline("lb_horas",LSHora$)
END IF
IF ButtonCheck("rb_29") = 1 THEN
LSHora$ = gettext("r29")
Addline("lb_horas",LSHora$)
END IF
IF ButtonCheck("rb_30") = 1 THEN
LSHora$ = gettext("r30")
Addline("lb_horas",LSHora$)
END IF
IF ButtonCheck("rb31") = 1 THEN
```

```

    LSHora$ = gettext$("r31")
    Addline("tb_horas",LSHora$)
END IF
IF ButtonCheck("rb_32") = 1 THEN
    LSHora$ = gettext$("r32")
    Addline("tb_horas",LSHora$)
END IF
s_conteo ()
LSSalon$ = Ltrim$(Rtrim$(gettext$("df_salon")))
LSCapacidad$ = Ltrim$(Rtrim$(gettext$("df_capacidad")))
LSDia$ = Ltrim$(Rtrim$(gettext$("df_dia")))
LSDiaDisp$ = gettext$("df_diadisp")
CLEAR("EB_QRY")
ADDLINE("EB_QRY","DELETE FROM salones WHERE salon = "
+""+LSSalon$+""+
" AND dia_disp = "+LSDiaDisp$)
sqlexec(Connect,"","EB_QRY")
LLBan = 0
CLEAR("EB_QRY")
CLEAR("tb_horas_ocu")
ADDLINE("EB_QRY","SELECT hora_disp FROM horario ")
ADDLINE("EB_QRY"," WHERE dia = "+LSDiaDisp$)
ADDLINE("EB_QRY"," AND salon = "+LSSalon$+""")
sqlquery(Connect,"tb_horas_ocu","EB_QRY")
FOR k = 1 TO GETNBROW("tb_horas_ocu")
    LSHi$ = MID$(GETTEXT$("tb_horas_ocu",k,1),1,5)
    LSHF$ = MID$(GETTEXT$("tb_horas_ocu",k,1),9,5)
    LLBan = 1
    FOR i = 1 TO GetNBROW("tb_rangos")
        LSHoraDisp$ = GETTEXT$("tb_rangos",i,1)
        LSHIa$ = MID$(LSHoraDisp$,1,5) 'hora inicial asignada
        LSHFa$ = MID$(LSHoraDisp$,9,5) 'hora final asignada
        IF (LSHI$ >= LSHIa$) AND (LSHF$ <= LSHFa$) THEN
            LLBan = 0
        END IF
    NEXT i
    IF LLBan = 1 THEN
        MESSAGE("Error","Existen grupos con el horario anterior ya asignados, no podr 
        hacer el cambio.")
    END SUB
END IF
NEXT k

FOR i = 1 TO GetNBROW("tb_rangos")
    LSHoraDisp$ = GETTEXT$("tb_rangos",i,1)
    CLEAR("EB_QRY")
    ADDLINE("EB_QRY","INSERT INTO salones VALUES
    ("+""+LSSalon$+""+LSCapacidad$+""+
    LSDiaDisp$+""+""+LSHoraDisp$+""")")
    sqlexec(Connect,"","EB_QRY")
NEXT i
hWndSrc = WinGetHandle ()
hWndDest = WinGetParent (hWndSrc)
ObjCopyDataW(hWndSrc, "tb_rangos", hWndDest, "tb_rangos")
CLOSE ()

END SUB

SUB s_todos ()

```

```
setcheck("rb_1","1")
setcheck("rb_2","1")
setcheck("rb_3","1")
setcheck("rb_4","1")
setcheck("rb_5","1")
setcheck("rb_6","1")
setcheck("rb_7","1")
setcheck("rb_8","1")
setcheck("rb_9","1")
setcheck("rb_10","1")
setcheck("rb_11","1")
setcheck("rb_12","1")
setcheck("rb_13","1")
setcheck("rb_14","1")
setcheck("rb_15","1")
setcheck("rb_18","1")
END SUB
```

```
SUB s_ninguno 0
  uncheck("rb_1")
  uncheck("rb_2")
  uncheck("rb_3")
  uncheck("rb_4")
  uncheck("rb_5")
  uncheck("rb_6")
  uncheck("rb_7")
  uncheck("rb_8")
  uncheck("rb_9")
  uncheck("rb_10")
  uncheck("rb_11")
  uncheck("rb_12")
  uncheck("rb_13")
  uncheck("rb_14")
  uncheck("rb_15")
  uncheck("rb_16")
END SUB
```

```
SUB s_ninguno1 0
  uncheck("rb_17")
  uncheck("rb_18")
  uncheck("rb_19")
  uncheck("rb_20")
  uncheck("rb_21")
  uncheck("rb_22")
  uncheck("rb_23")
  uncheck("rb_24")
  uncheck("rb_25")
  uncheck("rb_26")
  uncheck("rb_27")
  uncheck("rb_28")
  uncheck("rb_29")
  uncheck("rb_30")
  uncheck("rb31")
  uncheck("rb_32")
END SUB
```

```
SUB s_todos1 0
  setcheck("rb_17","1")
  setcheck("rb_18","1")
END SUB
```

```

setcheck("rb_19","1")
setcheck("rb_20","1")
setcheck("rb_21","1")
setcheck("rb_22","1")
setcheck("rb_23","1")
setcheck("rb_24","1")
setcheck("rb_25","1")
setcheck("rb_26","1")
setcheck("rb_27","1")
setcheck("rb_28","1")
setcheck("rb_29","1")
setcheck("rb_30","1")
setcheck("rb31","1")
setcheck("rb_32","1")
END SUB

```

