



# UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

## SISTEMA DE CONTROL DE BECARIOS PARA LA DGDC UNIVERSUM

### TESIS PROFESIONAL

QUE PARA OBTENER EL TITULO DE:  
INGENIERO EN COMPUTACION

PRESENTAN:

YAIR GOMEZ MENDEZ

JOSE ROBERTO HERNANDEZ MOLINA

DIRECTORA DE TESIS:

M. C MA. JAQUELINA LOPEZ BARRIENTOS



CIUDAD UNIVERSITARIA

2005

m. 345833



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.

# Agradecimientos

Autorizo a la Dirección General de Bibliotecas de la UNAM a difundir en formato electrónico e impreso el contenido de mi trabajo recepcional.  
NOMBRE: Yair Gómez Hernández

FECHA: 23/06/2005

FIRMA: [Firma]

## *Roberto*

Es hermoso caminar por largo tiempo y después voltear a ver el camino recorrido, si bien ha sido difícil y ha veces hemos tenido que librar obstáculos, al final del camino tendremos nuestra recompensa.

Esta tesis esta dedicada a Mis Padres Roberto y Socorro a mi Hermana Esperanza a todos mis Hermanos (Todos sin excepción), y mis Tías que me han apoyado siempre a lo largo del camino. ¡Gracias!

Quiero dedicar esta tesis a otra persona que me ha dado todo su amor y apoyo a mi Perla preciosa que siempre llevo en mi corazón Te amo.

## *Yair*

Empezar un proyecto de cualquier índole requiere conocimientos en el área, visión, disposición de trabajar, deseo de cumplir con los objetivitos planteados, entre otras cosas, pero hay algo más valioso, el apoyo y en muchas ocasiones la comprensión de esas personas que han estado y estarán junto a ti hasta que el tiempo y la vida lo permitan, por ello dedico estas líneas a mis padres y abuelita que en mayor o menor medida contribuyeron ha esta causa, ¡Gracias!.

## *Yair, Roberto*

No podíamos dejar de lado el apoyo y confianza que nos otorgó Ing. Addina Cuervo, gracias por confiar en nosotros.

Autorizo a la Dirección General de Bibliotecas de la UNAM a difundir en formato electrónico e impreso el contenido de mi trabajo recepcional.  
NOMBRE: José Roberto Hernández

Malina  
FECHA: 23/06/2005

FIRMA: [Firma]

Índice	Página
Capítulo 1 Introducción	001
1.1 Panorama General de la Problemática	002
1.1.1 Áreas dentro de Universum	006
1.1.2 Desarrollo	007
1.2 Objetivo de la Tesis	008
Capítulo 2 Antecedentes	010
2.1 Antecedentes Académicos	011
2.1.1 Ingeniería de Software	011
2.1.1.1 Modelo Lineal Secuencial	012
2.1.1.2 Modelo de Construcción de Prototipos	013
2.1.1.3 El Modelo en Espiral	015
2.1.2 Bases de Datos	018
2.1.2.1 Análisis del Concepto de Base de Datos	018
2.1.2.2 Las características elementales de una Base	019
de Datos	
2.1.2.3 Ventajas de las Bases de Datos	020
2.1.2.4 Componentes de los Sistemas de Bases de	021
Datos	
2.1.3 Programación Orientada a Objetos	024
2.1.3.1 Objetos	024
2.1.3.2 Mensajes	024
2.1.3.3 Métodos	025
2.1.3.4 Clases	025
2.1.3.5 Características de la POO	025
2.1.3.6 Abstracción	025
2.1.3.7 Encapsulamiento	026
2.1.3.8 Herencia	026
2.1.3.9 Polimorfismo	026
2.1.3.10 Constructores y Destructores	027
2.1.4 Redes De Computadoras	027
2.1.4.1 Topología Estrella	029
2.1.4.2 Topología Anillo	030
2.1.4.3 Topología Bus	031
2.1.4.4 Protocolos	032
2.1.4.5 TCP/IP	034
2.1.4.6 Protocolo Internet (IP)	035
2.1.4.7 Apple Talk	036
2.1.5 Arquitectura Cliente/Servidor	038



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**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.

2.1.5.1 Componentes Esenciales de la Infraestructura Cliente/Servidor	040
<b>Capítulo 3 Análisis del Problema</b>	<b>041</b>
3.1 Análisis del Problema	042
3.1.1 Las Características de los Sistemas Operativos	042
3.1.2 Microsoft Windows	045
3.1.2.1 Versiones basadas en MS-DOS	045
3.1.2.2 Versiones basadas en NT (New Technology)	045
3.1.2.3 Sistemas de Archivos de Windows	046
3.1.2.4 Windows XP	046
3.1.3 Apple Macintosh	047
3.1.3.1 Aplicaciones del Sistema	048
3.1.3.2 Interfaz de Usuario	048
3.1.3.3 Marcos para Aplicaciones	049
3.1.3.4 Gráficos y Multimedia	049
3.1.3.5 Gestión del Sistema	050
3.1.3.6 Cimientos del Sistema	050
3.1.4 Análisis del Sistema de Control de Becarios	050
Información Requerida del Sistema	
3.1.4.1 Relacionado a los Becarios	050
3.1.4.2 Relacionado a las Salas o Áreas del Museo	052
3.1.4.3 Relacionada a los Horarios	053
3.1.4.4 Información en Tiempo Real sobre los Anfitriones	053
3.1.4.5 Adquisición de la Información	054
3.1.4.6 El Entorno del Sistema	055
3.1.4.6.1 Red Interna	056
<b>Capítulo 4 Creación de un Modelo</b>	<b>059</b>
4.1 Creación de un Modelo (Ingeniería)	060
4.1.1 Reingeniería de Procesos	061
4.2 Asistencia de los Becarios	062
4.2.1 Funcionamiento del Sistema de Códigos de Barras	062
4.2.2 Simbologías del Código de Barras	063
4.2.2.1 Código EAN-13	064
4.2.2.2 Código Codabar	065
4.2.2.3 Código 39 (3 de 9)	065
4.3 Credencial e Identificación de los Becarios	066
4.4 Creación de la Base de Datos para el Control de Becarios	069

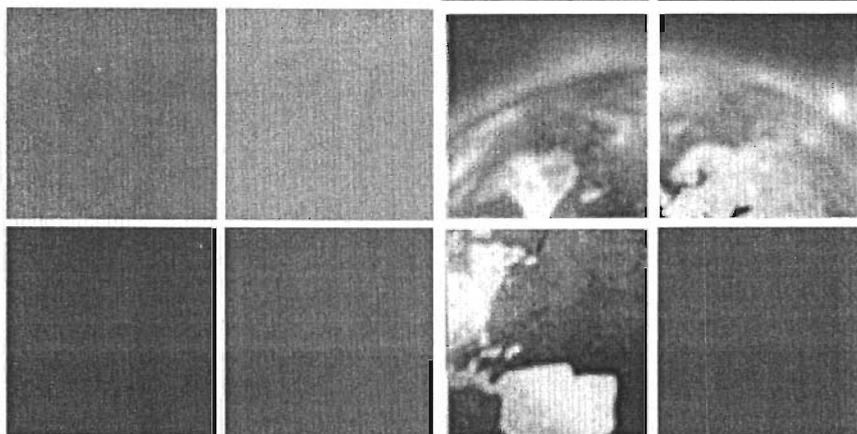
Capítulo 5 Construcción y Desarrollo	077
5.1 Visual Basic 6.0	077
5.1.1 Características Generales de Visual-Basic	078
5.1.2 Visual Basic en el Sistema de Control de	079
Becarios	
5.2 Java	080
5.2.1 Características Generales de Java	083
5.3 MYSQL	085
5.5.1 Tipos de Datos en MYSQL	085
5.4 Tomcat	090
5.4.1 Historia de Tomcat	090
5.4.1.1 Entorno Tomcat	090
5.4.1.2 Estado del Desarrollo de Tomcat	091
5.4.1.3 Estructura de Directorios	091
5.4.1.4 Características de Tomcat	091
5.5 Visión General de las JSP	092
5.5.1 Cómo funciona JSP	093
5.6 Credenciales Para Becarios de la DGDC	095
5.7 Sistema de Control de Becarios fase Red	104
5.7.1 Seguridad en el Sistema en Red	110
5.7.2 Modulo Para los Becarios	114
5.7.3 Diagrama Conceptual del Sistema en Red	115
5.8 Sistema de Control de Becarios fase Red Local	116
5.8.1 Servidor de Información de Asistencia (SIA)	117
5.8.1.1 Asistencia	117
5.8.1.2 Servicio de Envío de Información	123
5.8.1.3 Recepción de Información	129
5.8.1.4 Parar el Servicio de Información	131
5.8.2 Terminales	132
5.8.2.1 Ficha "Inicio"	135
5.8.2.2 Ficha "Asistencia"	145
5.8.2.2.1 Consulta Rápida	147
5.8.2.2.2 Consulta Avanzada	150
5.8.2.3 Fichas "Becarios", "Salas", "Horas de Inicio"	153
5.8.2.4 Ficha "Sábana"	159
5.8.2.4.1 Horarios	160
5.8.2.4.2 Asignar Horario	162
5.8.2.4.3 Consultar Horario	164
5.8.2.4.4 Modificar Horario y Asignar	166
Horario de Suplente	
5.8.2.4.5 Asignar Becario	167

5.8.2.5 Consultar Sábana por Sala	168
5.8.2.6 Sábana por Día	171
5.8.3 Mejorando Búsquedas	175
<b>Capítulo 6 Implantación del Sistema</b>	<b>179</b>
6.1 Requerimientos de Software	182
6.2 Requerimientos de Hardware	183
6.3 Capacitación de los Usuarios	183
6.4 Políticas de Seguridad	185
6.4.1 Políticas de Seguridad Para el Sistema de Control de Becarios	186
<b>Capítulo 7 Conclusiones</b>	<b>188</b>
<b>Capítulo 8 Bibliografía</b>	<b>193</b>
<b>Capítulo 9 Apéndices y Glosario</b>	<b>195</b>
Apéndice A Manual de Usuario C <sup>2</sup> WCAM	196
Apéndice B Manual de Usuario Ms Access a Mysql	197
Apéndice C Manual Sistema de Control de Becarios Red	198
Apéndice D Manual Sistema de Control de Becarios Red Local	199
Glosario de Términos	201

# Introducción

Capítulo

1





Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**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.

# CAPÍTULO 1 INTRODUCCIÓN

*En este capítulo se dejará ver el panorama general de la problemática del área de Becarios, así como también una breve reseña de la Dirección General de Divulgación de la Ciencia Universum, las áreas del Museo y el objetivo de esta tesis.*

## **1. Introducción**

### **1.1 Panorama General de la Problemática**

La Dirección General de Divulgación de la Ciencia (DGDC) tiene como fin difundir la cultura general, conocimientos básicos de la ciencia, avances de la ciencia y la tecnología del siglo pasado, así como lo que nos depara el siglo que empieza.

Para ello cuenta con diversos medios como publicaciones, eventos culturales, diversos recursos multimedia y exposición permanente en Universum.

Universum fue inaugurado el 12 de diciembre de 1992 empleando las antiguas instalaciones del Conacyt, en ese entonces contaba con 11 salas dedicadas a diversas áreas, más una de exposiciones temporales y con espacios donde crecer. En la antigua guardería del Conacyt, que se había adaptado previamente para albergar los gabinetes de ingeniería y diseño, se construyó su auditorio propio, un observatorio astronómico (Astrolab), un teatro laboratorio (Fisilab) y varias salitas para cursos y talleres, así como una sala de hidroponía; de esta forma quedó constituido lo que es conocido como la “Casita de las Ciencias”.

Actualmente cuenta con 12 salas con exposiciones permanentes y una de exposiciones temporales. También se imparte una maestría (Maestría en Divulgación de la Ciencia), para ello se adaptaron las instalaciones de la Casita de las Ciencias.

Una de las áreas importantes dentro del Universum es el área de Innovación Tecnológica la cual tiene como misión introducir a la DGDC las innovaciones tecnológicas que permitan incrementar la calidad y eficiencia de las labores de los integrantes de la institución; así como divulgar la ciencia por medio de las tecnologías digitales de cómputo y comunicaciones. Gracias a esta área se cuenta con una infraestructura de red de datos confiable y con posibilidades de expansión para migrar a nuevas tecnologías como es el caso de la fibra óptica.

Universum cuenta con un programa de Becarios los cuales son de gran importancia ya que son el puente de comunicación entre las exposiciones y los visitantes del museo.

Actualmente se cuenta con aproximadamente 200 Becarios, pero se ha llegado a contar hasta con 350. El Becario tiene un periodo de permanencia variable con un mínimo de 1 día y máximo de 2 años. Se publican dos convocatorias al año para aceptar becarios, pero si las necesidades de la dependencia requieren de uno o más becarios pueden entrar fuera de ese periodo de convocatoria. Esta convocatoria es abierta a cualquier alumno y sin importar la institución educativa de la que provenga, el alumno debe contar con un mínimo de 50% de créditos de su carrera y ser menor de 25 años, de lo contrario no pueden ser aceptados.

Al ser aceptados tienen la posibilidad de escoger un horario de acuerdo a sus necesidades, en total tienen que cubrir 20 horas a la semana, cabe señalar que la dependencia tiene un horario de Lunes a Domingo de 9:00 a 18:00 horas.

Los coordinadores del área de Becarios tienen diversas tareas a su cargo, entre ellas está coordinar a todos los Becarios, así como efectuar la planificación de horarios.

Se cuenta en esta área con archivos que contienen información personal de todos los Becarios como nombre, dirección, carrera a la que pertenece, escuela de procedencia, etc. Toda esta información se maneja de forma manual, es decir, se encuentra en papel.

Para llevar el control de la asistencia de los becarios, ellos tienen que firmar una bitácora lo cual implica llegar al tercer piso de la dependencia y anotar su nombre y firma cada que asisten.

Posteriormente los encargados tienen que registrar dicha información en un archivo de control de asistencia.

Los coordinadores realizan la planeación de horarios dos veces al año, ya que los becarios son estudiantes y sus horarios cambian cada semestre. Esto se plasma en varias hojas de control.

Para llevar a cabo este control se cuenta con un archivo en Excel y una hoja para cada sala, que contiene lo siguiente:

- Día de la semana
- Horario
- Becarios asignados en sala para cada horario

Como ejemplo práctico de lo que se hace actualmente se muestra la figura 1.1:

	A	B	C	D	E	F	G	H	I	J	K
1	HORA	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES	SÁBADO	DOMINGO			
2		Jorge	Fedex	Anahí	Karla	Anahí					
3			Jorge	Karla	Jorge	Fedex					
4	09:00		Karla	Vicente	Vicente	Vicente					
5	A										
6	10:00										
7											
8		Jorge	Fedex	Anahí	Karla	Anahí	Anahí	Angélica			
9		Jorge	Jorge	Karla	Jorge	Fedex	Angélica	Fernando			
10	10:00		Karla	Vicente	Vicente	Vicente	Adriana				
11	A		Moises								
12	11:00										
13											
14											
15		Jorge	Fedex	Anahí	Karla	Anahí	Anahí	Angélica			
16		Jorge	Jorge	Karla	Jorge	Fedex	Angélica	Fernando			
17	11:00		Karla	Vicente	Vicente	Vicente	Adriana				
18	A		Moises		Fedex		Ricardo				
19	12:00										
20											
21											
22		Jorge	Fedex	Karla	Karla	Anahí	Anahí	Angélica			
23		Jorge	Jorge	Vicente	Jorge	Fedex	Angélica	Fernando			
24	12:00		Karla		Vicente	Vicente	Adriana	Ricardo			
25	A		Moises		Fedex		Ricardo				
26	13:00										
27											
28											
29											
30		Adriana	Karla	Moises	Fedex	Fedex	Anahí	Angélica			

Figura 1.1 Hoja de Control de Becarios para una Sala

Además siete hojas para cada día de la semana con la siguiente información como se muestra en la figura 1.2:

- Sala de exposición
- Horario
- Becario

Microsoft Excel - Horarios\_sala

Archivo Edición Ver Insertar Formato Herramientas Datos Ventana ? Escribe una pregunta

Arial 10

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	HORAS	ESTRUCTURA	E. INFANTE	SENDA	MATEMÁTICAS	QUÍMICA	ENERGÍA	ANATOMÍA	CALCULO	BIOLOGÍA	UNIBALSA	CONCIENCIA	INFORMÁTICA	T. INTELIGEN	UNIVERSO	TAI
2																
3																
4	8:00															
5	9:00															
6																
7																
8																
9																
10																
11	10:00															
12	11:00															
13																
14																
15																
16																
17	11:00															
18	12:00															
19																
20																
21																
22																
23	12:00															
24	13:00															
25																
26																
27																
28																
29																
30	13:00															
31	14:00															
32																
33																
34	14:00															
35																
36																
37																
38																
39	14:00															
40																
41	15:00															
42																
43																

Hojas: Hoja14 / Hoja15 / Hoja16 / Hoja17 / Hoja18 / Hoja19 / hoja20

NUM

Figura 1.2 Hoja de control Para un Día

El manejo de este control es el siguiente:

Los coordinadores del área anotan, dependiendo del horario solicitado, el nombre del becario en la hoja correspondiente a la sala en la que fue asignado dentro de las horas y días acordados. Además se debe anotar esta información en las hojas de control para un día en todas las salas. Esta labor se realiza para cada uno de los becarios por cada periodo escolar.

Este control es flexible ya que un Becario puede cambiar su horario en forma temporal siempre y cuando cumpla con 20 horas a la semana, lo cual complica la labor de los coordinadores ya que los reportes se deben actualizar cuando hay un cambio de este tipo. No hay un límite para el número de becarios que pueden realizar estas modificaciones. Además un becario, en caso de requerirlo, puede darse de baja y esto se debe reflejar en las hojas de control.

### **1.1.1 Áreas dentro de Universum**

Por otra parte en el Universum y DGDC se cuentan con diversas Áreas:

- Actualización Técnica
- Actividades y Eventos Especiales
- Atención al Visitante
- Biblioteca
- Colecciones
- Compras
- Comunicación Gráfica
- Coordinación de Becarios
- Coordinación de la Maestría
- Coordinación de Mantenimiento y Obra Civil
- Colecciones
- Dirección de Museos
- Dirección General
- Exposiciones Itinerantes
- Fotografía
- Ingeniería
- Libros
- Multimedia
- Museografía
- Radio
- Espacio Infantil
- Fisilab- Astrolab
- Invernadero
- Jóvenes hacia la Investigación
- Mantenimiento Museográfico
- Medios Audiovisuales
- Prevención de Riesgos
- Relaciones Públicas
- Revista ¿Cómo Ves?
- Subdirección de Medios
- Subdirección de Universum
- Subdirección Innovación Tecnológica
- Subdirección Medios Comunicación

- Subdirección de Exposiciones
- Subdirección Museo de la Luz
- Taquilla
- Teatro
- Unidad Académica
- Unidad Administrativa

Las áreas dentro del Museo que requieren información de manera simultánea son la Coordinación de Becarios y Atención al Visitante.

En Atención al Visitante, se necesita conocer la información contenida en las hojas de control general con una semana de anticipación y confirmarla el día de la visita para saber si cuentan con Becarios disponibles en alguna sala, cuando hay visitas por parte de Instituciones Educativas. De esta forma proporcionar una mejor experiencia dentro del museo a los visitantes.

Puesto que los becarios pueden permanecer en una sala o estar en varias, dependiendo de las necesidades, es decir, un becario puede permanecer en una sala todo el tiempo o dar un recorrido por una o varias salas, las visitas guiadas son atendidas por uno o más becarios, por ejemplo, un becario puede acompañar a los visitantes por dos salas y a la tercera otro becario continuar con la visita; el caso extremo es que por cada sala que los visitantes recorran será necesario un becario, esta situación es muy común.

Considerando las situaciones anteriores, al llegar los becarios anotan su nombre en una hoja general en la hora (s) y sala que le corresponde, cabe señalar que este proceso es independiente de pasar lista, esta es la única forma rápida de saber quien asistió y quien esta disponible para guiar la visita. Es muy importante saber quien esta presente ya que de no estarlo es necesario asignar personal disponible a la sala o realizar un ajuste a la visita que se había programado.

### **1.1.2 Desarrollo**

Dentro del mundo de la informática existen diversos sistemas operativos y varias arquitecturas de computadoras. Dependiendo del área en que se empleará una computadora los usuarios tienen preferencia por alguna configuración en particular.

En Universum, como se mencionó, cuenta con varias áreas en las cuales se emplean computadoras para diversos fines, por ejemplo, diseño gráfico, edición de video, desarrollo de multimedia, ingeniería, etcétera. Para cada

actividad hay una configuración que se adecua mejor. Para multimedia, diseño gráfico en general es reconocido el desempeño que tienen las computadoras Apple, en particular se cuenta con equipos como iMac, G4 y G5; estos tienen diferentes versiones de sistema operativo: MacOS 9.X / OS X Jaguar y Panther. En el caso de otras actividades como las relacionadas a ingeniería se tiene mayor soporte para equipos PC y sistema operativo Windows.

Para los servicios tales como impresión, Internet, correo electrónico y DHCP se cuentan con una variedad de servidores dentro de la dependencia. Estos servidores cuentan con distintos sistemas operativos: Windows NT Server, OSX, Solaris y Linux.

Por lo que para el desarrollo del presente trabajo de tesis se debe considerar, además de la problemática antes mencionada, que el sistema debe desempeñarse bajo algunos de estos sistemas operativos y arquitecturas.

## **1.2 Objetivo de la Tesis**

Con base en lo expuesto anteriormente será necesario apoyarse en las tecnologías de la información para desarrollar un Sistema Integral de Información y Control de Becarios.

El objetivo de nuestra tesis es proveer a la dependencia de un Sistema de Información que cumpla cabalmente con los requerimientos y resuelva la problemática actual, así como contribuir al mejor desempeño de las funciones de la DGDC.

Requerimientos:

1. Registro de Becarios
2. Control de asistencia de Becarios
  - a. Encontrar la forma rápida y eficiente de registrar asistencia
  - b. Automatizar y procesar la información derivada del punto a
  - c. Transporte y Almacenamiento de información
  - d. Consulta de la información en los sistemas operativos existentes en la dependencia.

### 3. Logística

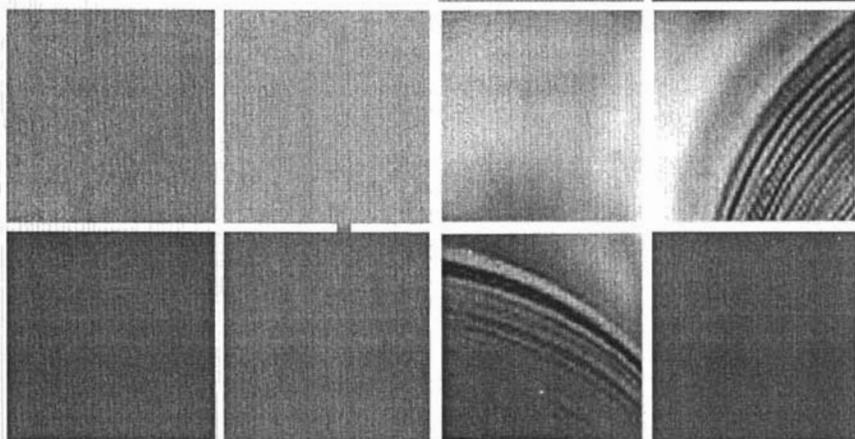
- a. Proveer información de utilidad para ambas áreas:
  - i. Ubicación de los Becarios asignados a cada sala y horario
  - ii. Manipular la información en forma dinámica
  - iii. Generar reportes estadísticos
- b. Consulta y manipulación de la información en múltiples plataformas.

Uno de los principales objetivos de esta tesis es resolver dicha problemática que existe en la dependencia aplicando los conocimientos obtenidos en el transcurso de la carrera de Ingeniería en Computación.

Antecedentes

Capítulo

2





Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**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.

# CAPÍTULO 2

## ANTECEDENTES

*En este capítulo propondremos algunas definiciones acerca de la Ingeniería de software, se describirán algunos modelos de la misma y se dejará ver cuál seguiremos para la realización del Sistema de Control de Becarios para la DGDC, se hablará también de las bases de datos, la programación orientada a objetos, los protocolos de transmisión de datos que nos interesan para el sistema, las redes de computadoras y sus topologías y por último la arquitectura cliente servidor.*

## 2.1 Antecedentes Académicos

En este capítulo se definirán algunos conceptos que debemos tener bien claros, tal es el caso de la Ingeniería de Software, las Bases de Datos, las Redes de computadoras, la programación Orientada a Objetos, además de la Arquitectura Cliente Servidor.

Comencemos por definir lo qué es la Ingeniería del Software:

### 2.1.1 Ingeniería de Software

[La Ingeniería del Software es una disciplina o área de la Informática o Ciencias de la Computación, que ofrece métodos y técnicas para desarrollar y mantener software de calidad que resuelvan problemas de todo tipo. La ingeniería de Software trata con áreas diversas de las ciencias de la Computación, tales como construcción de compiladores, sistemas operativos o desarrollos en Intranet/Internet, abordando todas las fases del ciclo de vida del desarrollo de cualquier tipo de sistemas de información y aplicables a una infinidad de áreas tales como: negocios, investigación científica, medicina, producción, logística, banca, control de personal (en nuestro caso), Internet, etc.]<sup>1</sup>

Para otros Ingeniería del Software se define como la aplicación práctica del conocimiento científico en el diseño y construcción de programas de computadora y la documentación asociada requerida para desarrollar, operar (funcionar) y mantenerlos. Se conoce también como desarrollo de software o producción de software [Bohem, 1976]<sup>2</sup>

Dadas estas dos definiciones de Ingeniería de Software podemos proponer la nuestra:

La Ingeniería del Software o Ingeniería de Programación es una disciplina que forma parte de las Ciencias de la computación, sirve para planear, diseñar y construir programas de cómputo para resolver algún problema práctico además de que el programa debe estar documentado adecuadamente para poder operar y mantenerlo en funcionamiento.

---

<sup>1</sup> PRESSMAN S. ROGER: <<Ingeniería del Software.>>, Mc Graw Hill, Prólogo, pp. XXIX

<sup>2</sup> BOEHM, B. W.: <<software Engineering>>, IEEE Transactions on Computers, C-25, núm. 12, pp. 1226-1241

Para el desarrollo de un sistema, generalmente se selecciona un modelo de proceso para la ingeniería de software según la naturaleza del proyecto, de la aplicación, los métodos y las herramientas a utilizarse, por tanto es necesario, definir la forma en como desarrollaremos el sistema de Control de Becarios dando una vista general de algunos modelos de proceso de software.

### 2.1.1.1 Modelo Lineal Secuencial

Llamado algunas veces ciclo de vida básico o modelo en cascada, el modelo lineal secuencial sugiere un enfoque sistemático, secuencial, para el desarrollo del software que comienza en un nivel de sistemas y progresa con el análisis, diseño, codificación, pruebas y mantenimiento la figura 2.1 muestra el modelo lineal secuencial para la ingeniería de software. Modelado según el ciclo de ingeniería convencional, el modelo lineal secuencial comprende las siguientes actividades:

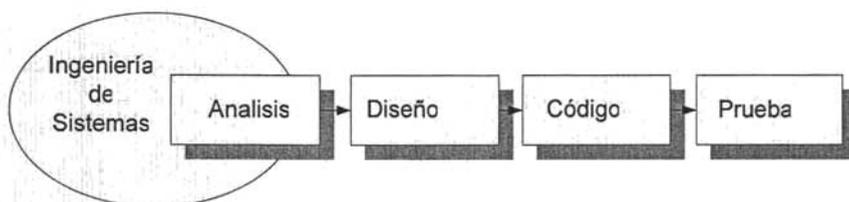


Figura 2.1. El modelo lineal secuencial.

- Ingeniería y modelado de Sistemas/Información. Como el software siempre forma parte de un sistema más grande(o empresa), el trabajo comienza estableciendo requisitos de todos los elementos del sistema y asignando al software algún subgrupo de estos requisitos. Esta visión del sistema es esencial cuando el software se debe interconectar con otros elementos como hardware, personas y bases de datos. La ingeniería de información abarca los requisitos que se recogen en el nivel de empresa estratégico y en el nivel del área de negocio.
- Análisis de los requisitos del Software. El proceso de reunión de requisitos se intensifica y se centra especialmente en el software. Para comprender la naturaleza de los programas a construirse, el ingeniero de software debe comprender el dominio de información del software, así como la función requerida comportamiento, rendimiento e interconexión.

- **Diseño.** El diseño de software es realmente un proceso de muchos pasos que se centra en cuatro atributos distintos de programa: estructura de datos, arquitectura de software, representaciones de interfaz y detalle procedimental (algoritmo). El proceso de diseño traduce requisitos en una representación del software donde se puede evaluar su calidad antes de que comience la codificación.
- **Generación de código.** El diseño se debe traducir en una forma legible por la maquina. El paso de generación de código lleva a cabo esta tarea. Si se lleva a cabo el diseño de una forma detallada, la generación de código se realiza mecánicamente.
- **Pruebas.** Una vez que se ha generado el código comienzan las pruebas del programa. El proceso de pruebas se centra en los procesos lógicos internos del software, asegurando que todas las sentencias se han comprobado, y en los procesos externos funcionales; es decir, realizar las pruebas para la detección de errores y asegurar que la entrada definida produce resultados reales de acuerdo con los resultados requeridos.
- **Mantenimiento.** El software indudablemente sufrirá cambios después de ser entregado al cliente. Se producirán cambios porque se han encontrado errores, porque el software debe acoplarse a los cambios de su entorno externo o porque el cliente requiere mejoras funcionales o de rendimiento. El soporte y mantenimiento del software vuelve a aplicar cada una de las fases precedentes a un programa ya existente y no a uno nuevo.

### 2.1.1.2 Modelo de Construcción de Prototipos

Un cliente, a menudo, define un conjunto de objetivos generales para el software, pero no identifica los requisitos detallados de entrada, proceso o salida. En otros casos, el responsable del desarrollo del software puede no estar seguro de la eficacia de un algoritmo, de la capacidad de adaptación de un sistema operativo, o de la forma en que debería tomarse la iteración hombre maquina. En estas y en otras muchas situaciones, un paradigma de construcción de prototipos puede ofrecer el mejor enfoque.

El paradigma de construcción de prototipos como se muestra (Ver figura 2.2), comienza con la recolección de requisitos. El desarrollador y el cliente encuentran y definen los objetivos globales para el software, identifican los requisitos conocidos y las áreas del esquema en donde es obligatoria más definición. Entonces aparece un diseño rápido. El diseño rápido se centra en una representación de esos aspectos del software que serán visibles para el usuario/cliente (por ejemplo: enfoques de entrada y formatos de salida). El diseño rápido lleva a la construcción de un prototipo. El prototipo lo evalúa el cliente/usuario y se utiliza para refinar los requisitos del software a desarrollar. La iteración ocurre cuando el prototipo se pone a punto para satisfacer las necesidades del cliente, permitiendo al mismo tiempo que el desarrollador comprenda mejor lo que se necesita hacer.



Figura 2.2 El paradigma de la construcción de prototipos.

Lo ideal sería que el prototipo sirviera como un mecanismo para identificar los requisitos del software. Si se construye un prototipo de trabajo, el desarrollador intenta hacer uso de los fragmentos del programa ya existente o aplica herramientas (por ejemplo: generadores de informes, gestores de ventanas, etc.) que permiten generar rápidamente programas de trabajo.

El prototipo puede servir como primer sistema. Es verdad que a los clientes y a los que desarrollan les gusta el paradigma de construcción de prototipos, ya que a los usuarios les gusta el sistema real y a los que desarrollan les gusta construir algo inmediatamente pero es recomendable trabajar con prototipos.

Para desarrollar el Sistema de Control de Becarios para la DGDC tomaremos como base El Modelo en Espiral ya que consideramos que es el mas adecuado debido a que en este modelo el cliente puede observar los avances del proyecto, así pues, definiendo la ruta a seguir comenzaremos por definir el modelo en espiral.

### 2.1.1.3 El Modelo en Espiral

El modelo en espiral, propuesto originalmente por Boehm, es un modelo de proceso de software que conjuga la naturaleza iterativa de construcción de prototipos con los aspectos controlados y sistemáticos del modelo lineal secuencial. Proporciona el potencial para el desarrollo rápido de versiones incrementales del software. En el modelo espiral, el software se desarrolla en una serie de versiones incrementales. Durante las primeras iteraciones, la versión incremental podría ser un modelo en papel o un prototipo. Durante las últimas iteraciones, se producen versiones cada vez más completas del sistema diseñado.

El modelo en espiral se divide en un número de actividades de marco de trabajo, también llamadas regiones de tareas.

- Comunicación con el cliente: las tareas requeridas para establecer comunicación entre el desarrollador y el cliente.
- Planificación: las tareas requeridas para definir recursos, el tiempo y otra información relacionadas con el proyecto
- Análisis de Riesgos: las tareas requeridas para evaluar riesgos técnicos y de gestión
- Ingeniería: las tareas requeridas para construir una o mas representaciones de la aplicación.
- Construcción y Acción: las tareas requeridas para construir, probar, instalar y proporcionar soporte al usuario (documentación y práctica).
- Evaluación del cliente: las tareas requeridas para obtener la reacción del cliente según la evaluación de las representaciones del software creadas durante la etapa de ingeniería e implementada durante la etapa de instalación.

Cada una de las regiones está compuesta por un conjunto de tareas del trabajo, llamado conjunto de tareas, que se adaptan a las características del proyecto que va a emprenderse. Para proyectos pequeños, el número de tareas de trabajo y su formalidad es bajo. Para proyectos mayores y más críticos cada región de tareas contiene tareas de trabajo que se definen para lograr un nivel más alto de formalidad.

Cuando empieza este proceso evolutivo, el equipo de ingeniería del software gira alrededor de la espiral en la dirección de las agujas del reloj, comenzando por el centro. El primer circuito de la espiral puede producir el desarrollo de una especificación de productos; los pasos siguientes en la espiral se podrían utilizar para desarrollar un prototipo y progresivamente versiones más sofisticadas del software. Cada paso por la región de planificación produce ajustes en el plan de proyección. El coste y la planificación se ajustan con la realimentación ante la evaluación del cliente. Además, el gestor del proyecto ajusta el número planificado de iteraciones requeridas para completar el software.

A diferencia del modelo de proceso clásico que termina cuando se entrega el software, el modelo en espiral puede adaptarse y aplicarse a lo largo de la vida del software de computadora. Una visión alternativa del modelo en espiral puede ser considerada examinando el eje de punto de entrada en el proyecto también reflejado en la figura 2.3.



Figura 2.3. Un modelo espiral típico

Cada uno de los cubos situados a lo largo del eje puede usarse para representar el punto de arranque para diferentes tipos de proyectos. Un

proyecto de desarrollo de conceptos comienza en el centro del espiral y continuará (aparecen múltiples iteraciones a lo largo de la espiral que limita la región sombreada central) hasta que se completa el desarrollo del concepto. Si el concepto se va a desarrollar dentro de un producto real, el proceso continúa a través del cubo siguiente (punto de entrada del proyecto de desarrollo del producto nuevo) y se inicia un nuevo proyecto de desarrollo. El producto nuevo evolucionara a través de iteraciones alrededor de la espiral siguiendo que limita la región algo más brillante que el centro. En esencia, la espiral, cuando se caracteriza de esta forma, permanece operativa hasta que el software se retira. Hay veces en que el proceso está inactivo, pero siempre que se inicie un cambio, el proceso arranca en el punto de entrada adecuado (por ejemplo: mejora del producto).

El modelo en espiral es un enfoque realista del desarrollo de sistemas y de software a gran escala. Como el software evoluciona, a medida que progresa el proceso el desarrollador y el cliente comprenden y reaccionan mejor ante riesgos en cada uno de los niveles evolutivos. El modelo en espiral utiliza la construcción de prototipos como mecanismo de reducción de riesgos, pero, lo que es más importante, permite a quien lo desarrolla aplicar el enfoque de construcción de prototipos en cualquier etapa de evolución del producto. Mantiene el enfoque sistemático de los pasos sugeridos por el ciclo de vida clásico, pero lo incorpora al marco de trabajo iterativo que refleja de forma más realista el mundo real.

El modelo en espiral demanda una consideración directa de los riesgos técnicos en todas las etapas del proyecto, y, si se aplica adecuadamente, debe reducir los riesgos antes de que éstos se conviertan en problemas.

Pero igual que otros paradigmas, el modelo en espiral no es la panacea. Puede resultar difícil convencer a grandes clientes (particularmente en situaciones bajo contrato) de que el enfoque evolutivo es controlable. Requiere una considerable habilidad para la evaluación del riesgo, y cuenta con esta habilidad para el éxito. Si un riesgo importante no es descubierto y gestionado, indudablemente surgirán problemas. Finalmente, en el modelo no se ha utilizado tanto como los paradigmas lineales secuenciales o de construcción de prototipos. Todavía tendrán que pasar muchos años antes de que se determine con absoluta certeza la eficacia de este nuevo e importante paradigma.

## 2.1.2 Bases de Datos

Podemos dar varias definiciones de Bases de Datos:

1. Conjunto de datos interrelacionados con independencia física y lógica, consistentes, íntegros y con redundancia controlada.
2. Colección integrada y generalizada de datos, estructurada atendiendo a las relaciones naturales, de modo que suministre todos los caminos de acceso necesario a cada unidad de datos con objeto de poder atender a todas las necesidades de los diferentes usuarios.
3. Colección de datos integrados, con redundancia controlada y una estructura que refleje las interrelaciones y restricciones existentes en el mundo real; los datos han de ser compartidos por diferentes usuarios y aplicaciones, manteniéndose independientes los datos de estas últimas. La definición y descripción serán únicas para cada tipo de dato, debiendo estar almacenadas junto a los datos. Los procedimientos de actualización y recuperación serán comunes y bien determinados, siendo capaces de conservar la integridad, seguridad y confidencialidad del conjunto de datos.

### 2.1.2.1 Análisis del Concepto de Base de Datos

El concepto de Base de Datos determina algunas características que le son propias, por ejemplo:

- El mundo real considera interrelaciones entre datos y restricciones semánticas que deben estar presentes en una base de datos. Una base de datos no solo debe almacenar entidades y atributos (recordar los sistemas tradicionales de archivos), sino que también debe almacenar interrelaciones entre datos. Por otro lado, actualmente se le está dando mucha importancia a las restricciones semánticas, de manera que éstas se almacenan junto con los datos.
- La redundancia de datos debe ser controlada, de forma que no existan duplicidades perjudiciales ni innecesarias. Las redundancias físicas, convenientes muchas veces a fin de responder a objetivos de eficiencia, sean tratadas por el mismo sistema, de modo que no puedan producirse incoherencias. Esto significa que en las bases de datos NO está permitida la redundancia lógica, pero si se admite cierta redundancia física por motivos de eficiencia.
- Las bases de datos pretenden servir a toda la organización, es decir a múltiples usuarios y a diferentes aplicaciones.

- La independencia, tanto lógica como física, de los tratamientos sobre los datos y estos mismos, ha tenido una enorme influencia en la arquitectura de los SGBD (recordar los sistemas tradicionales de archivos).
- La definición y descripción del conjunto de datos contenido en la base debe ser única e integrada con los mismos datos. (recordar los sistemas tradicionales de archivos). En las bases de datos, la descripción, y en algunos casos, también una definición y documentación completas (meta-datos) se almacenan junto con los datos, de modo que éstos están documentados, y cualquier cambio que se produzca en la documentación debe quedar recogido en el sistema.
- La actualización y recuperación de las bases de datos debe realizarse mediante procesos bien determinados, incluidos en el SGBD; procedimientos que han de estar diseñados de modo que se mantenga la integridad, seguridad y confidencialidad de la base.

### 2.1.2.2 Las Características elementales de una Base de Datos

El objetivo de disminuir la redundancia de un conjunto de datos determina dos características fundamentales que poseerá cualquier sistema de Bases de Datos:

Integrada: se entiende que una base de datos puede considerarse como una unificación de varios archivos de datos independientes, donde se elimina parcial o totalmente cualquier **redundancia** entre los mismos. Por ejemplo: una base de datos específica puede contener registros de EMPLEADO, que incluyen el nombre, dirección, departamento, salario, etc. y, existir registros de INSCRIPCIÓN que representan inscripciones de empleados en cursos de capacitación. Supongamos que para llevar a cabo el proceso de administración de los cursos se necesita conocer el departamento de cada estudiante inscrito. Desde luego, no hay necesidad de incluir este dato (redundante) en los registros de INSCRIPCIÓN, siempre se puede obtener recurriendo a los registros de EMPLEADO correspondiente.

Compartida: Se entiende que partes individuales de la Base de Datos pueden compartirse entre varios usuarios distintos, en el sentido que cada uno de ellos puede tener acceso a la misma parte de la Base de Datos y utilizarla con propósitos diferentes. Tal comportamiento es en verdad consecuencia del hecho de que la Base de Datos es integrada. En el caso

del ejemplo anterior se tiene que los datos de los registros de EMPLEADOS son compartidos por usuarios del departamento de personal y capacitación. Consecuencia del mismo hecho, que la Base de Datos es integrada, se advierte en que cualquier usuario tendrá acceso sólo a algún subconjunto de la Base completa, además, los subconjuntos de diferentes usuarios se procesarán de muy diversas maneras. En otras palabras, diferentes usuarios percibirán de modos muy distintos una Base de Datos.

### 2.1.2.3 Ventajas de las Bases de Datos

Tabla de resumen de las ventajas de las bases de datos.	
Referidas a	Ventajas
Los datos	<ul style="list-style-type: none"><li>• Independencia de estos respecto de los tratamientos y viceversa</li><li>• Mejor disponibilidad de los mismos</li><li>• Mayor eficiencia en la recogida, codificación y entrada</li></ul>
Los resultados	<ul style="list-style-type: none"><li>• Mayor coherencia</li><li>• Mayor valor informativo</li><li>• Mejor y más normalizada documentación de la información</li></ul>
Los usuarios	<ul style="list-style-type: none"><li>• Acceso más rápido y sencillo de los usuarios finales</li><li>• Más facilidades para compartir los datos por el conjunto de los usuarios</li><li>• Mayor flexibilidad para atender a demandas cambiantes.</li></ul>

Tabla 2.1

#### Análisis de la tabla anterior (tabla 2.1)

- Independencia de los datos respecto a los tratamientos y viceversa: esto supone que un cambio en los tratamientos no imponga un nuevo diseño lógico y/o físico de la base de datos. Por otro lado, cambios en la incorporación, desaparición de datos, cambios en la estructura física o caminos de acceso no deben obligar a alterar los programas. Así se evita la reprogramación de las aplicaciones. Es el punto de partida para la adaptación de los sistemas de información a la evolución de las organizaciones.

- Coherencia de los resultados: debido a que la información de la base de datos se recoge y se almacena una sola vez, en todos los tratamientos se utilizan los mismos datos, por lo que los resultados de estos son coherentes y comparables. Así, se eliminan las divergencias en los resultados.
- Mejor disponibilidad de los datos para el conjunto de los usuarios: en una base de datos ningún usuario es propietario de los datos, pues éstos se comparten entre las aplicaciones, existiendo una mayor disponibilidad y transparencia.
- Mayor valor informativo: esto se refiere al concepto de sinergia, en donde el valor informativo del conjunto de datos es superior a la suma del valor informativo de los elementos individuales.
- Mejor y más normalizada documentación: la mayoría de los SGBD proporcionan herramientas para reflejar el contenido semántico de los datos, es decir, incluyen una descripción de los datos dentro del sistema.
- Mayor eficiencia en la captura, validación e ingreso de datos al sistema: al no existir redundancias, los datos se capturan y validan una sola vez aumentando el rendimiento del proceso previo al almacenamiento
- Reducción del espacio de almacenamiento: por un lado, la disminución de redundancias y las técnicas de compactación hacen que disminuya el espacio en disco. Sin embargo, los diccionarios, referencias, punteros, listas invertidas también ocupan espacio.

#### **2.1.2.4 Componentes de los Sistemas de Bases de Datos**

Un sistema de Bases de Datos contempla los siguientes componentes:

- La base de datos
- El Sistema de Gestión de Bases de Datos(SGBD, DBMS) o motor, tal como Oracle, Sybase, etc.
- Programas de aplicación
- Un conjunto de usuarios (finales, DBA, programadores de aplicaciones, etc.)
- Máquinas
- Programas utilitarios (generadores de informes, de interfaces, herramientas de desarrollo, de administración, etc.)