```

SUB s_conteo ()
CLEAR("tb_rangos")
LSHoraii$ = left$(gettext$("tb_horas",1,1),5)
LSHorai$ = Mid$(gettext$("tb_horas",1,1),9,5)
FOR i = 2 TO GetNbrow("tb_horas")
LSHori$ = left$(gettext$("tb_horas",i,1),5)
IF LSHori$ = LSHorai$ THEN
LSHorai$ = Mid$(gettext$("tb_horas",i,1),9,5)
ELSE
LSRango$ = LSHoraii$ + " - " + LSHorai$
ADDLINE("tb_rangos",LSRango$)
LSHoraii$ = LSHori$
LSHorai$ = Mid$(gettext$("tb_horas",i,1),9,5)
END IF
NEXT i
LSRango$ = LSHoraii$ + " - " + LSHorai$
ADDLINE("tb_rangos",LSRango$)
END SUB

```

```

SUB s_llena_rangos ()
FOR i = 1 TO GETNBROW("tb_rangos")
LSRango$ = LEFT$(GETTEXT$("tb_rangos",i,1),5)
LSRango1$ = MID$(GETTEXT$("tb_rangos",i,1),9,5)
FOR j = 1 TO 32
LSTexto$ = LEFT$(GETTEXT$("r"+str$(j)),5)
LSTexto1$ = MID$(GETTEXT$("r"+str$(j)),9,5)
IF (LSRango$ <= LSTexto$ AND LSRango1$ >= LSTexto1$) THEN
setcheck("rb_"+str$(j),"1")
END IF
NEXT j
LSTexto$ = LEFT$(GETTEXT$("r31"),5)
LSTexto1$ = MID$(GETTEXT$("r31"),9,5)
IF (LSRango$ <= LSTexto$ AND LSRango1$ >= LSTexto1$) THEN
setcheck("rb31","1")
END IF
NEXT i
END SUB

```


PROGRAMA : ALTA GRUPOS
OBJETIVO: MANTENIMIENTO A DETALLE DE GRUPOS

```
$include : "EVENTS.INC"  
DECLARE SUB open (Event, p1$, p2$)  
DECLARE SUB s_alta ()  
DECLARE SUB s_alta_mat ()  
DECLARE SUB s_baja_mat ()  
DECLARE SUB s_mod_mat ()  
SUB open (Event, p1$, p2$)  
  
select case (Event)  
case EVN_OPEN  
    SQLSettitles(0)  
    Connect = GetVal("gws_connect")  
    LSOpera$ = GetString$("gwc_operas$")  
    ENABLEINPUT("df_grupo")  
    ENABLEINPUT("df_alumnos")  
    CLEAR("EB_QRY")  
    CLEAR("tb_coordina")  
    ADDLINE("EB_QRY", "SELECT DISTINCT clave, apellido1||' '||apellido2||' '||nombre1  
FROM maestros ")  
    sqlquery(Connect, "tb_coordina", "EB_QRY")  
    LLNuR = GETNBROW("tb_coordina")  
    FOR i = 1 TO LLNuR  
        LSNom$ = GETTEXT$("tb_coordina", i, 2)  
        ADDLINE("cb_coordinador", LSNom$)  
    NEXT i  
    IF LSOpera$ = "A" THEN  
        SETFOCUS("df_grupo")  
    ELSE  
        IF LSOpera$ = "B" OR LSOpera$ = "C" THEN  
            hWnd = WinGetHandle()  
            hParentWnd = WinGetParent(hWnd)  
            hOldWnd = SwitchContext(hParentWnd)  
            LLCurrow = getcurrow("TB_GRUPOS")  
            LSGrupo$ = gettext$("TB_GRUPOS", LLCurrow, 1)  
            LSAlumnos$ = gettext$("TB_GRUPOS", LLCurrow, 2)  
            LSCoordina$ = gettext$("TB_GRUPOS", LLCurrow, 3)  
            SwitchContext(hOldWnd)  
            Settext("df_grupo", LSGrupo$)  
            Settext("df_alumnos", LSAlumnos$)  
            LLNuR = GETNBROW("tb_coordina")  
  
            FOR i = 1 TO LLNuR  
                LSCla$ = GETTEXT$("tb_coordina", i, 1)  
                IF LSCla$ = LSCoordina$ THEN  
                    LLPos = i  
                END IF  
            NEXT i  
            SetCurPos ("cb_coordinador", LLPos)  
            DISABLEINPUT("df_grupo")  
            IF LSOpera$ = "C" THEN  
                SETFOCUS("df_alumnos")  
            ELSE  
                DISABLEINPUT("df_grupo")  
                DISABLEINPUT("df_alumnos")  
                DISABLEINPUT("cb_coordinador")  
            END IF  
        END IF  
    END IF  
END IF
```