En la Figura 2.4 - se puede observar un esquema general del la arquitectura de una base de datos, en la cual se detallan los principales componentes de ella además de las relaciones entre ellos y la base de datos lógica. Veamos una descripción simple de aquellos elementos:

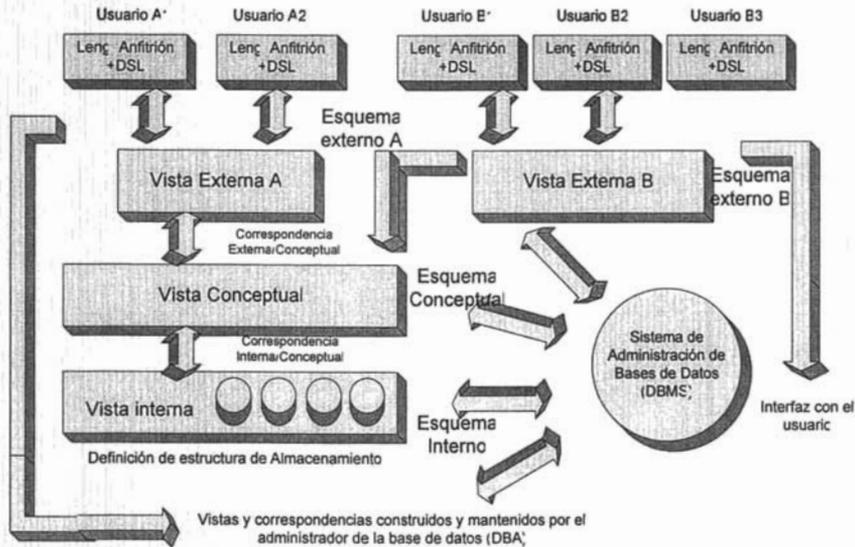


Figura 2.4. Arquitectura de una Base de Datos.

Una Vista Externa es una visión particular de un usuario o un grupo de usuarios de la Base de Datos. El Esquema Externo representa una forma de definición o formalización de esta vista externa.

La Vista Conceptual pretende ser la representación total y abstracta de los datos que componen la Base; la formalización de esta se logra mediante el Esquema Conceptual.

Por último, la Vista Interna es de un nivel muy bajo y corresponde al almacenamiento físico de los datos de la Base, sobre un Esquema Interno que es la formalización de esto, tipos de registros almacenados, índices, etc.

Las correspondencias se pueden definir como una asociación de distintas representaciones para un mismo dato.

Un **DSL** es un sublenguaje de datos, es una combinación de dos lenguajes: un lenguaje de definición de datos (**DDL**) y un lenguaje de manipulación de Datos (**DML**). Este lenguaje representa un nexo entre el Sistema de Base de Datos y algún lenguaje anfitrión (por ejemplo: COBOL, FORTRAN, C, etc.); el DSL provee herramientas a los lenguajes

---

tradicionales para que se integren al Sistema de Base de Datos. Puede haber distintos tipos de DSL para un mismo sistema.

**DBMS** es la sigla en inglés de Sistema de Administración de Bases de Datos, que corresponde al Software que maneja todos los accesos a la Base de Datos, cada solicitud de acceso de un usuario al SABD es interpretada e inspeccionadas las correspondencias, generando, a continuación, una respuesta coherente a las necesidades de la pregunta.

La interfaz con el Usuario es el límite de acceso que tiene un Usuario común a la Base, todo lo que está bajo este límite es transparente (desconocido) para él.

Por último, el Administrador de Bases de Datos (DBA) corresponde a la persona o grupo de personas encargada del control general del sistema. Sus responsabilidades o funciones incluyen:

- Decidir el contenido de la Base de Datos: comprende la identificación de entidades de interés para la organización y los datos a registrar de estas entidades. Luego se define el contenido de la Base de Datos generando un Modelo Conceptual.
- Decidir la estructura de almacenamiento y la estrategia de acceso: esto es decidir como deben representarse los datos en forma interna y hacer la correspondencia entre estos y el modelo conceptual ya definido.
- Vincularse con los usuarios: comprende toda una labor de prestación de servicios que busca garantizar la existencia, en la Base, de los datos necesarios y la formalización de los distintos esquemas externos.
- Definir los controles de autorización y procedimientos de validación: involucra la definición de restricciones de seguridad y protección para la conservación de la integridad de los Datos.
- Definir una estrategia de respaldo y recuperación: esto corresponde a un esquema de seguridad más amplio que lo anterior y, básicamente, su objetivo es la operación exitosa del sistema.
- Controles de desempeño y responder a los cambios de requerimiento: la idea aquí es lograr un desempeño aceptable, según expectativas, del Sistema mediante mecanismos de control.

### 2.1.3 Programación Orientada a Objetos

La programación orientada a objetos (POO) es una forma de programación que utiliza objetos, ligados mediante mensajes, para la solución de problemas. Puede considerarse como una extensión natural de la programación estructurada en un intento de potenciar los conceptos de modularidad y reutilización del código.

Los mecanismos básicos de la programación orientada a objetos son: objetos, mensajes, métodos y clases.

#### 2.1.3.1 Objetos

Un programa tradicional se compone de procedimientos y de datos. Un programa orientado a objetos se compone solamente de objetos. Un objeto es una encapsulación genérica de datos y de los procedimientos para manipularlos. Dicho de otra forma, un objeto es una entidad que tiene unos atributos particulares, las propiedades, y unas formas de operar sobre ellos, los métodos. Por lo tanto, un objeto contiene, por una parte, operaciones que definen su comportamiento, y por otra, variables manipuladas por esas operaciones que definen su estado.

Por ejemplo, un formulario es un objeto. El color de fondo del formulario, la anchura, la altura, etc. Son propiedades. Las rutinas, lógicamente transparentes al usuario, que permiten maximizar la ventana, minimizarla, etc., son métodos.

#### 2.1.3.2 Mensajes

Cuando se ejecuta un programa orientado a objetos, los objetos están recibiendo, interpretando y respondiendo a mensajes de otros objetos. Esto marca una clara diferencia con respecto a los elementos de datos pasivos de los sistemas tradicionales. Por ejemplo, en Visual Basic, un mensaje está asociado con un procedimiento, de tal forma que cuando un objeto recibe un mensaje la respuesta a ese mensaje es ejecutar el procedimiento asociado. Este procedimiento recibe el nombre de método. Otro ejemplo, cuando un usuario quiere maximizar una ventana, lo que hace simplemente es pulsar el botón de la misma que realiza esa acción. Eso, provoca que Windows envíe un mensaje a la ventana para indicar que tiene que maximizarse. Como respuesta a que este mensaje se ejecutara el método programado para ese fin.

### 2.1.3.3 Métodos

Un método se implementa en una clase de objetos y determina como tiene que actuar el objeto cuando recibe un mensaje. En adición, las propiedades permitirán almacenar información para dicho objeto. Un método puede también enviar mensajes a otros objetos solicitando una acción o información.

La estructura más interna de un objeto está oculta para otros usuarios y la única conexión que tiene con el exterior son los mensajes. Los datos que están dentro de un objeto solamente pueden ser manipulados por los métodos asociados al propio objeto.

La ejecución de un programa orientado a objetos realiza fundamentalmente tres cosas:

1. Crea los objetos necesarios
2. Los mensajes enviados a unos y a otros objetos dan lugar a que se procese internamente la información.
3. Finalmente, cuando los objetos no son necesarios, son borrados, liberándose la memoria ocupada por los mismos.

### 2.1.3.4 Clases

Una clase es un tipo de objeto definido por el usuario. Una clase equivale a la generalización de un tipo específico de objetos. Un objeto es una determinada clase se crea en el momento en que se defina una variable de dicha clase.

### 2.1.3.5 Características de la POO

Las características fundamentales de la POO son: abstracción, encapsulamiento, herencia y polimorfismo.

### 2.1.3.6 Abstracción

Por medio de la abstracción conseguimos no detenernos en los detalles concretos de las cosas que no interesen en cada momento, sino generalizar y centrarse en los aspectos que permitan tener una visión global del tema. Por ejemplo, el estudio de una computadora ponemos realizarlo a nivel de

---

funcionamiento de sus circuitos electrónicos, en términos de corriente, tensión, etc., o a nivel de transferencia entre registros, centrándose así el estudio del flujo de información entre las unidades que lo componen (memoria, unidad aritmética, unidad de control, registros, etc.), sin importarnos el comportamiento de los circuitos electrónicos que componen estas unidades.

### **2.1.3.7 Encapsulamiento**

Esta característica permite ver un objeto como una caja negra en la que se ha introducido de alguna manera toda la información relacionada con dicho objeto. Esto nos permitirá manipular los objetos como unidades básicas, permaneciendo oculta su estructura interna.

La abstracción y la encapsulación están representadas por la clase. La clase es una abstracción, porque en ella se definen las propiedades de un determinado conjunto de objetos con características comunes, y es una encapsulación porque constituye una caja negra que encierra tanto los datos que almacena cada objeto como los métodos que permiten manipularlos.

### **2.1.3.8 Herencia**

La herencia permite el acceso automático a la información contenida en otras clases. De esta forma, la reutilización del código está garantizada. Con la herencia todas las clases están clasificadas en una jerarquía estricta.

Cada lista tiene su superclase (la clase superior en la jerarquía), y cada clase puede tener una o más subclases (las clases inferiores en la jerarquía). Las clases que están en la parte inferior en la jerarquía se dice que heredan de las clases que están en la parte superior en la jerarquía.

### **2.1.3.9 Polimorfismo**

Esta característica permite implementar múltiples formas de un mismo método, dependiendo cada una de ellas de la clase sobre la que se realice la implementación. Esto hace que se pueda acceder a una variedad de métodos distintos (todos con el mismo nombre) utilizando exactamente el mismo medio de acceso.

### 2.1.3.10 Constructores y Destructores

Un constructor es un procedimiento especial de una clase que es llamado automáticamente siempre que se crea un objeto de esa clase. Su función es iniciar el objeto.

Un destructor es un procedimiento especial de una clase que es llamado automáticamente siempre que se destruye un objeto de esa clase. Su función es realizar cualquier tarea final en el momento de destruir el objeto.

### 2.1.4 Redes De Computadoras

Una red de computadoras es un conjunto de terminales, nodos, servidores y elementos de propósito especial que interaccionan entre sí con la finalidad de intercambiar información y compartir recursos.

Anteriormente (aunque existen todavía) la información se almacenaba en los llamados “Mainframes”. De este elemento, diferentes terminales eran conectadas con la finalidad de recibir información y compartir los recursos. El problema con este tipo de “red” es que se basaba en un sistema centralizado, es decir, del mainframe se extraía toda la información, teniendo de esta forma la limitante en cuanto a la capacidad de almacenamiento de datos (figura 2.5)

Por otro lado, con la utilización de un mainframe, el fallo del mismo provocaba el fallo de todo el sistema, lo cual traía consecuencias muy graves en el manejo de una organización.

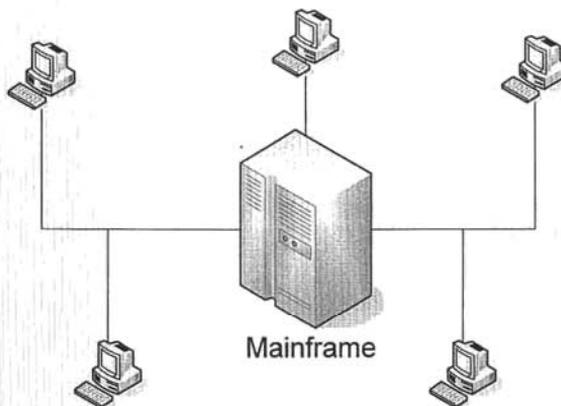


Figura 2.5 Mainframe

Con la introducción de las diferentes clases de redes de computadoras y tecnologías, surgió la posibilidad de utilizar diferentes “servidores” que, como su nombre lo indica, proveen servicios a un conjunto de nodos denominados “clientes”. De esta forma, no existe limitante en cuanto almacenamiento de información, ya que nuevos servidores pueden ser instalados (Ver figura 2.6), dando así, la facilidad en la expansión de las redes.

De esta forma, una “red de computadoras” se define como un sistema distribuido. En un sistema distribuido, la existencia de múltiples computadoras autónomas es transparente para el usuario. El usuario puede teclear una orden para ejecutar un programa y éste se ejecutará. La tarea de seleccionar al mejor o el correspondiente procesador y colocar los resultados en el lugar apropiado, corresponde al sistema operativo y a la red en sí. En un sistema distribuido, el usuario no está consciente de que existen múltiples procesadores. El sistema se ve como un monoprocesador virtual. El sistema distribuido es un sistema de software construido encima de una red. (Figura 2.6). Las razones principales que se pueden tomar en consideración para la utilización de una red de computadoras son:

**Integridad.** Hacer de un sistema de varios elementos una sola herramienta, en donde se utilicen las características y las aptitudes de cada uno de la mejor forma posible.

**Flexibilidad.** Fácil de utilizar, rápida comunicación de datos en cualquier momento, etc.

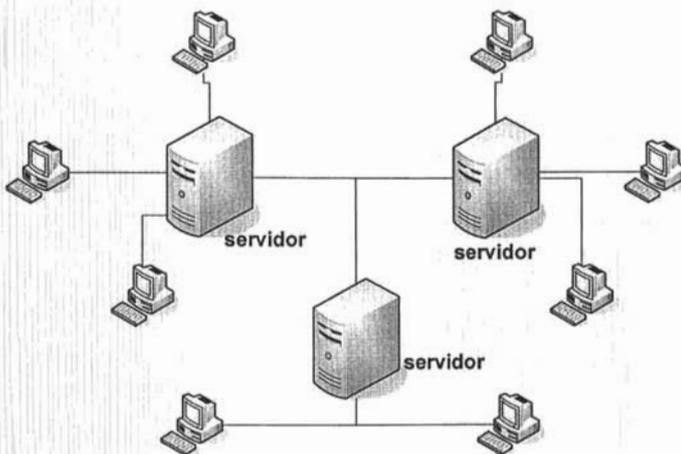


Figura 2.6. Ejemplo de una expansión de redes

Una topología de red va relacionada con el tipo de conexión entre los diferentes dispositivos que forman la red de computadoras, es decir, la forma como se interconectan todos los dispositivos para formar la red. Existen tres topologías básicas que son utilizadas para formar redes: Star (estrella), Ring (Anillo) y Bus. De estas tres topologías principales, es posible generar diferentes topologías “híbridas”, logrando así, una integración entre las topologías básicas, expandiendo las redes de computadoras hacia redes de cobertura global.

#### 2.1.4.1 Topología Estrella

La topología Estrella o Star (Ver figura 2.7) consta de una unidad central que controla el flujo de información a través de la red. La topología Star tiene limitaciones en cuanto a rendimiento y confiabilidad, ya que el tamaño de la red depende directamente de la capacidad del controlador central (número de conexiones que puede soportar) y en caso de fallar éste, todo el sistema deja de funcionar. Por otro lado, tiene la ventaja de poderse administrar únicamente administrando el dispositivo central.

En la topología estrella se tiene un control de transmisión Centralizado y una forma de transferencia de Conmutación.

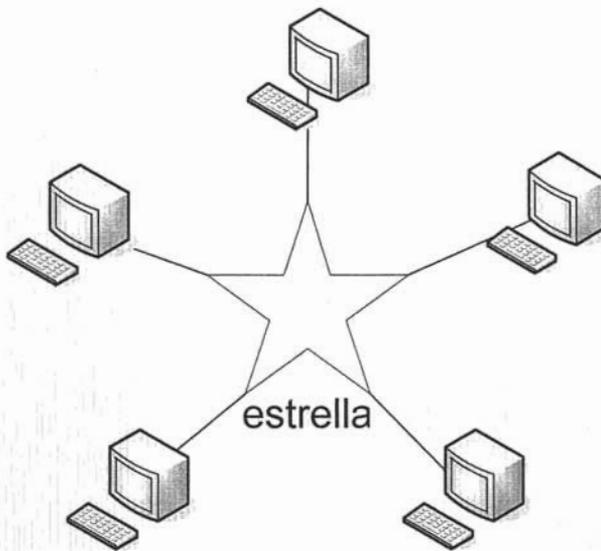


Figura 2.7. Topología estrella.

### 2.1.4.2 Topología Anillo

Una de sus características más importantes es que está formada por un conjunto de enlaces punto a punto (figura 2.8), lo cual es una topología bien entendida y probada, en donde la información es pasada a través de los nodos de uno a uno en una comunicación punto a punto. La ventaja que tiene esta topología es que no se requiere un cuarto de control central, aunque la desventaja es que si uno de los enlaces peer to peer que la forman se rompe (o se desconecta debido a errores en la transmisión, etc.), la red deja de funcionar.

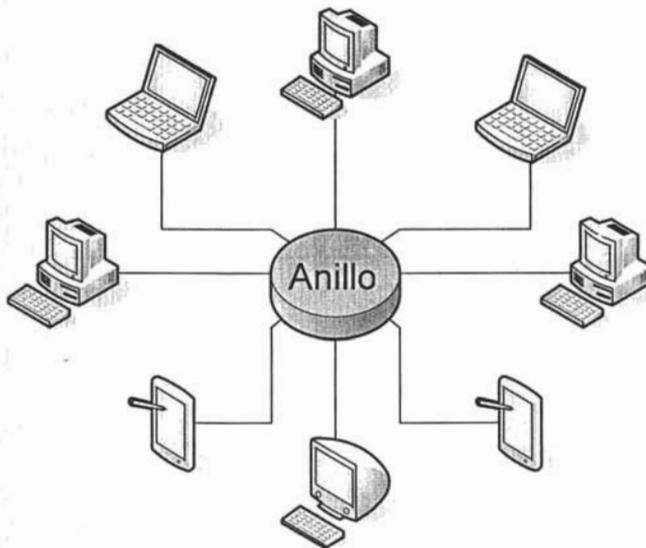


Figura 2.8. Topología anillo

El control de transmisión que usa esta topología es Distribuido y su modo de transferencia es de Conmutación. La tecnología común que utiliza dicha topología es denominada Token Ring.

Token Ring es una tecnología desarrollada por IBM, corresponde al estándar IEEE 802.5. El diseño básico es un anillo de nodos que no superan 256, operando a 4 ó 16 Mbps. En Token Ring se utiliza un código de autorización llamado Token que actúa como método de acceso al medio denominado Token Passing.

El método de acceso al medio Token Passing trabaja de la siguiente forma. Si no hay mensaje, el token (tres bytes) es enviado a través del anillo. Cuando un nodo A con un mensaje a enviar recibe el token, retiene éste y envía el mensaje, el cual incluye un código de identificación del destinatario. Los nodos ignoran el mensaje si no es para sí mismo, en caso contrario, obtienen la información. La información sigue viajando hasta que se completa su trayectoria alrededor del anillo hasta que llega al nodo A. Dicho nodo suelta el token para que pase nuevamente alrededor del anillo para futuros envíos de información.

### 2.1.4.3 Topología Bus

En esta topología no existe un CPU o similar que controle la comunicación entre los nodos. Cada nodo está conectado a un bus, donde cada nodo actúa como si fuera parte de una red anillo, pero ninguno depende del nodo siguiente para que el flujo de información continúe, ni tampoco depende del nodo anterior para que la información llegue a él (figura 2.9).

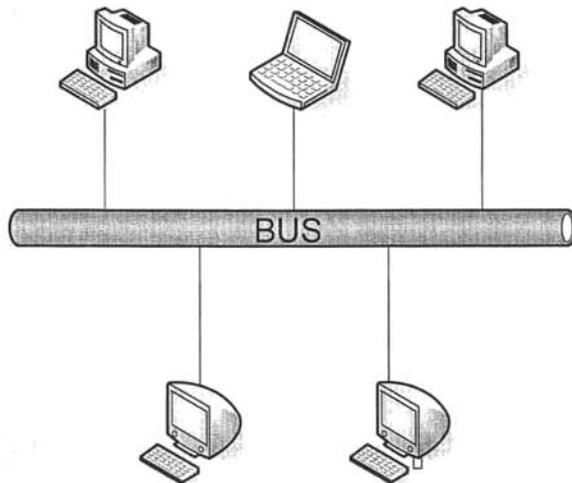


Figura 2.9. Topología bus

La tecnología común que trabaja bajo una topología Bus es denominada Ethernet.

Ethernet fue desarrollada por Digital, Intel y Xerox, normalizada con IEEE 802.3. Ethernet distribuye paquetes de datos de longitud variable con una

velocidad de 10 Mbps a los diferentes nodos dispersos a lo largo de un bus que comúnmente es cable coaxial.

Los nodos separados hasta un 50m de largo pueden ser también unidos por cable par trenzado. Una red Ethernet puede estar formada hasta por 1024 nodos.

Así como Token Ring utiliza un Token como acceso al medio, Ethernet se basa en el acceso al medio denominado CSMA/CD (Carrier Sense Multiple Access with Collision Detect). Es denominada Carrier Sense porque cada nodo es capaz de saber si la información que viaja en el bus es para si mismo o no. Multiple Access porque como se ha mencionado, un bus es compartido por todos los nodos que forman la red. Collision Detect porque cada nodo sabe si existe información que viaja en la red y es posible detectar y eliminar colisiones.

#### 2.1.4.4 Protocolos

Después de haber visto las topologías definamos qué es un protocolo: Los protocolos son reglas y procedimientos para la comunicación, es decir, cuando dos equipos están conectados en red, las reglas y procedimientos técnicos que dictan su comunicación e interacción se denominan protocolos.

##### ¿Cómo funciona un protocolo?

La operación técnica en la que los datos son transmitidos a través de la red se puede dividir en dos pasos discretos, sistemáticos. A cada paso se realizan ciertas acciones que no se pueden realizar en otro paso. Cada paso incluye sus propias reglas y procedimientos, o protocolo.

Los pasos del protocolo se tienen que llevar a cabo en un orden apropiado y que sea el mismo en cada uno de los equipos de la red. En el equipo origen, estos pasos se tienen que llevar a cabo de arriba hacia abajo. En el equipo de destino, estos pasos se tienen que llevar a cabo de abajo hacia arriba.

En el equipo origen:

1. Se dividen en secciones más pequeñas, denominadas paquetes.
2. Se añade a los paquetes información sobre la dirección, de forma que el equipo de destino pueda determinar si los datos le pertenecen.

3. Prepara los datos para transmitirlos a través de la NIC y enviarlos a través del cable de la red.

En el equipo destino: (se realizan los mismos pasos pero en sentido inverso):

1. Toma los paquetes de datos del cable y los introduce en el equipo a través de la NIC.
2. Extrae de los paquetes de datos toda la información transmitida eliminando la información añadida por el equipo origen.
3. Copia los datos de los paquetes en un búfer para reorganizarlos enviarlos a la aplicación.
4. Los equipos origen y destino necesitan realizar cada paso de la misma forma para que los datos tengan la misma estructura al recibirse que cuando se enviaron.

### Protocolos encaminados

Hasta mediados de los ochenta, la mayoría de las redes de área local (LAN) estaban aisladas. Una LAN servía a un departamento o a una compañía y rara vez se conectaba a entornos más grandes. Sin embargo, a medida que maduraba la tecnología LAN, y la comunicación de los datos necesitaba la expansión de los negocios, las LAN evolucionaron, haciéndose componentes de redes de comunicaciones más grandes en las que las LAN podían hablar entre sí.

Los datos se envían de una LAN a otra a lo largo de varios caminos disponibles, es decir, *se encaminan*. A los protocolos que permiten la comunicación LAN a LAN se les conoce como *protocolos encaminados*. Debido a que los protocolos encaminados se pueden utilizar para unir varias LAN y crear entornos de red de área extensa, han tomado gran importancia.

Los protocolos de red proporcionan lo que se denominan «servicios de enlace». Estos protocolos gestionan información sobre direccionamiento y encaminamiento, comprobación de errores y peticiones de retransmisión. Los protocolos de red también definen reglas para la comunicación en un entorno de red particular como es Ethernet o Token Ring.

**IP:** El protocolo de TCP/IP para el encaminamiento de paquetes.

**IPX:** El protocolo de Novell para el encaminamiento de paquetes.

**NWLink:** La implementación de Microsoft del protocolo IPX/SPX.

**NetBEUI:** Un protocolo de transporte que proporciona servicios de transporte de datos para sesiones y aplicaciones NetBIOS.

**DDP (Protocolo de entrega de datagramas):** Un protocolo de Apple Talk para el transporte de datos.

Veamos ahora los protocolos que nos interesan para el desarrollo del Sistema de Control de Becarios para la DGDC

#### 2.1.4.5 TCP/IP

El Protocolo de control de transmisión/Protocolo Internet (TCP/IP) es un conjunto de Protocolos aceptados por la industria que permiten la comunicación en un entorno heterogéneo (formado por elementos diferentes). Además, TCP/IP proporciona un protocolo de red encaminado y permite acceder a Internet y a sus recursos. Debido a su popularidad, TCP/IP se ha convertido en el estándar de hecho en lo que se conoce como *interconexión de redes*, la intercomunicación en una red que está formada por redes más pequeñas.

TCP/IP se ha convertido en el protocolo estándar para la interoperabilidad entre distintos tipos de equipos. La interoperabilidad es la principal ventaja de TCP/IP. La mayoría de las redes permiten TCP/IP como protocolo. TCP/IP también permite el encaminamiento y se suele utilizar como un protocolo de interconexión de redes.

Entre otros protocolos escritos específicamente para el conjunto TCP/IP se incluyen:

- **SMTP** (Protocolo básico de transferencia de correo). Correo electrónico.
- **FTP** (Protocolo de transferencia de archivos). Para la interconexión de archivos entre equipos que ejecutan TCP/IP.
- **SNMP** (Protocolo básico de gestión de red). Para la gestión de redes.

Diseñado para ser encaminado, robusto y funcionalmente eficiente, TCP/IP fue desarrollado por el Departamento de Defensa de Estados Unidos como un conjunto de protocolos para redes de área extensa (WAN). Su propósito era el de mantener enlaces de comunicación entre sitios en el caso de una guerra nuclear. Actualmente, la responsabilidad del desarrollo de TCP/IP reside en la propia comunidad de Internet. La utilización de TCP/IP ofrece varias ventajas:

---

**Es un estándar en la industria.** Como un estándar de la industria, es un protocolo abierto. Esto quiere decir que no está controlado por una única compañía, y está menos sujeto a cuestiones de compatibilidad. Es el protocolo, de hecho, de Internet.

**Contiene un conjunto de utilidades para la conexión de sistemas operativos diferentes.** La conectividad entre un equipo y otro no depende del sistema operativo de red que esté utilizando cada equipo.

**Utiliza una arquitectura escalable, cliente/servidor.** TCP/IP puede ampliarse (o reducirse) para ajustarse a las necesidades y circunstancias futuras.

### 2.1.4.6 Protocolo Internet (IP)

El Protocolo Internet (IP) es un protocolo de conmutación de paquetes que realiza direccionamiento y encaminamiento. Cuando se transmite un paquete, este protocolo añade una cabecera al paquete, de forma que pueda enviarse a través de la red utilizando las tablas de encaminamiento dinámico. IP es un protocolo no orientado a la conexión y envía paquetes sin esperar la señal de confirmación por parte del receptor. Además, IP es el responsable del empaquetado y fragmentación de los paquetes requeridos por los niveles físico y de enlace de datos del modelo OSI. Cada paquete IP está compuesto por una dirección de origen y una de destino, un identificador de protocolo, un checksum (un valor calculado) y un TTL (tiempo de vida, del inglés *time to live*). El TTL indica a cada uno de los routers de la red entre el origen y el destino cuánto tiempo le queda al paquete por estar en la red. Funciona como un contador o reloj de cuenta atrás. Cuando el paquete pasa por el router, éste reduce el valor en una unidad (un segundo) o el tiempo que llevaba esperando para ser entregado. Por ejemplo, si un paquete tiene un TTL de 128, puede estar en la red durante 128 segundos o 128 saltos (cada parada, o router, en la red), o una combinación de los dos. El propósito del TTL es prevenir que los paquetes perdidos o dañados (como correos electrónicos con una dirección equivocada) estén vagando en la red. Cuando la cuenta TTL llega a cero, se retira al paquete de la red.

Otro método utilizado por IP para incrementar la velocidad de transmisión es el conocido como «ANDing»<sup>3</sup>. La idea del ANDing es determinar si la dirección es de un sitio local o remoto. Si la dirección es local, IP preguntará al Protocolo de resolución de direcciones (ARP) por la dirección hardware de la máquina de destino. Si la dirección es remota, el IP comprueba su tabla de encaminamiento local para encaminarlo al destino. Si existe un camino, el paquete se envía por ahí. Si no existe el camino, el paquete se envía a través del gateway a su destino.

El TCP es el responsable de la transmisión fiable de datos desde un nodo a otro. Es un protocolo orientado a la conexión y establece una conexión (también conocida como una sesión, circuito virtual o enlace) entre dos máquinas antes de transferir ningún dato.

Para establecer una conexión fiable, TCP utiliza lo que se conoce como «acuerdo en tres pasos». Establece el número de puerto y los números de secuencia de inicio desde ambos lados de la transmisión. El acuerdo consta de tres pasos:

1. El solicitante envía al servidor un paquete especificando el número de puerto que él planea utilizar y el número de secuencia inicial (ISN).
2. El servidor responde con su ISN, que consiste en el ISN del solicitante más uno.
3. El solicitante responde a la respuesta del servidor con el ISN del servidor más uno.

En orden a mantener una conexión fiable, cada paquete tiene que contener:

- Un número de puerto TCP origen y destino.
- Un número de secuencia para mensajes que tienen que dividirse en partes más pequeñas.
- Un checksum que asegura que la información se ha recibido sin error.
- Un número de confirmación que indica a la máquina origen qué partes de la información han llegado.
- Ventanas deslizantes (*Sliding Windows*) TCP.

---

<sup>3</sup> Un AND es una operación lógica que combina los valores de dos bits (0, 1) o dos valores lógicos (verdadero, falso) y devuelve un 1 (verdadero) si los valores de ambas entradas son 1 (verdadero) y devuelve 0 (falso) en caso contrario.

---

### 2.1.4.7 Apple Talk

Apple Talk es la jerarquía de protocolos de Apple Computer para permitir que los equipos Apple Macintosh compartan archivos e impresoras en un entorno de red. Se introdujo en 1984 como una tecnología LAN autoconfigurable. Apple Talk también está disponible en muchos sistemas UNIX que utilizan paquetes comerciales y de libre distribución. El conjunto de protocolos AppleTalk permite compartir archivos a alto nivel utilizando AppleShare, los servicios de impresión y gestores de impresión de LaserWriter, junto con la secuencia de datos de bajo nivel y la entrega de datagramas básicos.

#### Protocolos AppleTalk

**AppleTalk:** Una colección de protocolos que se corresponde con el modelo OSI. Soporta LocalTalk, EtherTalk y TokenTalk.

**LocalTalk:** Describe el cable par trenzado apantallado utilizado para conectar equipos Macintosh con otros Macintosh o impresoras. Un segmento LocalTalk permite hasta un máximo de 32 dispositivos y opera a una velocidad de 230 Kbps. Es la forma original del protocolo. La comunicación se realiza por uno de los puertos serie del equipo. La velocidad de transmisión no es muy rápida pero es adecuada para los servicios que en principio se requieren de ella, principalmente compartir impresoras.

**Ether Talk:** AppleTalk sobre Ethernet. Opera a una velocidad de 10 Mbps. Fast Ethernet opera a una velocidad de 100 Mbps. Es la versión de Appletalk sobre Ethernet. Aumenta la velocidad de transmisión y facilita aplicaciones como la transferencia de archivos.

**Token Talk:** AppleTalk sobre Token Ring. Dependiendo del hardware, TokenTalk opera a 4 o a 16 Mbps. Es decir, es la versión de Appletalk para redes Tokenring.

### 2.1.5 Arquitectura Cliente/Servidor

La arquitectura cliente/servidor es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor, al proceso que responde a las solicitudes.

Es el modelo de interacción más común entre aplicaciones en una red. No forma parte de los conceptos de la Internet como los protocolos IP, TCP o UDP, sin embargo todos los servicios estándares de alto nivel propuestos en Internet funcionan según este modelo.

Los principales componentes del esquema cliente/servidor son entonces los Clientes, los Servidores y la infraestructura de comunicaciones. En este modelo, las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario.

Los Clientes interactúan con el usuario, usualmente en forma gráfica. Frecuentemente se comunican con procesos auxiliares que se encargan de establecer conexión con el servidor, enviar el pedido, recibir la respuesta, manejar las fallas y realizar actividades de sincronización y de seguridad.

Los clientes realizan generalmente funciones como:

- Manejo de las interfaces del usuario.
- Captura y validación de los datos de entrada.
- Generación de consultas e informes sobre las bases de datos.

Los Servidores proporcionan un servicio al cliente y devuelven los resultados. En algunos casos existen procesos auxiliares que se encargan de recibir las solicitudes del cliente, verificar la protección, activar un proceso servidor para satisfacer el pedido, recibir su respuesta y enviarla al cliente. Además, deben manejar los interbloques, la recuperación ante fallas, y otros aspectos afines. Por las razones anteriores, la plataforma computacional asociada con los servidores es más poderosa que la de los clientes. Por esta razón se utilizan PCs poderosas, estaciones de trabajo, minicomputadores o sistemas grandes. Además deben manejar servicios como administración de la red, mensajes, control y administración de la entrada al sistema ("login"), auditoría y recuperación y contabilidad. Usualmente en los servidores existe algún tipo de servicio de bases de

---

datos. En ciertas circunstancias, este término designará a una máquina. Este será el caso si dicha máquina está dedicada a un servicio particular, por ejemplo: servidores de impresión, servidor de archivos, servidor de correo electrónico, etc.

Por su parte los servidores realizan, entre otras, las siguientes funciones:

- Gestión de periféricos compartidos.
- Control de accesos concurrentes a bases de datos compartidas.
- Enlaces de comunicaciones con otras redes de área local o extensa.

Siempre que un cliente requiere un servicio lo solicita al servidor correspondiente y éste, le responde proporcionándolo. Normalmente, pero no necesariamente, el cliente y el servidor están ubicados en distintos procesadores. Los clientes se suelen situar en ordenadores personales y/o estaciones de trabajo y los servidores en procesadores departamentales o de grupo.

Para que los clientes y los servidores puedan comunicarse se requiere una infraestructura de comunicaciones, la cual proporciona los mecanismos básicos de direccionamiento y transporte. La mayoría de los sistemas Cliente/Servidor actuales, se basan en redes locales y por lo tanto utilizan protocolos no orientados a conexión, lo cual implica que las aplicaciones deben hacer las verificaciones. La red debe tener características adecuadas de desempeño, confiabilidad, transparencia y administración.

Entre las principales características de la arquitectura cliente / servidor, se pueden destacar las siguientes:

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.

Los cambios en el servidor implican pocos o ningún cambio en el cliente.

Como ejemplos de clientes pueden citarse interfaces de usuario para enviar comandos a un servidor, APIs para el desarrollo de aplicaciones distribuidas, herramientas en el cliente para hacer acceso a servidores remotos (por ejemplo, servidores de SQL) o aplicaciones que solicitan acceso a servidores para algunos servicios.

Como ejemplos de servidores pueden citarse servidores de ventanas como X-windows, servidores de archivos como NFS, servidores para el manejo de bases de datos (como los servidores de SQL), servidores de diseño y manufactura asistidos por computador, etc.

### **2.1.5.1. Componentes Esenciales de la Infraestructura Cliente/Servidor**

Una infraestructura Cliente/Servidor consta de tres componentes esenciales, todos ellos de igual importancia y estrechamente ligados:

**Plataforma Operativa.** La plataforma deberá soportar todos los modelos de distribución Cliente/Servidor, todos los servicios de comunicación, y deberá utilizar, preferentemente, componentes estándar de la industria para los servicios de distribución. Los desarrollos propios deben coexistir con las aplicaciones estándar y su integración deberá ser imperceptible para el usuario. Igualmente, podrán acomodarse programas escritos utilizando diferentes tecnologías y herramientas.

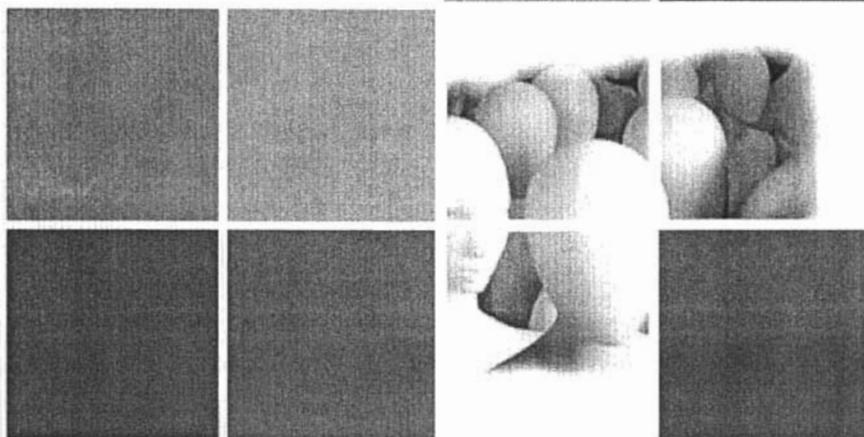
**Entorno de Desarrollo de Aplicaciones.** Debe elegirse después de la plataforma operativa. Aunque es conveniente evitar la proliferación de herramientas de desarrollo, se garantizará que el enlace entre éstas y el middleware no sea excesivamente rígido. Será posible utilizar diferentes herramientas para desarrollar partes de una aplicación. Un entorno de aplicación incremental, debe posibilitar la coexistencia de procesos cliente y servidor desarrollados con distintos lenguajes de programación y/o herramientas, así como utilizar distintas tecnologías (por ejemplo, lenguaje procedural, lenguaje orientado a objetos, multimedia), y que han sido puestas en explotación en distintos momentos del tiempo.

**Gestión de Sistemas.** Estas funciones aumentan considerablemente el costo de una solución, pero no se pueden evitar. Siempre deben adaptarse a las necesidades de la organización, y decidir la plataforma operativa y el entorno de desarrollo.

Análisis  
del  
Problema

Capítulo

3





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.

# CAPÍTULO 3

## ANÁLISIS DEL PROBLEMA

*En este capítulo se dará la definición de Sistema Operativo (S.O.), sus principales características, enfocándose particularmente en dos sistemas operativos: Windows y Mac OS; se asentarán las bases del análisis del problema esencial del Área de Becarios de la DGDC.*

## 3.1 Análisis del Problema

Una parte del enfoque de esta tesis será solucionar el problema multiplataforma, debido a que en la DGDC, hay diferentes sistemas operativos como son Microsoft Windows 98/XP y Apple Macintosh OS X.

Comencemos por definir lo que es un sistema Operativo:

Un sistema operativo (SO) es un conjunto de programas o software para permitir que el usuario se comunique con la computadora y gestionar sus recursos de manera cómoda y eficiente. Comienza a trabajar cuando se enciende la computadora, y gestiona el hardware de la máquina desde los niveles más básicos.

Otra definición posible y bastante aceptada indica que un sistema operativo es como una capa compleja entre el hardware y el usuario, concebible también como una máquina virtual, que facilita al usuario o al programador las herramientas e interfaces adecuadas para realizar sus tareas informáticas, abstrayéndole de los complicados procesos necesarios para llevarlas a cabo.

### 3.1.1 Las Características de los Sistemas Operativos

- Administración de tareas:

Monotarea: Si solamente puede ejecutar un programa (aparte de los procesos del propio S.O.) en un momento dado. Una vez que empieza a funcionar un programa, continuará haciéndolo hasta su finalización o interrupción.

Multitarea: Si es capaz de ejecutar varios programas al mismo tiempo. Este tipo de S.O. normalmente asigna los recursos disponibles (CPU, memoria, periféricos) de forma alternativa a los programas que los solicitan, de manera que el usuario percibe que todos funcionan a la vez.

- Administración de usuarios:

Monousuario: Si sólo permite ejecutar los programas de un usuario al mismo tiempo.

**Multiusuario:** Si permite que varios usuarios ejecuten simultáneamente sus programas, accediendo a la vez a los recursos de la computadora. Normalmente estos Sistemas operativos utilizan métodos de protección de datos, de manera que un programa no pueda usar o cambiar los datos de otro usuario.

- Manejo de recursos:

**Centralizado:** Si permite utilizar los recursos de una sola computadora.

**Distribuido:** Si permite utilizar los recursos (memoria, CPU, disco, periféricos, etc.) de más de una computadora al mismo tiempo.

- Organización interna:

**Kernel monolítico:** La estructura de esta arquitectura es simplemente no tener ninguna. A nivel de núcleo no se produce ninguna abstracción, es decir, si un procedimiento necesita a otro es libre de hacerlo en cualquier momento. Fue el primer enfoque en la historia, el resto son evoluciones. Como ejemplos de este tenemos: (Linux, BSD, skyOS, Windows, Syllable).

**Microkernel o micronúcleo:** En este caso, el S.O. se ocupa solo de unas pocas funciones, reduciendo el núcleo a su mínima expresión. El resto de funciones pasan a estar en el espacio de usuario. Como por ejemplo: (BeOS, Mach, MacOSX, newOS).

**Nanokernel:** Se basa en una cola de reparto de interrupciones, permitiendo así ejecutar dos kernels a la vez (como linux y windows). Se desarrollo en principio para cargar un kernel de tiempo real antes de linux, y tener así funciones de tiempo real como las de RTlinux como ejemplos tenemos: (AdeOS, Eros, KeyKOS, Brix-OS).

**VoID:** Es una forma de diseñar un sistema operativo en ausencia de kernel. El antikernel VOiD fue diseñado por un joven hacker llamado Dave Poirier, y consiste en un grupo de células o *cells* cargadas dinámicamente, su comportamiento es totalmente descentralizado, es una aplicación de la biología al diseño de sistemas operativos, algunos ejemplos de sistemas operativos de esta naturaleza son: (unununium, TUNES, Vapour).

SASOS: (Single Address Space Operating System) sistema adaptado a las necesidades de usos complejos, tales como CAD/CAM, donde un número de programas en cooperación manipulan y comparten una base de datos de gran tamaño. Ejemplo (Opal, Mungi, BriX).

Exokernel: es un sistema creado con fines de investigación en el Instituto Tecnológico de Massachussets (MIT) sobre OpenBSD. Su propósito es crear como una capa de software para otros sistemas virtuales.

Consta de 3 capas, la capa del Exokernel, la capa LibOS (núcleos de varios Sistemas Operativos corriendo sobre Exokernel) y la capa de aplicaciones, corriendo sobre LibOS.

El sistema es realmente robusto y flexible, debido a que la única tarea del Exokernel, es impedir que tengan errores de permisos al hardware y conflictos entre los LibOS. Ejemplo de este diseño se encuentra en ExOS y Extended OpenBSD.

El Cache Kernel: El cache kernel es un sistema que propone que el kernel del sistema sea básicamente una cache de los recursos empleados por las aplicaciones. De este modo, son las aplicaciones las que poseen los recursos físicos (aunque de modo virtualizado) y dentro del núcleo se mantienen sólo aquellos recursos más empleados. En la práctica, el cache kernel es un sistema similar al exokernel.

Los objetos de los que hace caching el Cache Kernel son espacios de direcciones y threads. El mecanismo de intercomunicación de procesos empleado se basa en la implementación de los espacios de direcciones. Básicamente se extiende el concepto de *traducción de direcciones* para que incluya opcionalmente un thread que deberá manejar mensajes relacionados con dicha dirección. Los clientes pueden *señalar* dicha dirección y el thread que lo sirve será notificado. La transferencia de datos entre clientes y servidores se implementa con ayuda del gestor de memoria virtual (universidad de stanford).

### **3.1.2 Microsoft Windows**

**Microsoft Windows** es el nombre de una familia de sistemas operativos desarrollados por la empresa de software Microsoft Corporation. Todos ellos tienen en común el estar basados en una interfaz gráfica de usuario basada en el paradigma de ventanas (de ahí su nombre en inglés). Las versiones de Windows que existen hasta el momento se basan en dos líneas separadas de desarrollo que finalmente convergen en una sola con la llegada de Windows XP.

#### **3.1.2.1 Versiones basadas en MS-DOS**

La primera de esas líneas mencionadas anteriormente conformaban la apariencia de un Sistema Operativo, aunque realmente requerían otro sobre el cual ejecutarse (MS DOS). Todos los Sistemas Operativos, desde Windows 1.0 a Windows ME necesitaban tener MS DOS instalado, aunque desde la aparición de Windows 95, podía instalarse Windows sobre un disco duro vacío, ya que durante su propia instalación, se instalaba además una versión reducida de MS DOS. La arquitectura de Windows comenzó siendo de 16 bits, hasta Windows 95, donde pasó a funcionar bajo una arquitectura de 32 bits, aunque manteniendo bastantes módulos de 16 bits por razones de compatibilidad.

Los sistemas operativos basados en MS-DOS son:

- Windows 1.0
- Windows 2.0
- Windows 3.0
- Windows 3.1
- Windows 3.11 (Windows Para trabajo en grupo)
- Windows 95
- Windows 95 OSR2
- Windows 98
- Windows 98 SE (Second Edition)
- Windows ME (Millennium Edition)

#### **3.1.2.2 Versiones basadas en NT (New Technology)**

La segunda línea de desarrollo (NT) se basaba en emplear desde el origen un sistema operativo en modo gráfico y con una arquitectura de 32 bits.

Este Sistema Operativo no requiere tener instalado ningún otro previamente. Incluye en todas sus versiones, un emulador de consola en modo texto. A modo de anécdota, Windows NT 4.0, en origen no era compatible con tarjetas gráficas AGP, requería la instalación de un Service Pack o conjunto de parches (de la versión 3 en adelante), que permitían su detección y la instalación de sus controladores.

Los sistemas operativos basados en NT son:

- NT 3.1
- NT 3.5
- NT 3.51
- NT 4.0
- Windows 2000
- Windows XP
- Windows Server 2003

### 3.1.2.3 Sistemas de Archivos de Windows

El sistema de archivos utilizado por estos sistemas operativos, comenzó siendo FAT16 o FAT solamente. La primera versión de Windows en incorporar soporte nativo para FAT32 fue Windows 95 OSR2. Por otro lado, los Sistemas Operativos basados en NT emplean los sistemas de archivos FAT32 y NTFS desde el origen.

### 3.1.2.4 Windows XP

**Windows XP** (cuyo nombre en clave inicial fue *Whistler*) fue hecho público el 25 de octubre de 2001 por Microsoft. Microsoft inicialmente sacó a la venta dos versiones: **Home** y **Professional**. La versión *Home* está destinada al mercado doméstico, mientras que la versión *Professional* dispone de características adicionales diseñadas para entornos empresariales, como la autenticación por red y el soporte multiprocesador.

Las letras "XP" provienen de la palabra *experience*.

Antes de XP, Microsoft producía dos líneas (como ya se vio en el capítulo 3.1.2) separadas de sistemas operativos. Una línea estaba dirigida a los ordenadores domésticos representada por Windows 95, Windows 98 y Windows Me, mientras que la otra, representada por Windows NT y Windows 2000, estaba pensada para el mercado corporativo y empresarial e incluía versiones especiales para servidores. Windows XP es el intento por parte de Microsoft de ofrecer un único sistema operativo multiuso, con el inconveniente de eliminar definitivamente el soporte para los programas basados en MS-DOS del sistema operativo.

Windows XP está basado en el código de Windows 2000 con un nuevo interfaz gráfico (llamado Luna), el cual incluye características ligeramente rediseñadas, algunas de las cuales parecen inspiradas por los modernos entornos de escritorio presentes en Linux, como KDE. La pantalla de login gráfica con imágenes para cada usuario es un buen ejemplo. Incluye también un reducido conjunto de las características de seguridad de Windows 2000 y un cortafuegos integrado (desactivado por defecto). Todo ello es parte de un esfuerzo mayor en hacer seguros los productos de Microsoft (sin demasiado éxito), conocidos por una muy larga lista de fallos de seguridad.

En noviembre de 2002, Microsoft sacó a la venta dos nuevas versiones de Windows XP para hardware específico:

Windows XP **Media Center Edition** para PCs especiales.

Actualmente, dichos PCs son los "HP Media Center Computer" y la serie "Alienware Navigator". "Windows XP Media Center Edition" debe ser vendido con uno de estos ordenadores y no puede encontrarse en tiendas.

Windows XP **Tablet PC Edition** para ordenadores portátiles especiales diseñados con una pantalla táctil que admiten escritura a mano y pantallas tamaño portarretratos. Adicionalmente, Microsoft hizo pública otra versión: Windows XP **64 Bit Edition** para fabricantes cuyo destino son los procesadores Intel Itanium.

### 3.1.3 Apple Macintosh

**Apple Macintosh** (abreviado *Mac*) es el nombre de una serie de computadoras fabricadas y comercializadas por Apple Computer desde 1984. Apple ha autorizado a otras compañías, como Umax o PowerComputing para la fabricación de clones Macintosh en los años 1990.

Los primeros microprocesadores Macintosh estaban basados en los de la familia 68000 de Motorola. Las máquinas más recientes emplean el PowerPC de IBM y Motorola. Las Apple Macintosh son comercializadas con el sistema operativo Mac OS pero también es posible instalar en ellos Linux o FreeBSD.

Mac OS X combina nuevas tecnologías basadas en estándares abiertos que lo convierten no sólo en el sistema operativo personal más estable, compatible e interactivo del mundo, sino también en el más apasionante e innovador.

La arquitectura modular de Mac OS consta de seis componentes principales: las aplicaciones del sistema, la interfaz de usuario Aqua, un variado conjunto de marcos de aplicaciones, sistemas gráficos estándar, servicios de gestión del sistema de amplio alcance y los cimientos Darwin.

### **3.1.3.1 Aplicaciones del Sistema**

Los programas del Mac son diseñados para trabajar en equipo, simplificar la experiencia informática y realizar tareas cada vez más complejas.

Mac OS X incluye todas las aplicaciones que se necesitan para aprovechar la computadora en un instante. Usa aplicaciones de estilo de vida digital: con iMovie, iPhoto e iTunes para gestionar cámaras, iPod y descargar canciones. iChat AV, Safari, Mail, QuickTime y Sherlock ofrecen la mejor Panther incluye aplicaciones para gestionar tipos de letra, ver PDFs y DVDs, editar texto, llevar a cabo cálculos, gestionar impresoras, conectarte a Internet, cambiar las preferencias del sistema y muchas más utilidades para cambiar la apariencia de la Mac.

### **3.1.3.2 Interfaz de Usuario**

La interfaz de usuario Aqua hace que el trabajo y el ocio en la Mac sean todavía más intuitivos para los usuarios primerizos, además de proporcionar potentes herramientas personalizables para los profesionales. Toda la interfaz, incluidos iconos, menús, ventanas y controles, constituye una innovadora continuación de la legendaria facilidad de uso de la Mac.

Colores, transparencias y animaciones mejoran la usabilidad y consistencia del sistema. Apple es líder del sector, como demuestra su aportación de avances como Exposé, que facilita la navegación por montones de ventanas abiertas. Además, Mac OS X cumple los criterios de accesibilidad

establecidos y ofrece reconocimiento de escritura manual, cambio rápido de usuario y prestaciones sin igual en entornos internacionales.

### 3.1.3.3 Marcos para Aplicaciones

Además de las tradicionales APIs de UNIX, Mac OS X incluye una amplia gama de completos marcos para aplicaciones. Esto ayuda a los desarrolladores de diferentes comunidades, puesto que pueden crear interfaces de usuario basadas en Aqua para aplicaciones Cocoa, Carbon, AppleScript y Java gracias al entorno de desarrollo de serie Xcode. Cocoa es un conjunto de marcos de desarrollo orientados a objetos diseñado para desarrollar aplicaciones rápidamente, por lo que resulta muy fácil dotar a los programas UNIX existentes de exuberantes interfaces Aqua o bien crear aplicaciones totalmente nuevas.

Carbon ofrece un armonioso método de migración para desarrolladores que hayan adaptado sus aplicaciones de Mac OS 9 a Mac OS X. AppleScript ayuda a crear aplicaciones propias para automatizar comandos en aplicaciones, extracción de información y tareas repetitivas. Java 2 Standard Edition v1.4.1 para Mac OS X es una versión totalmente compatible, enormemente optimizada y estrechamente integrada con el estilo y aspecto original de Java, por lo que es fácil ejecutar aplicaciones estándar de Java directamente.

### 3.1.3.4 Gráficos y Multimedia

El sistema gráfico de Mac OS X combina estándares 2D, 3D y temporales mediante un sistema de composición de ventanas líder del sector, con lo que se consigue una vivencia como usuario rica y sencilla. La librería de generación gráfica 2D de alto rendimiento Quartz es el principal modelo de imagen de Mac OS X. Basada en el estándar multiplataforma PDF (Portable Document Format), Quartz 2D muestra e imprime texto y gráficos suavizados de alta calidad y es absolutamente compatible con tipos OpenType, PostScript y TrueType. Para aquellos usuarios que dispongan de una tarjeta de vídeo compatible, Quartz Extreme mejora notablemente la velocidad y capacidad de respuesta de Mac OS X. OpenGL es el estándar para visualizar formas y texturas en 3D, y permite generar gráficos en 3D de primera calidad en Mac OS X, lo que lo convierte en la plataforma ideal para las animaciones y los efectos especiales en 3D. QuickTime ofrece un entorno totalmente basado en estándares para la creación, los juegos y la distribución de vídeo, audio e imágenes. El núcleo Core Audio de Mac OS X integra las funciones de audio directamente en el sistema operativo de

formas que antes eran impensables, posibilitando un rendimiento y facilidad de uso en tu estudio virtual sin precedentes.

### 3.1.3.5 Gestión del Sistema

Creado con muchas de las tecnologías que impulsan a la propia Internet, Mac OS X incluye conexiones de red modernas y es compatible con protocolos estándar. Esto hace que Mac OS X sea fácil de integrar en cualquier red estándar, incluidas las redes Macintosh, Windows y, por supuesto, Internet. Seguridad es un concepto clave en Mac OS X, como prueba la cantidad de detalles que permiten a los usuarios particulares e institucionales descansar algo más tranquilos. Dado que Apple diseña tanto el sistema operativo como los ordenadores, Mac OS X incluye tecnologías exclusivas que ponen a disposición de sus usuarios la mejor plataforma portátil del mundo.

### 3.1.3.6 Cimientos del Sistema

Bajo la interfaz fácil de usar y los gráficos detallados de Mac OS X se halla Darwin, los cimientos UNIX basados en tecnologías de código abierto como FreeBSD, Mach, Apache y GCC. Darwin proporciona un entorno completamente UNIX, con servidor X11 y servicios POSIX comparables a los de Linux o FreeBSD, como los viejos conocidos para kernel, librerías, redes y utilidades de línea de comandos.

## 3.1.4 Análisis del Sistema de Control de Becarios Información Requerida del Sistema

### 3.1.4.1 Relacionado a los Becarios

**Datos Personales del Becario:** Se trata de información básica del Becario como el nombre, apellidos, id, domicilio, la sala o área de la dependencia a la que se encuentra asignado; además del horario que tiene. Dicha información se debe:

- a) Consultar
- b) Modificar o actualizar; y
- c) Permanecer en archivos históricos

Los datos que se requieren almacenar en la base son los siguientes:

- Nombre completo

- Fecha de nacimiento
- Edad
- Estado civil
- Domicilio (calle y número, colonia, delegación, C.P.)
- Teléfono
- Correo electrónico
- Universidad de procedencia
- Facultad o escuela
- Carrera
- Número de cuenta (matricula)
- Créditos
- Promedio
- Otros estudios
- Idioma (porcentaje que habla, lee y escribe)
- Otras actividades que realice
- Fecha de ingreso

**Asistencia de los Becarios:** Por el papel que juegan los becarios dentro de la dependencia es importante saber qué becarios están presentes, cuáles tienen retardo o están ausentes.

Existe dos políticas al respecto: “sólo se permiten tres faltas injustificadas”, el tener tres faltas implica baja automática del Becario, y “tres retardos cuentan como una falta”, se considera como retardo el exceder la tolerancia de 15 minutos después de la hora de llegada.

Es importante resaltar que no todos los becarios se encuentran en las salas del museo, algunos ayudan en otras labores de la dependencia, por ejemplo, diseñando páginas Web.

Para los becarios que se encuentran en las salas (anfitriones), además de su estado en cuanto a la asistencia se refiere, es necesario ubicarlos en las mismas, es decir, saber qué becarios están asignados en una sala y quiénes se encuentran presentes.

Si los anfitriones cumplen con el tiempo semanal tienen derecho a solicitar mensualmente un día libre, además si por motivos escolares tienen que ausentarse pueden solicitar un permiso. El sistema debe ser capaz de aceptar o reconocer este cuarto estado de asistencia.

### 3.1.4.2 Relacionado a las Salas o Áreas del Museo

**Información de la Sala o Área del Museo:** Se refiere a datos como el nombre de la misma y asignar o crear de forma automática un identificador único. Las operaciones requeridas sobre la información son:

- a) Consulta
- b) Modificar o Actualizar

En este caso para efectuar las búsquedas sólo se realiza empleando los dos parámetros con los que se cuenta (nombre e id) o incluso seleccionado de una lista todas las que existen en la base de datos.

**Becarios Asignados a una Sala o Área:** Esta información se encuentra estrechamente ligada a los Becarios como ya fue mencionado, pero es necesario abundar un poco más.

Podemos clasificar esta información en dos tipos:

- I. Becarios asignados a un Área: La menor parte de los Becarios es la que se encuentra en este caso. Se necesita conocer de una forma más organizada y clara la asignación de los becarios dentro de la DGDC.
- II. Anfitriones: De forma similar al anterior caso se necesita tener la información de asignación a las salas de forma más clara.

De estos dos casos el segundo es el más complejo, puesto que tenemos dos variantes que llamaremos estática y dinámica. En cuanto a la primera se refiere, tiene un manejo más simple, puesto que las operaciones requeridas para este tipo de información son las siguientes:

- a) Consultar la información, poder ver la asignación de anfitriones en cada una de las salas en el transcurso de una semana.
- b) Modificar la información, poder cambiar la asignación de anfitriones en las salas cuando esto sea necesario.

La dinámica se refiere a que este tipo de información se crea en tiempo real, y cada día cambia puesto que depende de dos factores más: el humano (asistencia de los becarios) y la configuración de los horarios (realizada por jefes de área). A continuación se analizará lo concerniente a los horarios para después analizar esta información dinámica contando con todos sus elementos.

---

### 3.1.4.3 Relacionada a los Horarios

**Información sobre los horarios:** Hay un mínimo de horas a la semana que los Becarios están comprometidos a cumplir y para ello cada Becario puede configurar un horario que cumpla con este número de horas semanales. Para el caso de los anfitriones existe el siguiente criterio para establecer los horarios:

- a) Durante el horario del museo, en cada sala debe haber, mínimo un anfitrión.
- b) El número de becarios por sala será mayor en las mañanas que en las tardes.
- c) De lunes a viernes los anfitriones pueden asistir mínimo tres horas y máximo cuatro.
- d) Los sábados y domingos pueden asistir mínimo tres horas y máximo seis.
- e) El inicio de sus actividades puede ser a las 9:00, 10:00, 14:00 a 15:00 horas.

Los horarios se crean cada semestre, pero regularmente se modifican cada mes, todo depende de los Becarios, es decir, cambios en los horarios de los becarios, que termine su periodo dentro de la dependencia, etc. Incluso se puede dar cambios temporales, por ejemplo, si un anfitrión asiste a una práctica de campo alguien tiene que cubrir al becario en la sala y horario correspondiente.

Las operaciones que interesan sobre esta información son:

- a) Consulta
- b) Modificar o Actualizar

### 3.1.4.4 Información en Tiempo Real sobre los Anfitriones

Como lo mencionamos, es necesario considerar la asistencia, salas y horarios, esta información se encuentra muy relacionada entre sí por lo siguiente: los encargados del área necesitan saber qué becarios se encuentran presentes en una o varias salas del museo para atender a los visitantes y visitas guiadas en diversos horarios. Se trata de generar un informe que muestre la ubicación de aproximadamente 150 becarios en las

dieciséis salas del museo complementada con la asistencia. Por ejemplo, ver tabla 3.1:

Día: Lunes			
HORA	SENDA	MATEMÁTICAS	QUÍMICA
09:00 A 10:00	Felipe	Jimena	Pilar
		Jorge	Sofía

Tabla 3.1. Se muestran los anfitriones que pasaron asistencia y se encuentran a tiempo en tres salas del museo.

Se observa que de estos tres factores necesarios para este informe, el que no varía es el de las salas, es decir, se mantiene estático, pero es indispensable. Los horarios, sí pueden cambiar pero es menos frecuente que esto ocurra. Por otra parte la asistencia es la más variable, ya que depende del factor humano, el cual no se puede controlar.

Ahora bien, si observamos más detenidamente, los horarios son la base de este informe, ya que la asistencia implica estar en la sala que le corresponde al becario dentro del horario que le corresponde, es decir, la asistencia depende de los horarios. Y con fines de logística esta información se empaata con la asignación de anfitriones en las salas.

### 3.1.4.5 Adquisición de la Información

Este proceso implica algunas operaciones que no se han mencionado de forma explícita hasta el momento:

- Dar de alta a un becario
- Registrar la asistencia de un becario
- Dar de alta una sala
- Dar de alta un horario

La forma más simple de efectuar un alta en el sistema es teclear los datos para así obtener información de ellos. Esta acción sólo se efectuará una vez

para cada registro, esto tomará tiempo al principio del ciclo vida del sistema, después no implicará mayor problema. Entonces este proceso puede ser efectuado por los encargados y / o asistentes del área.

Para registrar la asistencia en el sistema de todos los becarios lo más eficiente es que ellos mismos lo hagan, es decir, cada becario registrará su asistencia al llegar al museo. Para esto se debe buscar la forma más fácil y simple; pero eficiente a la vez.

Actualmente el sistema opera de la forma descrita con detalle en el Capítulo 1 Introducción; La automatización del sistema es necesaria ya que al dejar el proceso en manos del personal tenemos que estar sujetos al factor humano, es decir, estamos sujetos a que la persona que registra a los becarios en ese momento no se encuentre, o que no llegue en todo el día, etc.; al automatizar este sistema tendremos un control adecuado de cada uno de los Becarios de la DGDC ya que el sistema será imparcial, es decir, no tendrá distinción entre los becarios (todos serán tratados de la misma forma), de esta manera no se tomaran decisiones erróneas ya que el sistema no estará sujeto a factores humanos.

### 3.1.4.6 El Entorno del Sistema

Es importante conocer y analizar las características del entorno de un sistema si queremos que este se adapte mejor a las condiciones que imperan. Si pasamos por alto alguna característica en *especial* puede afectar de forma negativa durante el ciclo de vida del software, lo cual puede implicar: pérdida de tiempo durante alguna etapa de creación del sistema y llegar al extremo de fracasar, es decir, que el sistema provoque problemas en lugar de generar soluciones. Por lo tanto analizaremos que puede afectar en el entorno al éxito del proyecto.

Retomando las áreas de la dependencia tenemos que son dos áreas las que se verán beneficiadas por el sistema: Becarios y Atención al Visitante. Sólo parte de todo el personal que trabaja en estas áreas tiene acceso completo a toda la información sobre los becarios.

Pero recordemos que existe otra área que aunque empleará el sistema si puede tener influencia en él, se trata de Innovación Tecnológica. Dicha área es la encargada de todo lo referente a Tecnologías de la Información, por lo tanto se debe tener comunicación con ésta área.

La comunicación con los usuarios finales es vital, pero no siempre se puede obtener todos los datos que son necesarios. En este caso, las áreas interesadas no conocen como se encuentra la infraestructura de la red, el tipo de servidores que se tienen, etc. Por lo tanto, es necesario establecer comunicación con Innovación Tecnológica, para obtener este tipo de datos y poder tener acceso los recursos que sean necesarios.

Además hay ciertas políticas en cuanto a software se refiere. En particular se debe tomar en cuenta que para este proyecto no se debe depender de software con licencia. Es decir, como parte de la solución al problema que se presenta no se debe incluir software por el cual se tenga que pagar una o más licencias.

Después de tener un acercamiento con Innovación Tecnológica se obtuvo más información sobre la red interna. Sólo se presentan los datos que sean utilidad pero que no vulneren la seguridad informática de la dependencia.

#### 3.1.4.6.1 Red Interna

Se trata de una red LAN, la conexión a Internet llega por parte de DGSCA, como se muestra en la figura 3.1:

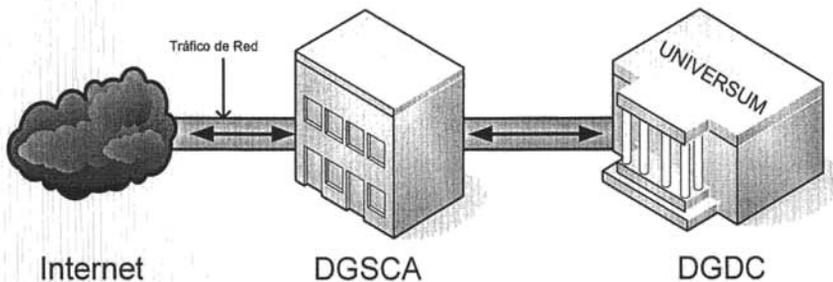


Figura 3.1. Esquema de conexión a Internet

Las características generales de la red interna son:

- a) Su topología es del tipo estrella.
- b) Cuenta con cableado UTP categoría 5 y fibra óptica en el back bone.

- c) Se maneja conmutadores con las velocidades de 10/100/1000 Mbps.
- d) Se cuenta con una red inalámbrica.
- e) Hay un servidor de impresión con Windows NT
- f) Existe un servidor de correo en la plataforma Solaris 8.
- g) Para hospedar las páginas de la DGDC cuenta con servidores Intel (sistema operativo Red Hat 9) y servidores Sun (con sistema operativo Solaris 8).
- h) Un software enrutador sobre el sistema Mac OSX.
- i) Se tiene el siguiente rango de IP: 132.248.66.XX

El esquema general se muestra en la siguiente figura:

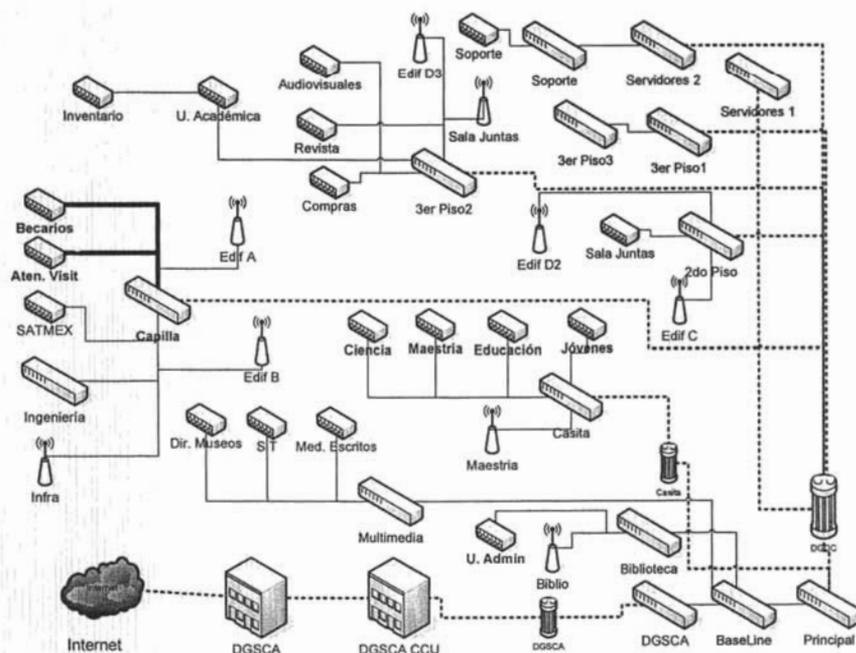


Figura 3.2. Esquema general de la red interna de la DGDC

En la figura 3.2 se muestran con líneas continuas el cableado UTP y con líneas punteadas el cableado con fibra óptica. Las líneas con un grosor mayor son el segmento de la red que da servicio a las dos áreas interesadas.

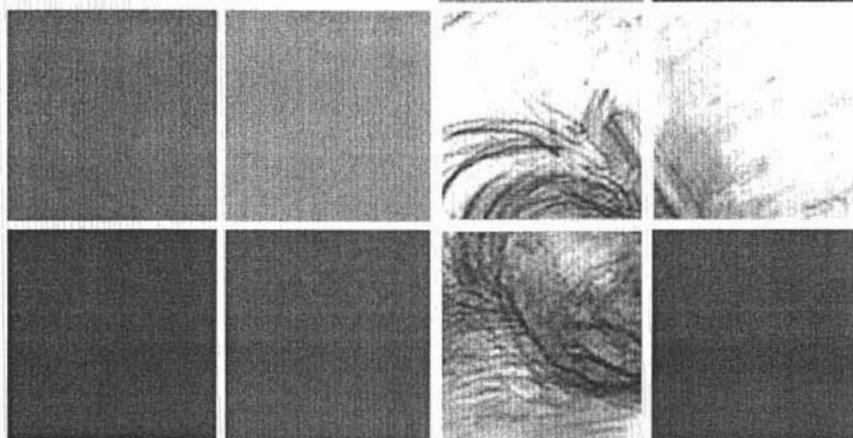
El porcentaje de ancho de banda que se ocupa es máximo del 13 %, según datos proporcionados por DGSCA, en el año 2003. Como lo mencionamos se pueden manejar velocidades de conexión 10/100 Mps.

Se puede aprovechar estas características, ya que se cuenta con tecnología actual en el backbone de la red, el uso del ancho de banda no es una limitante. Además se cuentan con servidores robustos capaces de soportar grandes cargas de trabajo y se puede solicitar permiso para emplear aquel que Innovación Tecnológica designe en caso de ser necesario.

# Creación de un Modelo

Capítulo

4





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.

# CAPÍTULO 4

## CREACIÓN DE UN

### MODELO (INGENIERÍA)

*En este capítulo realizaremos el análisis a fondo de lo que será nuestro modelo de ingeniería analizando como haremos el control de los Becarios y que tipo de código será el que ocuparemos para pasar lista de forma eficiente y rápida, dejando ver cual es el proceso que se tiene actualmente y cual es el proceso que vamos a realizar; veremos el análisis de la Base de Datos para el Sistema de Control de Becarios para la DGDC.*

## 4.1 Creación de un Modelo (Ingeniería)

Después de conocer que tipo de información se requiere obtener del sistema, las operaciones sobre ella y analizarla, podemos comenzar a elaborar un modelo.

Para crear un modelo debemos recordar los puntos cruciales:

- Se manejan dos sistemas operativos (Windows y Macintosh)
- Se requiere que la información se encuentre en tiempo real
- Se cuenta con infraestructura de red
- Se cuenta con Servidores
- La forma de pasar lista a los Becarios debe ser sencilla
- Se desea almacenar información

Como ya sabemos actualmente se cuenta con “un sistema” para registrar a un nuevo Becario. Para esto se necesitan fotos y crear un reporte con los datos del mismo. El proceso que siguen actualmente es el siguiente:

1. Primero se toman los datos del Becario en un formato el cual se llena a mano.
2. Se le pide al Becario dos fotografías; una para su credencial y otra para el documento antes mencionado.
3. Para la credencial se tiene un formato, el cual se llena a máquina en los espacios en blanco y se agrega la fotografía del Becario.

Lo anterior se muestra en la figura 4.1

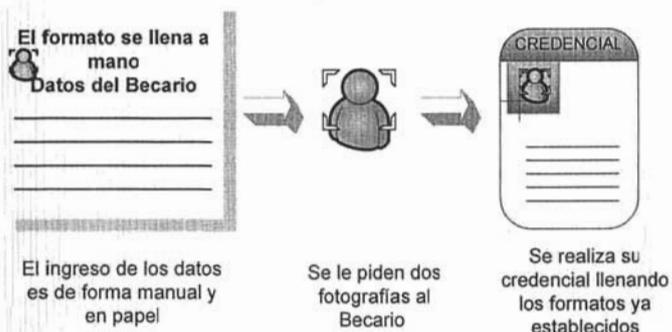


Figura 4.1 Proceso Actual de ingreso del Becario

Considerando el esquema actual, observamos que sería más lento si le agregamos un paso más, es decir, dar de alta a los Becarios en el nuevo Sistema, como lo podemos ver en la figura 4.2:



Figura 4.2 Proceso Actual de ingreso del Becario y captura de datos en el nuevo formato digital

Esto lejos de agilizar un proceso incrementa el tiempo que se lleva en realizarlo. Para solucionar este problema, nos ayudaremos de técnica conocida como reingeniería de procesos.

### 4.1.1 Reingeniería de Procesos

La Reingeniería de Procesos es una herramienta gerencial moderna, orientada al mejoramiento de los procesos. Su adecuada aplicación seguida de innovación y mejoramientos continuos nos permitirá mantenernos competitivos.

Propiamente hablando: "reingeniería es la revisión fundamental y el rediseño radical de procesos para alcanzar mejoras espectaculares en medidas críticas y actuales de rendimiento, tales como costos, calidad, servicio y rapidez"<sup>1</sup>.

Un proceso rediseñado es muy distinto a un proceso tradicional. Los procesos rediseñados toman diferentes formas, pero en general podemos dar algunas características típicas de los mismos:

<sup>1</sup> Definición tomada del libro REENGINEERING, THE CORPORATION, escrito por: HAMMER & CHAMPY, ED. HARPER BUSSINESS.

---

Varios oficios se combinan en uno: desaparece el trabajo en serie, y muchos oficios o tareas que antes eran distintos, ahora se integran en uno solo. Es una compresión horizontal del trabajo. Por ejemplo, una persona, llamada trabajador del caso, ejecuta todo el proceso. Al eliminar fases de tareas se eliminan errores y ahorra tiempo. También hay 'equipos de caso': un grupo ejecuta todas las tareas.

Los trabajadores toman decisiones: Es una compresión vertical, porque ahora los trabajadores ya no consultan al superior jerárquico y pueden tomar sus propias decisiones. Hay así menos demoras, costos más bajos, mejor atención hacia la persona que se atiende (en nuestro caso el Becario).

Los pasos del proceso se ejecutan en orden natural: No hay por qué respetar un orden lineal artificial, donde la tarea 2 se hacía al finalizar la tarea 1. Ahora ambas pueden hacerse simultáneamente, con lo cual se gana tiempo.

## **4.2 Asistencia de los Becarios**

Como se requiere pasar lista a los Becarios de forma ágil y eficaz, proponemos que con su credencial (en donde hay impreso un código de barras) el Becario pueda registrar su asistencia al pasar la credencial por el lector de código, en ese momento el sistema reconozca el Id del Becario y mande la información al servidor con la hora de llegada del Becario.

Vamos a ver un poco acerca de los códigos de Barras y cual será el más adecuado para realizar la identificación del Becario. Actualmente la mayoría de los lectores de código de barras leen las siguientes codificaciones: Codabar, EAN 39 o código 3 de 9,RS2-32.

### **4.2.1 Funcionamiento del Sistema de Códigos de Barras**

El lector de código de barras decodifica la información a través de la digitalización proveniente de una fuente de luz reflejada en el código y luego de ser procesados los impulsos se envía la información a una computadora como si la información hubiese sido ingresada por teclado.

El procedimiento (Ver figura 4.3): el símbolo de código de barras es iluminado por una fuente de luz visible o infrarroja, las barras oscuras absorben la luz y los espacios las reflejan nuevamente hacia un escáner. El escáner transforma las fluctuaciones de luz en impulsos eléctricos los cuales copian las barras y el modelo de espacio en el código de barras. Un decodificador usa algoritmos matemáticos para traducir los impulsos eléctricos en un código binario y transmite el mensaje decodificado a una Terminal manual, PC, o sistema centralizado de computación. El decodificador puede estar integrado al escáner o ser externo al mismo.

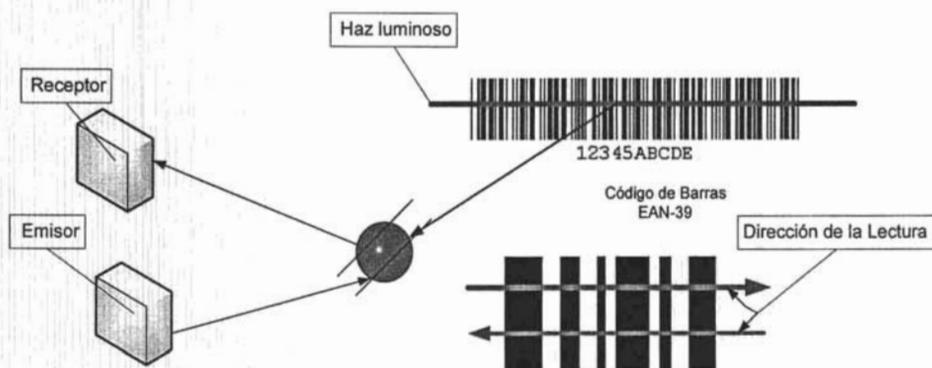


Figura 4.3 Descripción del funcionamiento de un scanner lector de código de barras.

Algunos de los beneficios de los códigos de barras son:

- Entrada de Datos con Rapidez
- Costos de Mano de Obra más bajo
- Un acceso rápido a la información.
- Reducción de errores en la recolección de datos.

### 4.2.2 Simbologías del Código de Barras

Los sistemas de códigos de barras pueden usar varias simbologías. Una simbología es equivalente a un lenguaje. Cada simbología tiene fortalezas y debilidades. Muchas de ellas están en uso por razones históricas o por motivos políticos, mientras que otras tienen ventajas técnicas definidas. En muchos sistemas, se tiene que satisfacer la simbología específica de la empresa.

**Estándares:** La simbología EAN tiene un comité de estándares que establece códigos para una variedad de aplicaciones. Esta es la razón por la que un fabricante de productos puede poner un código de barras en un empaque sin temor de duplicación.

**Densidad:** Cada simbología puede codificar un número diferente de caracteres sobre un espacio determinado. Aún si una simbología es capaz de hacer un código de barras del tamaño requerido, se puede conseguir una mejor relación de lectura usando diferente simbología. Algunos scáners, particularmente los de Lápiz, están construidos para una densidad fija. Es importante asegurarse que el dispositivo de lectura sea compatible con la densidad del código de barras que se va a usar.

**Legibilidad:** Algunos códigos de barras son más legibles que otros. Por ejemplo, algunas pruebas muestran que el Code 128 (el cual es uno de los que tienen simbología más flexible) es fácil y exitosamente leído por la mayoría de los lectores. En algunos casos se puede sacrificar legibilidad a cambio de otros requisitos.

#### 4.2.2.1 Código EAN-13

EAN-13 se utiliza por todo el mundo para las mercancías de venta al por menor de alguna marca. El símbolo codifica 13 caracteres: los primeros dos o tres son un código de país que identifican el país en el cual se coloca el fabricante (no necesariamente donde el producto se hace realmente). El código de país es seguido por 9 o 10 dígitos de los datos (dependiendo de la longitud del código de país) y una sola suma de comprobación del dígito. Las barras suplementales 2-digit y 5-digit se pueden agregar para un total de 14 o 17 dígitos de los datos como se muestra en la figura 4.4.



Figura 4.4 Código EAN-13

---

### 4.2.2.2 Código Codabar

También conocido como USD-4, NW-7, y 2 del código 7.

Codabar puede codificar los seis símbolos (-:.\$/+), de los dígitos 0 a 9, y los caracteres A, B, C, D, E, \*, N, o T. Codabar (ver figura 4.5) se utiliza en bibliotecas, los bancos de sangre, la industria de entrega de paquetes, y una variedad de otros usos de tratamiento de la información.



Figura 4.5 Ejemplo de un Código de Barras Codabar

### 4.2.2.3 Código 39 (3 de 9)

El código 39 se utiliza extensamente en muchas industrias y es el estándar para muchas especificaciones del código de barras del gobierno, incluyendo el departamento de defensa de ESTADOS UNIDOS. El código 39 se define en el American National Standards Institute (ANSI) MH10.8M-1983 estándar, y también se conoce como USD-3 y 3 de 9.

El juego de caracteres del código 39 incluye los dígitos 0-9, las letras A-Z (mayúscula solamente), y los símbolos siguientes: espacio, menos (-), más (+), punto (.), signo de dólar (\$), raya vertical (/), y el carácter partida/parada especial de los por ciento (%). El código puede ser de cualquier longitud, aunque más de 25 caracteres realmente comienzan a rebasar los límites.

Cada carácter consiste en 9 elementos: 5 barras y 4 espacios. Cada carácter incluye 3 elementos anchos y 6 estrechos (Ver figura 4.6). Los caracteres son separados por un boquete del inter-carácter que sea de la misma anchura que una barra estrecha. El cociente de la anchura de la barra de wide:narrow puede estar en la gama de 1.8 a 3.4. Códigos de barra con una anchura estrecha de la barra de menos de 0.020 pulgada (0.508m m) debe tener un cociente de por lo

menos 2.5. Un cociente de 3.0 se recomienda. Cada código de barras del código 39 se debe preceder y seguir por una zona reservada la anchura por lo menos de 10 barras estrechas.



Figura 4.6 Ejemplo de un código de barras EAN-39

El código de barras que se elegirá para realizar la identificación del Becario es EAN 39 ya que este código como ya se vio es de tipo alfanumérico acepta varios caracteres y dado que el Id del Becario será su RFC mas un número consecutivo se tomara este código.

### **4.3 Credencial e Identificación de los Becarios**

Se requiere tener la información de los Becarios así como generar su credencial y tener un número de identificación único (que será el Código de barras); esto para tomar asistencia de forma ágil y eficaz. Después de aplicar la Reingeniería de Procesos, encontramos la forma de optimizar el modelo que se vio al principio del capítulo incluyendo al nuevo sistema:

Al realizar la credencial del Becario vamos a adquirir la información del mismo, ejemplo: todos sus datos personales; en ese momento toda esta información almacenada se puede migrar hacia el servidor evitando así tener que volver a tomar todos los datos del Becario para el sistema, ver figura 4.7. Es decir, con el registro del becario y la impresión de su credencial tendremos toda la información que necesitamos para el Sistema de Control de Becarios para la DGDC.

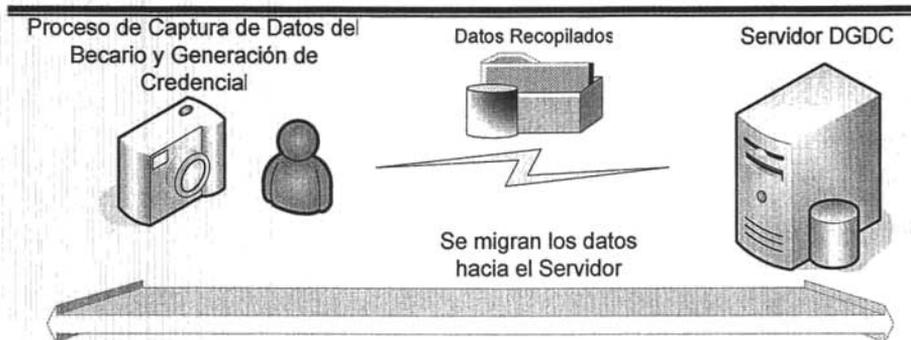


Figura 4.7 Esquema conceptual de Procesos para el Sistema

Proponemos una forma rápida, eficaz y ágil de pasar lista, tener el control de los Becarios, generar los reportes necesarios de los datos del Becario, actualizar una Base de Datos para los Becarios, desde el mismo momento que el Becario se registre.

De forma detallada esto sería así: Ver figura 4.8



Figura 4.8 Diagrama del Proceso Automatizado

Considerando que tenemos la información suficiente en la credencial podemos describir el modelo detallado para tomar asistencia.

La forma de pasar asistencia a la llegada del Becario a Universum sería como se muestra a continuación en la figura 4.9

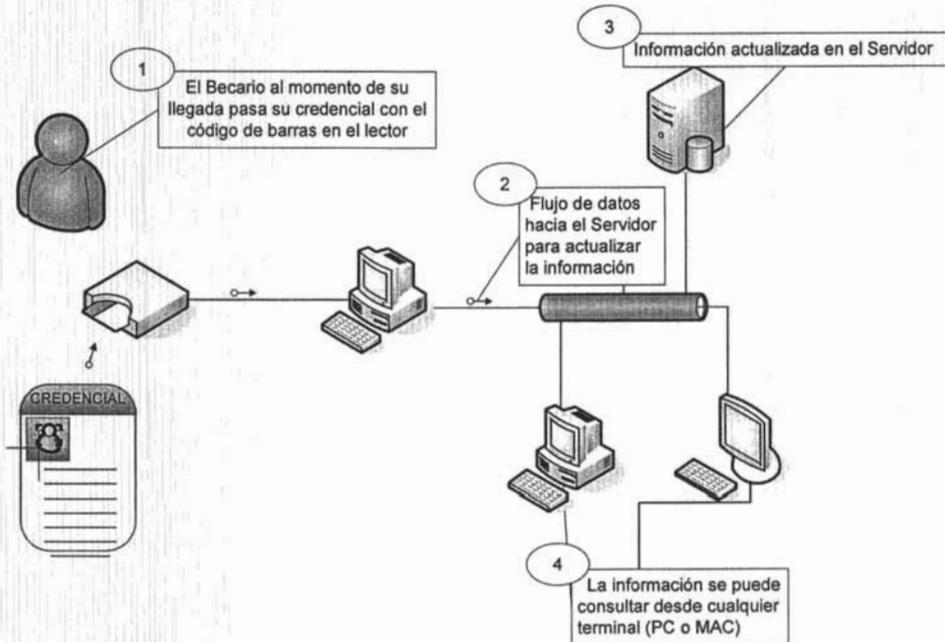


Figura 4.9 Proceso de identificación del Becario

## 4.4 Creación de la Base de Datos para el Control de Becarios DGDC

Para hacer eficiente la Base de Datos tenemos que realizar un análisis de la misma, para no tener una redundancia en las búsquedas ni arrojar algún error. Comenzamos por la creación de una tabla que llamaremos tabla de **Becarios**, en la que tendremos una llave primaria (PK) la cual será el **Id \_ becario**, además de contar con todos los datos solicitados para el registro del Becario como se ejemplifica en la figura 4.10

Becarios	
PK	<u>id becario</u>
	nombre
	appat
	apmat
	fecha_de_nacimiento
	edad
	edo_civil
	direccion
	telefono
	email
	universidad
	facultad
	carrera
	no_cuenta
	Idiomas
	creditos
	promedio
	Otros_edos
	Otras_Acti

Figura 4.10 Tabla o Entidad Becarios

Contamos también con la tabla o Entidad **Salas** con su respectiva llave primaria **id\_sala** y como registro tendremos todas las Salas dentro del museo; al relacionar estas dos tablas tenemos, como se ve en la figura 4.11, una relación de los becarios hacia las salas como sigue:

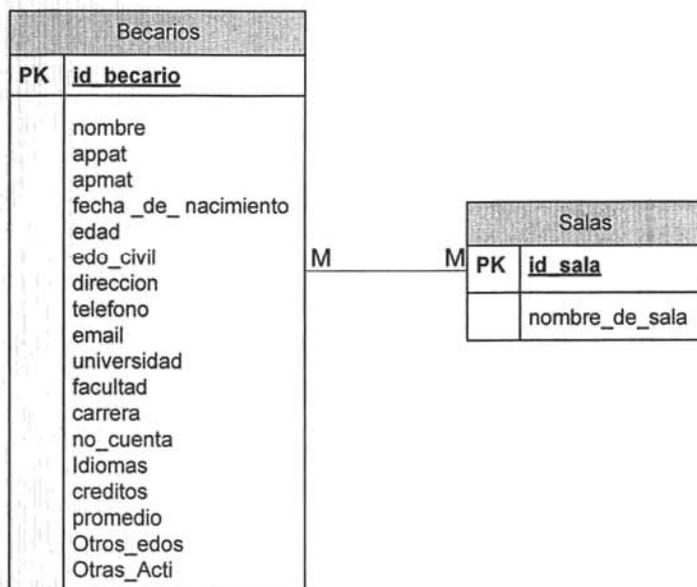


Figura 4.11 Relación M : M entre las entidades Becarios y Salas

- Un becario puede asignarse a una o más salas
- En una sala puede haber uno o mas becarios

Considerando estas premisas tenemos una asociación entre las dos entidades anteriores (Becarios y Salas) de muchos a muchos o M : M (esto ocurre cuando se asocian una ocurrencia en una entidad con muchas ocurrencias de otra entidad y viceversa.)