```

        END IF
    END IF

    exit sub
case EVN_CLOSE

case else

end select
END SUB

```

```

SUB s_alta()
LLNuR = GETNBROW("tb_coordina")
FOR i = 1 TO LLNuR
    IF GETTEXT$("tb_coordina",i,2) = GETCURTEXT$("cb_coordinador") THEN
        LSClave$ = GETTEXT$("tb_coordina",i,1)
    END IF
NEXT i
LSGrupo$ = Ltrim$(Rtrim$(GETTEXT$("df_grupo")))
IF LSGrupo$ = "" THEN
    MESSAGE("Error","Debe capturar el grupo.")
    SETFOCUS("df_grupo")
    EXIT SUB
END IF
LSAlumnos$ = Ltrim$(Rtrim$(GETTEXT$("df_alumnos")))
IF LSAlumnos$ = "" THEN
    MESSAGE("Error","Dobe capturar el número de alumnos.")
    SETFOCUS("df_alumnos")
    EXIT SUB
END IF
IF LSOpera$ = "A" THEN
    CLEAR("EB_QRY")
    CLEAR("df_qry")
    ADDLINE("EB_QRY","SELECT DISTINCT grupo FROM grupos ")
    ADDLINE("EB_QRY"," WHERE grupo = " + "" + LSGrupo$ + "" )
    sqlquery(Connect,"df_qry","EB_QRY")
    IF GETTEXT$("df_qry") <> "" THEN
        MESSAGE("Error","El grupo "+LSGrupo$+" ya fué capturado anteriormente.")
        SETFOCUS("df_grupo")
        EXIT SUB
    END IF
    CLEAR("EB_QRY")
    ADDLINE("EB_QRY","INSERT INTO grupos VALUES
    (" + "" + LSGrupo$ + "," + LSAlumnos$ + "," + LSClave$ + ")")
    sqlexec(Connect,"","EB_QRY")
ELSE
    IF LSOpera$ = "B" THEN
        CLEAR("EB_QRY")
        ADDLINE("EB_QRY","SELECT grupo FROM horario WHERE grupo = "
        + "" + LSGrupo$ + """)
        sqlquery(Connect,"df_qry","EB_QRY")
        IF GETTEXT$("df_qry") <> "" THEN
            MESSAGE("Error","El grupo "+LSGrupo$+" está asignado en horario, no podrá
            borrarlo.")
            EXIT SUB
        END IF
        CLEAR("EB_QRY")
        ADDLINE("EB_QRY","DELETE FROM grupos WHERE grupo = "
        + "" + LSGrupo$ + """)
    END IF

```

```
sqlexec(Connec,"","EB_QRY")
ELSE
CLEAR("EB_QRY")
ADDLINE("EB_QRY","UPDATE grupos SET numero = " + LSAalumnos$ + ",")
ADDLINE("EB_QRY"," coordinador = "+ LSClave$)
ADDLINE("EB_QRY"," WHERE grupo = " + "" + LSGrupo$ + "" )
sqlexec(Connec,"","EB_QRY")
END IF
close()
END SUB
```

PROGRAMA : SALONES
OBJETIVO : PANTALLA PRINCIPAL CATALOGO SALONES

```
$include : "EVENTS.INC"  
DECLARE SUB open (Event, p1$, p2$)  
DECLARE SUB s_alta ()  
DECLARE SUB s_baja ()  
DECLARE SUB s_cambio ()  
DECLARE SUB s_llena ()  
SUB open (Event, p1$, p2$)  
  
select case (Event)  
case EVN_OPEN  
    SQLSettitles(0)  
    ServerName$ = "sqlbase_10"  
    UserName$ = ""  
    Password$ = ""  
    DatabaseName$ = "DEMO"  
    ConnectNbr =  
SQLAutoConnect(ServerName$,UserName$,Password$,DatabaseName$)  
    SetVal("gws_connect",ConnectNbr)  
    s_llena()  
    exit sub  
  
case EVN_CLOSE  
  
case else  
  
end select  
END SUB  
  
SUB s_alta()  
    SetString("gwc_operas","A")  
    MODALPOPUP ("ALTASAL")  
    s_llena()  
END SUB  
  
SUB s_baja()  
    IF GETCURROW("TB_SALON") > 0 THEN  
        SetString("gwc_operas","B")  
        MODALPOPUP ("ALTASAL")  
        s_llena()  
    ELSE  
        MESSAGE("Error","Debe seleccionar un salón para ser borrado.")  
    EXIT SUB  
    END IF  
END SUB  
  
SUB s_cambio()  
    IF GETCURROW("TB_SALON") > 0 THEN  
        SetString("gwc_operas","C")  
        MODALPOPUP ("ALTASAL")  
        s_llena()  
    ELSE  
        MESSAGE("Error","Debe seleccionar un salón para ser modificado.")  
    EXIT SUB  
    END IF  
END SUB
```

END SUB

SUB s_llena()

CLEAR("EB_QRY")

ADDLINE("EB_QRY","SELECT DISTINCT SALON,CAPACIDAD "+
" FROM SALONES")

sqlquery(ConnectNbr,"TB_SALON","EB_QRY")

END SUB

PROGRAMA : ALTA SALONES
OBJETIVO : MANTENIMIENTO DETALLE DE SALONES

```
$include : "EVENTS.INC"  
DECLARE SUB open (Event, p1$, p2$)  
DECLARE SUB s_alta ()  
DECLARE SUB s_rangos ()  
DECLARE SUB s_dia ()  
DECLARE SUB s_salon ()  
SUB open (Event, p1$, p2$)  
  
select case (Event)  
case EVN_OPEN  
    SQLSetTitles(0)  
    Connect = GetVal("gws_connect")  
    LSOpera$ = GetString$("gwc_operas$")  
    CLEAR("df_diadisp")  
    DISABLEINPUT("df_diadisp")  
    IF LSOpera$ = "A" THEN  
        SETFOCUS("df_salon")  
    ELSE  
        IF LSOpera$ = "B" OR LSOpera$ = "C" THEN  
            hWnd = WinGetHandle()  
            hParentWnd = WinGetParent(hWnd)  
            hOldWnd = SwitchContext(hParentWnd)  
            LLCurrow = getcurrow("TB_SALON")  
            LSSalon$ = gettext$("TB_SALON",LLCurrow,1)  
            LSCapacidad$ = gettext$("TB_SALON",LLCurrow,2)  
            LSDiaDisp$ = gettext$("TB_SALON",LLCurrow,3)  
            LSHoraDisp$ = gettext$("TB_SALON",LLCurrow,4)  
            SwitchContext(hOldWnd)  
            Settext("df_salon",LSSalon$)  
            Settext("df_capacidad",LSCapacidad$)  
            Settext("df_diadisp",LSDiaDisp$)  
            Settext("df_horadisp",LSHoraDisp$)  
            IF LSOpera$ = "B" THEN  
                DISABLEINPUT("df_salon")  
                DISABLEINPUT("df_capacidad")  
                DISABLEINPUT("df_horadisp")  
                DISABLEINPUT("bu_rangos")  
            ELSE  
                DISABLEINPUT("df_salon")  
                ENABLEINPUT("df_capacidad")  
                ENABLEINPUT("df_horadisp")  
                ENABLEINPUT("bu_rangos")  
                SETFOCUS("df_capacidad")  
            END IF  
        END IF  
    END IF  
    SetCurPos("cb_dia",1)  
    s_dia ()  
    exit sub  
case EVN_CLOSE  
  
case else  
  
end select  
END SUB
```

```
SUB s_alla()
```

```
LSSalon$ = Ltrim$(Rtrim$(gettext$("df_salon")))
LSCapacidad$ = Ltrim$(Rtrim$(gettext$("df_capacidad")))
LSDiaDisp$ = Ltrim$(Rtrim$(getCurtext$("cb_dia")))
IF LSSalon$ = "" THEN
    MESSAGE("Error","Debe capturar un salón.")
    SETFOCUS("df_salon")
    EXIT SUB
END IF
IF LSDiaDisp$ = "" THEN
    MESSAGE("Error","Debe capturar el día disponible.")
    SETFOCUS("cb_dia")
    EXIT SUB
END IF
IF LSCapacidad$ = "" THEN
    MESSAGE("Error","Debe capturar la capacidad del salón.")
    SETFOCUS("df_capacidad")
    EXIT SUB
END IF
IF LSOpera$ = "A" THEN
    CLEAR("EB_QRY")
    LSSalon$ = gettext$("df_salon")
    CLEAR("EB_QRY")
    CLEAR("df_qry")
ELSE
    IF LSOpera$ = "B" THEN
        CLEAR("EB_QRY")
        CLEAR("df_qry")
        ADDLINE("EB_QRY","SELECT salon FROM horario WHERE salon = "
            +"""+LSSalon$+""")
        sqlquery(Connect,"df_qry","EB_QRY")
        IF GETTEXT$("df_qry") <> "" THEN
            MESSAGE("Error","El salón "+LSSalon$+ " tiene un grupo seleccionado.")
            EXIT SUB
        END IF
        CLEAR("EB_QRY")
        ADDLINE("EB_QRY","DELETE FROM salones WHERE salon = "
            +"""+LSSalon$+""")
        sqlexec(Connect,"","EB_QRY")
    ELSE
        CLEAR("EB_QRY")
        ADDLINE("EB_QRY","UPDATE salones SET capacidad = " +LSCapacidad$)
        ADDLINE("EB_QRY"," WHERE salon = " + """+ LSSalon$ + """)
        sqlexec(Connect,"","EB_QRY")
    END IF
END IF
close()
END SUB
```

```
SUB s_rangos ()
```

```
LSSalon$ = gettext$("df_salon")
LSDiadisp$ = GetCurText$("cb_dia")
LSDia$ = GETTEXT$("df_diadisp")
LSCapacidad$ = GetText$("df_capacidad")
IF LSOpera$ = "A" THEN
    LSSalon$ = gettext$("df_salon")
    CLEAR("EB_QRY")
    CLEAR("df_qry")
    ADDLINE("EB_QRY","SELECT salon FROM salones ")
```

```

ADDLINE("EB_QRY"," WHERE salon = " + "" + LSSalon$ + """)
ADDLINE("EB_QRY"," AND dia_disp = " + LSDiadiasp$)
SQLQUERY(Connect,"df_qry","EB_QRY")
IF GETTEXT$("df_qry") <> "" THEN
    MESSAGE("Error","El salón "+LSSalon$+" ya fué capturado anteriormente.")
    CLEAR("df_salon")
    SETFOCUS("df_salon")
    EXIT SUB
END IF
END IF
IF LSSalon$ = "" THEN
    MESSAGE("Error","Debe capturar un salón.")
    SETFOCUS("df_salon")
    EXIT SUB
END IF
IF LSDia$ = "" THEN
    MESSAGE("Error","Debe capturar el día disponible.")
    SETFOCUS("cb_dia")
    EXIT SUB
END IF
IF LSCapacidad$ = "" THEN
    MESSAGE("Error","Debe capturar la capacidad del salón.")
    SETFOCUS("df_capacidad")
    EXIT SUB
END IF
SETSTRING("gwc_salon",LSSalon$)
SETSTRING("gwc_dia",LSDia$)
SETSTRING("gwc_diadiasp",LSDiadiasp$)
SETSTRING("gwc_capacidad",LSCapacidad$)
IF GETTEXT$("lb_rangos",1,1) <> "" THEN
    LSRangos$ = "1"
ELSE
    LSRangos$ = "0"
END IF
SETSTRING("gwc_rangos$",LSRangos$)
MODALPOPUP("RANGOS")
END SUB

SUB s_dia()
CLEAR("lb_rangos")
LSSalon$ = gettext$("df_salon")
LSDia$ = GetCurText$("cb_dia")
IF LSDia$ = "1" THEN
    SetText("df_diadiasp","LUNES")
END IF
IF LSDia$ = "2" THEN
    SetText("df_diadiasp","MARTES")
END IF
IF LSDia$ = "3" THEN
    SetText("df_diadiasp","MIERCOLES")
END IF
IF LSDia$ = "4" THEN
    SetText("df_diadiasp","JUEVES")
END IF
IF LSDia$ = "5" THEN
    SetText("df_diadiasp","VIERNES")
END IF
IF LSDia$ = "6" THEN
    SetText("df_diadiasp","SABADO")
END IF

```