El diseño de Bases de Datos nos dice que se deben romper todas las relaciones muchos a muchos por tanto realizamos el rompimiento de esta asociación de la siguiente forma (Ver figura 4.12) y obtenemos la nueva entidad que llamaremos **Becarios X Salas** a la cual se le heredan las llaves primarias **id\_becario** e **id\_sala**, esta nueva tabla se une a la tabla Salas y a Becarios mediante una relación 1:M (uno a muchos) y también se cambia esa relación de M:M por una 1:M entre Becarios y Salas; se dice que una relación entre entidades es 1:M, si las ocurrencias de una entidad están relacionadas con varias ocurrencias de otra entidad.

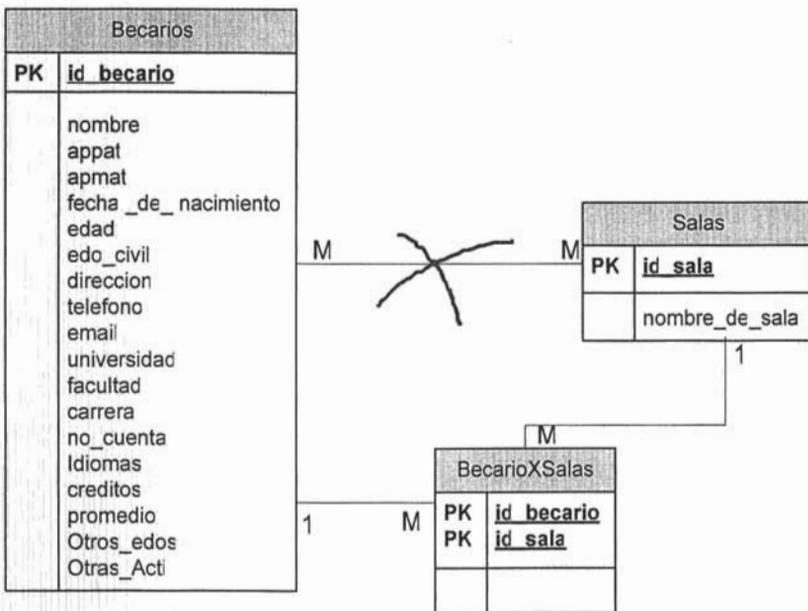


Figura 4.12 Rompimiento de la asociación M:M y generación de la nueva entidad.

La tabla **BecarioXSalas** se relacionará 1:1 con la tabla Horario de Becario y heredará el **id\_horario\_becario** como se dejará ver más adelante en el diseño final (figura 4.15)

Este mismo análisis se realiza para la entidad o tabla Horarios de Becario como se ejemplifica en la figura 4.12 excepto que aquí tenemos una relación 1:1 de Becarios hacia Horarios de Becario así como una llave primaria **id\_horario\_de\_becario** además de los atributos días de la semana (lunes, martes, miércoles, jueves, viernes, sábado, domingo) para esta tabla. Una relación 1:1 (uno a uno) es en donde las ocurrencias de una entidad se pueden relacionar solo a una ocurrencia de la otra entidad. En esta figura 4.13 podemos ver como se hereda la llave primaria **id\_becario**

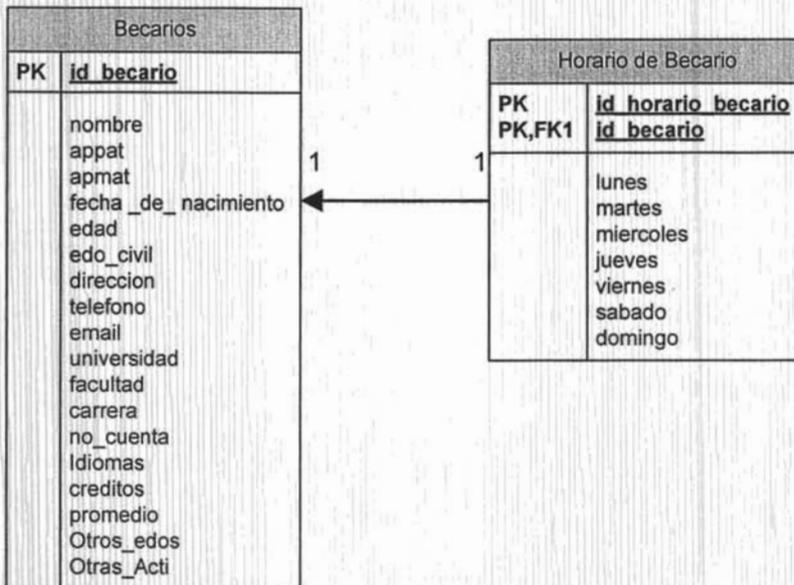


Figura 4.13 Relación 1:1 entre la tabla Becarios y la tabla Horarios de Becario

Esta relación 1:1 la obtenemos ya que a la entidad Becario solo le interesa saber los horarios del Becario.

Otra entidad que necesitamos es la **Asistencia** como registros tendremos **presente\_hoy** (para saber si el Becario esta presente en el día), **num\_faltas** (número de faltas del Becario), **num\_retardos** (numero de retardos del Becario), **num\_asistencias** (número de asistencias); este ultimo solo para estadísticas.

A esta entidad Asistencia la asociamos con la entidad Becarios ya que se trata de una característica de los becarios (si están presentes o no), la relación de las tablas o entidades será 1: 1 tal como se muestra en la siguiente figura 4.14.

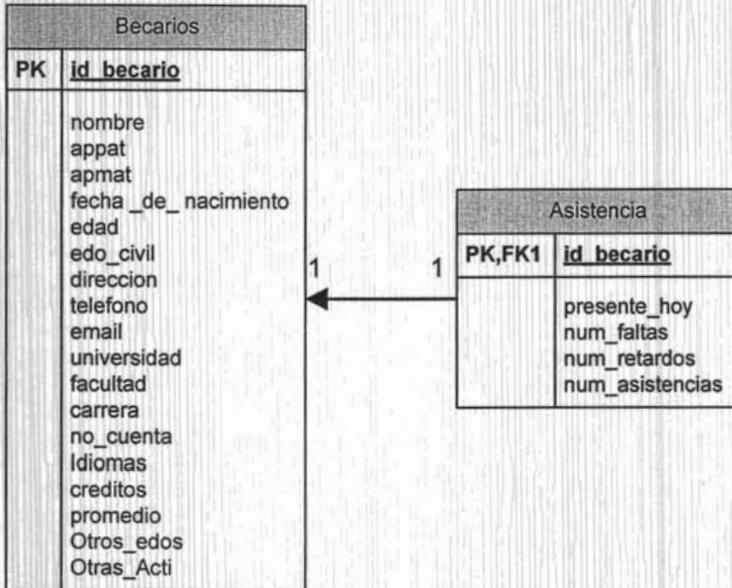


Figura 4.14 Relación 1:1 entre la tabla Becarios y la tabla Asistencia

No podemos olvidarnos de la Seguridad en la Base de Datos por tanto es indispensable crear una tabla adicional a la que llamaremos Usuarios en la que pondremos como llave primaria un Login y como atributo un Password. Esto para asegurarnos de que las personas que ingresen al sistema o a la base de datos tengan permiso para acceder a estas. También tendremos otra tabla que llamaremos Horas de Llegada esta tabla servirá para guardar la información de la hora de llegada del Becario.

Así el diseño de la Base de Datos se presenta de la siguiente forma: (Ver figura 4.15)

Diseño Base de Datos Sistema de Control de Becarios DGDC

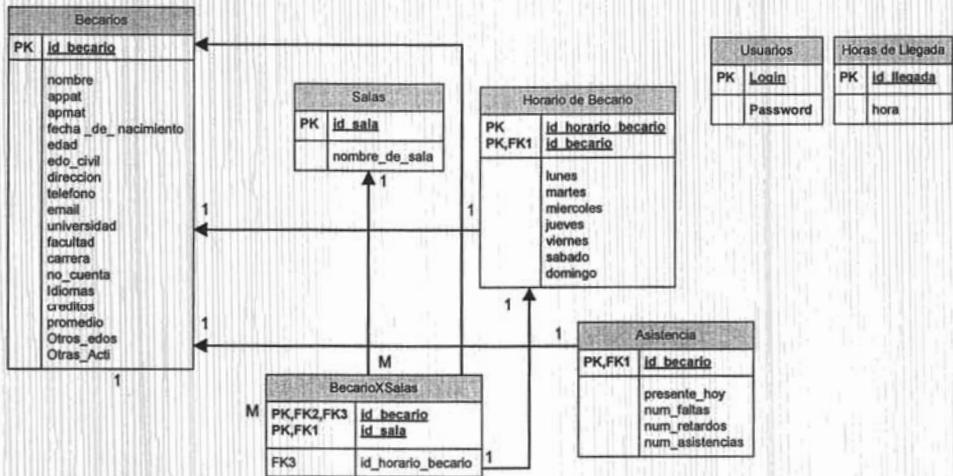


Figura 4.15 Diseño de la Base de Datos para la DGDC

En la siguiente figura (ver figura 4.16) se muestra como será el proceso desde que el Becario ingresa a la Dependencia hasta la visualización de la información por el personal autorizado (solo usuarios con acceso).

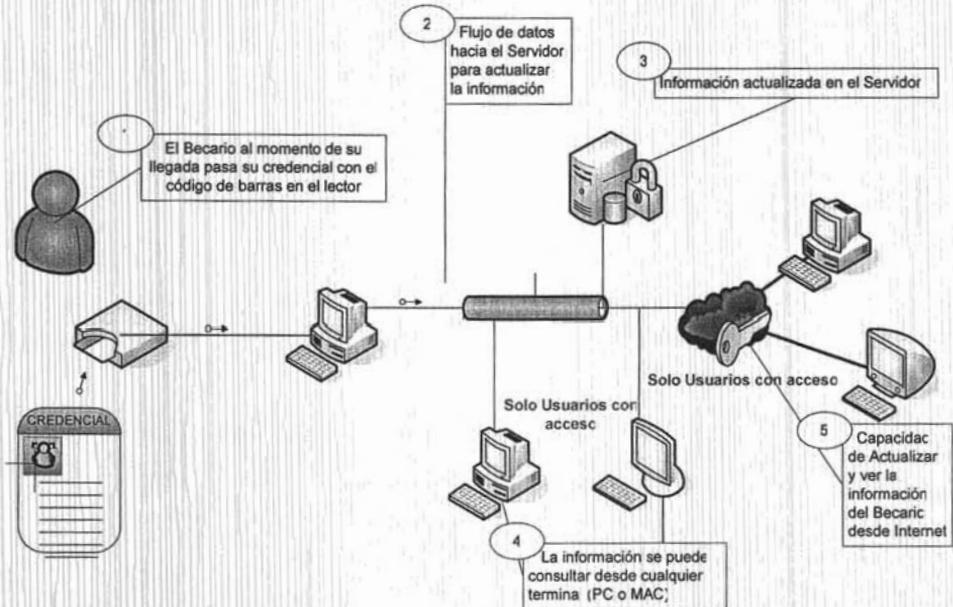
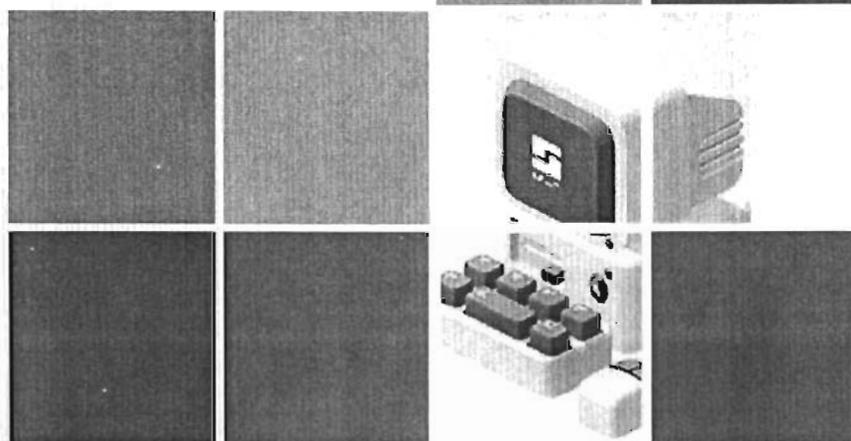


Figura 4.16 Modelo conceptual del proceso Automatizado

# Construcción y Desarrollo

Capítulo

5





Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**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.

# CAPÍTULO 5

## CONSTRUCCIÓN Y

## DESARROLLO

*En este capítulo se realizará la construcción del Sistema Propiamente dicho, se explicaran y se expondrán los diagramas de flujo necesarios para cada parte del Sistema de Control de Becarios, es decir, explicaremos el funcionamiento esencial de cada Programa así como una vista rápida de los lenguajes en los que se desarrollo el sistema.*

## **5 Construcción y Desarrollo**

Después de analizar el problema al que nos enfrentamos y crear un modelo que resuelva esta problemática el siguiente paso natural es la construcción y desarrollo del sistema. Para ello es necesario elegir las herramientas que ayuden a llevar a buen término este proyecto. Gracias al trabajo hecho en los anteriores capítulos, ahora tenemos más claros los requerimientos del sistema y las condiciones en las que debe operar. Así bien partiendo del modelo podemos separar este sistema en tres partes:

- a) Elaboración de las credenciales
- b) Tomar asistencia, manejo de información en red local
- c) Consulta de información vía Internet

La anterior división nos ayuda en el desarrollo como lo veremos más adelante.

Después de analizar todas las herramientas para desarrollo se llegó a la conclusión de emplear para el punto a) Visual Basic 6.0 y para los dos puntos restantes usar Java. A continuación se plantean los puntos importantes por los cuales se escogieron.

### **5.1 Visual Basic 6.0**

Es un sistema de desarrollo diseñado especialmente para crear aplicaciones con interfaz gráfica, de una forma rápida y sencilla. Para ello se vale de dos herramientas (principalmente), la primera permite realizar diseños gráficos para crear la Interfaz Gráfica de Usuario.

También podemos ver que Visual-Basic es una herramienta de diseño de aplicaciones para Windows, en la que estas aplicaciones se desarrollan en gran parte a partir del diseño de una interfase gráfica. En una aplicación Visual Basic, el programa está formado por una parte de código puro, y otras partes asociadas a los objetos que forman la interfase gráfica.

El lenguaje esta basado en BASIC (Beginner's All purpose Symbolic Instruction Code ) que nació en el año 1964 como una herramienta destinada a principiantes, buscando una forma sencilla de realizar programas, empleando un lenguaje casi igual al usado en la vida ordinaria (en inglés), y con instrucciones muy sencillas y escasas. Pero después de varias evoluciones esta

herramienta conserva del BASIC únicamente su nombre y su sencillez. Y aunque no se tiene prestaciones como otros lenguajes cuando deseamos llegar al fondo de la máquina y controlar uno a uno sus registros, en la mayor parte de las aplicaciones, las herramientas aportadas por VB son mas que suficiente para lograr un programa fácil de realizar y de altas prestaciones.

Es por tanto un término medio entre la programación tradicional, formada por una sucesión lineal de código estructurado, y la programación orientada a objetos. Combina ambas tendencias. Ya que no podemos decir que VB pertenezca por completo a uno de esos dos tipos de programación, debemos inventar una palabra que la defina: programación visual.

### **5.1.1 Características Generales de Visual-Basic**

Cuenta con un entorno de desarrollo integrado: editor de Texto, intérprete, depurador, examinador de objetos, explorador de proyectos, compilador, etc. Visual Basic fue diseñado para ser intérprete, lo que favorece la creación y la depuración de una aplicación, y a partir de la versión 5 incluyó también un compilador que permite generar .exe favoreciendo así la ejecución de los programas. En adición, puede crear ficheros .ocx y .dll lo que permitirá manipular controles ActiveX.

La creación de un programa bajo Visual Basic lleva los siguientes pasos:

*Análisis.* Es el estudio de las necesidades que han dado origen a la creación de ese programa. Es lo que se llama análisis de la aplicación. Es la primera fase que debe tener siempre un programa y es también la más olvidada entre los programadores. Una aplicación no se inicia con el teclado, sino sobre un papel.

*Creación de una interfase gráfica de usuario.* Esta interfase será la principal vía de comunicación hombre máquina, tanto para salida de datos como para entrada. Será necesario partir de una o varias ventanas - Formularios - a las que les iremos añadiendo los controles necesarios.

*Definición de las propiedades de los controles.* Se dará la forma, posición, y todas las características necesarias a los controles que hayamos colocado en ese formulario. Estas propiedades determinarán la forma estática de los controles, es decir, como son los controles y para qué sirven.

***Generación del código asociado a los eventos que ocurran a estos controles.***  
A la respuesta a estos eventos (click, doble click, una tecla pulsada, etc.) le llamamos Procedimiento, y deberá generarse de acuerdo a las necesidades del programa.

***Generación del código del programa.*** Un programa puede hacerse solamente con la programación de los distintos procedimientos que acompañan a cada objeto. Sin embargo, VB ofrece la posibilidad de establecer un código de programa separado de estos eventos. Este código puede introducirse en unos bloques llamados Módulos, en otros bloques llamados Funciones, y otros llamados Procedimientos. Estos Procedimientos no responden a un evento acaecido a un control o formulario, sino que responden a un evento producido durante la ejecución del programa.

Las anteriores son características generales de la herramienta, cuenta con otras entre las destacadas tenemos: Acceso a bases de datos a través del control de datos ADO, utilizando el motor de Access o controladores ODBC, además de acceso utilizando OLE DB. Cuenta con una biblioteca que permite manipular bases de datos relacionales. Y un administrador visual de datos para manipular bases de datos.

### **5.1.2 Visual Basic en el Sistema de Control de Becarios**

Después de ver las características generales de esta herramienta, podemos ir a puntos más específicos:

- Para adquirir la fotografía se pueden tomar en cuenta tres formas: digitalizarla, tomarla desde una cámara digital y emplear una cámara Web. Pero las dos primeras toman tiempo y no resulta práctico.

Con la última se puede aprovechar la documentación, librerías disponibles y la facilidad que tiene Visual Basic para conectarse a distintos dispositivos periféricos. La mayoría de las Web Cam son fácilmente instaladas en una PC y con Visual Basic se pueden aprovechar los controladores del sistema y así ahorrar tiempo, puesto que no es necesario escribir un controlador.

Aprovechando lo anterior es posible tener una pre-visualización de la fotografía, lo cual no es posible en muchos sistemas de credencialización existentes en el mercado.

Además la forma en que se maneja la información facilita el almacenamiento de la fotografía en una base de datos.

- La impresión de la credencial es muy fácil y sencilla, ya que con Visual Basic se utiliza la interfaz de impresión que los usuarios ya conocen. Para la impresión del código de barras se tiene la ventaja de un fácil al dispositivo de impresión.
- Para la hoja con los datos del becario que deben firmar, se cuenta con un generador de reportes en Visual Basic. Empleando esta herramienta se puede crear un formato que se llene de forma automática.
- La información se puede almacenar en una base de datos interna al programa y después ser migrada a un servidor

Contemplando estas características y otras bondades del lenguaje, aunado a la experiencia que tenemos programando con esta herramienta se decidió crear el sistema de credenciales en Visual Basic 6.

## 5.2 Java

En 1991 Sun Microsystems patrocinó un proyecto de investigación denominado Green, el cuál desembocó en el desarrollo de un lenguaje basado en C++, al que su creador James Gosling, llamó *Oak*. Posteriormente se descubrió que ya existía un lenguaje de programación con el mismo nombre y posteriormente se sugirió *Java*, que finalmente fue aceptado.

Los programas en Java constan de varias piezas llamadas clases. Estas clases incluyen piezas llamadas métodos, las cuales realizan tareas y devuelven información cuando completan esas tareas. Se puede crear dichas piezas pero en Java se cuenta con una gran variedad de colecciones de clases existentes en las bibliotecas de clases de Java, que también se conocen como APIs (Interfaces de programación de aplicaciones) de Java.

Los programas en Java se crean usando editores de Texto ASCII como el *vi*, *emacs* o *notepad* y por lo general para poder ejecutarse pasan por cinco fases: edición, compilación, carga, verificación y ejecución. Estas etapas se explican en la siguiente figura (Ver figura 5.1):

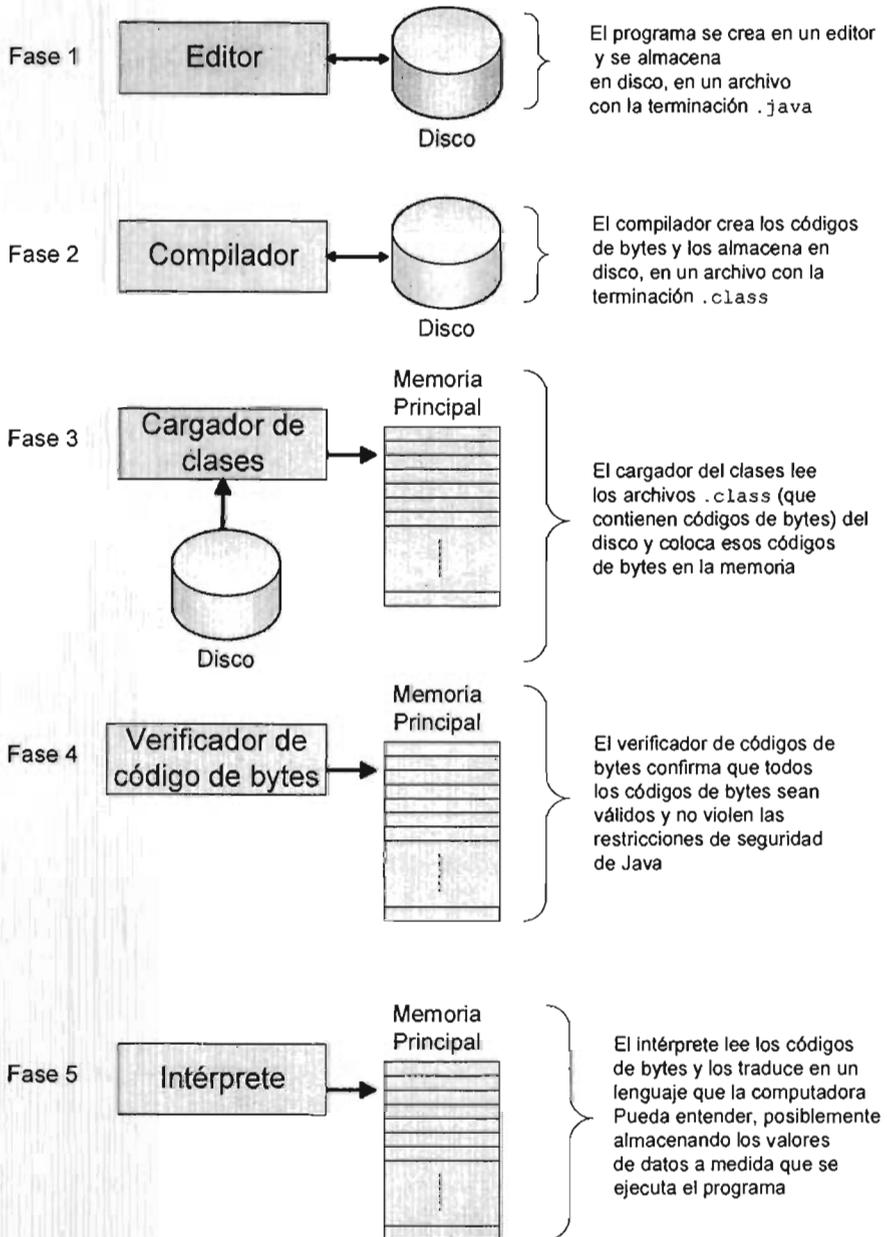


Figura 5.1 Entorno típico en Java

## 5.2.1 Características Generales de Java

**Orientado a objetos:** Java implementa la tecnología básica de C++ con algunas mejoras y elimina algunas cosas para mantener el objetivo de la simplicidad del lenguaje. Java trabaja con sus datos como objetos y con interfaces a esos objetos. Soporta las tres características propias del paradigma de la orientación a objetos: encapsulación, herencia y polimorfismo. Las plantillas de objetos son llamadas, como en C++, clases y sus copias, instancias. Estas instancias, como en C++, necesitan ser construidas y destruidas en espacios de memoria.

**Distribuido:** Java se ha construido con extensas capacidades de interconexión TCP/IP. Existen librerías de rutinas para acceder e interactuar con protocolos como HTTP y FTP. Esto permite a los programadores acceder a la información a través de la red con tanta facilidad como a los archivos locales.

**Robusto:** Java realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución. La comprobación de tipos en Java ayuda a detectar errores, lo antes posible, en el ciclo de desarrollo. Java obliga a la declaración explícita de métodos, reduciendo así las posibilidades de error. Maneja la memoria para eliminar las preocupaciones por parte del programador de la liberación o corrupción de memoria. También implementa los Arrays auténticos, en vez de listas enlazadas de punteros, con comprobación de límites, para evitar la posibilidad de sobrescribir o corromper memoria resultado de punteros que señalan a zonas equivocadas. Estas características reducen drásticamente el tiempo de desarrollo de aplicaciones en Java. Además, para asegurar el funcionamiento de la aplicación, realiza una verificación de los byte codes.

**Interpretado:** El intérprete Java (sistema run-time) puede ejecutar directamente el código objeto. El lenguaje Java es a la vez compilado e interpretado. Con el compilador se convierte el código fuente que reside en archivos cuya extensión es *.java*, a un conjunto de instrucciones que recibe el nombre de byte codes que se guardan en un archivo cuya extensión es *.class*. Estas instrucciones son independientes del tipo de ordenador. El intérprete ejecuta cada una de estas instrucciones en una computadora específica (Windows, Macintosh, etc). Solamente es necesario, por tanto, compilar una vez el programa, pero se interpreta cada vez que se ejecuta en una computadora. Como lo podemos ver en la figura 5.2:

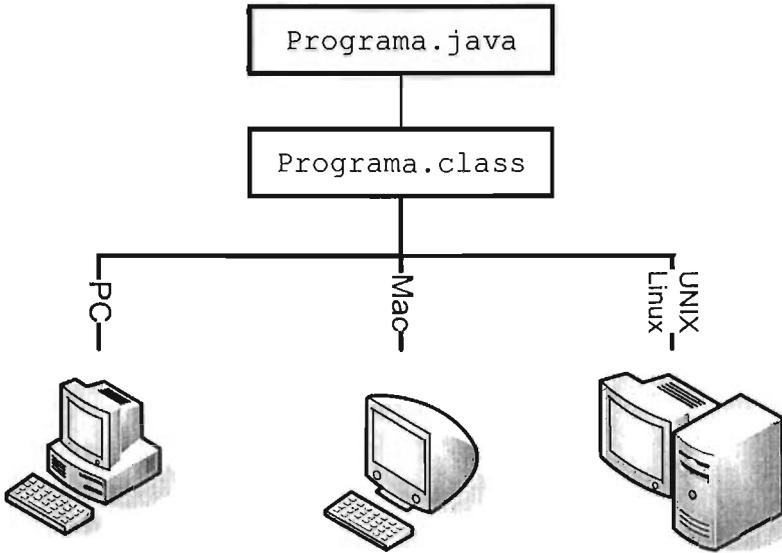


Figura 5.2 Portabilidad de Java

Cada intérprete Java es una implementación de la Máquina Virtual Java (JVM). Los byte codes posibilitan el objetivo de "write once, run anywhere", de escribir el programa una vez y que se pueda correr en cualquier plataforma que disponga de una implementación de la JVM. Por ejemplo, el mismo programa Java puede correr en Windows 98, Solaris, Macintosh, etc.

Como vemos en la figura 5.3, el API de Java y la Máquina Virtual Java forman una capa intermedia (Java platform) que aísla el programa Java de las especificidades del hardware (hardware-based platform).

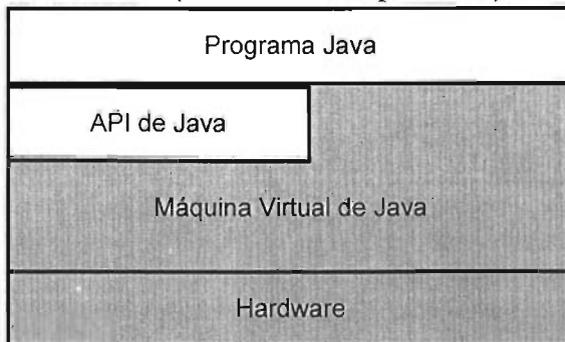


Figura 5.3 Máquina Virtual de Java

**Conexión a Bases de Datos:** Los programas en Java se comunican con las bases de datos y manipulan sus datos utilizando la *API JDBC*. Un controlador de JDBC implementa la interfaz para una base de datos específica. Al separar la API de los controladores específicos, los desarrolladores pueden cambiar la base de datos subyacente sin necesidad de modificar el código Java para tener acceso a dicha base. La mayoría de los sistemas de administración de bases de datos populares incluyen controladores de JDBC y también los hay de terceros.

## 5.3 MYSQL

MySQL es el sistema gestor de bases de datos de código abierto más popular del mundo. La razón: las empresas que utilizan MySQL consiguen recortar el gasto dedicado a este tipo de software en un 90%, y por regla general su uso es gratuito. La facilidad de uso, la velocidad y la flexibilidad en diferentes sistemas operativos hacen popular a MySQL. Por la razón de costos es que hemos elegido este gestor de bases de datos.

### 5.3.1 Tipos de Datos en MYSQL

MySQL soporta varios tipos de datos, que pueden ser agrupados en tres categorías: numéricos, fecha y hora, y cadenas (caracteres). En este capítulo se presenta un breve resumen de estos tipos, se mencionan sus requerimientos de almacenamiento, y se hace una descripción más detallada de las propiedades de cada uno de ellos. Los tipos de datos soportados por MySQL son listados a continuación.

Las siguientes convenciones son usadas en las descripciones:

M - Indica el tamaño máximo que se puede desplegar (mostrar). El mayor valor legal es de 255.

D - Aplica a los datos de punto flotante e indica el número de dígitos que siguen al punto decimal. El valor máximo posible es de 30, pero no debe ser mayor de M-2.

Los paréntesis cuadrados indican que un elemento es opcional.

Se debe notar que cuando se especifica **ZEROFILL** para una columna, MySQL agregará automáticamente el atributo **UNSIGNED**.

- **TINYINT[(M)] [UNSIGNED] [ZEROFILL]**

Un entero muy pequeño. Su rango es de -128 a 127. Para enteros sin signo el rango es de 0 a 255.

- **BIT y BOOL**

Es un sinónimo para **TINYINT(1)**.

- **SMALLINT[(M)] [UNSIGNED] [ZEROFILL]**

Un entero pequeño. Su rango es de -32768 a 32767. Para enteros sin signo el rango es de 0 a 65535.

- **MEDIUMINT[(M)] [UNSIGNED] [ZEROFILL]**

Un entero de tamaño medio. Su rango es de -8388608 a 8388607. Para enteros sin signo el rango es de 0 a 16777215.

- **INT[(M)] [UNSIGNED] [ZEROFILL]**

Un entero normal. Su rango es de -2147483648 a 2147483647. Para enteros sin signo el rango es de 0 a 4294967295.

- **INTEGER[(M)] [UNSIGNED] [ZEROFILL]**

Es un sinónimo para **INT**.

- **BIGINT[(M)] [UNSIGNED] [ZEROFILL]**

Es un entero grande. Su rango es de -9223372036854775808 a 9223372036854775807. Para enteros sin signo el rango es de 0 a 18446744073709551615.

- **FLOAT(precisión) [UNSIGNED] [ZEROFILL]**

Un número de punto flotante. La precisión puede ser  $\leq 24$  para un número de precisión sencilla y entre 25 y 53 para un número de doble precisión. Este tipo es como el **FLOAT** y **DOUBLE** que se describen a continuación. Un **FLOAT(X)** tiene el mismo rango que los correspondientes tipos **FLOAT** y **DOUBLE**, pero el tamaño de despliegue y el número de decimales no están definidos.

- `FLOAT[(M,D)] [UNSIGNED] [ZEROFILL]`

Un número pequeño de punto flotante (o bien, de precisión sencilla). El rango de valores permitidos es de  $-3.402823466E+38$  a  $-1.175494351E-38$ , 0, y de  $1.175494351E-38$  a  $3.402823466E+38$ . Si se usa el modificador `UNSIGNED`, los valores negativos son deshabilitados. M es el tamaño de despliegue y D el número de decimales. `FLOAT` sin argumentos o `FLOAT(X)` donde  $X \leq 24$  se entiende como un número de punto flotante de precisión sencilla.

- `DOUBLE[(M,D)] [UNSIGNED] [ZEROFILL]`

Un número normal de punto flotante (o bien, de precisión doble). El rango de valores permitidos es de  $-1.7976931348623157E+308$  a  $-2.2250738585072014E-308$ , 0, y de  $2.2250738585072014E-308$  a  $1.7976931348623157E+308$ . Si el modificador `UNSIGNED` es especificado, los valores negativos son deshabilitados. M es el tamaño de despliegue y D el número de decimales.

- `DOUBLE` sin argumentos o `DOUBLE(X)` donde  $25 \leq X \leq 53$  se entiende por un número de punto flotante de doble precisión.
  - `DOUBLE PRECISION[(M,D)] [UNSIGNED] [ZEROFILL]`
  - `REAL[(M,D)] [UNSIGNED] [ZEROFILL]`
- Son sinónimos para `DOUBLE`.

- `DECIMAL[(M[,D])] [UNSIGNED] [ZEROFILL]`

Un número de punto flotante sin empaquetar. Como en las columnas `CHAR`, "sin empaquetar" significa que el número es almacenado como una cadena con un carácter para cada dígito del valor. El punto decimal, para números negativos el signo - no son contados en M (sin embargo se les reserva espacio). Si D vale 0, los valores no tendrán punto decimal.

El máximo rango de valores es el mismo que para `DOUBLE`, pero el rango para una columna `DECIMAL` puede ser restringido con la elección de valores en M y D. Si el atributo `UNSIGNED` es usado, los valores negativos son deshabilitados. Si D es omitido, el valor por omisión es 0. Si M es omitido el valor por omisión es 10.

- `DEC[(M[,D])] [UNSIGNED] [ZEROFILL]`
  - `NUMERIC[(M[,D])] [UNSIGNED] [ZEROFILL]`
- Son sinónimos para `DECIMAL`.

- **DATE**

Valores de fecha. El rango válido es de '1000-01-01' a '9999-12-31'. MySQL muestra valores DATE en el formato 'YYYY-MM-DD', y permite asignar valores a columnas DATE usando cadenas o números.

- **DATETIME**

Una combinación de fecha y hora. El rango de valores válidos es de '1000-01-01 00:00:00' a '9999-12-31 23:59:59'. MySQL muestra los valores DATETIME en el formato 'YYYY-MM-DD HH:MM:SS', y permite asignarles valores usando cadenas y números.

- **TIMESTAMP[(M)]**

Un *timestamp*. El rango de valores es de '1970-01-01 00:00:00' a algo antes del año 2037. MySQL muestra estos valores en el formato YYYYMMDDHHMMSS, YMMDDHHMMSS, YYYYMMDD, o YYMMDD, dependiendo de si M es 14 (o hace falta), 12, 8, o 6, y permite asignar estos valores usando cadenas y números.

Las columnas **TIMESTAMP** son útiles para grabar la fecha y la hora en que se efectuó una operación **INSERT** o **UPDATE**, ya que automáticamente se le asigna el valor de la fecha y la hora de la operación más reciente si es que no se le asigna un valor diferente. El argumento **M** afecta únicamente como se muestra una columna **TIMESTAMP**; sus valores son almacenados usando 4 bytes. Se debe notar que cuando **M** es 8 o 14 se reportan como números, mientras que si **M** tiene otros valores, se reportan como cadenas.

- **TIME**

Una hora. Su rango es de '-838:59:59' a '838:59:59'. MySQL muestra valores **TIME** en el formato 'HH:MM:SS', y permite asignar valores usando cadenas o números.

- **YEAR[(2|4)]**

Un **año** en formato de 2 o 4 dígitos. Los valores permitidos son de 1901 a 2155, 0000 en el formato de 4 dígitos, y 1970-2069 si se usa el formato de 2 dígitos (70-69). MySQL muestra valores **YEAR** en el formato **YYYY**, y permite asignarles valores usando cadenas o números.

- [NATIONAL] CHAR(M) [BINARY]

Una cadena de longitud fija que siempre es rellenada con espacios a la derecha para completar la longitud específica. El rango de M es de 0 a 255 caracteres. Los espacios sobrantes son removidos cuando se recuperan los valores. Los valores CHAR son almacenados y comparados en modo no sensitivo a menos que se especifique el modificador BINARY. NATIONAL CHAR es la manera de definir que una columna CHAR debe usar el conjunto CHARACTER por default.

- CHAR

Es un sinónimo para CHAR(1).

- [NATIONAL] VARCHAR(M) [BINARY]

Una cadena de longitud variable. Los espacios sobrantes son removidos cuando el valor es almacenado. El rango de M es de 0 a 255 caracteres. Los valores VARCHAR son ordenados y comparados en modo no sensitivo a menos que se use el modificador BINARY.

- TINYBLOB

- TINYTEXT

Una columna BLOB o TEXT con una longitud máxima de 255 ( $2^8 - 1$ ) caracteres.

- BLOB

- TEXT

Una columna BLOB o TEXT con una longitud máxima de 65535 ( $2^{16} - 1$ ) caracteres.

- MEDIUMBLOB

- MEDIUMTEXT

Una columna BLOB o TEXT con una longitud máxima de 16777215 ( $2^{24} - 1$ ) caracteres.

- LONGBLOB

- LONGTEXT

- Una columna BLOB o TEXT con una longitud máxima de 4294967295 ( $2^{32} - 1$ ) caracteres.

- ENUM('valor1','valor2',...)
- Una enumeración. Un objeto cadena que puede tener sólo un valor, elegido de la lista de valores 'valor1', 'valor2',... NULL o el valor especial de error "". Un valor ENUM puede tener un máximo de 65535 valores distintos.
  
- SET('valor1','valor2',...)
- Un conjunto. Un objeto cadena que tiene cero o más valores, cada uno de los cuales debe ser elegido de la lista de valores 'valor1','valor2',.... Un valor SET puede tener un máximo de 64 miembros.

## 5.4 Tomcat

Tomcat (Jakarta Tomcat) funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Sun Microsystems. Se le considera un servidor de aplicaciones.

### 5.4.1 Historia de Tomcat

Tomcat empezó siendo una implementación de la especificación de los servlets comenzada por James Duncan Davidson, que trabajaba como arquitecto de software en Sun y que posteriormente ayudó a hacer el proyecto *open source* y en su donación a la Apache Software Foundation.

Duncan Davidson inicialmente esperaba que el proyecto se convirtiese en *open source* y dado que la mayoría de los proyectos open source tienen libros de O'Reilly asociados con un animal en la portada, quiso ponerle al proyecto nombre de animal. Eligió *Tomcat* (gato), pretendiendo representar la capacidad de cuidarse por sí mismo, de ser independiente.

#### 5.4.1.1 Entorno Tomcat

Tomcat funciona con cualquier servidor web con soporte para servlets y JSPs. Tomcat incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache.

Tomcat puede, funcionar como servidor web por sí mismo. Opera de tal manera en entornos de desarrollo poco exigentes en términos de velocidad y de manejo de transacciones. Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.

### **5.4.1.2 Estado del Desarrollo de Tomcat**

Tomcat lo desarrollan y lo mantienen miembros de la Apache Software Foundation y voluntarios independientes. Los usuarios disponen de libre acceso a su código fuente y a su forma binaria en los términos establecidos en la *Apache Software Licence*. Las primeras distribuciones de Tomcat fueron las versiones 3.0.x. Las versiones más recientes son las 5.x, que implementan las especificaciones de Servlet 2.4 y de JSP 2.0. En las versiones 4.x, Jakarta Tomcat utiliza el contenedor de servlets Catalina.

### **5.4.1.3 Estructura de Directorios**

La jerarquía de directorios de instalación de Tomcat incluye:

- Bin : arranque, cierre, y otros scripts y ejecutables.
- Common: clases comunes que pueden utilizar Catalina y las aplicaciones web.
- Conf: ficheros XML y los correspondientes DTDs para la configuración de Tomcat.
- Logs: logs de Catalina y de las aplicaciones.
- Server: clases utilizadas solamente por Catalina.
- Share: clases compartidas por todas las aplicaciones web.
- Webapps: directorio que contiene las aplicaciones web.
- Work: almacenamiento temporal de ficheros y directorios.

### **5.4.1.4 Características de Tomcat**

**Tomcat 3.x** (distribución inicial)

- Implementado a partir de las especificaciones Servlet 2.2 y JSP 1.1
- Recarga de servlets

### **Tomcat 4.x**

- Implementado a partir de las especificaciones Servlet 2.3 y JSP 1.2
- Contenedor de servlets rediseñado como Catalina
- Motor JSP rediseñado como Jasper
- Conector Coyote
- Java Management Extensions (JMX), JSP Y administración basada en Struts

### **Tomcat 5.x**

- Implementado a partir de las especificaciones Servlet 2.4 y JSP 2.0
- Recolección de basura reducida
- Capa envolvente nativa para Windows y Unix para la integración de las plataformas

## **5.5 Visión General de las JSP**

Una página Java en servidor (JSP, acrónimo del inglés Java Server Pages) es una plantilla para una página Web que emplea código Java para generar un documento HTML dinámicamente. Las paginas JSP se ejecutan en un componente de servidor conocido como contenedor de JSP, que las traduce a servlets Java equivalentes.

Por esta razón los servlets y las paginas JSP están íntimamente relacionados. Lo que se puede hacer con una tecnología es, en gran medida, también posible con otra; aunque cada una tiene capacidades propias. Como son servlets, las paginas JSP tienen todas las ventajas de los servlets:

- Tienen un mayor rendimiento y capacidad de adaptación (escalabilidad) que las secuencias de comandos CGI porque se conservan en la memoria y admiten múltiples subprocesos.
- No es necesaria una configuración especial por parte del cliente.
- Incorporan soporte para sesiones HTTP, lo que hace posible la programación de aplicaciones.
- Tienen pleno acceso a la tecnología Java –capacidad de reconocimiento del trabajo en red, subprocesos y conectividad a bases de datos- sin las limitaciones de los applets del cliente.

Pero además las páginas JSP tienen ventajas propias:

- Se vuelven a compilar automáticamente cuando es necesario
- Como están en el espacio común de documentos del servidor Web, dirigirse a ellas es más fácil que dirigirse a los servlets.
- Como las páginas JSP son similares al HTML, tienen mayor compatibilidad con las herramientas de desarrollo Web.

### 5.5.1 Cómo funciona JSP

La página JSP pasa por tres etapas en la evolución de su código:

- **Código fuente JSP.** Este código es el que realmente escribe el desarrollador. Se encuentra en un archivo de texto con extensión `.jsp` y consiste en una mezcla de código de plantilla HTML, instrucciones en el lenguaje Java, directivas JSP y acciones que describen como generar una página Web para dar servicio a una petición concreta.
- **Código fuente Java.** El contenedor de JSP traduce el código fuente JSP al código fuente de un servlet Java equivalente. Este código fuente guarda en un área de trabajo y suele ser útil en el proceso de depuración de errores.
- **Clase Java compilada.** Como cualquier otra clase Java, el código del servlet generado se compila en código de bytes en un archivo `.class`, preparado para ser cargado y ejecutado.

El contenedor de JSP administra cada una de estas etapas de la página JSP automáticamente, basándose en la situación temporal de cada archivo. Como respuesta a la petición HTTP, el contenedor comprueba si el archivo fuente `.jsp` ha sufrido modificaciones desde que el código fuente `.java` se compiló por última vez.

La figura 5.4 muestra el proceso empleado por el contenedor de JSP. Cuando se realiza una petición de una página JSP, el contenedor determina primero el nombre de la clase correspondiente al archivo `.jsp`. Si la clase no existe o es anterior al archivo `.jsp` (lo que significa que el código fuente

JSP ha cambiado desde que fue compilado por ultima vez), entonces el contenedor crea el código fuente Java para un servlet equivalente y lo compila. Si no hay aun un ejemplar o instancia del servlet en funcionamiento, el contenedor carga la clase servlet y crea un ejemplar. Finalmente, el contenedor lanza un subproceso para que gestione la petición HTTP actual del ejemplar cargado.