```

IF.LSDia$ = "7" THEN
  SetText("df_diadisp","DOMINGO")
END IF
CLEAR("EB_QRY")
ADDLINE("EB_QRY","SELECT hora_disp FROM salones ")
ADDLINE("EB_QRY"," WHERE salon = " + "" + LSSalon$ + "" )
ADDLINE("EB_QRY"," AND dia_disp = " + "" + LSDia$ + "" )
SQLQUERY(Connect,"tb_rangos","EB_QRY")
END SUB

SUB s_salon ()
IF LSOpera$ = "A" THEN
  LSSalon$ = gettext$("df_salon")
  CLEAR("EB_QRY")
  CLEAR("df_qry")
  ADDLINE("EB_QRY","SELECT salon FROM salones ")
  ADDLINE("EB_QRY"," WHERE salon = " + "" + LSSalon$ + "" )
  SQLQUERY(Connect,"df_qry","EB_QRY")
  IF GETTEXT$("df_qry") <> "" THEN
    MESSAGE("Error","El salón "+LSSalon$+" ya fué capturado anteriormente.")
    CLEAR("df_salon")
    SETFOCUS ("df_salon")
  END IF
END IF
END SUB

```

PROGRAMA : MATERIAS
OBJETIVO : PANTALLA PRINCIPAL CATALOGO MATERIAS

```
$include : "EVENTS.INC"  
DECLARE SUB open (Event, p1$, p2$)  
DECLARE SUB s_alta ()  
DECLARE SUB s_baja ()  
DECLARE SUB s_cambio ()  
DECLARE SUB s_llena ()  
SUB open (Event, p1$, p2$)  
  
select case (Event)  
case EVN_OPEN  
SQLSetTitles(0)  
ServerName$ = "sqlbase_10"  
UserName$ = ""  
Password$ = ""  
DatabaseName$ = "DEMO"  
ConnectNbr = GetVal("gws_connect")  
s_llena()  
exit sub  
  
case EVN_CLOSE  
  
case else  
  
end select  
END SUB  
  
SUB s_alta()  
SetString("gwc_operas","A")  
MODALPOPUP ("ALTAMAT")  
s_llena()  
END SUB  
  
SUB s_baja()  
IF GETCURROW("TB_MATERIA") > 0 THEN  
SetString("gwc_operas","B")  
MODALPOPUP ("ALTAMAT")  
s_llena()  
ELSE  
MESSAGE("Error","Debe seleccionar un área para ser borrada.")  
EXIT SUB  
END IF  
END SUB  
  
SUB s_cambio()  
IF GETCURROW("TB_MATERIA") > 0 THEN  
SetString("gwc_operas","C")  
MODALPOPUP ("ALTAMAT")  
s_llena()  
ELSE  
MESSAGE("Error","Debe seleccionar un área para ser modificada.")  
EXIT SUB  
END IF  
  
END SUB
```

```
SUB s_llena()
  CLEAR("EB_QRY")
  ADDLINE("EB_QRY","SELECT DISTINCT AREA "+
    " FROM MATERIAS")
  sqlquery(ConnectNbr,"TB_MATERIA","EB_QRY")
END SUB
```

PROGRAMA : ALTA MATERIAS
OBJETIVO : MANTENIMIENTO DETALLE PARA MATERIAS

```
$include : "EVENTS.INC"  
DECLARE SUB open (Event, p1$, p2$)  
DECLARE SUB s_alta ()  
DECLARE SUB s_alta_mat ()  
DECLARE SUB s_baja_mat ()  
DECLARE SUB s_baja_area ()  
DECLARE SUB s_mod_mat ()  
DECLARE SUB s_llena ()  
SUB open (Event, p1$, p2$)  
  
select case (Event)  
case EVN_OPEN  
    SQLSetTitles(0)  
    Connect = GetVal("gws_connect")  
    LSOpera$ = GetString$("gwc_operas$")  
    SHOW("bu_nueva")  
    SHOW("bu_baja")  
    SHOW("bu_mod")  
    IF LSOpera$ = "A" THEN  
        SETFOCUS("df_area")  
    ELSE  
        IF LSOpera$ = "B" OR LSOpera$ = "C" THEN  
            hWnd = WinGetHandle()  
            hParentWnd = WinGetParent(hWnd)  
            hOldWnd = SwitchContext(hParentWnd)  
            LLCurrow = getcurrow("TB_MATERIA")  
            LSArea$ = gettext$("TB_MATERIA", LLCurrow, 1)  
            SwitchContext(hOldWnd)  
            Settext("df_area", LSArea$)  
            DISABLEINPUT("df_area")  
            IF LSOpera$ = "B" THEN  
                HIDE("bu_nueva")  
                HIDE("bu_baja")  
                HIDE("bu_mod")  
            END IF  
  
            s_llena ()  
        END IF  
    END IF  
  
    exit sub  
case EVN_CLOSE  
  
case else  
  
end select  
END SUB  
  
SUB s_alta()  
    LSArea$ = Ltrim$(Rtrim$(GETTEXT$("df_area"))) )  
    IF LSArea$ = "" THEN  
        MESSAGE("Error", "Debe capturar el área de la materia.")  
        SETFOCUS("df_salon")  
    EXIT SUB  
    END IF  
    IF LSOpera$ = "A" THEN
```

```

CLEAR("EB_QRY")
ADDLINE("EB_QRY","DELETE FROM materias WHERE area = "
+""+LSArea$+""")
sqlxec(Connect,","EB_QRY")
FOR i = 1 TO GetNBROW("tb_rangos")
CLEAR("EB_QRY")
LSHoraDisp$ = GETTEXT$("tb_rangos",i,1)
ADDLINE("EB_QRY","INSERT INTO salones VALUES
(+""+LSSalon$+""+LSCapacidad$+""+
LSDiaDisp$+""+""+LSHoraDisp$+""")
sqlxec(Connect,","EB_QRY")
NEXT i
END IF
close()
END SUB

SUB s_alta_mat ()
LSArea$ = Ltrim$(Rtrim$(GETTEXT$("df_area")))
IF LSArea$ = "" THEN
MESSAGE("Error","Debe capturar el área de la materia.")
SETFOCUS("df_area")
EXIT SUB
END IF
SETSTRING("gwc_opcion","A")
MODALPOPUP("ALTAMAT1")
s_llena()
END SUB

SUB s_baja_mat ()
LSArea$ = Ltrim$(Rtrim$(GETTEXT$("df_area")))
IF LSArea$ = "" THEN
MESSAGE("Error","Debe capturar el área de la materia.")
SETFOCUS("df_area")
EXIT SUB
END IF
IF GETCURREW("tb_materias") < 1 THEN
MESSAGE("Error","Debe seleccionar una materia para borrarla.")
EXIT SUB
END IF
SETSTRING("gwc_opcion","B")
MODALPOPUP("ALTAMAT1")
s_llena()
END SUB

SUB s_baja_area ()
LSArea$ = Ltrim$(Rtrim$(GETTEXT$("df_area")))
IF LSArea$ = "" THEN
MESSAGE("Error","Debe capturar el área de la materia.")
SETFOCUS("df_salon")
EXIT SUB
END IF
LLNu = GETNBROW("tb_materias")
IF LLNu <> 0 THEN
FOR i = 1 TO LLNu
LSMat$ = GETTEXT$("tb_materias",i,1)
CLEAR("EB_QRY")
ADDLINE("EB_QRY","SELECT materia FROM horario WHERE materia = "
+""+LSMat$+""")
sqlquery(Connect,"df_qry","EB_QRY")
IF GETTEXT$("df_qry") <> "" THEN

```

```

        MESSAGE("Error", "La materia "+LSMat$+" está asignada en horario, no podrá
        borrarla.")
    EXIT SUB
END IF
NEXT I
ELSE
    LSArea$ = GetText$("df_area")
    CLEAR("EB_QRY")
    ADDLINE("EB_QRY", "DELETE FROM materias WHERE area = "
    +"""+LSArea$+""")
    sqlquery(Connect, "df_qry", "EB_QRY")
END IF
CLEAR("EB_QRY")
ADDLINE("EB_QRY", "DELETE FROM materias WHERE area = " +"""+LSArea$+""")
sqlexec(Connect, "", "EB_QRY")
close()
END SUB

```

```

SUB s_mod_mat ()
    LSArea$ = Ltrim$(Rtrim$(GETTEXT$("df_area")))
    IF LSArea$ = "" THEN
        MESSAGE("Error", "Debe capturar el área de la materia.")
        SETFOCUS("df_area")
        EXIT SUB
    END IF
    IF GETCURREW("tb_materias") < 1 THEN
        MESSAGE("Error", "Debe seleccionar una materia para modificarla.")
        EXIT SUB
    END IF
    SETSTRING("gwc_opcion", "C")
    MODALPOPUP("ALTAMAT1")
    s_llena()
END SUB

```

```

SUB s_llena()
    LSArea$ = Ltrim$(Rtrim$(GETTEXT$("df_area")))
    CLEAR("EB_QRY")
    ADDLINE("EB_QRY", "SELECT materia, semestre, horas_sem FROM materias ")
    ADDLINE("EB_QRY", " WHERE area = " + """+LSArea$+""")
    ADDLINE("EB_QRY", " ORDER BY 1")
    SQLQUERY(Connect, "tb_materias", "EB_QRY")
END SUB

```

APÉNDICE B

PROGRAMA : HORARIO
OBJETIVO : PANTALLA PRINCIPAL CATALOGO HORARIOS

```
$include : "EVENTS.INC"
DECLARE SUB open (Event, p1$, p2$)
DECLARE SUB s_alta ()
DECLARE SUB s_baja ()
DECLARE SUB s_cambio ()
DECLARE SUB s_llena ()
SUB open (Event, p1$, p2$)

select case (Event)
case EVN_OPEN
  SQLSetTitles(0)
  ServerName$ = "sqlbase_10"
  UserName$ = ""
  Password$ = ""
  DatabaseName$ = "DEMO"
  ConnectNbr = GetVal("gws_connect")
  s_llena()
  exit sub

case EVN_CLOSE

case else

end select
END SUB

SUB s_alta()
  SetString("gwc_operas$", "A")
  LSGrupo$ = gettext$("TB_GRUPOS", GETCURROW("TB_GRUPOS"))
  SetString("gwc_grupo$", LSGrupo$)
  MODALPOPUP ("ALTAHOR")
  s_llena()
END SUB

SUB s_baja()
  IF GETCURROW("TB_GRUPOS") > 0 THEN
    LSGrupo$ = gettext$("TB_GRUPOS", GETCURROW("TB_GRUPOS"))
    SetString("gwc_grupo$", LSGrupo$)
    SetString("gwc_operas$", "B")
    MODALPOPUP ("ALTAHOR")
    s_llena()
  ELSE
    MESSAGE("Error", "Debe seleccionar un grupo para ser borrado.")
  EXIT SUB
  END IF
END SUB