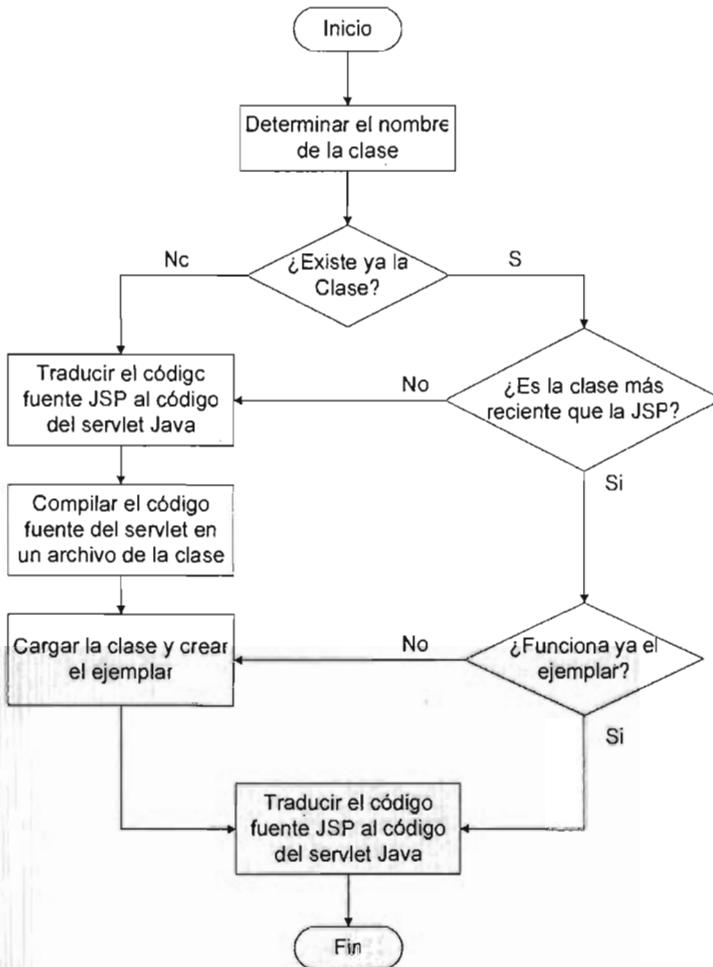


Figura 5.4. Lógica empleada por el contenedor JSP para gestionar la conversión JSP.

## 5.6 Credenciales Para Becarios de la DGDC

Con las herramientas a emplear definidas, detallaremos más las actividades que este programa realizará. En la figura 5.5 se muestran en el diagrama la estructura de este programa.

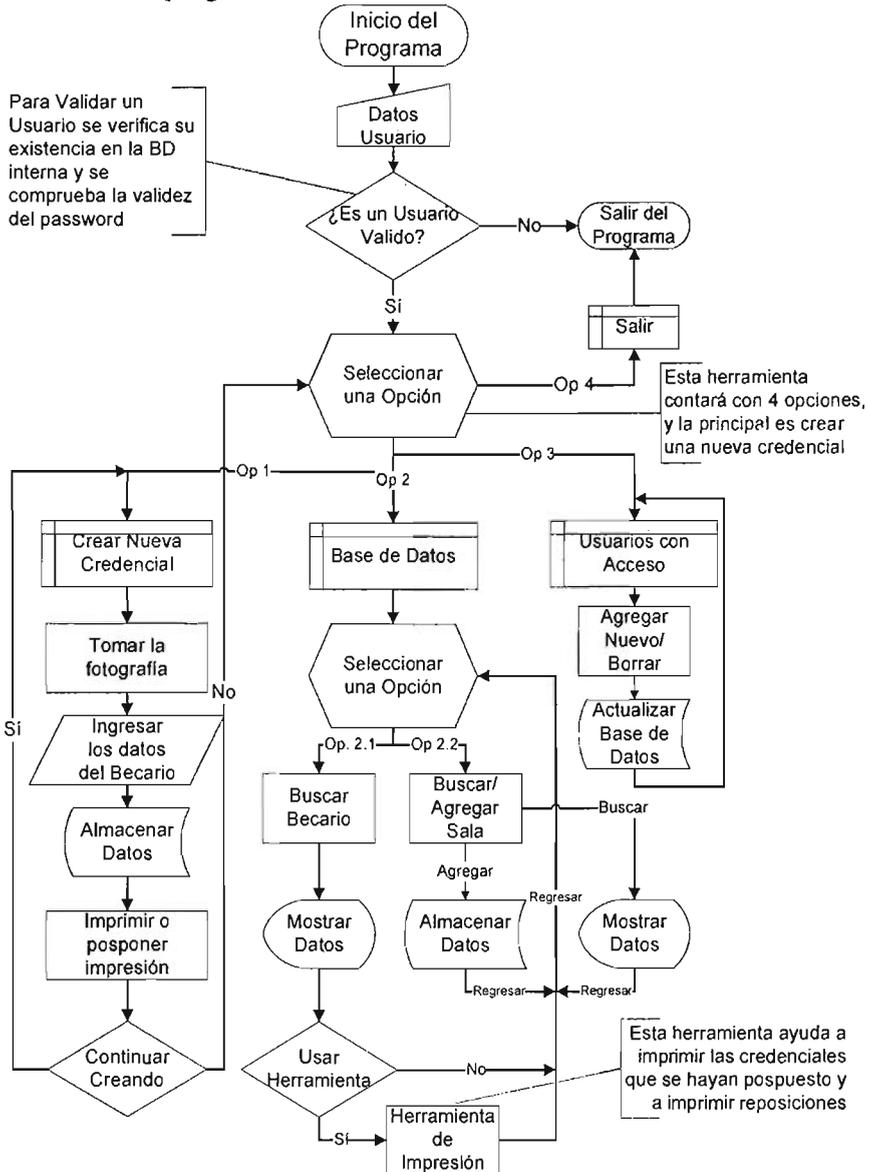


Figura 5.5 Diagrama General para la elaboración de credenciales

En el anterior esquema se mostraron los procesos más relevantes dentro del programa para generar credenciales. Como se observa, algunos se refieren a almacenaje y búsqueda información, para ello se siguen los procedimientos tradicionales y en algunos casos sólo se requiere de una instrucción, por lo cual no los veremos más a fondo. El resto tiene diferente funcionalidad y propósitos; estos son los procesos que veremos más a detalle. Para ellos nos valemos de diagramas de flujo de funciones cruzadas; empleamos estos diagramas en vez de mostrar todo el código ya que pensamos que una imagen puede ilustrar mejor el funcionamiento del programa, más aun si no se está familiarizado con el lenguaje de programación empleado.

El primero es el proceso de *Tomar Fotografía*, en la figura 5.6 se muestra el diagrama de este proceso y dentro de él podemos encontrar tres funciones básicas:

1. Iniciar cámara
2. Mostrar imagen de la cámara
3. Capturar fotografía

En el proceso 1 lo que se busca es establecer una conexión con la WebCam. Con el fin de tener un programa versátil y sin necesidad de que opere sólo con un modelo y /o marca se tiene la posibilidad de operar con cualquier dispositivo de este tipo que sea compatible con el sistema operativo. Esto se logra al reconocer todos los drivers a fin de que estén instalados, es decir, si el usuario tiene configurado en su sistema dos WebCam, podrá operar con la que desee. En adición a lo anterior, se pueden configurar los parámetros que el driver nos permita, como puede ser la resolución, el contraste etc.

Al detectar un driver instalado lo siguiente es determinar si está conectado el dispositivo, si no es el caso se maneja el error para que no colapse el sistema y se informa al usuario que no se encontró un dispositivo de captura. Otro problema relacionado es que otra aplicación esté empleando el dispositivo, en este caso se solicita seleccionar otra fuente de captura disponible, si no la hay canaliza el problema de forma similar al caso anterior.

Si se logra una conexión exitosa con el dispositivo se inicia la pre – visualización, el proceso 2. En la ventana creada se muestra lo que la cámara está captando, para ello se establece una velocidad de muestreo, que en este caso es de 66 milisegundos.

Y finalmente se puede pasar al proceso 3, tomar la fotografía. Consiste en mostrar lo que la cámara esta captando en el instante que se hizo clic en un botón, de forma análoga a una foto tradicional. Para fines del programa se coloca en un `PictureBox` la imagen de la cámara, de hecho la operación es similar a un “copiar y pegar”, es decir, se coloca la información en el porta papeles del sistema y mediante una operación se obtiene la misma, para luego colocarse en el objeto antes mencionado. Todo este se realiza en segundos. El código es el siguiente:

```
capEditCopiar vntnaC  
Picture1.Picture = Clipboard.GetData
```

Parecería muy sencillo pero si se conoce del lenguaje se sabrá que la primera función no existe, fue necesario agrupar varias operaciones para llegar a ella y que al final todo se traduzca en una línea. La clave es saber colocar la información en el porta papeles como si se tratara de una imagen, para que con la segunda línea se coloque dicha información en el objeto contenedor de imágenes. Y para ello se emplea la técnica de recorte Ventana – Puerto, la cuál no explicaremos ya que sale del objetivo de esta Tesis.

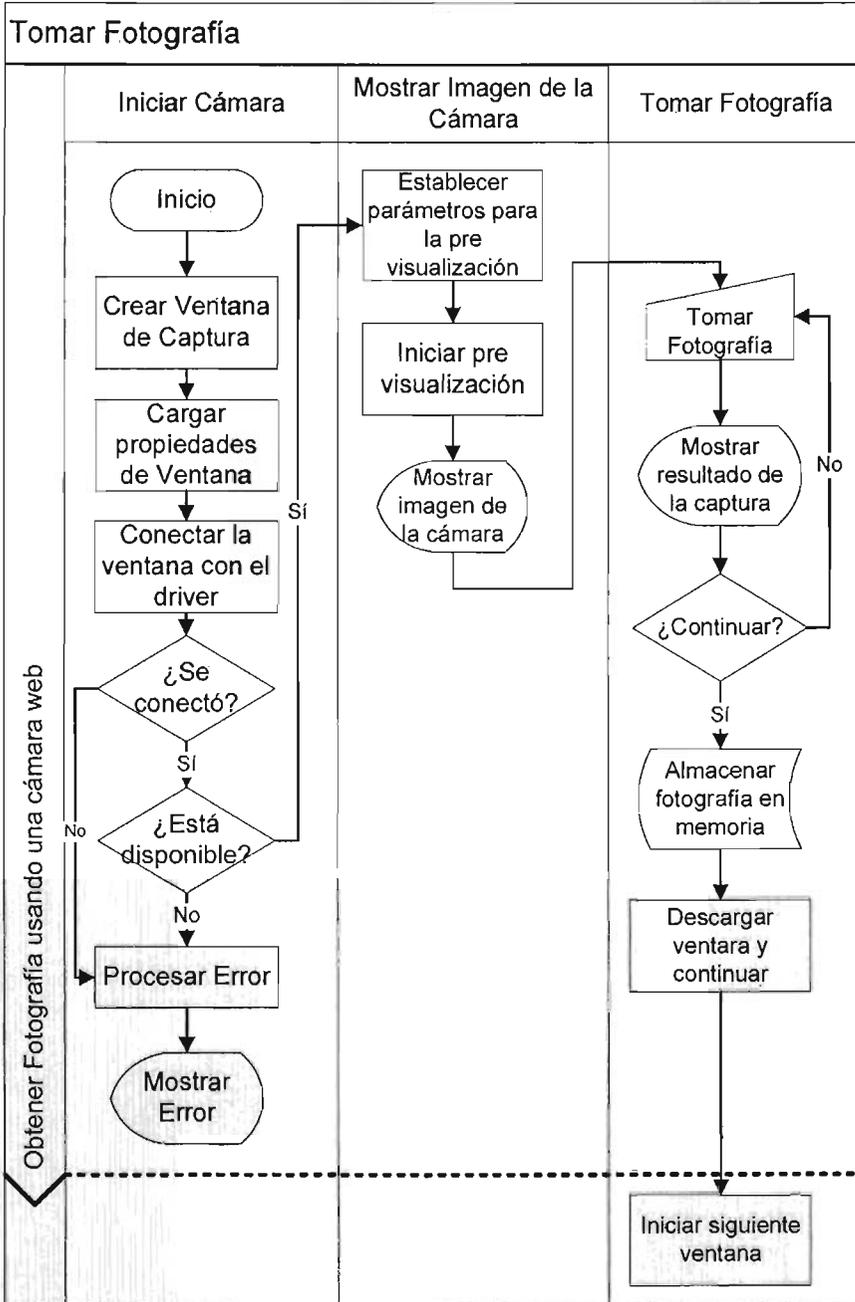


Figura 5.6 Diagrama para tomar fotografía

El siguiente proceso que analizaremos es *Imprimir o posponer impresión*, en la figura 5.7 se observa el diagrama:

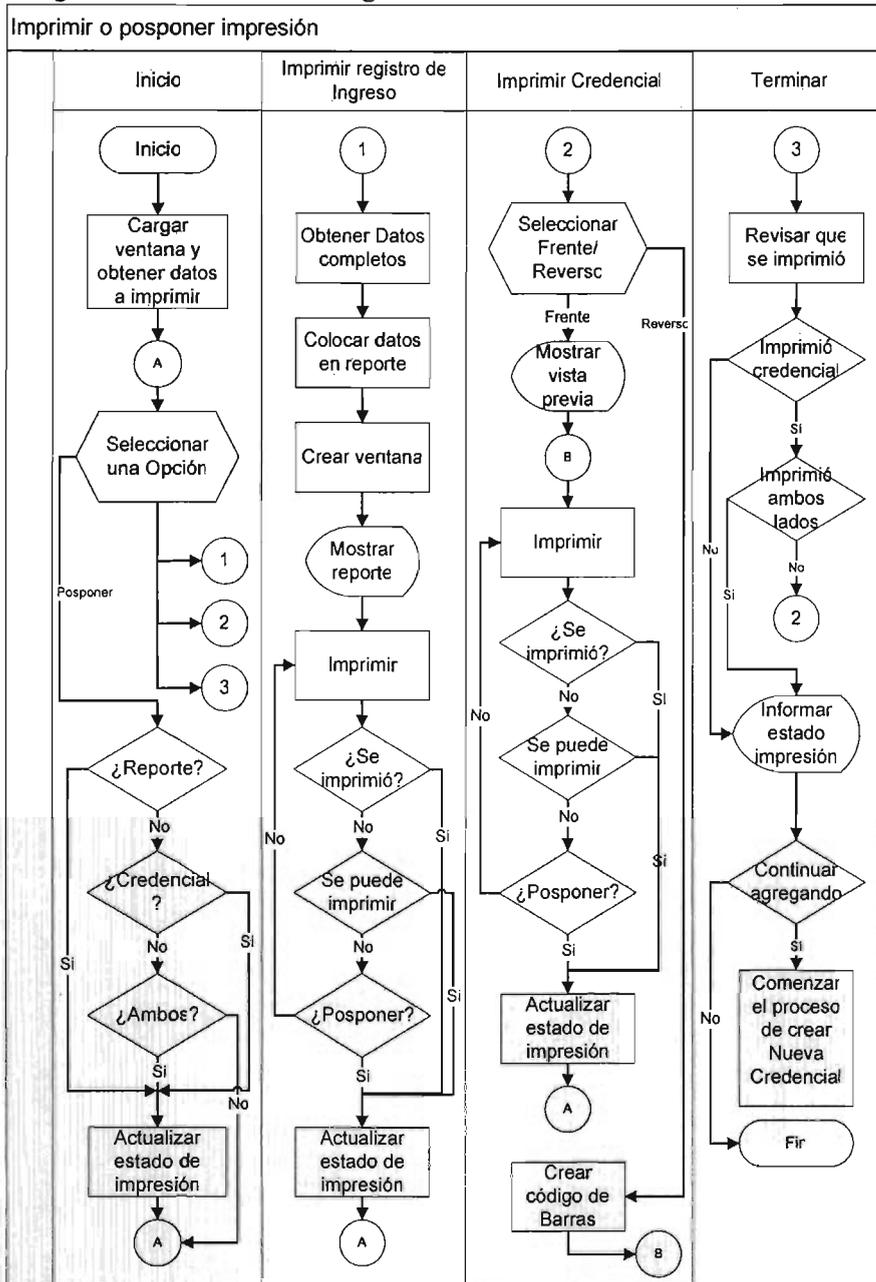


Figura 5.7 Diagrama de Impresión

Sí recordamos la figura 5.5 un paso antes de imprimir es el de almacenar los datos del becario, por eso en el primer proceso de la figura 5.7 se tiene que recuperar los datos, pero para optimizar memoria sólo se obtiene el ID del becario, ya que con él se tiene acceso al resto de la información y cuando se requiera el resto se obtiene de la base de datos interna.

Para imprimir se tienen básicamente dos opciones: imprimir el registro de ingreso y la credencial, si se elige el primero se carga la información completa del becario en un reporte, el cual tiene un formato predefinido, se muestra al usuario una vista previa de lo que se va a imprimir y se imprime. Si se tiene problemas con la impresora o cualquier causa que impida que se lleve a cabo la impresión del documento se puede posponer. Al terminar cualquiera de estas acciones el sistema actualiza la bandera de impresión del reporte con los siguientes valores:

`imprimioReporte = True` >> Se imprimió el reporte

`imprimioReporte = False` >> No se imprimió el reporte

Para la credencial se tienen dos banderas como esta, puesto que se imprimen los dos lados de la credencial y el camino a seguir es similar al que se lleva un reporte. Para este caso se tiene el formato de la figura 5.8:

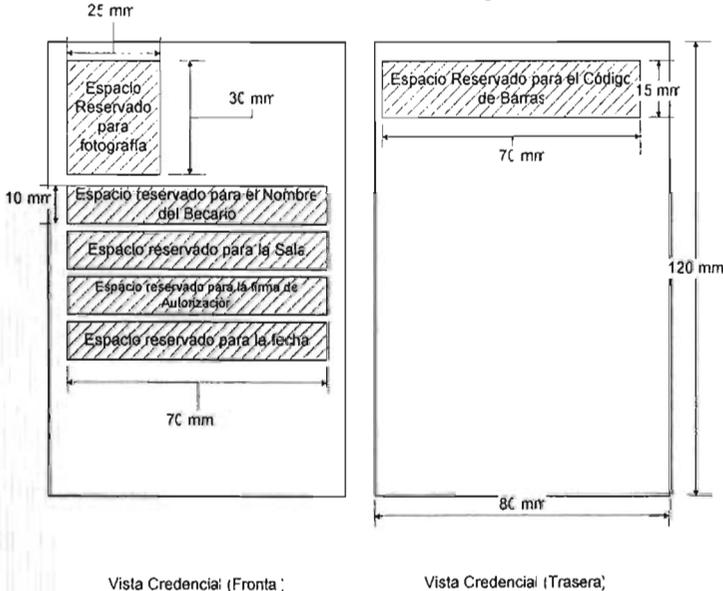


Figura 5.8 Formato de impresión de la credencial

En la figura 5.8 se observa el formato que se llena de forma automática por el sistema cuando se imprime, para la vista frontal se obtienen los datos y se colocan en el espacio reservado, cabe señalar que imprime respetando las medidas indicadas en la figura 5.8.

Para la parte trasera de la credencial se crea el código de barras, para ello se traduce el ID del Becario a unos y ceros según el estándar EAN 39, por ejemplo:

*GOMY810608001* a código EAN 39

G = 101010011011  
O = 110101101001  
M = 110110101001  
Y = 110010110101  
8 = 110100101101  
1 = 110100101011  
0 = 101001101101  
6 = 101100110101  
0 = 101001101101  
8 = 110100101101  
0 = 101001101101  
0 = 101001101101  
1 = 110100101011

Estos unos y ceros se traducen en las líneas que se observan en un código de barras, cómo lo vimos en el capítulo 4. Para el código de barras no se tiene una vista previa por que se crea en el instante que se manda imprimir, es decir, a la impresora se le pide que imprima las líneas que componen el código de barras y no se dibujan en pantalla. Cuando se imprime un lado de la credencial se hace una verificación cómo en el reporte, modificar las banderas para saber si se logró la impresión.

Cuando se pospone una impresión lo que se hace es poner el valor `False` a la bandera de impresión, para el caso de la credencial se modifican las dos banderas que se tienen. Así cuando se pide terminar el proceso o continuar, se verifica el estado de las banderas y se coloca un valor que representa el estado de la impresión para después tener acceso a la *Herramienta de Impresión*, los valores son los siguientes:

- 0 - Imprimió todo, credencial y registro de registro
- 1 - Imprimió sólo credencial
- 2 - Imprimió sólo registro de ingreso
- 3 - No imprimió

El esquema de la *Herramienta de Impresión* se muestra en la figura 5.9, esta herramienta tiene el objetivo de imprimir las credenciales o registros de ingreso que no se pudieron imprimir, además de poder hacer reposiciones de credenciales en caso de ser necesario. Se tienen dos formas de llamar a esta herramienta, una es de forma automática, cuando un becario es buscado. Al mostrar la información se determina el estado de impresión y se habilita el uso de la herramienta para que el usuario pueda imprimir lo que esta pendiente.

La otra forma es imprimir una reposición de credencial, lo cual habilita la impresión de la credencial sin importar el estado de impresión

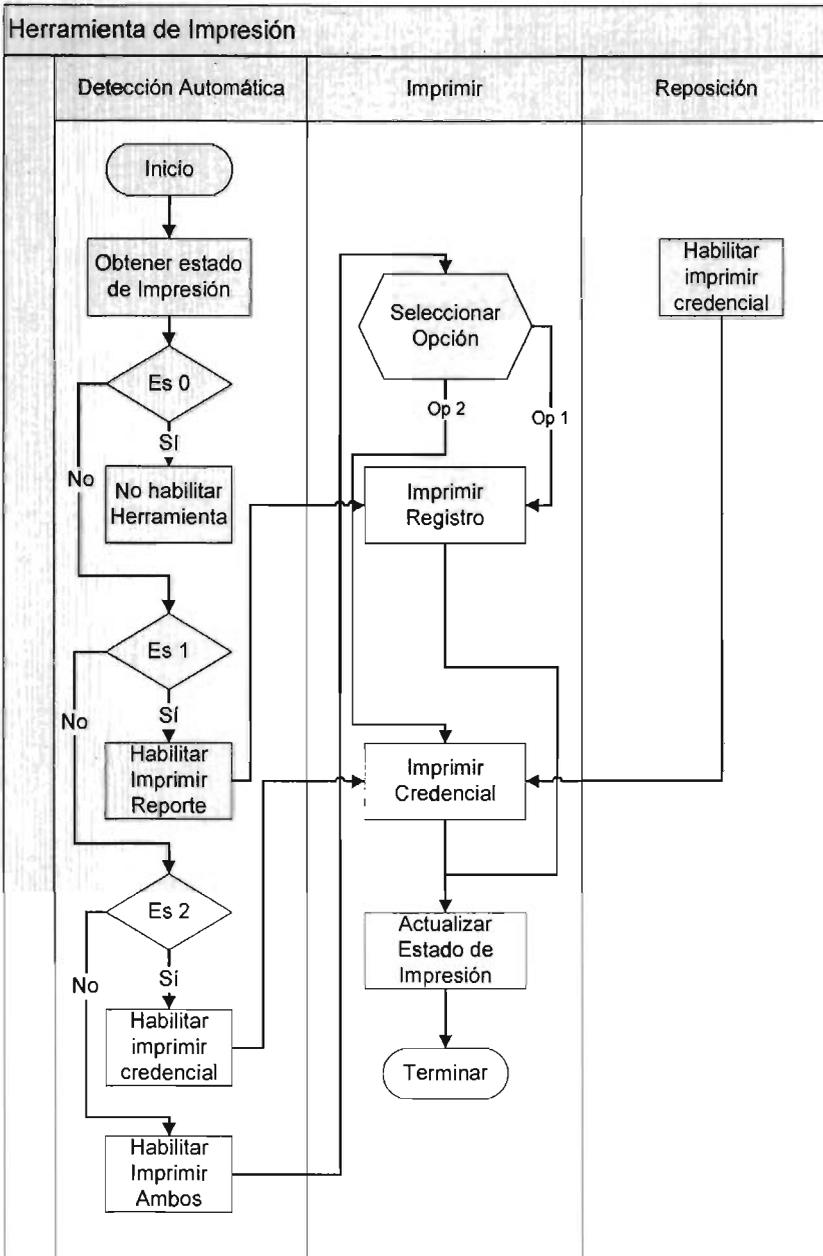


Figura 5.9 Diagrama de la Herramienta de Impresión

Los procesos de impresión de credencial y registro de ingreso son los mismos que se observaron en la figura 5.7, por ello no se muestran cómo un solo proceso.

Finalmente este programa de creación de credenciales es una herramienta que complementa el Sistema de Control de Becarios, para ver el producto final se puede ver el manual de usuario en el Apéndice A.

## **5.7 Sistema de Control de Becarios fase Red**

Para la parte de desarrollo en Red, de inicio se levanto un servidor en una maquina de forma local para realizar pruebas, esto se hizo a través de Tomcat y Mysql que lo describiremos un poco mas adelante; gracias al estudio de la base de datos para nuestro sistema (Sistema de Control de Becarios) se crearon las tablas correspondientes a dicha base de datos.

Para el programa de consulta en Red; ya que el personal del área de Becarios es el único que tendrá acceso a este sistema decidimos protegerlo por contraseña el funcionamiento del programa se describe en el diagrama siguiente (Ver figura 5.10)

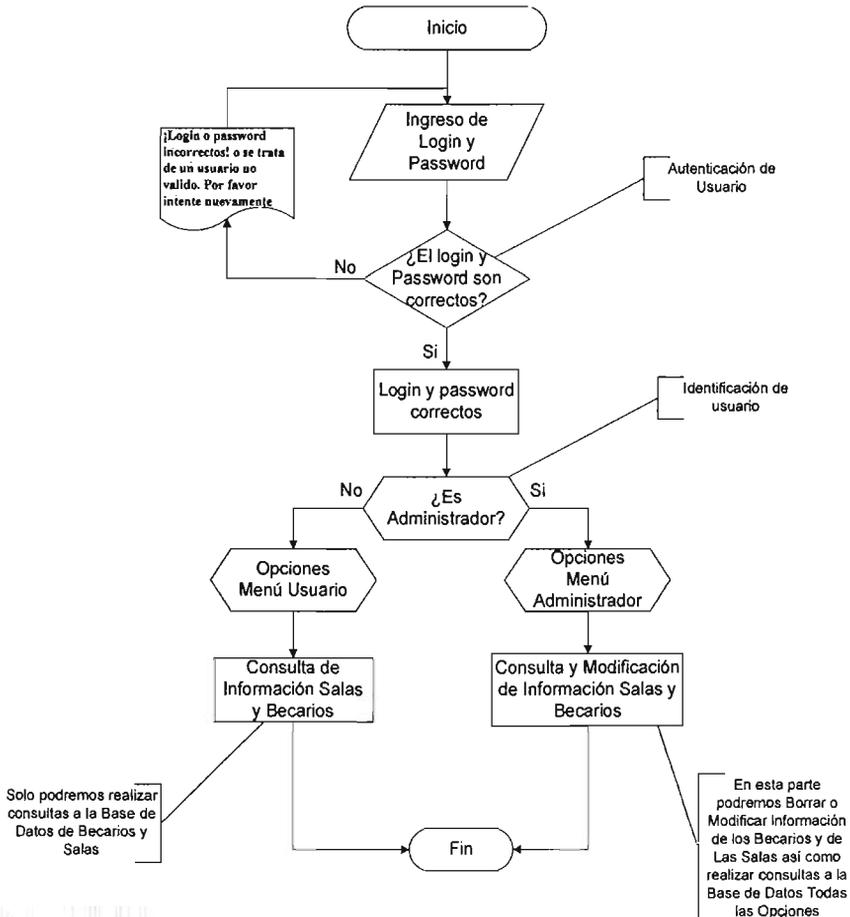


Figura 5.10. Diagrama de Flujo del Sistema en Red

De inicio el programa pide un nombre de usuario y una contraseña si el “login” y “password” son correctos el sistema nos dará acceso, de lo contrario tendremos un mensaje como el descrito en la figura 5.5 “Login y password incorrectos o se trata de un usuario no valido. Por favor intente nuevamente”. En esta parte del desarrollo hay unos archivos claves para esta función de autorizar un usuario estas son login.jsp y loginVprueba.jsp, en los cuales se importa un paquete de clases llamadas VerUsuarios mediante la instrucción que se encuentra en el archivo loginVprueba.jsp:

```
<%@ page language="java" import="java.util.*, VerUsuarios.*" %>
```

Una vez dentro del sistema y ya autorizados por el mismo las opciones a elegir serán las siguientes siempre y cuando el usuario se haya identificado como Administrador:

- Información de Becarios
  - En este apartado podremos ver todos los datos personales de los Becarios contenidos en la Base de Datos, esto se puede hacer gracias a una clase realizada en Java (a la cual llamaremos Servlet) en la cual se hace una conexión y una consulta a la Base de Datos. El servlet o la clase de java se llama `MostrarBecario.class` esta clase realiza una consulta `SELECT` a la tabla de Becarios, se muestra el Id del Becario y su nombre, al dar clic al nombre accedemos a los datos del Becario.
  
- Consulta Horario
  - En esta parte del menú de opciones tenemos la opción de Introducir el id del Becario para poder obtener su horario completo. La clase que realiza esta función es `Horario.class` esta clase esta en un paquete que se llama `horarios` el cual esta importado en la pagina JSP `ImprimirHorario.jsp` `<%@ page import="horarios.*"%>` La clase `Horario` devuelve una consulta a la base de datos `dgdc` de la tabla `horario_de_becario` dado el id del Becario, esta consulta devuelve una cadena la cual ocupamos para generar visualmente el horario.
  - Además se cuenta dentro de la página dinámica `ImprimirHorario.jsp` con la opción de imprimir el horario del becario si así se solicita.
  
- Modificar Datos de un Becario
  - En este modulo del sistema podremos modificar los datos personales de los Becarios (dando el id del Becario) contenidos en el sistema, las paginas dinámicas que se utilizan aquí son `frmModificarBec.jsp`, `ModificarB.jsp` el funcionamiento de la primera pagina jsp consiste en consultar la base de datos y mostrar los resultados de la búsqueda (datos personales del becario seleccionado) y mostrarlos en cajas de texto para que el usuario pueda modificar la información que desee; la segunda pagina muestra la información ya modificada y un mensaje el cual indica si la información se guardo correctamente o hubo un

error en la base de datos esta página utiliza un Update (Actualización) en la base de datos.

- Información de Salas
  - Podremos ver todas las salas registradas en la Base de Datos, esto gracias a una clase más hecha en Java (Servlet MostrarSala.class) la cual muestra el contenido de la Base de Datos de las Salas registradas, esta clase devuelve el resultado en una tabla HTML, la clase hace una consulta SELECT a la Base de Datos.
- Alta de una Sala
  - Aquí podremos registrar una nueva Sala, el Id de sala se muestra automáticamente ya que aquí se cuenta con una clase mas en java la cual genera automáticamente el id, el administrador solo agrega el nombre de la sala. La sala se agrega al sistema.
- Modificar Datos de una Sala
  - En esta parte del sistema podremos modificar la información de una sala cambiando el nombre de la misma si así se desea solo tecleando el id de la sala, esto es similar al modulo de Modificar Datos de un Becario, aquí las páginas cambian frmModificarSala.jsp y ModificarS.jsp.
- Agrega Horarios
  - Aquí podremos agregar id de horario al sistema el id de horario se da con base en las horas de entrada de los Becarios, esto nos sirve para identificar la hora de llegada y la hora de salida de los Becarios.
  - El id de horario se muestra automáticamente esto gracias a una clase que es similar al modulo de Alta de una Sala.
- Información Horarios
  - Se muestra la información contenida en la base de datos acerca de los id de horario ya mencionados anteriormente aquí se ocupa un servlet MostrarHora.class para mostrar la información.
- Manuales
  - Aquí se muestran todos los manuales que comprenden el Sistema de Control de becarios, los formatos son en HTML y en PDF,

esto con la versatilidad de que los usuarios que no cuenten con Acrobat Reader puedan ver los archivos desde el servidor y los que cuenten con el los puedan copiar a su computadora.

- **Lista de Becarios**
  - Este es un modulo en el cual generamos las listas por cada día de la semana (de Lunes a Domingo), donde podemos obtener una lista de asistencia de todos los Becarios que asisten en ese día. La pagina dinámica jsp se llama Listas.jsp para generar las listas tenemos un servlet para cada día de la semana. Se cuenta con la opción de imprimir las listas por cada día de la semana.
  
- **Sábana por Sala**
  - Aquí se genera la Sábana por cada Sala a seleccionar, tenemos la opción de generar la Sábana seleccionando primero la sala en una lista de opciones cabe señalar que para realizar la Sábana el programa consta de aproximadamente 1500 líneas de código y que se utilizan clases del sistema en red local esto se verá en el punto 5.8.2.5 de este capítulo.
  - En la selección de la Sábana por Sala se da la opción para imprimir dicha sábana
  
- **Sábana por Día**
  - Para esta opción se genera la Sábana de todas las Salas por día de Lunes a Domingo, seleccionando primero un día de la semana de la lista en seguida se muestra la Sábana se hace el uso de clases del sistema de red local esto se vera en el punto 5.8.2.6 de este capitulo.
  - Al igual que en sábana por día aquí también se da la opción para imprimir la Sábana por Día la cual se genera dinámicamente.
  
- **Usuarios del Sistema**
  - En esta parte del menú de opciones es donde el usuario del sistema tendrá la opción de cambiar su login y su password así como crear un nuevo usuario para que pueda acceder al sistema, en este modulo el usuario teclea su login y su password actuales y a continuación, puede cambiar estos parámetros por los nuevos, aquí se solicita la confirmación del password y si los passwords o contraseñas son las mismas entonces se actualiza en la base de

datos el login y el password nuevos del usuario, de lo contrario regresa al menú de opciones para volver a esta opción.

- Salir
  - En esta opción podremos salir del sistema y cerrar la sesión el Manejo de la sesión lo explicaremos en el siguiente apartado de este capítulo Seguridad en el Sistema de Red punto 5.7.1.

Cabe señalar que se realizaron dos sistemas; como se ha visto, en la dependencia se cuenta con equipos PC y Mac, por tanto los sistemas quedan de la siguiente forma:

1. Versión PC (Ver figura 5.11 a.)
2. Versión Mac. (Ver figura 5.11 b.)

Esto se tuvo que hacer ya que algunos Scripts de Java (Java Script) no son compatibles con las versiones de Mac, por lo que se tuvieron que modificar los menús.

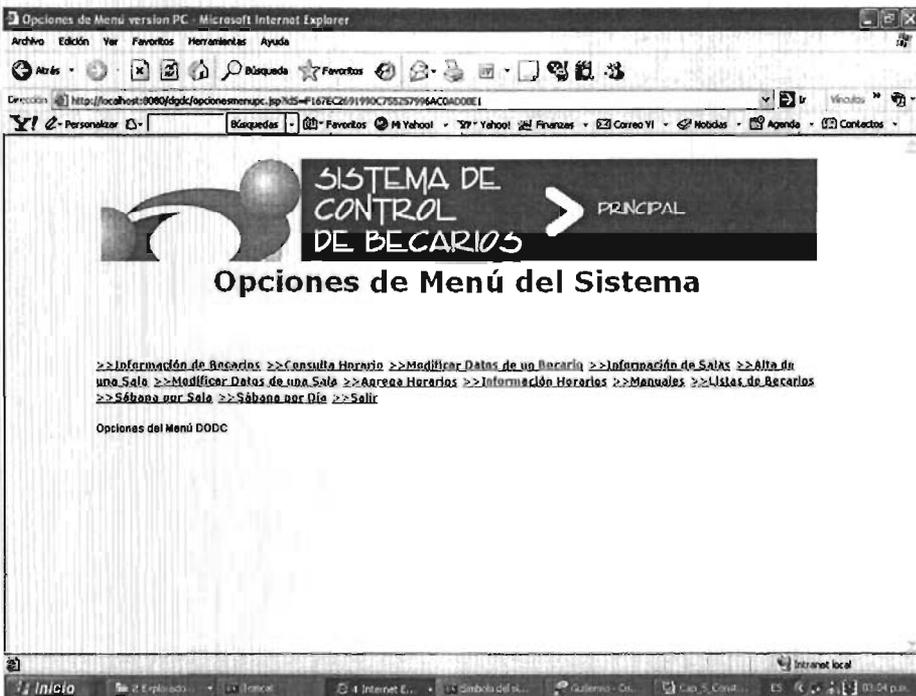


Figura 5.11 a. Ejemplo del sistema Versión PC

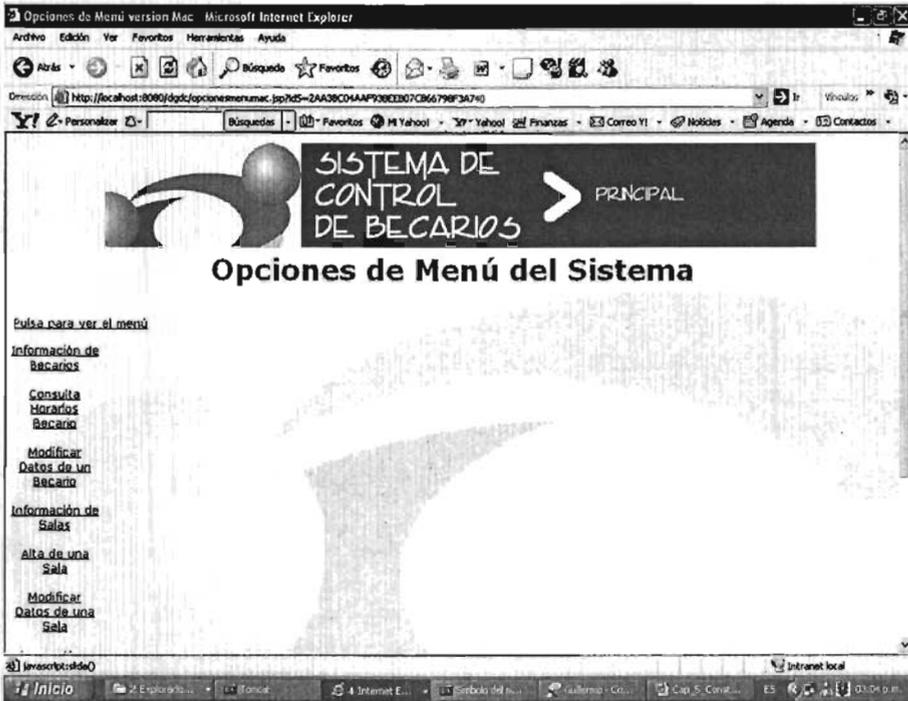


Figura 5.11 b. Ejemplo del sistema Versión Mac

Como se muestra en las figuras anteriores (Figura 5.11 a. y 5.11 b.) las versiones cambian de una a otra solamente por el menú, es decir, el funcionamiento del sistema es el mismo por tanto el diagrama de flujo de la figura 5.10 no cambia.

### 5.7.1 Seguridad en el Sistema de Red

Para la parte de seguridad en red, pudimos percatarnos de que el sistema en sus inicios no era seguro ya que al teclear nosotros toda la dirección de la pagina podíamos acceder a cualquier modulo del sistema independientemente de habernos logeado como administrador o no, es decir, que si cualquier persona copiara la dirección completa de la página podría atacar nuestro sistema; por tal motivo se opto por incluir dentro de las paginas dinámicas del sistema un par de identificadores “IdS” e “Id” ambos para seguridad, el funcionamiento del id de Sesión es el siguiente:

1. Al momento de autorizarse como administrador creamos una sesión a través de la siguiente instrucción:

```
HttpSession sesion = request.getSession();
```

En esta página donde se crea la sesión así como en todas las demás del sistema pondremos una instrucción:

```
<%@ page session="true" %> // el valor true indica que  
el valor de la sesión esta activo.
```

2. Cuando se inicia una sesión en la primera solicitud, el servidor emite un Id. de sesión único para el usuario. Para que persista el Id. de sesión lo incrustamos en las URL (o referencias) de cada pagina dinámica jsp.

```
String urlA = "opcionesmenupc.jsp?idS="+sesion.getId();  
out.println("<input type='hidden' name='id' value='"+  
sesion.getId()+ "'");
```

3. En cada pagina dinámica tendremos que comparar el Id del servidor con el id de la sesión este método se realiza para cada pagina tal como lo muestra el siguiente código:

```
<%  
String idS = request.getParameter("idS");  
String id = (String)session.getId();  
  
if(idS.equals(id))  
{  
%>  
    // continua la dinámica de la página  
    // Aquí va todo lo que tiene que hacer la página
```

4. Entonces cuando la sesión es rescrita en la dirección de otro navegador la sesión no es valida (id de Servidor y de sesión son diferentes); al no ser valida el sistema regresa automáticamente a logear al usuario.

```
<%  
  
    }  
    else  
    {  
        out.println("<H2>Sesion no valida</h2>");  
        response.sendRedirect("SesionNoValida.jsp");  
    }  
%>
```

Así podemos afirmar que nuestro sistema cumple con lineamientos de seguridad indispensables para su óptimo funcionamiento. El ID de seguridad se puede ver en el explorador Web y tiene la siguiente estructura:

<http://localhost:8080/dgdc/opcionesmenupc.jsp?idS=35FBFF665CD64980B0CAAFF9747ECB9E>

- localhost:8080: es el servidor de pruebas
  - dgdc : es la carpeta que contiene todas nuestras páginas dinámicas
  - opcionesmenupc.jsp : es una de las páginas donde están las opciones del sistema.
  - ?idS=35FBFF665CD...: es el id generado por el servidor el cual es diferente para cada sesión.
5. Cuando queremos salir de la sesión de forma adecuada elegimos la opción salir, aquí nos enviara a una pagina CerrarSesion.jsp, al seleccionar la opción Regresar al Sistema la sesión se mantiene activa gracias a que en cada referencia estamos llevando con cada pagina nuestro idS y podremos utilizar todas las opciones (Ver figura 5.12). Si lo que deseamos es salir del sistema utilizaremos la opción Salir del Sistema aquí aparecerá una pantalla emergente del navegador en el cual nos preguntará si realmente queremos salir; al aceptar la pregunta saldremos de la sesión y se cerrara el navegador este método se utiliza comúnmente en las paginas que manejan sesiones Bancarias este método lo utiliza Bancomer por mencionar uno de tantos. Con esto aseguramos que la sesión termine por completo. Ver figura 5.13

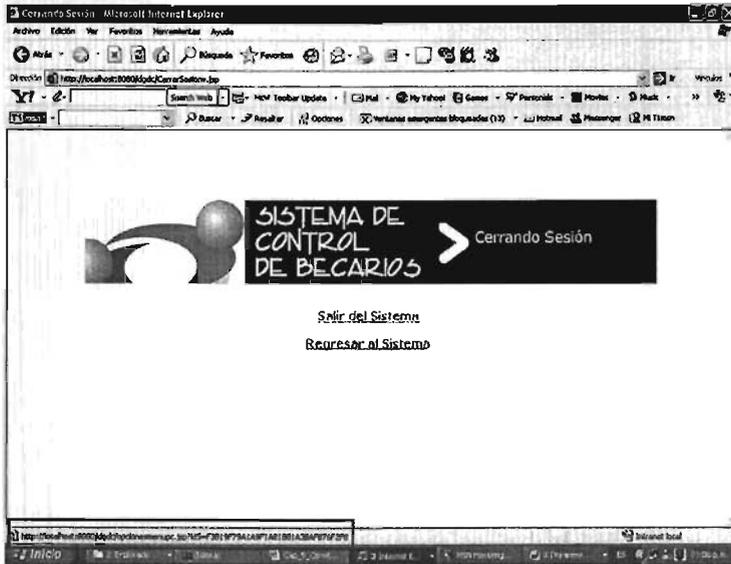


Figura 5.12 Regresando al sistema de Control se mantiene la Sesion



Figura 5.13 Cerrando Sesion Correctamente

## 5.7.2 Modulo para los Becarios

Dentro del Sistema también se creó un modulo que facilita a los Becarios la consulta de sus faltas, retardos, datos personales y la impresión de su horario.