SUB s_cambio()
  IF GETCURROW("TB_GRUPOS") > 0 THEN
    LSGrupo$ = gettext$("TB_GRUPOS", GETCURROW("TB_GRUPOS"))
    SetString("gwc_grupo$", LSGrupo$)
    SetString("gwc_operas$", "C")
    MODALPOPUP ("ALTAHOR")
    s_llena()
  ELSE
  END IF
END SUB
```



```
MESSAGE("Error","Debe seleccionar un grupo para ser modificado ")
EXIT SUB
END IF
```

```
END SUB
```

```
SUB s_llena()
CLEAR("EB_QRY")
CLEAR("TB_GRUPOS")
ADDLINE("EB_QRY","SELECT DISTINCT GRUPO "+
        " FROM GRUPOS")
sqlquery(ConnectNbr,"TB_GRUPOS","EB_QRY")
END SUB
```

APÉNDICE C

PROGRAMA : ALTA HORARIO
OBJETIVO : MANTENIMIENTO PARA DETALLE DE HORARIO

```
$include : "EVENTS.INC"  
DECLARE SUB open (Event, p1$, p2$)  
DECLARE SUB s_alta ()  
DECLARE SUB s_rangos ()  
DECLARE SUB s_dia ()  
DECLARE SUB s_area ()  
DECLARE SUB s_salon ()  
DECLARE SUB s_materias ()  
DECLARE SUB s_detalle ()  
DECLARE SUB s_cambio ()  
DECLARE SUB s_borra ()  
DECLARE SUB s_otras_horas()  
SUB open (Event, p1$, p2$)  
  
select case (Event)  
case EVN_OPEN  
SQL SetTitles(0)  
Connect = GetVal("gws_connect")  
LSOpera$ = GetString$("gwc_operas$")  
LSGrupo$ = GetString$("gwc_grupo$")  
SETTEXT("df_grupo",LSGrupo$)  
CLEAR("df_diadisp")  
DISABLEINPUT("df_diadisp")  
IF LSOpera$ = "A" THEN  
SETFOCUS("cb_dia")  
ELSE  
IF LSOpera$ = "B" OR LSOpera$ = "C" THEN  
hWnd = WinGetHandle()  
hParentWnd = WinGetParent(hWnd)  
hOldWnd = SwitchContext(hParentWnd)  
LLCurrow = getcurrow("TB_GRUPOS")  
LSClave$ = gettext$("TB_GRUPOS",LLCurrow,1)  
SwitchContext(hOldWnd)  
Settext("df_clave",LSClave$)  
LSClave$ = gettext$("df_clave")  
IF LSOpera$ = "B" THEN  
DISABLEINPUT("df_clave")  
DISABLEINPUT("df_nombre2")  
DISABLEINPUT("df_apellido1")  
DISABLEINPUT("bu_apellido2")  
DISABLEINPUT("bu_rangos")  
ELSE  
SETFOCUS("df_nombre1")  
END IF  
END IF  
END IF  
SetCurPos("cb_dia",1)  
s_dia ()  
exit sub  
case EVN_CLOSE  
  
case else  
  
end select  
END SUB
```

```

SUB s_rangos ()
  LSGrupo$ = gettext$("df_grupo")
  LSDia$ = getCurtext$("cb_dia")
  LSDiadiasp$ = gettext$("df_diadiasp")
  LSArea$ = getCurtext$("cb_area")
  LSMateria$ = getCurtext$("cb_materias")
  SETSTRING("gwc_grupo", LSGrupo$)
  SETSTRING("gwc_dia", LSDia$)
  SETSTRING("gwc_diadiasp", LSDiadiasp$)
  SETSTRING("gwc_area", LSArea$)
  SETSTRING("gwc_materia", LSMateria$)
  SETSTRING("gwc_op_sel", "A") 'Opción seleccionada (Alta)
  s_otras_horas()
  MODALPOPUP("RANGOSHO")
  s_detalle()
END SUB

SUB s_cambio ()
  LLCurRow = GETCURROW("tb_detalle")
  IF LLCurRow > 0 THEN
    IF GETTEXT$("tb_detalle", LLCurRow, 1) = "" THEN
      MESSAGE("Error", "Debe seleccionar un detalle del horario para ser
        modificado.")
      EXIT SUB
    END IF
  ELSE
    MESSAGE("Error", "Debe seleccionar un detalle del horario para ser modificado.")
    EXIT SUB
  END IF
  s_otras_horas()
  LSGrupo$ = gettext$("df_grupo")
  LSDia$ = getCurtext$("cb_dia")
  LSDiadiasp$ = gettext$("df_diadiasp")
  LSArea$ = getCurtext$("cb_area")
  LSMateria$ = getCurtext$("cb_materias")
  LSSalon$ = gettext$("tb_detalle", LLCurRow, 1)
  LSCveMaestro$ = gettext$("tb_detalle", LLCurRow, 2)
  LSNmMaestro$ = gettext$("tb_detalle", LLCurRow, 3)
  LSHora$ = gettext$("tb_detalle", LLCurRow, 4)
  FOR i = 1 TO GETNBROW("tb_otras_horas")
    IF GETTEXT$("tb_otras_horas", i, 1) = LSHora$ THEN
      DELETELINE("tb_otras_horas", i)
    END IF
  NEXT i
  SETSTRING("gwc_grupo", LSGrupo$)
  SETSTRING("gwc_dia", LSDia$)
  SETSTRING("gwc_diadiasp", LSDiadiasp$)
  SETSTRING("gwc_area", LSArea$)
  SETSTRING("gwc_materia", LSMateria$)
  SETSTRING("gwc_salon", LSSalon$)
  SETSTRING("gwc_cve_disp", LSCveMaestro$)
  SETSTRING("gwc_maestro", LSNmMaestro$)
  SETSTRING("gwc_hora", LSHora$)
  SETSTRING("gwc_op_sel", "C") 'Opción seleccionada (Cambio)
  MODALPOPUP("RANGOSHO")
  s_detalle()
END SUB

SUB s_borra ()
  LLCurRow = GETCURROW("tb_detalle")

```

```

IF LLCurRow > 0 THEN
  IF GETTEXT$("tb_detalle",LLCurRow,1) = "" THEN
    MESSAGE("Error","Debe seleccionar un detalle del horario para ser borrado.")
    EXIT SUB
  END IF
ELSE
  MESSAGE("Error","Debe seleccionar un detalle del horario para ser borrado.")
  EXIT SUB
END IF
END IF
LSGrupo$ = gettext$("df_grupo")
LSDia$ = getCurtext$("cb_dia")
LSDiadi$ = gettext$("df_diadi")
LSArea$ = getCurtext$("cb_area")
LSMateria$ = getCurtext$("cb_materias")
LSSalon$ = gettext$("tb_detalle",LLCurRow,1)
LSCveMaestro$ = gettext$("tb_detalle",LLCurRow,2)
LSNmMaestro$ = gettext$("tb_detalle",LLCurRow,3)
LSHora$ = gettext$("tb_detalle",LLCurRow,4)

Buton = MessageBox ("Desea borrar el detalle seleccionado.", "Horario", 4)
IF Buton = 6 THEN
  CLEAR("EB_QRY")
  ADDLINE("EB_QRY","DELETE FROM horario ")
  ADDLINE("EB_QRY"," WHERE grupo = '"+LSGrupo$+"'")
  ADDLINE("EB_QRY"," AND dia = '"+LSDia$"'")
  ADDLINE("EB_QRY"," AND materia = '"+LSMateria$+"'")
  ADDLINE("EB_QRY"," AND maestro = '"+LSCveMaestro$"'")
  ADDLINE("EB_QRY"," AND hora_disp = '"+LSHora$+"'")
  ADDLINE("EB_QRY"," AND salon = '"+LSSalon$+"'")
  SQLEXEC(Connect,"","EB_QRY")
  s_detalle()
END IF
END SUB

SUB s_dia()
  CLEAR("tb_rangos")
  LSDia$ = GetCurText$("cb_dia")
  LSClave$ = Gettext$("df_clave")
  IF LSDia$ = "1" THEN
    SetText("df_diadi","LUNES")
  END IF
  IF LSDia$ = "2" THEN
    SetText("df_diadi","MARTES")
  END IF
  IF LSDia$ = "3" THEN
    SetText("df_diadi","MIERCOLES")
  END IF
  IF LSDia$ = "4" THEN
    SetText("df_diadi","JUEVES")
  END IF
  IF LSDia$ = "5" THEN
    SetText("df_diadi","VIERNES")
  END IF
  IF LSDia$ = "6" THEN
    SetText("df_diadi","SABADO")
  END IF
  IF LSDia$ = "7" THEN
    SetText("df_diadi","DOMINGO")
  END IF
  s_area()

```

END SUB

```
SUB s_area()
  CLEAR("EB_QRY")
  ADDLINE("EB_QRY","SELECT DISTINCT area FROM materias ")
  SQLQUERY(Connect,"cb_area","EB_QRY")
  SetCurPos("cb_area",1)
  s_materias()
END SUB
```

```
SUB s_materias()
  LSArea$ = GETCURTEXT$("cb_area")
  CLEAR("EB_QRY")
  ADDLINE("EB_QRY","SELECT DISTINCT materia FROM materias ")
  ADDLINE("EB_QRY"," WHERE area = '"+LSArea$+"'")
  SQLQUERY(Connect,"cb_materias","EB_QRY")
  SetCurPos("cb_materias",1)
  s_detalle()
END SUB
```

```
SUB s_detalle()
  LSGrupo$ = gettext$("df_grupo")
  LSDia$ = getCurtext$("cb_dia")
  LSArea$ = getCurtext$("cb_area")
  LSMateria$ = getCurtext$("cb_materias")
  CLEAR("EB_QRY")
  ADDLINE("EB_QRY","SELECT DISTINCT salon,clave,apellido1||' '|apellido2||'
  '|nombre1,horario.hora_disp ")
  ADDLINE("EB_QRY"," FROM horario,maestros ")
  ADDLINE("EB_QRY"," WHERE grupo = '"+LSGrupo$+"'")
  ADDLINE("EB_QRY"," AND maestros.clave = horario.maestro ")
  ADDLINE("EB_QRY"," AND dia = '"+LSDia$"'")
  ADDLINE("EB_QRY"," AND materia = '"+LSMateria$+"'")
  ADDLINE("EB_QRY"," ORDER BY 4")
  SQLQUERY(Connect,"lb_detalle","EB_QRY")
END SUB
```

```
SUB s_otras_horas()
  LSGrupo$ = gettext$("df_grupo")
  LSDia$ = getCurtext$("cb_dia")
  CLEAR("EB_QRY")
  ADDLINE("EB_QRY","SELECT DISTINCT hora_disp ")
  ADDLINE("EB_QRY"," FROM horario ")
  ADDLINE("EB_QRY"," WHERE dia = '"+LSDia$"'")
  ADDLINE("EB_QRY"," AND grupo = '"+LSGrupo$+"'")
  SQLQUERY(Connect,"lb_otras_horas","EB_QRY")
END SUB
```

END SUB