Así tenemos que este modulo se encuentra en la dirección:

<http://localhost:8080/dgdc/iniciobec.jsp>

Lo que realiza este modulo es pedirle al Becario que ingrese su ID (id Becario), después manda una consulta hacia la Base de Datos la cual mostrara el resultado de la búsqueda devolviendo el nombre del Becario, su Id y las opciones que se quieren ver. Esto se muestra en la figura 5.14

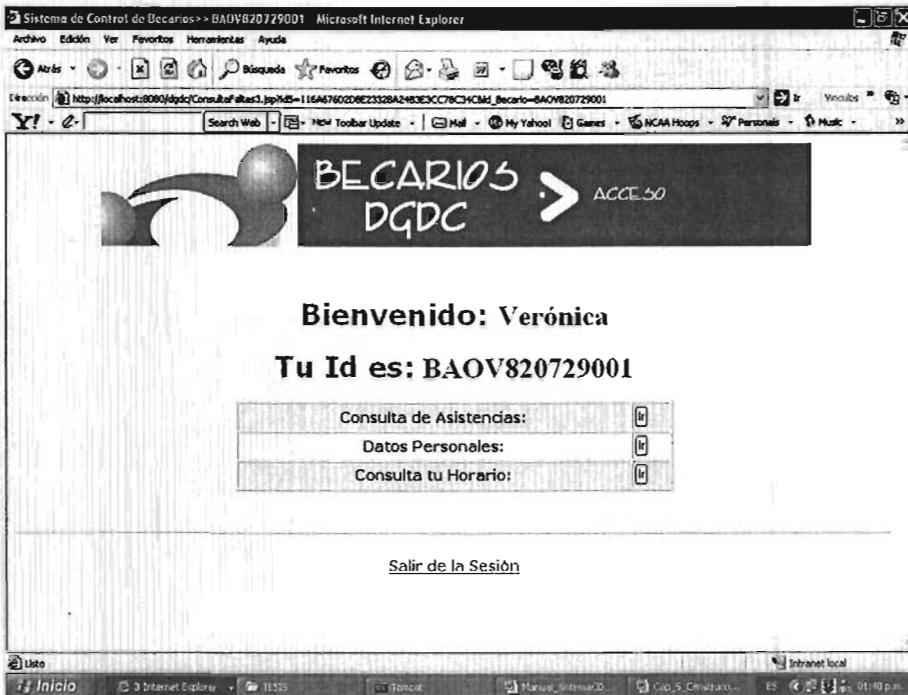


Figura 5.14 Modulo de Opciones disponibles para un Becario

- Consulta de Asistencias
  - Aquí el Becario podrá ver Sus faltas asistencias y retardos, esto gracias a una página dinámica en la cual se hace una consulta a la base de datos en la tabla asistencia.

- Datos Personales
  - El Becario podrá ver todos sus datos personales, para verificar si están correctos.
  
- Consulta tu Horario
  - El Becario podrá ver su horario completo e imprimirlo si así lo desea ya que se agrego esta opción al modulo del Becario.

### 5.7.3 Diagrama Conceptual del Sistema en Red

A continuación mostramos un mapa o diagrama (Figura 5.15) conceptual para nuestro sistema en Red, las páginas son todas en formato .jsp, cabe señalar que este diagrama conceptual es para el sistema en PC, el de Mac es exactamente el mismo.

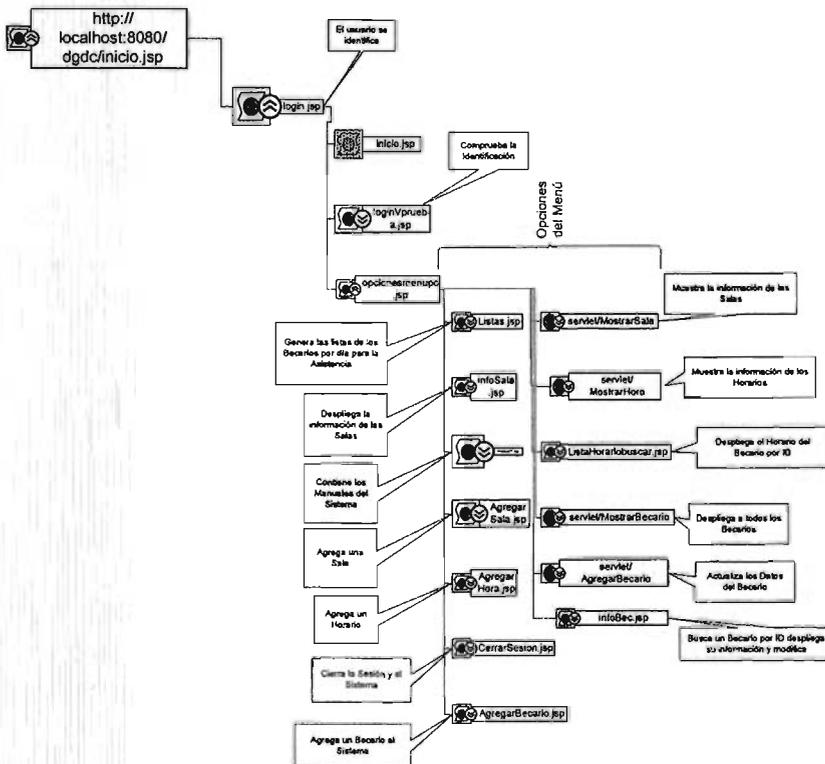


Figura 5.15 Diagrama del Sistema

## 5.8 Sistema de Control de Becarios Fase Red Local

Se le llama de Red Local porque aprovecha la posibilidad de compartir información dentro de la red interna del museo cómo ya se explicó al principio de este capítulo. Esta fase la podemos considerar con el corazón del sistema, puesto que aquí se genera la información que el usuario requiere a partir de los datos recopilados en la fase anterior y los que se obtienen dentro de esta fase.

Para poder comprender mejor el desarrollo recordemos el modelo que se planteo para resolver la problemática y sólo concentrándonos en esta fase tendría como resultado la figura 5.16:

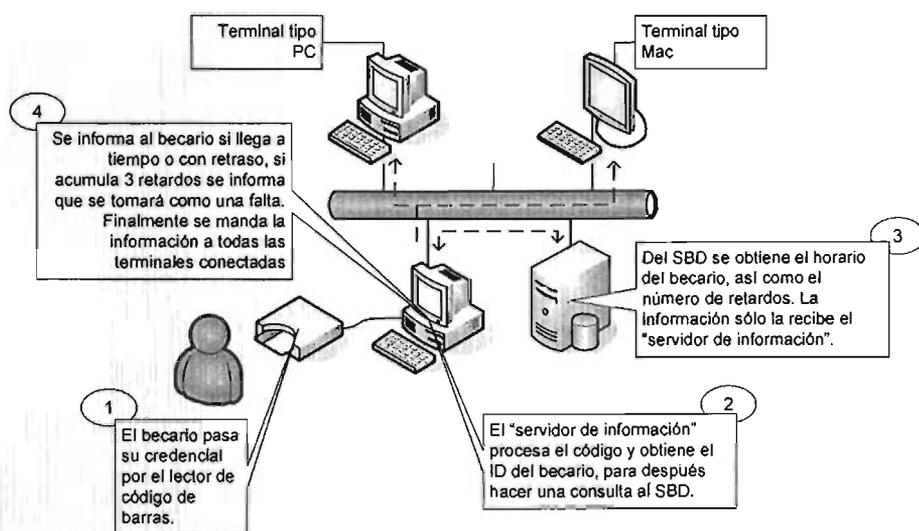


Figura 5.16. Modelo para la fase de red local

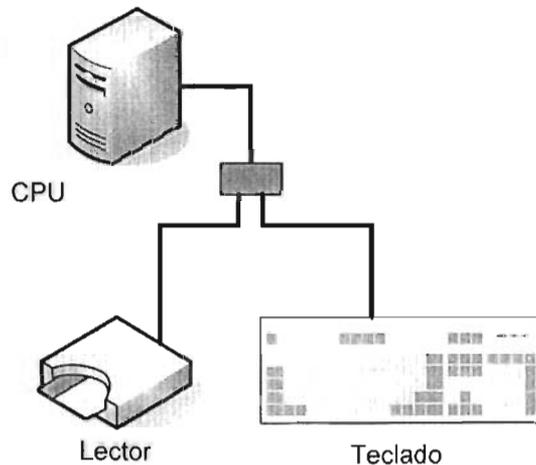
Del modelo podemos distinguir tres elementos: el Servidor de Información de Asistencia (SIA), el Servidor de Base de Datos (SBD) y las terminales. El segundo se tocará en el punto 5.8.1 de este capítulo, por lo tanto nos centraremos en el primero y tercero, los cuales veremos en los siguientes puntos.

## 5.8.1 Servidor de Información de Asistencia (SIA)

El SIA tiene como labor el recolectar datos sobre la asistencia de Becarios día con día. Cómo ya se ha mencionado esto se hace empleando un lector de códigos de barra, para saber el estado del becario se consulta la BD, después se analiza si esta a tiempo o tiene retardo, finalmente se hacen actualizaciones y se informa a las terminales. Pero veamos esto más a detalle.

### 5.8.1.1 Asistencia

El primer paso es obtener el identificador único del becario, para ello se utiliza el lector, el cual se encuentra conectado al mismo puerto del teclado, se utiliza un cable especial conocido como “y” para tener ambos dispositivos conectados al CPU, esto lo podemos ver en la figura 5.17:



5.17. Conexión del Lector al CPU

Esta configuración nos ayuda a la obtención del identificador del becario, ya que, podemos programar con los eventos asociados al teclado, sabemos que la información siempre vendrá de ese puerto. Con esto en mente lo siguiente es tener comunicación con el usuario, para ello se utiliza un GUI, esta interfaz esta compuesta básicamente de tres partes:

- a. Información del Sistema
- b. Datos del Becario

c. Resultado

En la figura 5.18 podemos ver cómo están organizados, si se desea ver la apariencia final se puede consultar el manual de usuario correspondiente que se encuentra en el apéndice.

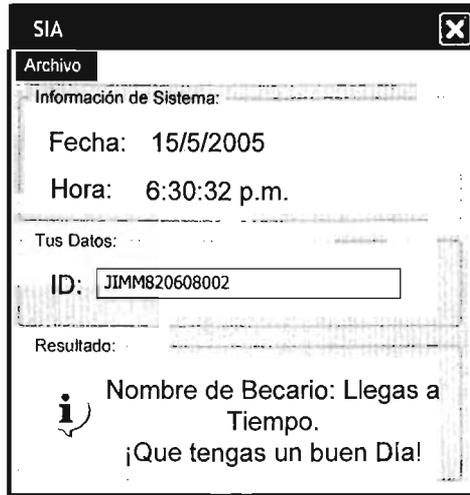


Figura 5.18. GUI de SIA

Cómo su nombre lo indica la Información de Sistema nos muestra la fecha y la hora del sistema, pero en el caso de la hora no es totalmente la del sistema, si depende del mismo pero tiene un defasamiento, si solicitamos un objeto del tipo fecha lo que obtenemos es la hora pero con la configuración de la máquina virtual, esto depende totalmente de la versión instalada. Esta variación depende del estándar que se ha empleado, puede ser el GMT o el UTC. Para evitar este problema se define que sea el estándar GMT -6 que es correspondiente para el Distrito Federal.

El siguiente elemento de la interfaz son los datos del becario, para esta parte se colocó una caja de texto previniendo una posible falla del lector, así se puede teclear el identificador, es decir, cuentan con las dos opciones trabajando de forma simultánea.

La última parte tiene fines informativos, se le informa al becario los resultados que básicamente pueden ser de tres tipos: a tiempo, retardo o error. Estos mensajes se verán más adelante.

Ya con esta idea que se tiene de la interfaz podemos continuar viendo lo que tiene que hacer este servidor. Cómo ya lo mencionamos al llegar el becario pasa su credencial por el lector, este último se encarga de interpretar el código y es enviado por el puerto a la computadora. Esta información llega al programa como un evento ocurrido sobre el teclado, es decir, como si el becario hubiera teclado el identificador, por lo tanto esta información será mandada a la caja de texto en donde se tiene implementado un “escucha”, este último se encarga de detectar cualquier cambio en la caja de texto, pero no todos los cambios nos interesan, es decir, cuando presionamos una tecla y la información llega a la caja de texto este “escucha” se activa, entonces sería un total desperdicio hacer algo por cada tecla que sea presionada, por ello solo nos concentraremos el contenido de toda la caja y en la longitud del mismo.

Sabemos que el identificador del becario es su RFC y una homoclave de tres dígitos, en total tendremos 13 caracteres. Entonces verificaremos la longitud de dicho contenido y además que esté conformada por letras concatenadas con caracteres numéricos, para ser más concretos en el siguiente orden: primero letras seguidos por los números. Esta última verificación se efectúa para evitar optimizar los recursos de red y uso de procesador, veamos dos casos para comprender mejor: el uso de códigos de barras es muy socorrido para identificar productos de cualquier tipo y en particular algunos emplean un código con una longitud de trece caracteres, por ejemplo las botellas de agua. Si por alguna razón algún becario pasa ese código por el lector el sistema lo podría tomar como bueno si sólo nos fijamos en su longitud, entonces desperdiciaríamos recursos. El segundo caso se presentaría con una posible falla del lector, al no tener el lector disponible el sistema puede continuar su funcionamiento, los becarios tendrían que teclear su identificador, si digitaran uno incorrecto una primera forma de detectarlo es ver como esta compuesto, claro que si esta escrito con las características antes mencionadas, pero aún así es incorrecto la única forma de detectar esta situación es verificando en la base.

En el estado normal de trabajo del sistema tendremos un identificador que se encuentre en la base, al obtener este dato el siguiente paso es registrar la hora exacta en que llegó el Becario, obviamente es necesaria para determinar si llega a tiempo, en la figura 5.19 vemos como están determinados los rangos.

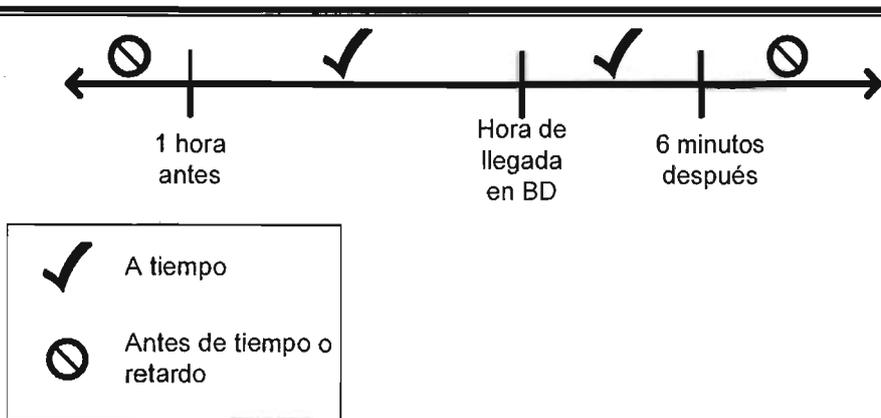


Figura 5.19. Rango tiempo para considerar un asistencia o retardo

Con la hora y el identificador procedemos a consultar a la BD, de la cual necesitamos la siguiente información:

- Nombre: este dato es empleado para mejorar la forma de comunicación con el usuario, en este caso el becario.
- Horario: con el horario podemos saber su hora de entrada, la cual necesitamos para saber si llegó a tiempo.
- Número de retardos: cuando llega tarde se tiene que aumentar un retardo, cuando se acumulan tres se aumenta en 1 el número de faltas.
- Número de faltas: cuando el número de faltas es igual a tres el becario esta dado de baja.

Cuando obtenemos la respuesta de la base y la analizamos podemos tener los siguientes casos, empezando para los casos negativos:

- El becario no esta registrado:* se entra en este caso si el identificador que se proporciono aún no esta registrado en la base o simplemente se usó el teclado y se digitó uno incorrecto.
- El becario no tiene horario:* aún no tiene un horario asignado en el sistema, estos dos primeros casos son factibles cuando un becario nuevo es registrado.
- No es un día de trabajo para el becario:* recordemos que los becarios pueden elegir (bajo ciertas condiciones) las horas y días que asisten al museo, por lo tanto si alguien pasa su credencial por el

- lector en un día que no le corresponde no puede ser admitido por es sistema.
- iv. *Baja temporal*: si el estado es igual a baja temporal no puede ser admitido, ya que, dicho estado implica que se encuentra deshabilitado para efectuar actividades como becario dentro del museo.
  - v. *Tiene permiso para ausentarse*: cuando un becario solicita permiso para no asistir al museo se coloca bajo dicho estado para no afectarlo con las faltas, pero si asiste tampoco se le puede considerar.
  - vi. *Error en la consulta*: puede ser que ocurra algún problema inesperado en el servidor, entonces obtendremos como respuesta un error SQL, el cual tiene que ser manejado.
  - vii. *Error general*: esto ocurre por problemas ajenos al sistema o la BD, puede ser una falla de red o algún otro error inesperado, estos problemas no son una respuesta directa del servidor pero la clase encargada de la consulta puede experimentarlos y los regresará como una respuesta, también tienen que ser manejados.
  - viii. *Llega retrasado*: después de comparar la hora de llegada del becario con la que se tiene en su horario se detectó que llegó tarde.
  - ix. *Llega antes de tiempo*: una política con respecto a las asistencias es que si el becario llega con mucho tiempo de anticipación no se puede registrar, esto es más factible en los horarios vespertinos.
  - x. *El becario llega a tiempo*: cuando el becario llega dentro del intervalo aceptado es registrada su asistencia.

En este momento no veremos como crear el horario de un becario, por el momento solo consideraremos que podemos extraer los datos que necesitamos, como la hora de entrada, los horarios se verán en el punto 5.8.2.4.1

El punto *i* y *ii* son fáciles determinar, si el becario aún no esta registrado, no tiene horario o simplemente no existe, cuando se efectuó la consulta no generará ningún error en el manejador, pero obtendremos como respuesta un *Empty set*. Al tener esta respuesta por ende sabemos que no esta en la base o no tiene horario. Con el punto *iii* pasa algo similar, aunque existe una gran diferencia, si obtendremos datos pero el campo correspondiente al horario no tendrá valor.

Como lo veremos en el punto referente al horario la cabecera del mismo tiene información con respecto a su estado, por ejemplo normal, baja temporal y

permiso. El campo correspondiente en la base de datos contiene una cadena, para ver el estado del becario leemos la cabecera (que son las seis primeras letras) y podemos saber si esta en el caso *iv* y *v*.

Los casos *vi* y *vii* son posibles situaciones que se presentan y deben ser consideradas. Para la clase estos en realidad no son respuestas directas del servidor, se tratan de errores que impiden que se continuara con la ejecución del programa, se les conoce como excepciones, cuando se produce una de ellas pueden ser “atrapadas” para no terminar con la ejecución. Por ejemplo, una excepción SQL (caso *vi*) en java se le conoce como *SQLException*; cuando ocurra algún problema y esté contemplado será atrapado y podemos solicitar que la clase regrese un valor, en este caso de error.

Los casos *ix* y *x* se determinan comparando la hora de llegada en ese momento (hora de sistema o *hA*) y la que se tiene registrada en la base dentro de su horario (*hB*). Con estas dos horas hacemos una comparación directa de la cual podemos obtener tres resultados:

- $hA < hB$  :: el becario llega antes de su hora de entrada
- $hA = hB$  :: el becario llega exactamente a su hora de entrada
- $hA > hB$  :: el becario llega después de su hora de entrada

Si llega exactamente entonces estamos de forma directa en el caso *x*, es decir, que esta a tiempo o puntual para ser más precisos. Pero si recordamos la figura 5.19, podemos ver que aún puede no tener retraso o estar antes de tiempo, para este último tenemos que fijarnos en la diferencia de horas, si no excede a una hora puede ser registrado pero se considera como a tiempo. Cuando se presenta con una anticipación mayor a una hora no se registra y entramos en el caso *ix*. Estas determinaciones obedecen a que los becarios están registrados y serán necesitados en el horario que ellos solicitaron, no antes y no después, por lo tanto tienen que cumplir con él en la medida de lo posible.

Cuando el becario llega a tiempo se aumenta en uno el número de asistencias, esto es con fines estadísticos.

Al llegar después de la hora aún existe la posibilidad de estar a tiempo, cuando estamos en este caso el sistema puede tener más tareas que ejecutar, es decir, si llega antes de los seis minutos aún se considera dentro del caso *x*. Si esto no ocurre la situación cambia, el becario tendrá un retardo más y tenemos que considerar los siguientes escenarios:

Escenario 1: El becario ya tiene un retardo, este caso aún no es grave, simplemente aumentamos el número de retardos, tendría dos, actualizamos la base e informamos al becario el total de faltas acumuladas.

Escenario 2: El becario acumula dos retardos y máximo una falta, sería un caso de alerta amarilla, al llegar tarde en este día se tiene que aumentar el número de retardos que serían tres en total, por lo tanto se considera como una falta, si tiene cero o una falta, aún no hay un problema grave puesto que máximo tendría dos faltas, pero tendrá que tener mucho cuidado con su asistencia. También se informa al becario el total de retardos y faltas, además de una breve explicación del caso en que se encuentra.

Escenario 3: El becario acumula dos retardos y dos faltas, es un caso grave o alerta roja, al llegar retrasado se aumenta en uno el número de retardos y tenemos tres, por lo tanto, se aumenta el número de faltas que llegaría a tres, que es el número de faltas necesario para ser considerado como baja automática del programa de becarios. Este caso para el sistema es baja temporal, ya que, aún puede ser retraso justificable y en caso de no serlo aún puede ser considerado su caso por parte de los encargados del área, se le puede dar una segunda oportunidad.

En todos los casos se actualiza la base y se informa al becario, puede ser un simple saludo de bienvenida o un informe de sus retardos y faltas. Complementado con todos los casos que se acaban de mencionar son las tareas que se ejecutan para tomar asistencia.

### **5.8.1.2 Servicio de Envío de Información.**

Cómo ya lo vimos uno de los servicios que ofrece este sistema es el de tener información en tiempo real, esto es, cuando un becario llega se puede informar a las terminales. Esto se logra empleando un esquema parecido al de suscripción, que solo se envíe la información a quién la solicite, es decir, un servicio multicast. Con esta idea se pretende hacer más eficiente el servicio de información, ya que no es necesario que el servidor recuerde o tenga almacenado en un archivo las direcciones IP de las máquinas a las que tiene que enviar la información, es más fácil tener la dirección del servidor en las

terminales. Así cada programa que solicite la información deberá conectarse y el servidor enviará información a una sola dirección IP, una reservada para envío multicast. Como lo podemos observar en la figura 5.20.

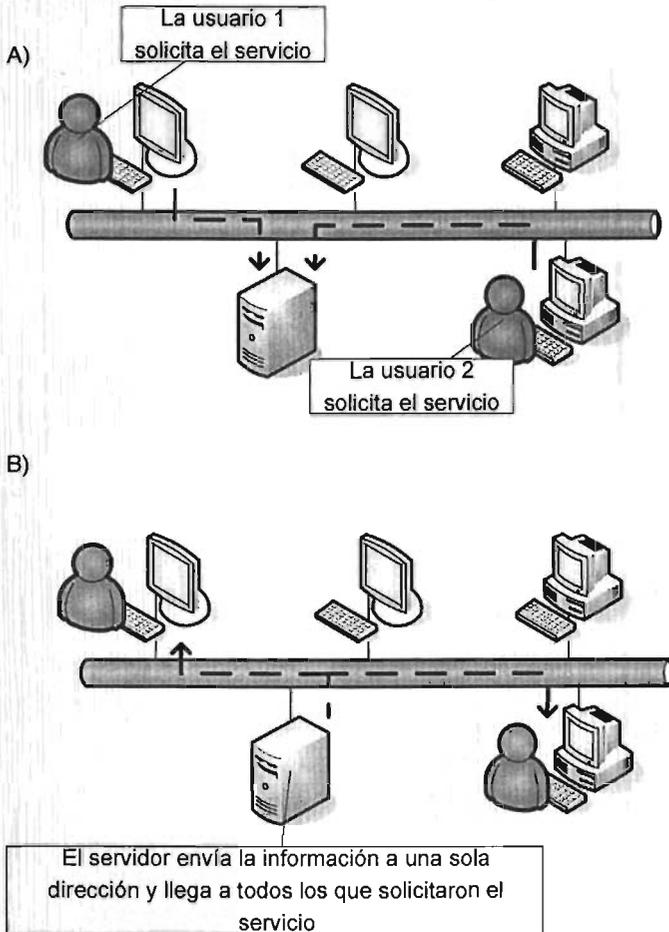


Figura 5.20. Envío de Información usando Multicast

Las direcciones para este tipo de envío se encuentran en el rango de 224.0.0.0 a 239.255.255.255, en particular las que empiezan con 239 esta reservada para intranets, se empleará una de estas direcciones para la transmisión.

El servidor SIA será el encargado de proporcionar el servicio, obviamente tendrá que ser un proceso más ejecutándose de forma paralela, recordemos que ya tenemos una aplicación multiproceso, esta posibilidad es permitida en java. En particular es conveniente que este subproceso sea del tipo "daemon",

ya que estos no evitan que el programa termine hasta que ellos terminen. Este subproceso se encargará de escuchar peticiones de otros programas, pero también de enviar la información.

En total se tendrán tres procesos ejecutándose de forma simultánea, en la figura 5.21 se ejemplifica de forma gráfica, se trata del proceso principal, el encargado de escuchar los eventos del teclado que a su vez es el encargado de disparar los eventos para tomar asistencia y el servicio de mensajería. Todos se ejecutan en paralelo y terminan cuando el usuario cierra la aplicación.

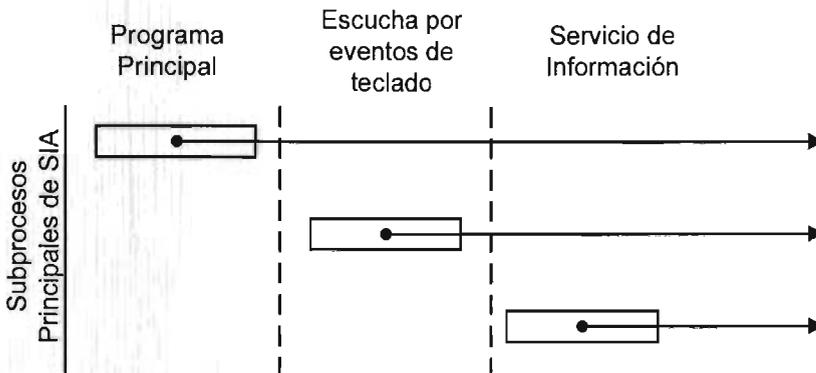


Figura 5.21. Procesos Principales de SIA

El subproceso de servicio de información a su vez esta conformado por tres subprocesos:

- *subp\_SI\_1*, se encarga de escuchar las peticiones de servicio.
- *subp\_SI\_2*, se encarga de escuchar y recibir la información que el subproceso “escucha por eventos de teclado” solicita enviar a los suscriptores.
- *subp\_SI\_3*, el encargado de preparar y enviar la información.

El sistema tiene como una constante el puerto por el cuál se escuchará, tanto las terminales como el servidor emplean el mismo puerto para escuchar información. Cuando un usuario pide el servicio, este es solicitado al puerto X que es el puerto que *subp\_SI\_1* escucha (figura 5.22).

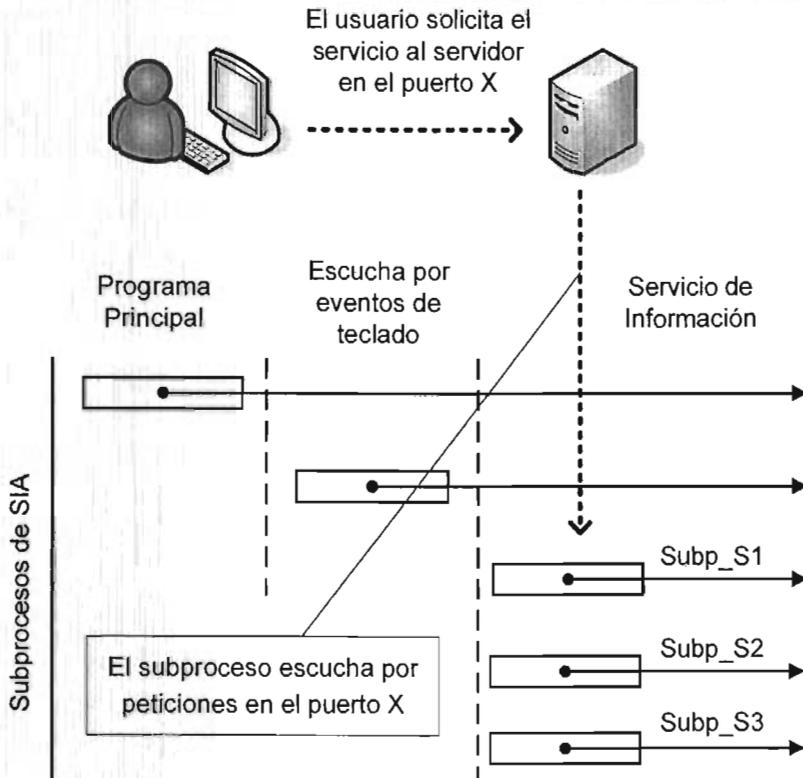


Figura 5.22. Subproceso dedicado a escuchar peticiones

Cuando un becario pasa su credencial por el lector “el escucha de eventos de teclado” solicita que se procese y se obtenga un resultado, el becario está a tiempo o con retardo, cuando se llega a una conclusión se envía información al *subp\_SI\_2*(figura 5.23).

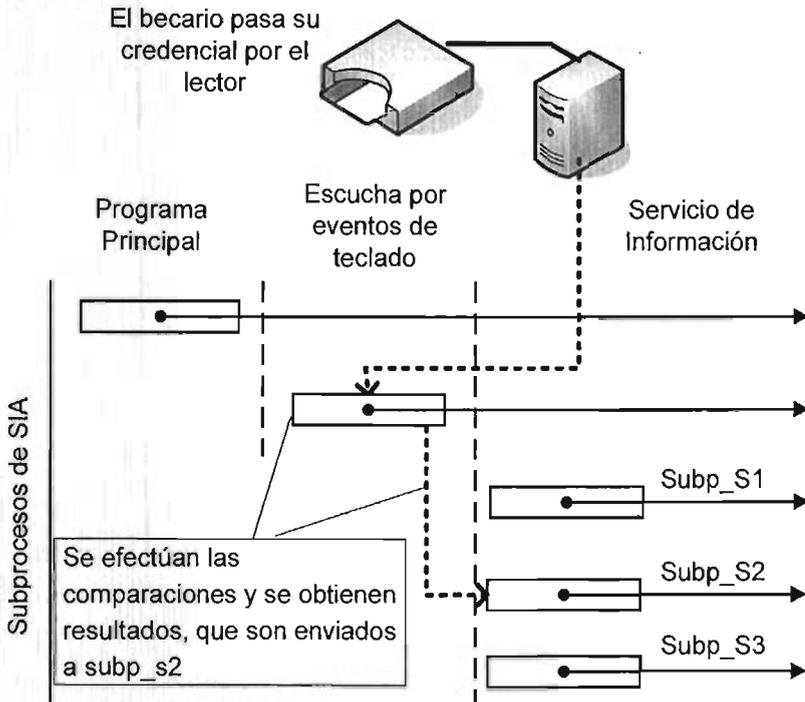


Figura 5.23. Subproceso que escucha por resultados

Finalmente se prepara para ser enviada la información a los suscriptores, para ello se genera un paquete compuesto por: el mensaje, la longitud del mensaje, la dirección a la cual debe entregarse y el puerto. En el siguiente orden:



Figura 5.24. Paquete a enviar

Obviamente será complementada con más información que requiera el protocolo, como puede ser dirección de origen, puerto origen, checksum, etc. Se aprovechan las herramientas que nos proporciona Java, es decir, se emplean sockets para comunicación en red, muchas de las características son transparentes para el programador, pero sabemos que se emplea el protocolo UDP (User Datagram Protocol) y el IP. Suponiendo que enviamos el mensaje “Juan>>>Hola” y analizamos la red, podríamos ver lo que se muestra en la figura 5.25.

```

Frame 113 (53 bytes on wire (53 bytes captured)
Arrival Time: May 23, 2005 18:43:11.721714000
Time delta from previous packet: 0.004198000 seconds
Time since reference or first frame: 5.294594000 seconds
Frame Number: 113
Packet Length: 53 bytes
Capture Length: 53 bytes
Ethernet II, Src: 00:11:11:29:24:72, Dst: 01:00:5e:00:00:01
Destination: 01:00:5e:00:00:01 (01:00:5e:00:00:01)
Source: 00:11:11:29:24:72 (Intel_29:24:72)
Type: IP (0x0800)
Internet Protocol, Src Addr: 132.248.66.221 (132.248.66.221), Dst Addr: 239.0.0.1 (239.0.0.1)
Version: 4
Header length: 20 bytes
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
Total Length: 39
Identification: 0x7220 (29216)
Flags: 0x00
Fragment offset: 0
Time to live: 1
Protocol: UDP (0x11)
Header checksum: 0x90cf (correct)
Source: 132.248.66.221 (132.248.66.221)
Destination: 239.0.0.1 (239.0.0.1)
User Datagram Protocol, Src Port: 5554 (5554), Dst Port: 5555 (5555)
source port: 5554 (5554)
destination port: 5555 (5555)
Length: 19
Checksum: 0x2495 (correct)
Data (11 bytes)
.....
0000 01 00 5e 00 00 01 00 11 11 29 24 72 08 00 45 00  .A.....)$.E.
0010 00 27 72 20 00 00 01 11 90 cf 84 f8 42 dd ef 00  .P.....B...
0020 00 01 15 b2 15 b3 00 13 24 95 4a 75 61 6e 3e 3e  .....$.Juan>>
0030 3e 68 6f 6c 61                                     :hola

```

Figura 5.25. Análisis del un paquete enviado

El contenido del mensaje es el identificador del becario, separador, nombre, separador y un valor de asistencia (a tiempo o con retraso).

Los caracteres sombreados en negro de la figura 5.25 representan la forma final del paquete que es transmitido por *subp\_SI\_3*. Así entonces la información se transmite a todos los suscriptores del servicio (figura 5.26).

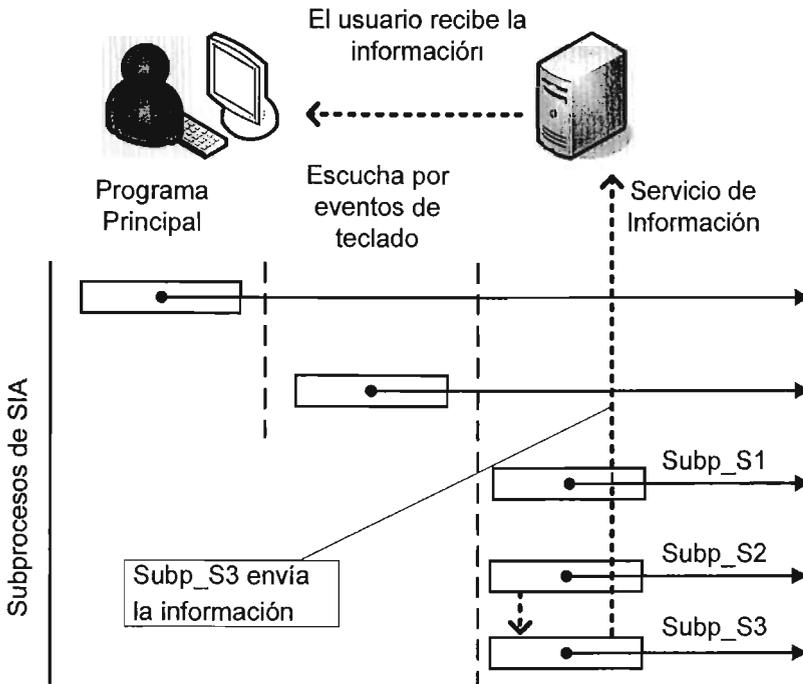


Figura 5.26. Envío de Información

Una vez que la información es enviada los suscriptores recibirán la información emitida.

### 5.8.1.3 Recepción de Información

Cuando las terminales reciben los datos enviados desde SIA tienen dos opciones de mostrar la información, una es en modo texto y la segunda en modo gráfico. La forma de recibir la información es la misma para ambos casos. Para ello también se emplean subprocesos, se tiene la funcionalidad como lo veremos en el punto correspondiente a las terminales y un subproceso tipo daemon que es activado por el usuario, también determina el modo de mostrar la información, es decir, si será modo texto o gráfico.

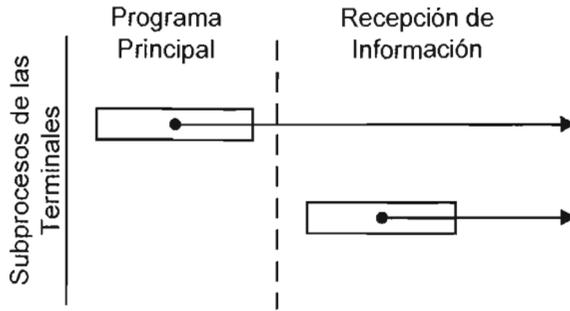


Figura 5.27. Subprocesos en las terminales

Para recibir la información se usa una técnica similar a la del servidor, pero esta vez no se escucha por peticiones, lo que se recibe es información y esta es enviada a otro subproceso que la muestra según lo solicitó el usuario, es decir, *subp\_s1* y *subp\_s2* se convierte en un solo subproceso (*subp\_t1*) y *subp\_s3* envía la información a la interfaz de usuario (*subp\_t2*). Gráficamente lo podemos ver en la figura 5.28.

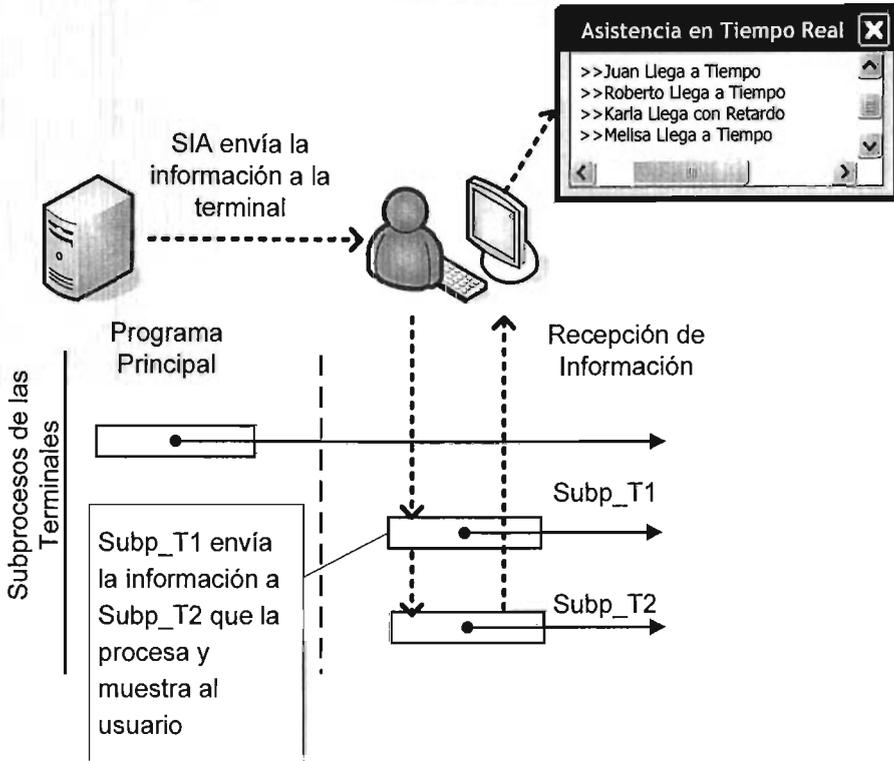


Figura 5.28. Subprocesos escucha de información.

El subproceso *subp\_t1* se encarga de escuchar por información entrante en el puerto X, cuando esta llega es preprocesada, es decir, se extraen del paquete y se separan los datos del mensaje. Los datos son enviados a *subp\_t2* que es el encargado de mostrar información al usuario. Cuando es en modo texto se imprime el nombre y si llegó a tiempo o no, para el modo gráfico se aprovecha la sabana por día (que se verá más adelante) en esta interfaz se coloca un icono al becario que corresponda.

#### 5.8.1.4 Parar el Servicio de Información

Cuando el servicio de información no puede continuar se debe a dos razones, la primera es: el usuario de una Terminal cierra la venta o solicita terminar el servicio, en este caso las implicaciones son locales, es decir, al parar el servicio en una Terminal se envía una señal de desconexión y ya no recibirá información pero el servidor continúa funcionando y el resto de suscriptores contará con el servicio.

Cuando se termina la aplicación (SIA) automáticamente se interrumpe el servicio, esta es la segunda razón, y afecta a todos los suscriptores, obviamente todos dejarán de recibir información. Para que las terminales no estén ejecutando un subproceso que ya no tiene sentido, una tarea que se debe ejecutar al cerrar SIA es enviar un mensaje final a los suscriptores informando sobre la terminación del servicio y así poder liberar recursos.

La última tarea que ejecuta SIA es el verificar que becarios no asistieron el día de hoy, esto se hace sólo cuando se termina la aplicación y está cercana la hora de cerrar el museo. La última que se puede dar de alta son las cinco de la tarde, por lo tanto se puede dar como “falta” a los Becarios que no se registraron antes de esa hora. Así se terminan los aspectos relacionados a la asistencia y el envío de información.

### **5.8.2 Terminales**

Las terminales tienen más funciones que solo mostrar la información que el Servidor de Información de Asistencia les envía. Con estas se tiene acceso a los datos de la base y también la posibilidad de modificar dichos datos. Por lo tanto tendrán mayor variedad de opciones y un GUI más elaborada que el SI.

Debido a la variedad de información que se maneja y para que el usuario tenga mayor facilidad, además de un acceso rápido a la información que requiere, se crearon cinco secciones y un menú de principal, el esquema que podemos ver en la figura 5.29 es la estructura general de una Terminal del sistema.

Por seguridad, es necesario ingresar un nombre de usuario (login) y contraseña (password); además cómo se puede ver se restringe el número de intentos a tres. Cabe señalar que un usuario puede estar registrado en el servidor, sólo en la máquina en donde está instalada la Terminal o ambos, se tiene esta opción considerando las siguientes cuatro posibilidades de un fallo: una caída del servidor, un problema con la conexión de red o que se cambiará la dirección IP del servidor en que se tiene la base de datos y que no se tenga configurado el conector MySQL. Así se tendrá acceso al sistema y probar las herramientas incluidas, de las cuales se hablará más adelante cuando se hable de la opción *Inicio*.

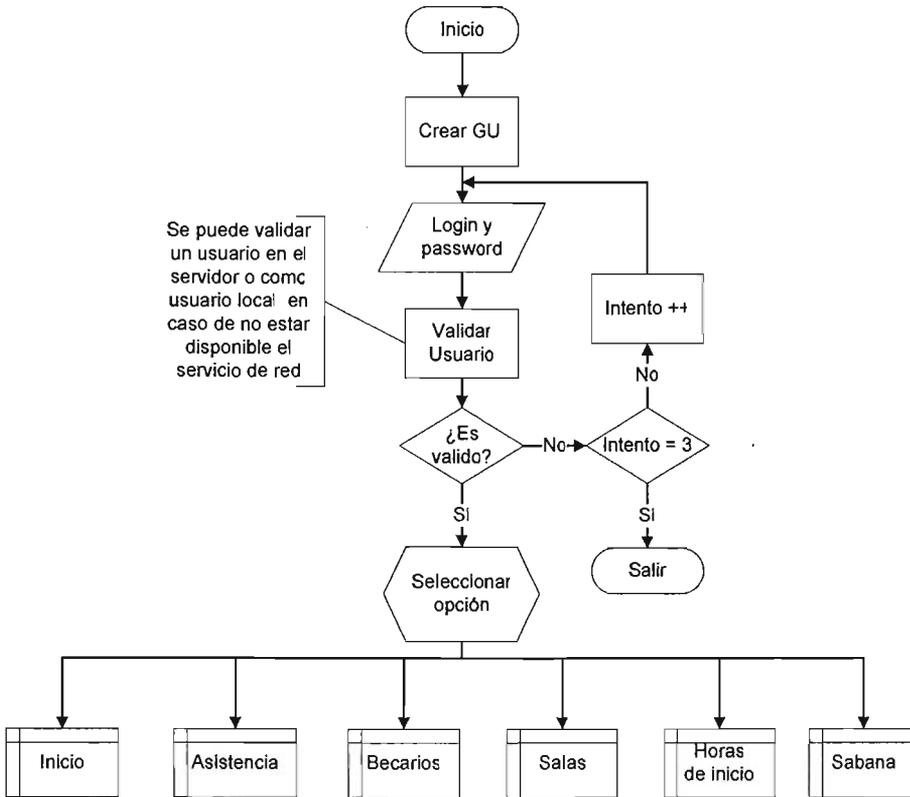


Figura 5.29. Esquema general Sistema de Control de Becarios fase red local

Una vez validado un usuario se tiene acceso al sistema, cómo se mencionó antes se tienen 6 opciones, el siguiente paso importante a tratar es el tener una *GUI* que cumpla con el propósito del esquema de la figura 5.29: mayor facilidad y un acceso rápido a la información. Para ello se propone la *GUI* que se muestra en la figura 5.30. En la cual se puede observar que las opciones y por ende sus funciones están separadas por “fichas”, el objetivo es que al hacer clic en una de estas fichas se muestren las funciones principales o de uso frecuente.

Otra característica importante es que se tiene un “área de salida de información” común a todas las fichas, es decir, la información resultado de una búsqueda o resultado de una operación se mostrará en dicha área, aun que no toda la información se mostrará en dicha área. Para que se tenga una mejor

presentación y resulte más amigable al usuario se colocará la información con formato HTML.

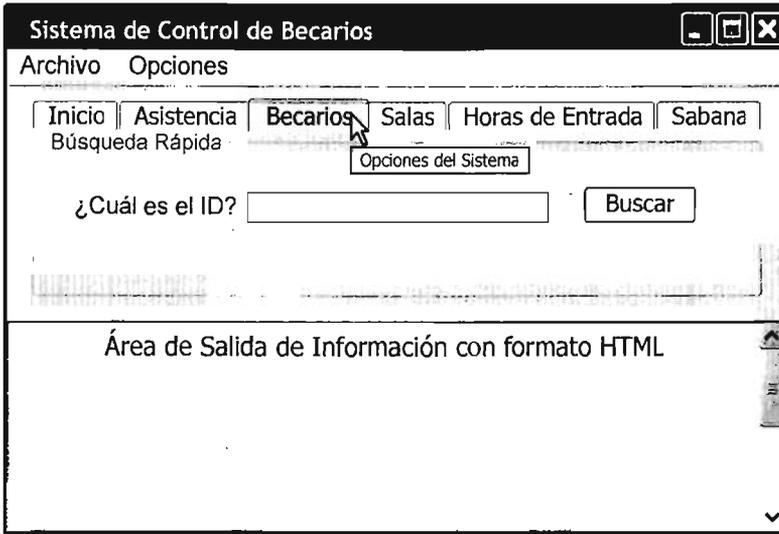


Figura 5.30. GUI de una Terminal

En este caso no se muestra su apariencia final puesto que se empleará Java para crearla, lo anterior se explica si abundamos más en lo mencionado acerca de la portabilidad de Java.

Podemos mencionar que para crear una *GUI* con Java se tienen los *componentes de la GUI de Swing* del paquete `javax.swing`, la mayoría de estos están escritos completamente en Java. Además se tiene los componentes de *GUI* del paquete `java.awt` del AWT (*Abstract Windows Toolkit*), los cuales están enlazados directamente con las herramientas de interfaz gráfica de usuario de la plataforma local. Cuando se ejecuta un programa con una *GUI* del AWT en distintas plataformas los componentes se mostrarán de manera distinta, asumirá la apariencia de cada sistema operativo.

Por lo tanto, la apariencia del sistema cambiará dependiendo del tipo de máquina en el que este instalado, PC o Macintosh. Para conocer la apariencia en estos dos sistemas en el Apéndice 2 se muestra el manual de usuario.

A continuación hablaremos sobre las opciones que se muestran en la figura 5.29, que para el usuario se presentan como fichas.

### 5.8.2.1 Ficha “Inicio”

En la figura 5.31 se muestra la estructura de las funciones que se tienen en esta ficha

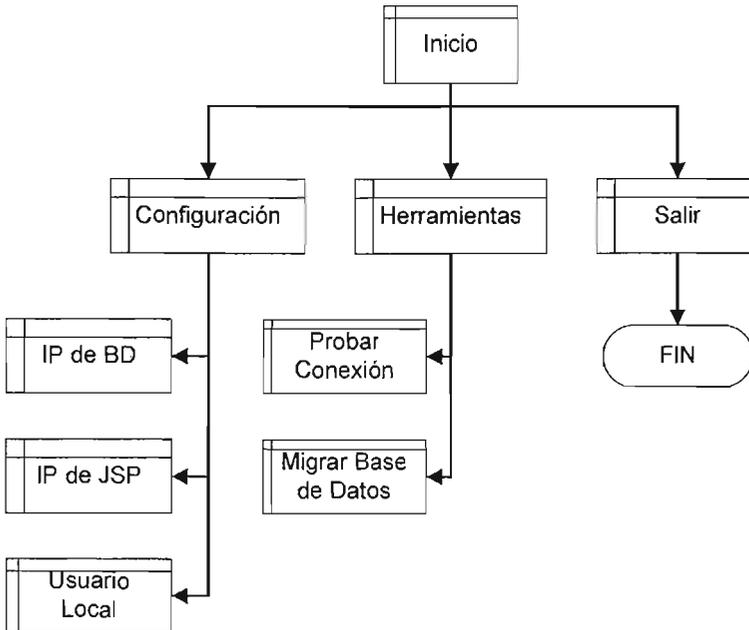


Figura 5.31. Estructura de la Ficha *Inicio*

Como podemos observar en la figura 5.31 son tres las opciones que tenemos en esta ficha. Aunque no son de uso frecuente si es necesario tener acceso fácil a ellas por la utilidad que tienen: La primera se llama *Configuración* porque en ella se colocaron funciones que afectan directamente el funcionamiento del sistema, se modifican algunos parámetros que usa el sistema para su funcionamiento, cómo son la dirección IP del servidor de Bases de Datos, el servidor JSP y el usuario de acceso del cual se habló en el punto 5.4.2 Ahora para entender el porque de los dos primeros veamos como se realiza la conexión a un Servidor de Bases de Datos (SBD) con Java, tenemos algo como este fragmento de código<sup>1</sup>:

<sup>1</sup> El código esta escrito según las convenciones de código para el lenguaje de programación Java, ver bibliografía.

```
1  /**
2   * Ejemplo de conexión a una Base de Datos
3   */
4   public class Consulta() {
5
6       Connection conn=null;
7       try {
8           Class.forName("com.mysql.jdbc.Driver");
9           conn = DriverManager.getConnection(url, login, password);
10          if(conn!=null) {
11              hacerAlgo();
12          }
13      }
14      ...
15  }
```

Figura 5.32. Ejemplo de Conexión a una Base de Datos en Java

En la figura 5.32 vimos dos de los cinco pasos básicos para ejecutar una sentencia SQL, pero lo que nos interesa de esta figura es la línea 9 en la cual aparece la variable “url” que es la dirección a la cual se debe conectar el programa para tener acceso al SBD, pero si se deja esta forma se tiene el inconveniente de tener que volver a compilar las clases que tengan esta dirección si por algún motivo se cambia la IP del SBD. Por ello es necesario dar la posibilidad al usuario de cambiar esta IP. Cabe mencionar que solo se guarda la IP puesto que el resto de la URL no cambia, que en este caso sería como la que se muestra a continuación:

**jdbc:mysql://XXX.XXX.XXX.XXX/Nombre\_BD**

Entonces las clases leerán la IP del servidor desde un archivo y con esta información se puede construir fácilmente la URL del SBD. Así no será necesario volver a compilar las clases que necesiten una conexión. Y para no provocar lentitud en las clases que necesiten una conexión, se usan las características introducidas en el SDK 1.4, más concretamente en la API conocida como NIO (Nueva I/O). En particular se emplean búferes y canales. Los búferes ayudan a agilizar las operaciones de entra y salida (E/S) de datos (que por lo general las más lentas) ya que sustituyen muchas operaciones de E/S que son lentas por varias operaciones sobre el búfer rápidas y sólo una operación E/S lenta. Y los canales son una conexión hacia un dispositivo de E/S que están diseñados para interactuar eficientemente con los búferes.

Recordando también que se desea tener un sistema multiplataforma, dichos archivos se guardaran de forma binaria, ya que esto garantizaría la independencia del sistema operativo. Así una, por ejemplo, la IP 132.248.66.221 se vería como se muestra en la figura 5.33:

```
00000000 00 00 00 84 00 00 00 F8 00 00 00 42 00 00 00 DD .....B....
00000010 00 00 1F 90
```

Figura 5.33. Ejemplo de archivo binario

Este archivo tiene un tamaño de 1 KB en memoria, lo cual no representa un aumento significativo en el espacio requerido por es el sistema en una computadora. Es importante mencionar que en el archivo se guardan los cuatro octetos que componen una IP, estos se almacenan como enteros, no es necesario almacenar los puntos de la IP ya que con código se puede agregar fácilmente, además se hace más fácil la lectura del archivo, puesto que solo se tiene que recuperar enteros de él y esto es muy sencillo y rápido usando NIO. Pero como lo que se necesita es una cadena(String), la clase encargada de leer el archivo y mandar la cadena a quien se lo solicite, lee el archivo y genera la dirección, esto lo podemos solucionar de una manera sencilla como se muestra en el código de la figura 5.34:

```
/**
 * Ejemplo para leer un archivo compuesto de enteros
 * y crear una dirección IP como cadena
 */
public String leerArchivo() throws IOException
{
    String ipString="";
    ByteBuffer buffer = ByteBuffer.allocate( 24 );
    canalArchivo.position( 0 );
    canalArchivo.read( buffer );
    buffer.flip();
    ipString = buffer.getInt()+ "." +buffer.getInt()
              + "." +buffer.getInt()+ "." +buffer.getInt();
    canalArchivo.close();
    return ipString;
}
```

Figura 5.34. Ejemplo para lectura de un archivo binario

Obviamente en este fragmento de código falta la declaración de canalArchivo esa parte se deja al lector.

Después de explicar el fin que tienen estas opciones describiremos el funcionamiento de las tres usando como ejemplo a “IP de BD”, en la figura 5.35 se observa el diagrama de flujo:

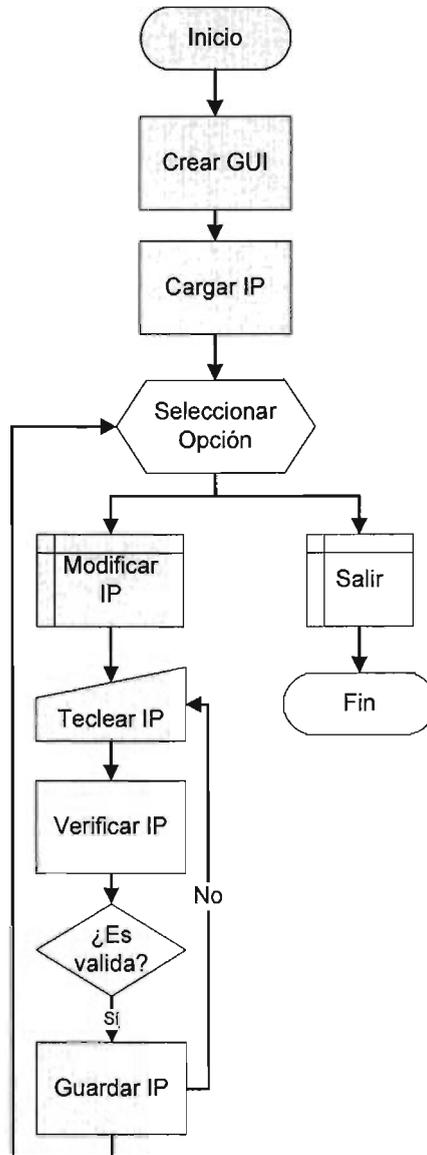


Figura 5.35. Diagrama de Flujo para guardar una IP

Los procesos que veremos más a fondo son los de *Cargar IP* y *Guardar IP*, ya que el resto de los procesos son simples. Después de que se crea la GUI se intenta mostrar la IP que se tiene almacenada actualmente en el archivo,

pueden ocurrir tres cosas, que no exista el archivo, que ocurra un error al leerlo o que se lea exitosamente. Las dos primeras se pueden manejar de manera similar, es decir, informar al usuario que ocurrió un error ya sea por problemas al leer el archivo o que no existe y después se procede a cargar una IP predefinida (que viene dentro del código), pero este manejo de error no es hecho directamente por la clase actual para ella es transparente; como ya lo mencionamos hay una clase encargada mandar la URL a las demás, por lo cual, el hecho de detectar los errores de lectura y mandar la IP predefinida son su tarea. A la clase actual llega una IP, pero si es posible informar a la clase que ocurrió un error, así se puede avisar al usuario que sucedió y la acción a tomar.

En este caso lo que necesitamos son los cuatro octetos de la dirección y no la URL completa, por ello debemos separar extraer esta información de la cadena. Se pueden usar diversas técnicas para extraer los octetos y la que empleamos fue usar expresiones regulares. El manejo de expresiones regulares se introduce en la API NIO que se mencionó anteriormente y provee varios métodos los cuales facilitan la operación que necesitamos efectuar sobre la cadena.

Para obtener los datos que necesitamos declaramos una expresión regular como la siguiente:

**[0-9]?[0-9]?[0-9]**

Esta expresión busca abarcar las posibles formas que puede adoptar los octetos que forma la dirección IP. Se buscan que sean números en el rango cero a nueve que estén en grupos de tres, dos o sólo un número. Así buscamos coincidencias con la cadena; cuando se encuentra una se extrae y el resto se ignora. Pueden haber varias coincidencias en una cadena, por lo tanto, se continua el análisis de toda la cadena, por omisión sabemos que habrán 4.

Al final obtenemos la IP en partes y las mostramos en cajas de texto, así se sólo se puede cambiar la parte que haya cambiado.

El diagrama de flujo de la figura 5.36 se puede observar el proceso que se acaba de explicar.

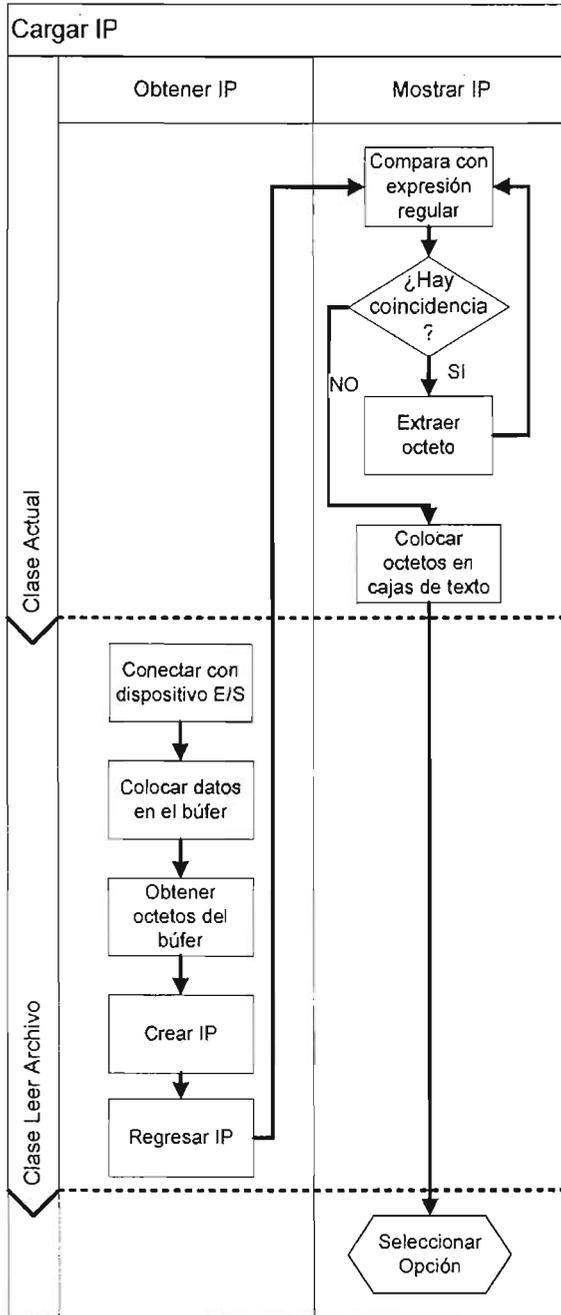


Figura 5.36. Diagrama de flujo para Cargar una IP

El proceso para guardar una IP es más sencillo, primero se tiene que comprobar que se introdujeron valores válidos, después los octetos son mandados directamente a la clase que los guardará en el archivo, como no es un archivo con bloqueo se sobre escribe completamente con los nuevos valores.

Las dos opciones de configuración restantes tienen un comportamiento similar a esta, por lo tanto podemos continuar con las *Herramientas*.

### *Herramientas del Sistema*

La primera herramienta se llama *Probar Conexión*, su propósito como su nombre lo indica es probar la conexión con la Base de Datos. Puesto que para poder realizar sentencias SQL (que son necesarias para obtener información) se tienen 5 pasos, como lo mencionamos antes. Estos pasos son:

1. Cargar el Driver
2. Obtener una conexión
3. Obtener el objeto "Statement"
4. Ejecutamos la sentencia SQL y obtenemos resultado
5. Cerramos la conexión a la Base de Datos

Es obvio que si no podemos completar alguno de los dos primeros pasos, no podemos continuar y no obtendremos resultados, pero para el usuario esto es transparente. Si existe algún problema con el servidor esto se traducirá al usuario como una Excepción SQL o un error con la Base de Datos; este falló no es culpa del sistema, pero si lo afecta. También puede ocurrir que el driver no esta instalado, así entonces tendríamos un problema de tipo local y otro foráneo (en el servidor). Por lo tanto es necesario proveer una herramienta que informe: en que pasó ocurrió el problema y así se encuentre una solución más rápido. Esto se hace en dos pasos, primero buscando el problema local y luego el de servidor:

1. Verificar el conector J (Connector/J)
2. Intentar la conexión con la Base de Datos

Para que las clases Java se conecten a una Base de Datos se requiere de un controlador. El conector, controlador o driver son partes referentes a JDBC (Java Data Base Conectcity) que permite conexión a diferentes DBMS y cada

uno tendrá su controlador específico. En la figura 5.37 podemos apreciar gráficamente esta explicación:

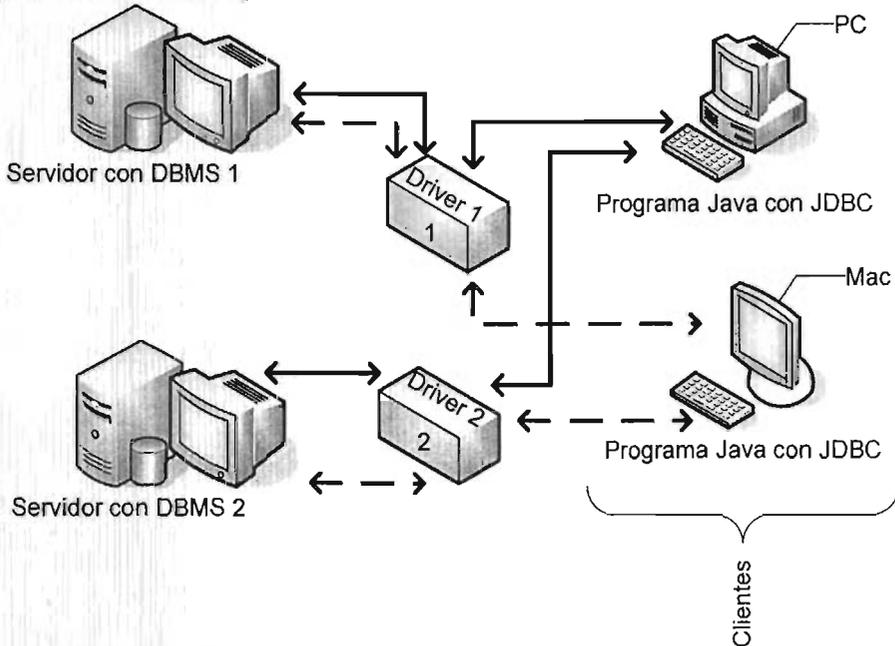


Figura 5.37. Modelo de conexión a un DBMS con Java

En una clase, la sentencia SQL es una cadena y el driver se encarga de traducir esa cadena Java en instrucciones de SQL y a su vez la respuesta en código Java.

Por lo tanto es necesario contar con el driver para la el DBMS al que se desea conectar, que en este caso es MySQL y el conector que ellos proporcionan en su página es llamado Connector/J.

Si no se cuenta con él no será posible establecer una conexión, por ello es que se comprueba que esté instalado y configurado en el ambiente de Java, esto es el primer paso. Se informa al usuario que el problema radica en el driver.

Pero si no hay un problema en el paso 1, lo siguiente es ver que hay comunicación con el DBMS que se encuentra en el servidor. Se intenta establecer comunicación con el servidor si fue exitosa no hay problema. Si no fue posible el problema radica fuera del sistema; las causas pueden ser muchas y no es tan fácil determinar, pero las podemos desglosar en tres principales: no

se tiene acceso al DBMS, hay un problema con el servidor o que exista un problema con la red. Se informa al usuario que el problema es en el paso 2 (no se logró conexión) y las posibles causas.

Visto de forma gráfica, tenemos la figura 5.38:

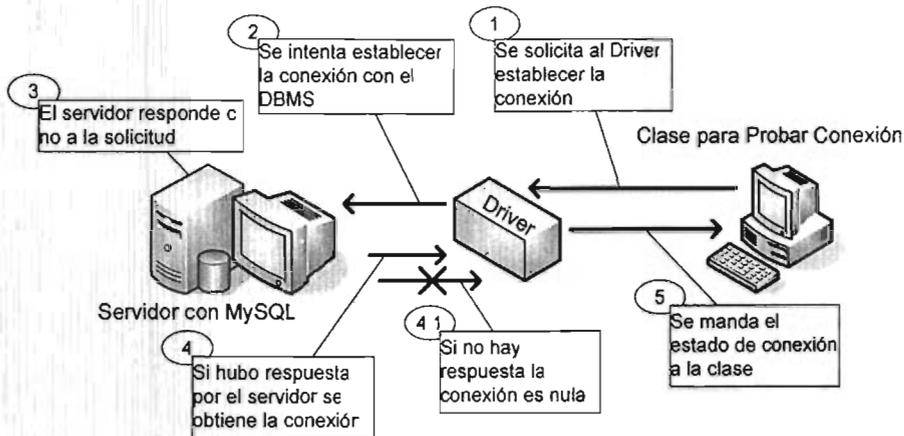
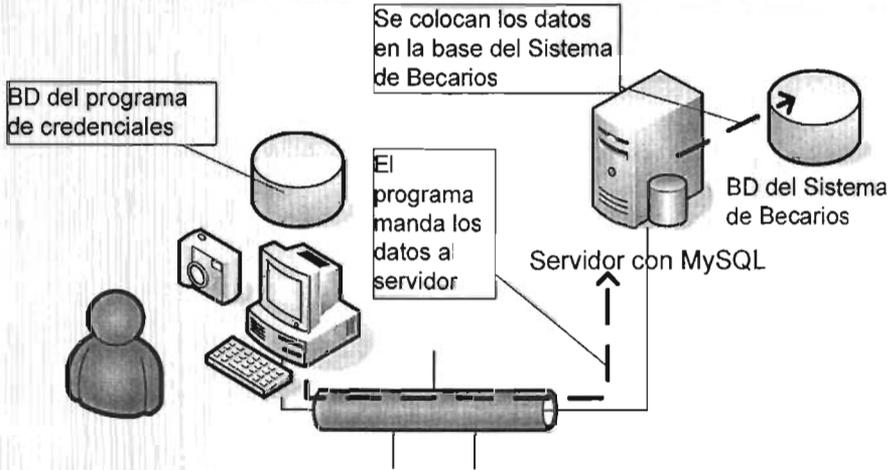


Figura 5.38. Forma de Probar la Conexión con la Base de Datos

La segunda Herramienta es *Migrar Base de Datos*, recordemos que los datos son colectados por el programa de credenciales en una base de datos local, como ellos son más útiles en el servidor es necesario colocarlos en él. Para ello se hace lo siguiente:

1. Poner los datos en el servidor.
2. Colocar los datos en la tabla específica.

En la figura 5.39 podemos observarlo gráficamente:



5.39. Migrando a la Base de Datos

Para ejecutar el primer paso se emplea una herramienta que proporciona el equipo de desarrollo de MySQL. Esta herramienta se llama “MS Access to MySQL” y se encarga de leer la base local y colocarla en un servidor, además de homologar los tipos de datos. Esta herramienta crea una nueva base en el servidor con los datos obtenidos.

El siguiente paso es poner los datos en la base que usa el sistema, para ello se leen los datos de la base creado por el programa y después se colocan en la base que se desea. Se informa al usuario si fue afectado o no cada registro y el total de registros afectados. No necesariamente se puede efectuar este paso sólo en el equipo que tiene el programa de credenciales, es posible hacerlo desde otro equipo. Así entonces la figura 5.39 se quedaría de la siguiente forma:

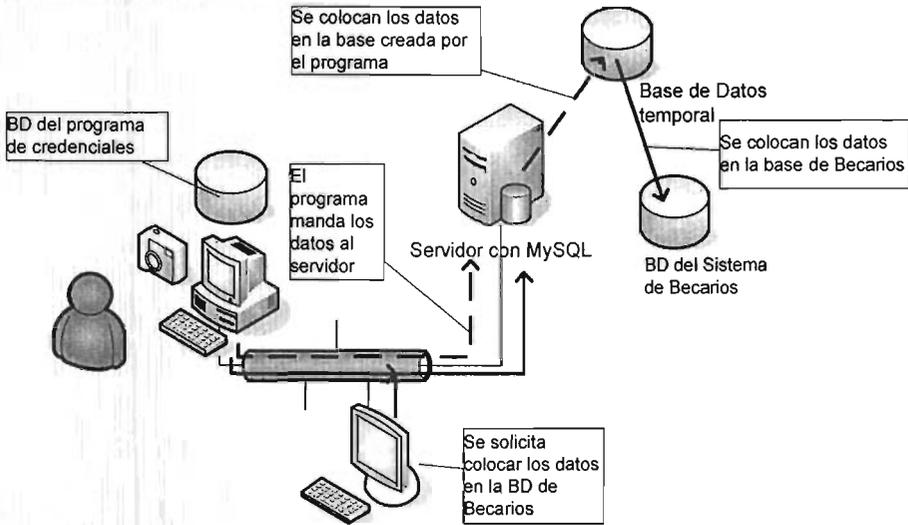


Figura 5.40. Migrando la Base de Datos

### 5.8.2.2 Ficha “Asistencia”

En esta ficha se agrupan los aspectos relacionado a la asistencia. Su principal propósito es consulta de información, pero también se puede modificarla. En la figura 5.41 podemos observar su estructura.

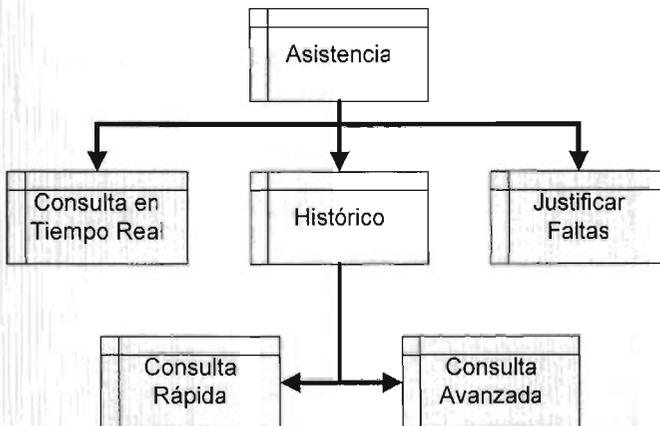


Figura 5.41. Estructura de la Ficha “Asistencia”

La primera opción que podemos ver es llamada “Consulta en Tiempo Real”, es llamada así puesto que al activar esta opción se escucha por la información que manda el SIA y se despliega en el área de salida de información. Como lo vimos antes, esta información es mandada unos segundos después que el becario pasa asistencia, entonces podemos considerar que esta información se obtiene en tiempo real, es decir, en el momento que ocurre el hecho de pasar asistencia. En el punto 5.4.1 ya se habló de esta característica, por lo tanto continuaremos con la siguiente opción.

Primero hablaremos de “Justificar Faltas”, esta opción se debe a que un becario tendrá una falta al no asistir, pero puede ser que la falta sea justificada, es decir, enfermedad, prácticas escolares, etc. Por lo tanto es necesario poder quitar esa falta del record de faltas, recordemos que se tiene un máximo de faltas y es importante el número de ellas. En términos de programación esto se refleja restando uno al número total de faltas que se tiene en la base de datos, lo cual es muy simple.

A la segunda opción se le llama “Histórico” porque se consulta la base de datos buscando el record de asistencias, faltas y retardos. Es posible ver información de un becario en particular o de todos los becarios. Existen dos tipos de búsquedas, la primera es llamada rápida, es necesario teclear el ID del Becario y seleccionar que se quiere buscar de las tres opciones disponibles; la información se desplegará en el área de información. La segunda es llamada avanzada porque además de lo anterior en esta se pueden ver todos los becarios almacenados, seleccionar alguna de las tres opciones o las tres; la información se mostrará en una nueva ventana.

Estas opciones emplean las clases con fines SQL creadas para el sistema, estas son usadas por todo el sistema cuando se requiere una consulta de información, exceptuando al SIA y la validación de un usuario en el servidor, estas emplean sus propias consultas a la Base de Datos. El motivo es muy simple, la información que es consultada tiene diferentes usos, en el caso de las excepciones será empleada para el sistema y no se mostrará dicha consulta, pero el resto de las consultas son hechas por el usuario y el resultado será mostrado al usuario. Como estas clases de consulta son de propósito general hablaremos más acerca de ellas.

### **5.8.2.2.1 Consulta Rápida**

Cuando se hace este tipo de consulta se espera obtener cierta información en particular de un becario, por lo tanto, se tienen dos posibles resultados: cero o un registro encontrado. Como la cantidad de información no es mucha, es posible mostrarla en el área de información sin mayor dificultad, ello también nos lleva a que no es necesario crear una nueva ventana, esto agiliza el tiempo de respuesta lo cual se hace notorio en computadoras de recursos limitados.

El diagrama de flujo se muestra en la figura 5.42. En ella podemos ver que después de crear la instancia a esta clase se manda una cadena SQL, para poder crearla se tienen diferentes formas, esto se debe a que obviamente las tablas y nombres de columnas no son iguales, por ello se crea la cadena SQL con la que se espera obtener los resultados deseados.

El siguiente paso mencionado es mandar la cadena, siendo más apeados al código esto se hace cuando se llama al método de Consulta Rápida que realiza la consulta (el único que tiene), en su firma viene incluida la declaración de esta cadena que se ejecutará en el servidor.

Después de esto comienza a trabajar la clase de consulta, se lee la cadena que es mandada por la clase que llama(o clase original) y se lee la URL desde el archivo como ya se mencionó en el punto 5.8.2.1. Ya que se tienen estos elementos necesarios para la conexión, se procede a intentar establecer una conexión con el servidor en el que se encuentra la base de datos.

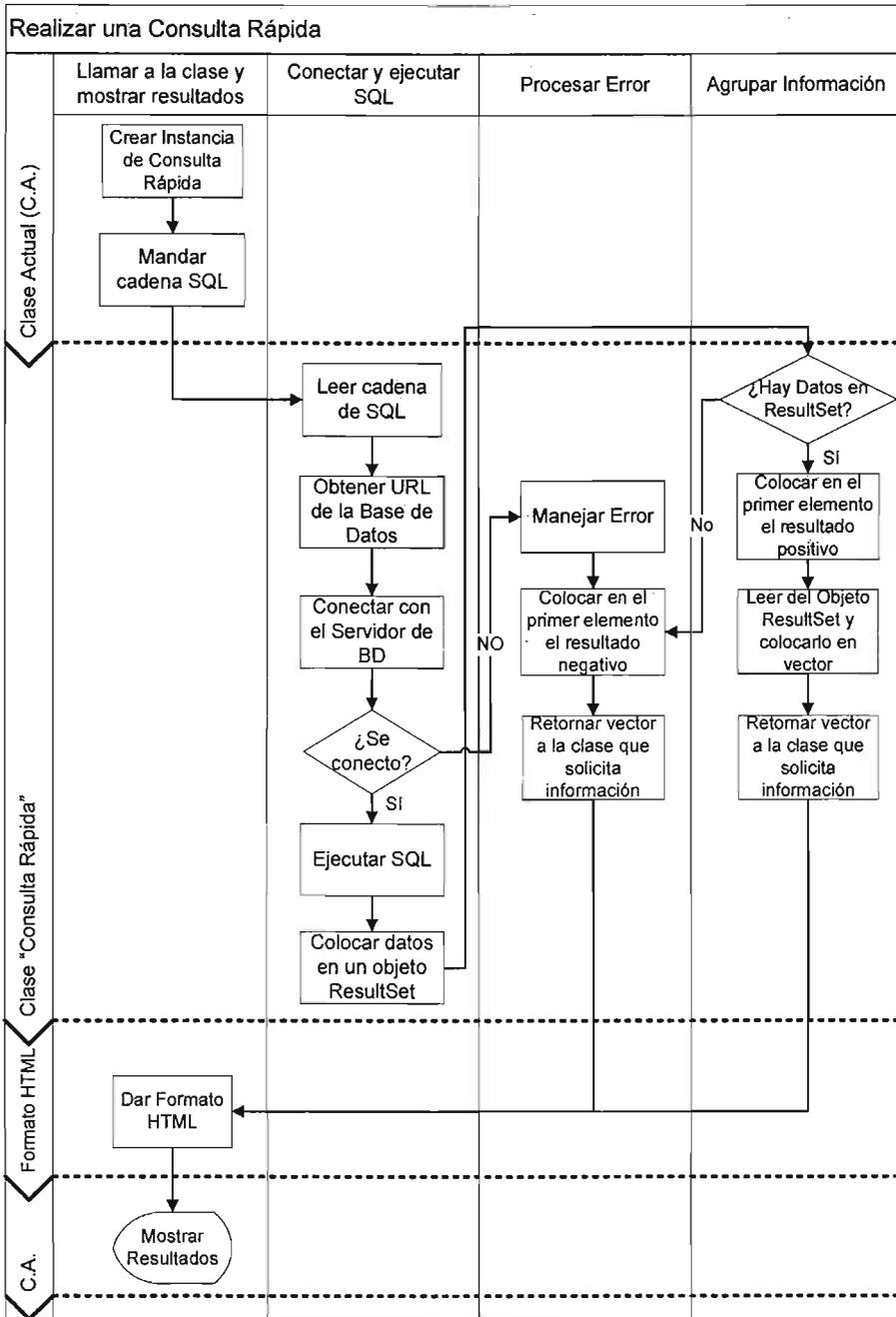


Figura 5.42. Diagrama de flujo de una Consulta Rápida

Si no fue posible establecer la conexión se tiene que manejar el error y el siguiente paso es colocar el resultado en el vector. Esta información se manda a la clase original para poder informar al usuario lo que ocurrió.

Los datos son colocados en un vector ya que así se pueden empaquetar todos y facilita su manejo al extraerlos mediante ciclos, así como darles formato. Los elementos de un vector son de tipo `Object` de Java, esto implica que no existe problema para colocar dentro de él cadenas de tipo `String`, las cuales son convertidas al tipo `Object` y cuando se extraen se pueden convertir nuevamente a su tipo original, esto podría parecer más trabajo, pero es una practica muy común dentro del ambiente Java, de hecho siempre que se manda a imprimir se hace una operación similar, todo lo que se desee imprimir se convierte al tipo `String`, lo cual es transparente para el programador.

Regresando al diagrama, cuando fue posible la conexión con la base es momento de ejecutar la cadena SQL. Los resultados son colocados por Java en un objeto `ResultSet`, como ya lo mencionamos solamente obtendremos cero o un resultado, es fácil verificar si hay datos o no, así entonces, si no hay datos en el objeto se coloca en el primer elemento del vector que el resultado fue negativo, es decir, que no se encontró el becario o lo que fue solicitado. El caso contrario es obtener un resultado, también se informa que el resultado en el primer elemento y en los siguientes se colocan los datos. El vector resultado es devuelto a la clase original, esta a su vez solicita que se le de formato, lo cual se describe en el siguiente párrafo. Después se manda imprimir en pantalla los resultados. La figura 5.43, se muestra un ejemplo de una posible consulta rápida.

---

<b>Resultado de Búsqueda</b>	
Nombre	Salvador Mendoza
Edad	23
Teléfono	582163 48

Figura 5.43. Ejemplo de una consulta rápida

La clase que da formato HTML crea una cadena la cual tiene la información y el código HTML necesario para que tenga la apariencia deseada, recordemos que el área de salida de información soporta este formato (vea punto 5.8.2),

cuando se manda imprimir solo se solicita colocar la cadena y Java se encarga de interpretar dicho formato.

Sólo es necesario proporcionar dos vectores, el que ya se tiene con los datos y uno para etiquetar la información. Se emplean de esta forma puesto que en esta clase se crean tablas para mostrar la información de una manera ordenada. Esta clase regresará una cadena, por ejemplo, la que se muestra en la figura 5.44, que este caso sería el código necesario para poder mostrar la información como se observa en la figura 5.43. Obviamente los caracteres como vuelta de carro y fin de cadena no se crean ya que no es necesario, se colocaron con fines ilustrativos.

```
<table width="100%" border="0" cellspacing="0" cellpadding="0">
<tr bgcolor="#CCCCCC">
  <td><strong><font face="verdana, Arial, Helvetica, sans-serif">
    Nombre</font></strong></td>
  <td><font face="verdana, Arial, Helvetica, sans-serif">
    Salvador Mendoza</font></td>
</tr>
<tr>
  <td><strong><font face="verdana, Arial, Helvetica, sans-serif">
    Edad</font></strong></td>
  <td><font face="verdana, Arial, Helvetica, sans-serif">
    23</font></td>
</tr>
<tr bgcolor="#CCCCCC">
  <td><strong><font face="verdana, Arial, Helvetica, sans-serif">
    Teléfono</font></strong></td>
  <td><font face="verdana, Arial, Helvetica, sans-serif">
    58216340</font></td>
</tr>
</table>
```

Figura 5.44. Ejemplo de clase Formato HTML

Así se tiene la información etiquetada, ordenada y con formato, lo cual proporciona una mejor apariencia al usuario.

### 5.8.2.2.2 Consulta Avanzada

Como ya se comentó en el punto 5.8.2.2, esta opción tiene mayor capacidad. Esto se debe primordialmente a dos razones, la primera es que se puede ver todo el contenido de la tabla deseada y que la instrucción SQL se genera de una manera más dinámica. La información se muestra con la configuración mostrada en la figura 5.45:



ID Becario	Nombre	Apellido Paterno	Apellido Materno	Domicilio	Edad	Facultad o Escuela
HEPJ810623	Juan	Hernández	Pérez	Calle 23, No 82	23	Química
MULE821215	Erica	Murillo	López	Av. Coyoacán No 569	22	ENAH
LUCM820904	María	Luna	Cruz	Calle 101 No 182	22	Biología

Figura 5.45. Ejemplo de configuración de Consulta Avanzada

La figura 5.45 nos ayuda a ver características que podrían parecer irrelevantes pero son muy útiles para esta consulta; obviamente se trata de una tabla que tiene cierto número de renglones, de columnas y el nombre de las últimas. Para poder crear una tabla semejante a la mostrada en la figura 5.45 podemos emplear la clase `AbstractTableModel` del paquete `javax.swing.table` en particular la interfaz `TableModel` (si se desea conocer más acerca de esta clase consulte la bibliografía proporcionada en el capítulo 8).

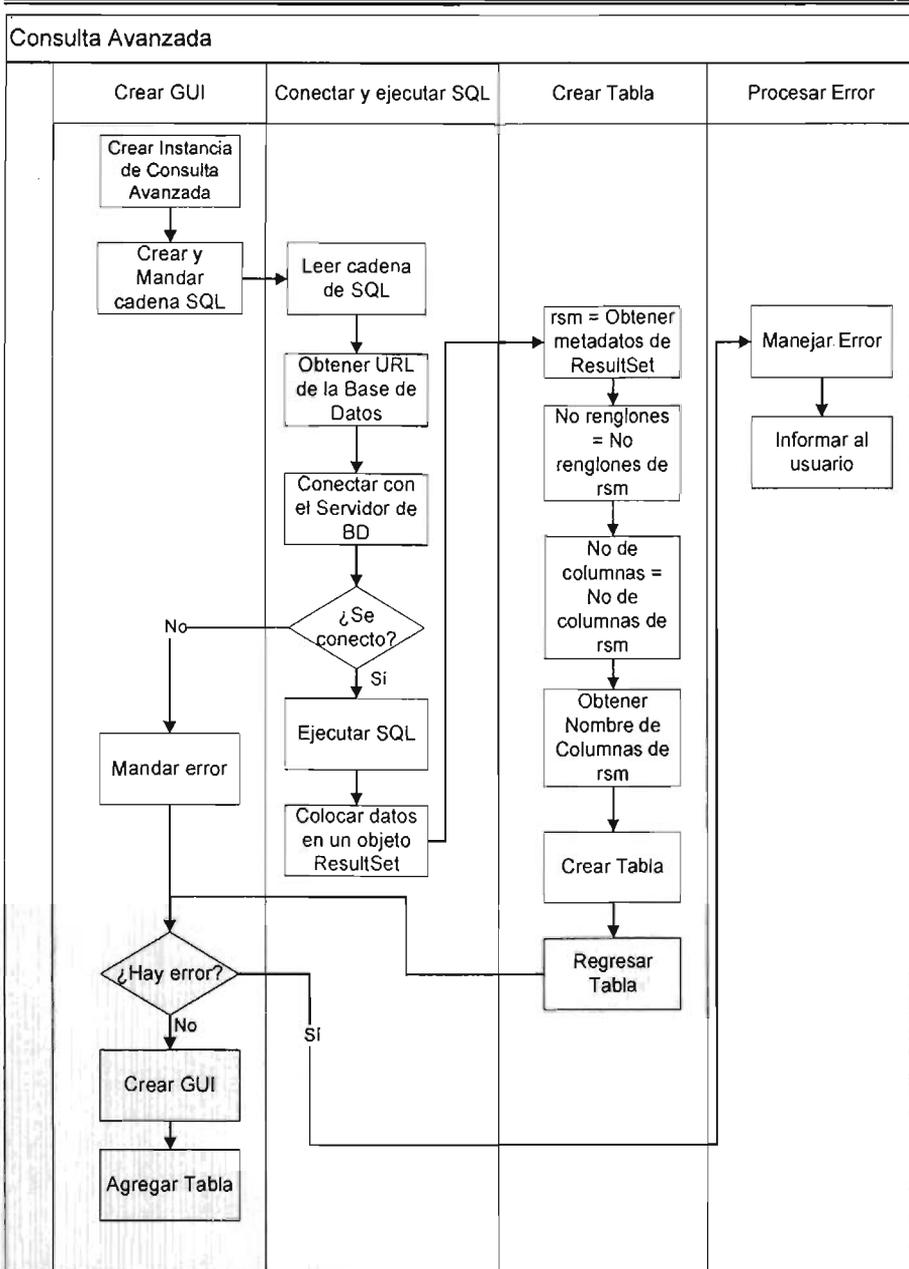


Figura 5.46. Diagrama de Consulta Avanzada

En la figura 5.46 podemos observar el diagrama de flujo para esta consulta, la primera parte es la conexión a la base de datos, esta parte es muy parecida a lo

que se vio en el punto 5.8.2.2.1. Después de tener el objeto `ResultSet` podemos obtener las características que necesita `TableModel` para poder construir la tabla. Primero es necesario obtener los datos del objeto a una forma que sea manejable por `TableModel` y esto se hace obteniendo los meta datos de `ResultSet`; esto es necesario ya que en una consulta podemos encontrar varios tipos de datos, es decir, pueden haber cadenas, enteros, fechas, etc. y para poder extraerlos de forma normal se tiene que especificar el tipo del dato que se desea extraer del `ResultSet` obtenido de una consulta, por ello al homologar los datos transformándolos a un mismo tipo de dato es más fácil crear la tabla sin preocuparse del tipo de dato que se esta mostrando en la tabla.

Ya que tenemos los meta datos, podemos crear la estructura de la tabla, es decir, saber cuantas columnas y renglones tendrá la tabla que se mostrará, es necesario proporcionar estos parámetros a `TableModel` ya que necesita las dimensiones que tendrá la tabla. Otro parámetro, es el nombre que tendrán las columnas, en este caso será el nombre que tiene el campo de la tabla consultada en la base de datos.

Finalmente es posible crear la tabla, pero antes de crear la GUI se debe comprobar que no ocurriera algún error en el proceso de creación o en la conexión a la base de datos, en caso de pasar se maneja y se informa que ocurrió al usuario, en este caso ya no se continua la construcción de la GUI, pero en caso de no existir algún problema se puede crear la interfaz de usuario para poder mostrar al tabla.

Así se obtiene una tabla con la información solicitada, un dato que no se menciona es que si la consulta no arroja resultados se mostrará una tabla vacía. Esto no es un error, ya que todo se ejecuta correctamente, solo que el criterio de búsqueda del usuario no arrojó algún resultado.

### **5.8.2.3 Fichas “Becarios”, “Salas”, “Horas de Inicio”**

En este caso agrupamos estas fichas, ya que, como se explico las consultas emplean las clases Consulta Rápida y Avanzada para desplegar información, cada una tiene su manera particular de llamarlas, pero esto no aporta información nueva a este trabajo. Si mencionaremos la estructura de cada ficha y las características particulares.

Becarios

Para Becarios tenemos una estructura como se muestra en la figura 5.47, las primeras dos opciones ya fueron mencionadas, la última es modificar la información que se tiene en la base de datos.

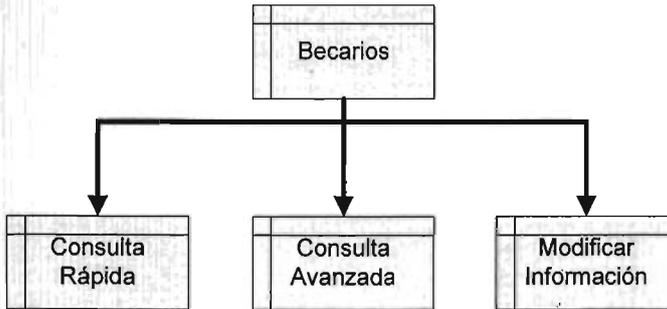


Figura 5.47. Estructura de Becarios

Para esto se solicita el ID del Becario y a partir de ello se busca al becario, esto se hace ya que se cargarán los datos que se tienen almacenados en cajas de texto para que el usuario modifique solo los deseados. Esto lo podemos apreciar en la figura 5.48.

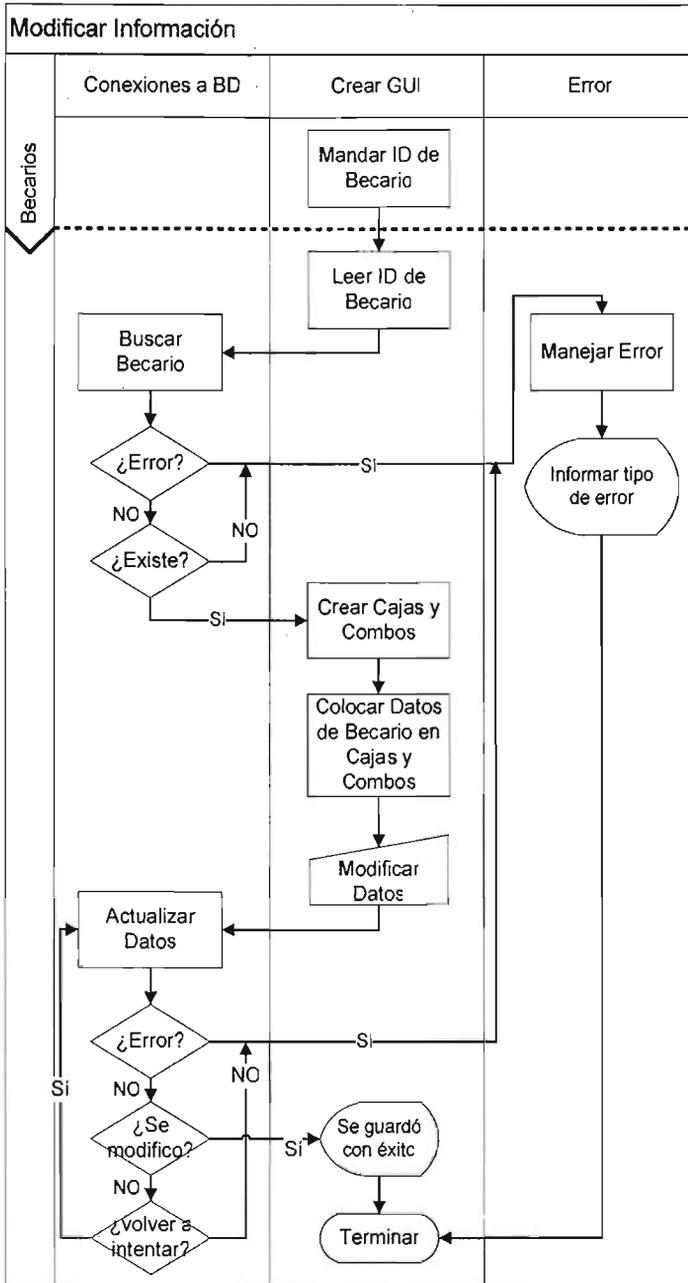


Figura 5.48. Diagrama para Modificar Información

Los bloques llamados Buscar Becario y Actualizar Datos engloban varios procesos que ya se mencionaron, como leer de un archivo la IP del servidor o establecer conexión. Para no complicar más este diagrama se muestran como bloques.

Cuando se encuentra al becario y no ocurrió un error se puede crear la interfaz gráfica de usuario, para ello se crean cajas de texto en donde se colocarán los datos del becario y en el caso de datos que estén ligados a otras tablas o que se quiere forzar a que se escoja solo ciertos valores se crean listas desplegables. Así cuando se ha creado la interfaz se colocan los datos.

Sí hay un error, este se maneja y se informa al usuario lo ocurrido. Los errores pueden ser básicamente dos, que no exista el becario solicitado o que se tengan problemas de conexión. El primero no se trata de un error crítico pero es mejor manejarlo para evitar crear una interfaz sin datos. Como ya no es necesario continuar creando la interfaz se salta este paso y se termina la ejecución de este bloque de código.

Al no ocurrir un error se continúa de forma normal y el usuario modifica los datos que desea cambiar y procede a actualizar la información. Se vuelve a ejecutar una conexión a la base de datos para actualizar la información. Si se logró o no guardar se informa al usuario, así termina este proceso.

### Salas

Para las Salas se tiene tres operaciones básicas: agregar, consultar y modificar. Como lo podemos ver en la figura 5.49. En esta se muestra como se encuentran agrupadas estas opciones en ficha Salas.

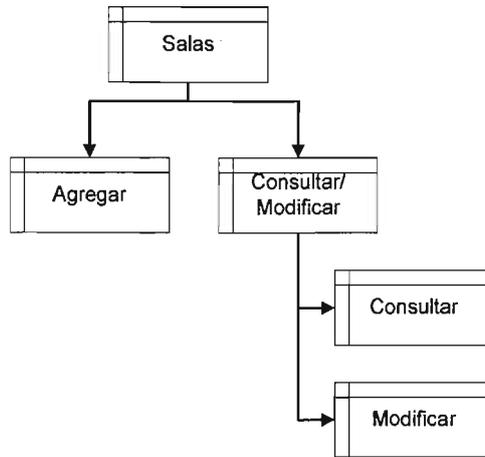


Figura 5.49. Estructura de Salas

Agregar una sala a la Base de Datos es una tarea muy sencilla, solo requerimos un identificador único (id) y el nombre de la sala. Para hacer la tarea aún mas simple al usuario, el sistema puede crear el id de la sala. Cuando el id de la sala es creado por el sistema sigue el siguiente algoritmo:

1. Obtener el número de salas almacenadas
2. Contador = número de salas almacenadas +1
3. idSala = "sala" & convertirACadena(Contador)

Es decir se obtiene el número de registros almacenados en la Base de Datos y a esto se le agrega uno para tener un id del tipo: sala1, sala2, sala3, etc. Cabe destacar que el símbolo "&" se emplea en este caso para indicar la concatenación del resultado obtenido al convertir a cadena el número de registros. Esto en Java no se hace exactamente así, cambia un poco como lo podemos ver en la figura 5.50:

```
numeroDeReglones = rs.getRow() + 1 ;  
resultado= "sala" + numeroDeReglones ;
```

Figura 5.50. Creando el Id de Sala en Java

Observamos que el operador "+" se usa para sumar, pero también para concatenar cadenas, además la concatenación se hace de forma directa, aparentemente.

Finalmente cuando se tiene el identificador y el nombre de la sala se puede guardar; los resultados se informan al usuario imprimiéndolos en el área común con formato HTML.

Siguiendo con las opciones, en este caso Consultar y Modificar se encuentran agrupadas para no complicar la interfaz, ya que, ambas solo llaman a otras clases. En el caso de consultar se llama a Consulta Avanzada, la cual ya fue comentada en el punto 5.8.2.2.2. Podemos decir que en este caso la sentencia SQL es la siguiente:

```
SELECT * FROM salas.
```

Para modificar la información se lleva un procedimiento muy similar al que se lleva cuando se quiere modificar a un Becario, solo que en este caso, solo se crea un Combo o lista desplegable, el cual tiene los nombres de todas las salas almacenadas; así el usuario selecciona cuál desea modificar y hace el cambio que desee.

### Horas de Inicio

Al igual que la otra opción sólo tenemos tres operaciones: agregar, consultar y modificar. En la figura 5.51 podemos ver su estructura.

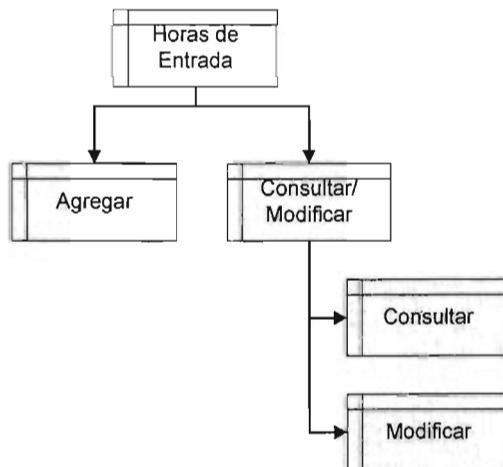


Figura 5.51. Estructura de Horas de Entrada

Como ya se pudo observar tiene una estructura igual a la que tiene Salas, de hecho la única diferencia es que todas estas operaciones se hacen a una tabla diferente, por no hablaremos más sobre esta opción.

### 5.8.2.4 Ficha “Sábana”

Quizás sea una de las partes que más uso tendrán por parte del usuario, aquí es en donde se realiza la asignación o registro del horario en el sistema y la ubicación del becario dentro del museo, es decir, a que sala se encuentra asignado el becario. También se tienen funciones de consulta de esta información, individual o grupal. Aquí se puede ver todos los becarios que asisten en un día en especial o todos los que están asignados a una sala en particular. Cómo lo podemos ver en la figura 5.52 la estructura de esta ficha cuenta con las opciones antes mencionadas.

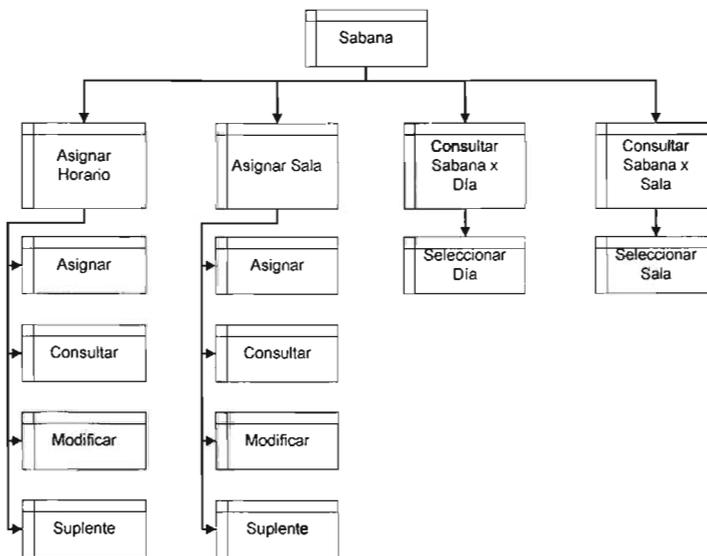


Figura 5.52. Estructura de Sabana

Las dos primeras opciones (de izquierda a derecha) que se observan en la figura 5.52 son las encargadas de hacer modificaciones de forma directa a los datos que están almacenados en la base.

Pero antes de mencionar el funcionamiento de las opciones antes mencionadas, es necesario hablar sobre los horarios, ya que, de ellos depende el funcionamiento principal del Sistema.

### 5.8.2.4.1 Horarios

Para entender que es lo que se tiene que almacenar se muestra la figura 5.53, se trata de un ejemplo de como el becario puede organizar su horario.

Hora\Día	Lune	Marte	Miércoles	Jueve	Vierne	Sábado	Domingo
a	s	s	s	s	s	o	o
9 – 10	x					x	x
10 – 11	x		x			x	x
11 – 12	x		x			x	x
12 - 13	x		x			x	x
13 – 14	x						
14 – 15		x					
15 – 16		x					
16 – 17		x					

Figura 5.53. Ejemplo de un horario

En la anterior figura se busca ejemplificar que los horarios pueden ser casi cómo el becario lo desee. También hay ciertas normas, que ya se mencionaron en la introducción, una de las principales es que se tienen ciertas horas de entrada ya establecidas.

Ahora tomando en cuenta el “tipo” de horario que se desea guardar, las horas de entrada y que este se tiene que almacenar en una Base de Datos se llegó a la conclusión que una forma de guardar los horarios sería una cadena que tenga cierta estructura y facilite las búsquedas. Para agilizar las búsquedas, se tienen cadenas por cada día, es posible juntar todo el horario en una cadena, pero es más recomendable separarlas para tener búsquedas más rápidas y ahorrar tiempo de programación.

Esta cadena tiene una estructura que ayuda a extraer la información que se necesita. Para describirla emplearemos el metalenguaje EBNF, así la cadena tiene la siguiente forma:

```
< Cadena > : = < Estado > < EstructuraHorario > % <
EstructuraHorario >
< EstructuraHorario > := { < HoraEntrada > # < Horas > } | $
< Horas > := < Hora > < Separador > < Horas > | < FinHorario >
< HoraEntrada > := < IDHoraEntrada >
< IDHoraEntrada > := [a-zA-Z]+[0-9]+
< Estado > := NORMAL | PERMIS | SUPLEN | BAJATL
< Hora > := (1)?[0-9]
< FinHorario > := [!]
< Separador > := [,]
```

Empezando de Izquierda a Derecha observamos que la etiqueta tiene un elemento llamado “Estado” esto describe las cuatro posibles situaciones de un becario dentro del museo:

NORMAL, está en activo dentro del museo y es su horario habitual.

PERMIS, tiene permiso para ausentarse.

SUPLEN, está en activo y aparte de su horario normal está supliendo a otro becario.

BAJATL, está dado de baja temporal de sus actividades.

El elemento que hace referencia a la estructura se puede repetir separado por “%” ya que la segunda parte es el horario de suplente. La estructura del horario esta conformado por delimitadores, por un ID de la hora de entrada y las horas que componen el horario. La mejor manera de entender esta estructura es con un ejemplo, para ello empleamos el horario del la figura 5.53, en particular el día lunes y suponiendo que las 9 de la mañana es una hora de entrada registrada en el Sistema con un identificador igual a Hora1.

```
NORMAL {
    Hora1#
    10, 11, 12, 13, !
}
```

Esto en la Base se guarda sin elementos superfluos de la siguiente forma:

```
NORMAL{Hora1#10,11,12,13,!}
```

Así esta cadena nos proporciona los elementos suficientes para poder reconstruir el horario, además de algunos delimitadores para encontrar datos sobre el horario del becario que sean requeridos.

### 5.8.2.4.2 Asignar Horario

Lo importante al asignar un horario es proporcionar una interfaz que facilite la asignación y la creación de la cadena. La solución es la que se muestra en la figura 5.54, en ella podemos ver que utilizando casillas de verificación se vuelve muy sencillo la asignación, sólo es cuestión de seleccionar las horas deseadas y remover una hora en caso de haber sido seleccionada por error las dos opciones son igual de simples.



Hora\Día	Lunes	Martes
9 a 10	<input checked="" type="checkbox"/> L	<input type="checkbox"/> M
10 a 11	<input checked="" type="checkbox"/> L	<input checked="" type="checkbox"/> M
11 a 12	<input checked="" type="checkbox"/> L	<input checked="" type="checkbox"/> M
12 a 13	<input type="checkbox"/> L	<input checked="" type="checkbox"/> M
13 a 14	<input type="checkbox"/> L	<input checked="" type="checkbox"/> M
14 a 15	<input type="checkbox"/> L	<input checked="" type="checkbox"/> M

Figura 5.54. Ejemplo de Interfaz para Asignar Horario

Se podría pensar que en términos de programación se complica, pero no es así. Lo mejor es hacer un arreglo de casillas de verificación para cada día. Empleando un ciclo se crean los objetos, se dan propiedades y se colocan en el lugar que les corresponde; al final lo que se tiene es la GUI.

Para hacer la asignación, también se hace por medio de un ciclo, en el cual se analiza el estado del arreglo, es decir, se verifica el valor de cada uno de los

elementos del arreglo, que puede ser seleccionado o no. Si esta seleccionado se agrega la hora a la que corresponde ese elemento. La manera de saber a que hora corresponde un elemento se entiende ya que podemos observar que hay una relación uno a uno (vea Figura 5.55), entonces, la hora se determina agregando 9 al valor actual del contador usado en el ciclo.



Pero primero se tiene que analizar en cada día el primer elemento seleccionado, este primer elemento que sea seleccionado corresponde a la hora de entrada al museo, por lo tanto se tiene que verificar que esté dentro de las almacenadas en la Base de Datos (BD). Si se solicita una hora no válida, el sistema rechazará la misma, en caso contrario se puede crear la cadena del horario. Cabe destacar que los becarios no asisten todos los días, si observamos la descripción de la cadena esto es tolerado por la misma. Cuando se ha analizado los siete arreglos y ninguno provocó un error tendremos siete cadenas, todas por omisión tienen el valor de estado NORMAL, estas cadenas creadas están listas para ser almacenadas, antes de ello se debe verificar que el becario no tenga asignado un horario previamente, esto se debe hacer ya que la operación de inserción en la BD provocaría un error al tener llaves duplicadas. Cuando todo ocurre sin errores se procede a guardar el horario en la BD.

Como valor agregado es posible asignar el mismo horario a varios Becarios, esto se hace proporcionando los identificadores separados por coma, por ejemplo:

**ID1, ID2, ID3, ... , IDn**

Usando una expresión regular podemos determinar si el usuario esta haciendo uso de esta opción y lo hizo de la forma correcta, es decir, usó el separador correcto y que esta introduciendo un identificador, esta es:

([a-zA-Z]+[0-9]+,)\*([a-zA-Z]+[0-9]+)

Después de analizar la cadena y obtener resultados positivos se extraen los identificadores de los becarios, cuando se solicita asignar el horario se asigna a cada uno de ellos.

Podemos ver en el diagrama de la figura 5.55 lo anteriormente explicado.

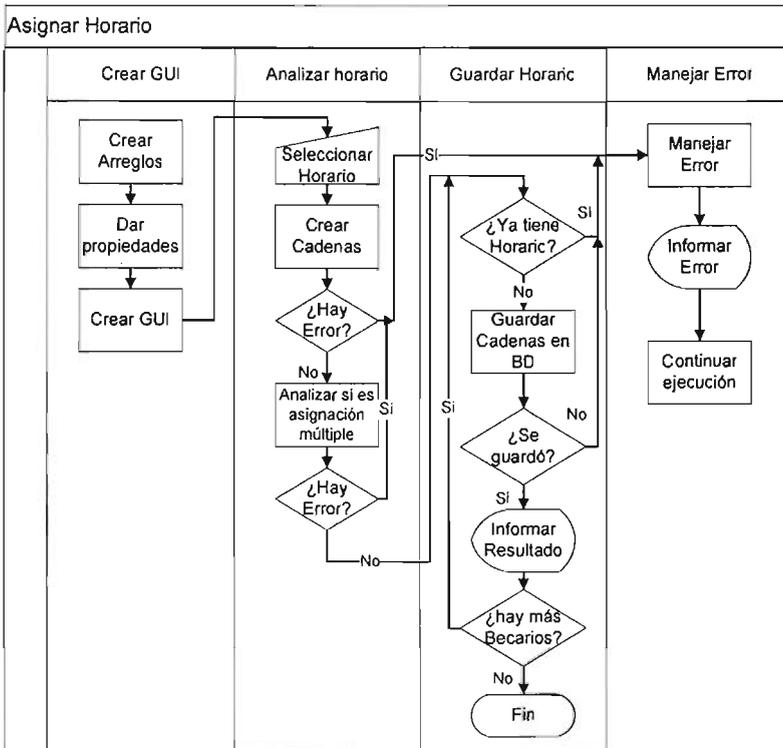


Figura 5.55. Asignar Horario

### 5.8.2.4.3 Consultar Horario

Para consultar un horario es necesario tener el identificar del becario, con él se puede obtener las cadenas para reconstruir el horario. Este proceso es el inverso a la creación, y nuevamente hacemos uso de una expresión regular que nos ayude a buscar los números dentro de la cadena y así obtener las horas en las que asiste el becario.

Este proceso de reconstrucción del horario no es destructivo con la cadena, al usar una expresión regular y buscar coincidencias Java nos proporciona métodos que nos dan el inicio y el fin de una coincidencia en caso de ser encontrada. Así podemos crear un subcadena con el dato que nos interesa sin necesidad de afectar a la cadena original. Por ejemplo:

```
String cadena = "NORMAL{hora1#9,10,11,12}";  
int inicio = 13;  
int fin = 14;  
String subCadena = cadena.substring(inicio,fin);
```

Resultado

```
subCadena = 9
```

En el ejemplo anterior se indicó de forma manual el inicio y fin de la subcadena pero en nuestro caso se hace de forma dinámica. Así obtenemos las horas de la estructura del horario.

Recordando la relación uno a uno vista en el punto 5.8.2.4.2, observamos que podemos emplear el mismo método para desplegar gráficamente el horario. Creamos una “plantilla” que para fines de programación se trata de un arreglo de enteros, cada uno de sus elementos representa una hora, en la figura 5.56 se puede ver más claramente, en esta figura se acomodó en un arreglo las horas para hacer más evidente esta relación.

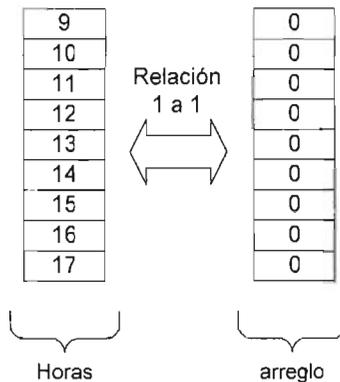


Figura 5.56. Relación entre horas y arreglo

De la cadena original obtenemos las horas y cambiamos el valor del arreglo en la ubicación según corresponda. Así obtenemos la plantilla que nos sirve para poder crear la GUI, para ello volvemos a crear un arreglo de objetos pertenecientes a la API swing y dependiendo del valor del arreglo de enteros se coloca o no un icono al objeto, en este caso si es uno al objeto se le coloca un icono. Esto lo hacemos para cada una de las cadenas y así tenemos la reconstrucción del horario completo de becario de forma gráfica, algo similar a la figura 5.52. En la figura 5.57 podemos ver la descripción gráfica de este proceso.

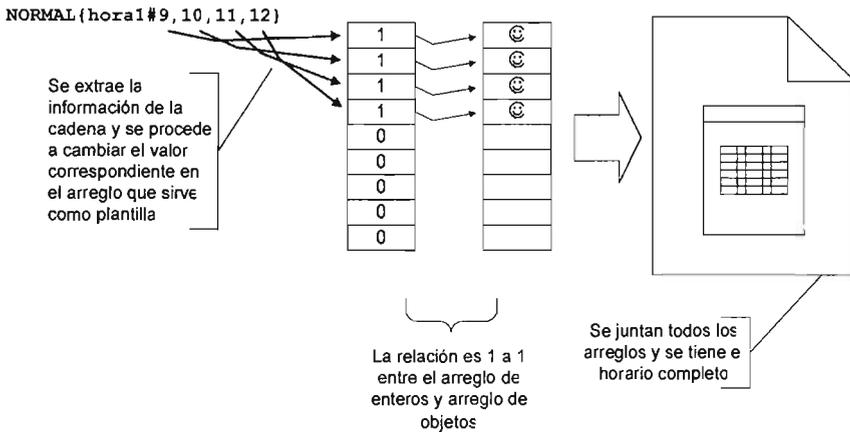


Figura 5.57.Reconstrucción Gráfica del Horario

Para completar esta descripción, se tiene que efectuar una conexión a la BD para obtener las cadenas y se pueda afectar el proceso de análisis de cada una de ellas.

#### 5.8.2.4.4 Modificar Horario y Asignar Horario de Suplente.

Para estas dos operaciones se usan los procedimientos para asignar y consulta. Por ello se agrupan.

Para modificar se lleva un proceso similar al de mostrar, pero esta vez se cambia el objeto de la API de swing, se usan casillas de verificación y no se les coloca un icono sino que su estado se coloca ha seleccionado. Para hacer efectiva la modificación en la BD, se crean las cadenas siguiendo el procedimiento llevado para asignar el horario pero no se verifica si ya tiene un horario asignado.

Desde el punto de vista del usuario no hay diferencia entre asignar un horario de suplente y el horario habitual del becario. Pero si las hay, la diferencia radica en que sí es importante que ya tenga un horario, en caso de no tenerlo no se puede asignar el horario de suplente, puesto que no es posible tener solo suplentes y se provocaría un error al querer actualizar la BD. Cuando si lo hay, las cadenas se crean sin valor de estado, ya que, es necesario conservar el horario base y sólo agregar el de suplente. En la figura 5.58 vemos como se genera la nueva cadena.

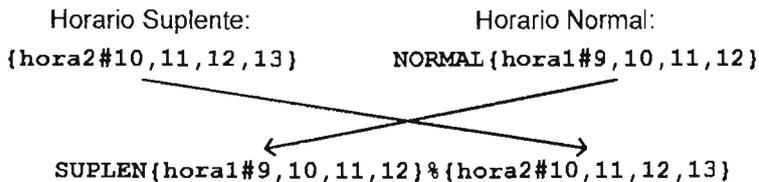


Figura 5.58. Agregar Horario de Suplente.

#### 5.8.2.4.5 Asignar Becario

El funcionamiento básico de esta ficha es hacer inserción, modificar y consultar los datos de la tabla "becarios por sala", se trata de operaciones estándar y no aportan algo nuevo a este trabajo. Pero es prudente señalar que tiene implementadas algunas herramientas ya antes vistas y otras próximas a ser mencionadas. En particular ya fue descrita la asignación múltiple de horarios, este caso se tiene la misma asignación múltiple implementada para asignar a una sala a los becarios. Como en el anterior caso se solicita el identificador del becario y se crea una lista con las salas disponibles en las BD. En la figura 5.59 podemos observar como se hace la asignación.

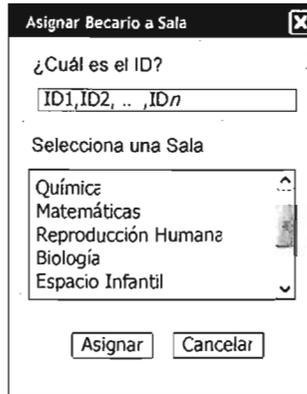


Figura 5.59. Asignar un Becario a una Sala

Otra característica son las búsquedas, para ello se usa el análisis llamado “Mejorando Búsquedas” que se verá más adelante.

### 5.8.2.5 Consultar Sábana por Sala

Se trata de la sábana más “sencilla” de crear, ya que tenemos sus dimensiones fijas como lo podemos observar en la figura 5.60. En total se tiene una “tabla” con 10 renglones y 8 columnas, pero un renglón y una columna son empleados para etiquetar, el resto es usado para agregar la información de los becarios.

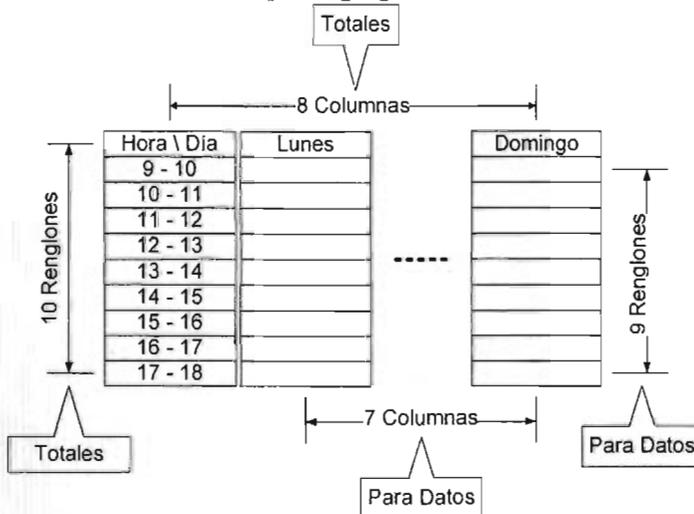


Figura 5.60. Dimensiones y Estructura de la Sábana por Sala

Para poder construir esta sabana necesitamos la siguiente información:

- a. Sala que se desea consultar (SC).
- b. Todos los becarios que están asignados a la SC.
- c. Horario de cada uno de los becarios del punto b.

Con estos datos podemos dar la información que el usuario necesita, es decir saber que becarios están en esa sala y en que horario. Para obtener la sala se crea una lista con todas las salas en la BD, el usuario selecciona la que desee ver, así podemos obtener el identificar único de la sala, el resto de la información se obtiene con este dato.

Para construir la tabla recordemos el método descrito en la figura 29 y la relación descrita en el punto 5.8.2.4.2, tenemos las ideas suficientes para crear esta tabla, solo hay unos cambios que son necesarios para lograr este objetivo. En los anteriores casos se crearon arreglos de casillas de verificación y etiquetas, en este caso se utilizaran objetos contenedores, es decir, objetos que son capaces almacenar otros objetos, que en este caso son botones. La idea es agregar un botón en vez de poner un icono o cambiar de propiedad, en la figura 5.61 se puede observar esto. El objeto contenedor cambia de tamaño cada vez que se agrega un botón, por lo tanto es posible agregar los que sean necesarios.

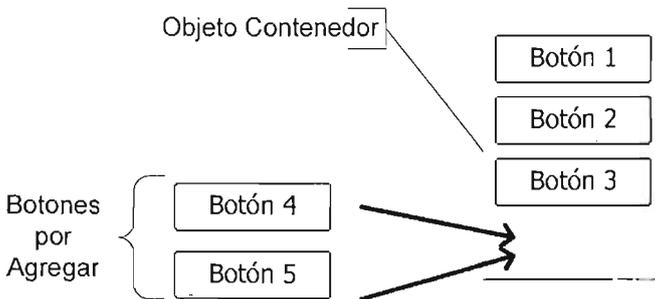


Figura 5.61. Objeto Contenedor

Los botones tienen la finalidad de poder acceder la información individual de los becarios, entonces la figura 5.61 quedaría como lo podemos ver en la figura 5.62. La etiqueta del botón sería el nombre de cada becario.

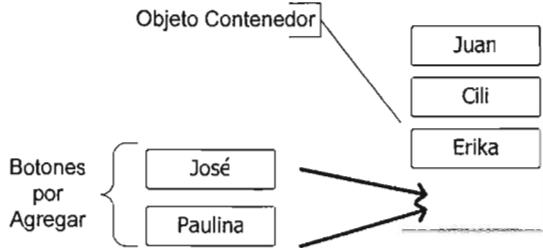


Figura 5.62. Objeto Contenedor

Con este cambio a lo antes visto es posible crear la sábana. Se analiza el horario y se agregan a la interfaz un botón por cada hora que el becario asiste al museo, este procedimiento se lleva a cabo para cada uno de los que se encuentren asignados a esa sala en particular. De lo anterior deducimos que el número de becarios será variable, así como el número de botones necesarios, pero en este caso solo se crea un arreglo de botones y cada que se necesita uno es creado.

Como ya se mencionó los botones tienen el propósito de consultar y / o modificar la información de un becario en particular, por ello es necesario saber qué botones pertenecen a quién. Al tener un arreglo de botones podemos saber el intervalo de ellos asignados a un becario guardando el inicio y fin del mismo. Estos límites los podemos relacionar con el identificador único del becario. Se efectúa esta operación cuando se empiezan y terminan de agregar los botones. En la figura 5.63 lo podemos ver gráficamente.

Cuando un botón es accionado podemos saber precisamente cuál fue y que índice del arreglo tiene, por lo tanto podemos saber que identificador de becario corresponde.

El usuario puede configurar el sistema para que al hacer clic sobre el botón se acceda a una de las cuatro opciones posibles:

- Ver su Horario
- Modificar su Horario
- Cambiar de Asignación de Sala
- Ver información personal.

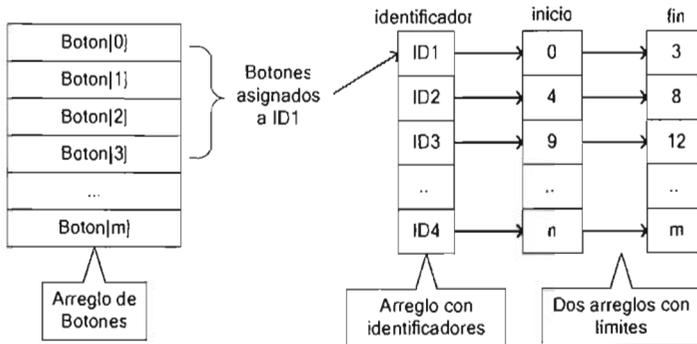


Figura 5.63. Arreglo de Botones

### 5.8.2.6 Sábana por Día

En esta caso tenemos una sábana con dimensiones más variables ya que el número de columnas que corresponde al número de salas puede variar, en la figura 5.64 podemos ver una descripción de la tabla.

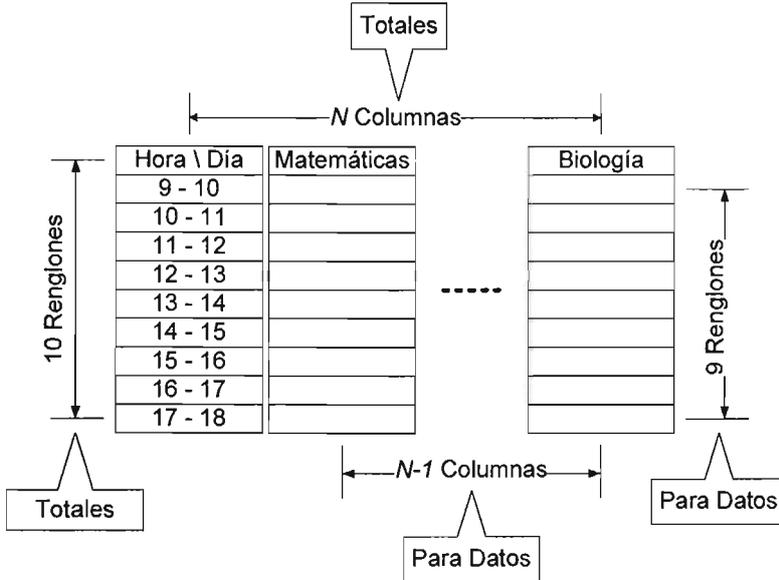


Figura 5.64. Descripción de Sabana por Día

En la figura anterior observamos que esta sábana crece de forma horizontal, se tiene una dependencia total del número de salas que se tengan almacenadas de la BD, todos los datos que se necesitan para crear esta sábana son los siguientes:

- El día que se desea consultar
- Número y nombre de salas almacenadas en la BD
- Todos lo becarios y su horario en este día en particular

El primer dato lo obtenemos por medio de una lista, el usuario selecciona el día deseado y lo siguiente es conocer cuantas salas están almacenadas en la BD, ya que, así tendremos las dimensiones de la tabla. También es necesario el identificador y nombre de cada una, puesto que necesitamos etiquetar cada una de las columnas y esto obviamente lo aremos con el nombre de las salas, el identificador será usado de forma interna.

Si recordamos que los resultados de una consulta a la BD pueden ser extraídos de forma cíclica y como ya lo mencionamos antes el crecimiento de la sábana es de manera horizontal, esto facilita el etiquetar las columnas, en un ciclo se puede etiquetar las columnas. Esto lo podemos observar en la figura 5.65, en

ella podemos ver que los resultados son colocados en un arreglo horizontal y este a su vez representa las cabeceras de la tabla.

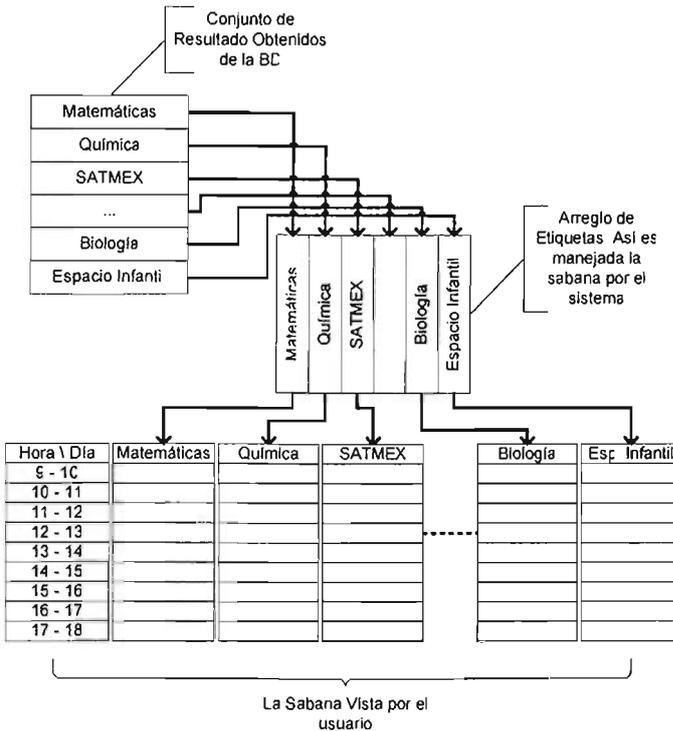


Figura 5.65. Etiquetando las Columnas

Es muy importante señalar que para el sistema y en términos de programación no se trata de un arreglo de etiquetas horizontal, se trata simplemente de un arreglo. El sentido horizontal o vertical se le da a la hora de ser mostrados en pantalla, es decir, a la hora de mostrarlos se hace en el sentido deseado.

El gran cambio es el acomodar a los becarios, en esta ocasión se tiene que agregar en varios arreglos y lo importante es saber en qué parte del arreglo le corresponde ser acomodado, esto lo solucionamos con la asignación a salas. El identificar de becario esta asociado con el de sala en la tabla becarios por sala, entonces creamos un arreglo temporal que tenga una relación uno a uno con el de etiquetas, así cuando agreguemos un becario a la sala empleamos el identificador de sala asociado a él y ubicaremos en que parte del arreglo deber colocado. En la figura 5.66 podemos ver una descripción gráfica de cómo se hace la reconstrucción del horario de becario de un solo día.

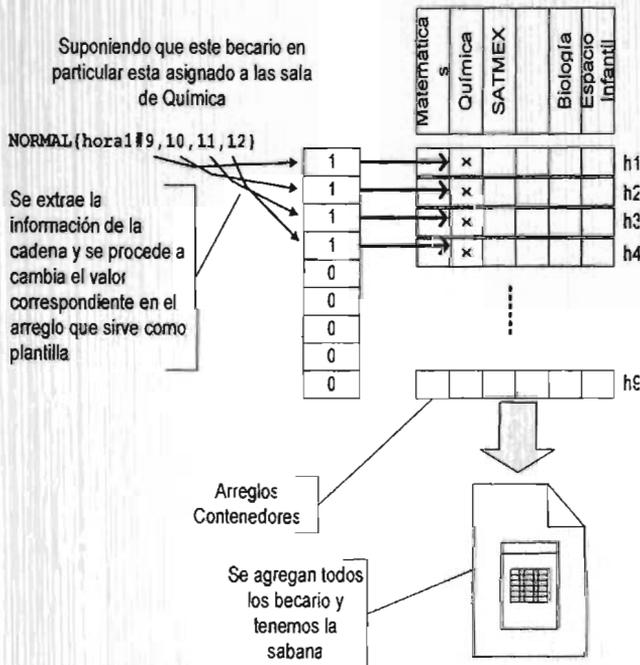


Figura 5.66. Agregando un Becario a la Sábana por Día

Recordemos que se usan arreglos contenedores para colocar a los becarios, así podemos colocar cada uno de los becario que vienen el día seleccionado. En este caso también se usa la misma técnica de usar botones para obtener más información de los becarios, no se hacen modificación se usa tal cual el caso de la sabana por sala, además se cuentan con los mismas opciones.

### 5.8.3 Mejorando Búsquedas

Normalmente cuando queremos hacer una búsqueda en una BD tenemos que señalar explícitamente el criterio de búsqueda, por ejemplo, si deseamos un libro y lo solicitamos en una librería o biblioteca podemos proporcionar el ISBN, el nombre del autor, el título, la editorial, etc. como criterios y así efectuar búsqueda. El Sistema de Becarios no es la excepción, es necesario que el usuario proporcione datos como el identificador, el nombre, apellido, número de faltas o retardos, etc. No podemos obtener información sin que el usuario proporcione datos, pero si podemos encubrir de cierta forma que se diga al sistema que tipo de dato proporciona, es decir, efectuar búsquedas sin necesidad de señalar explícitamente cuál es el criterio usado.

El usuario proporciona datos como el nombre o identificador sin necesidad de señalar que se está usando, el sistema es capaz de diferenciar entre ellos. Para poder lograr esto se hizo una adaptación del proceso de compilación. Podemos recordar se tienen varias etapas, las empleadas fueron: análisis léxico, sintáctico, semántico. No hablaremos de ellos para evitar alejarnos del tema central de este capítulo. Y continuando con las analogías podemos decir que se trata de un pequeño *Front End*, pero para poder comprender mejor veamos lo que es capaz de reconocer el sistema.

1. Un identificador único de becario (*id*)
2. Nombre de becario (*nom*)
3. Apellido paterno precedido de la etiqueta "*ap.*" (*ap.appat*)
4. Solicitar Todo el contenido de una tabla
5. Todos los becarios que tengan Faltas  $<, >, =, \diamond X$
6. Todos los becarios que tengan Retardos  $<, >, =, \diamond X$
7. Usar hasta tres criterios separados por coma, por ejemplo *id, nom, ap.appat*

Ahora veamos porque se hicieron las analogías, los pasos seguidos para estudiar la cadena de entrada (descritos de una forma muy general) son:

1. Identificar si se escribió algo válido, qué caso se trata y empezar a eliminar elementos superfluos:
  - a. hay un solo criterio
  - b. hay varios criterios
  - c. se solicita búsqueda por faltas

d. se solicita búsqueda por retardos

Esto se hace empleando 4 expresiones regulares:

- Expresión para el caso a

$[aA][pP][. ]?[a-zA-Z]+([0-9]+)?$

- Expresión para el caso b

$(([aA][pP][. ]?[a-zA-Z]+([0-9]+)?), )^*$   
 $(([aA][pP][. ]?[a-zA-Z]+([0-9]+)?))^*$

Aquí también se comienzan a extraer cada uno de los elementos.

- Expresión para el caso c

$[fF][aA][lL][tT][aA][ ]^*([<|>|?)[ ]^*[0-9]+|$   
 $[fF][aA][lL][tT][aA][ ]^*(>)[ ]^*[0-9]+|$   
 $[fF][aA][lL][tT][aA][ ]^*(=)[ ]^*[0-9]+$

- Expresión para el caso d

$[rR][eE][tT][aA][rR][dD][oO][ ]^*([<|>|?)[ ]^*[0-9]+|$   
 $[rR][eE][tT][aA][rR][dD][oO][ ]^*>[ ]^*[0-9]+|$   
 $[rR][eE][tT][aA][rR][dD][oO][ ]^*(=)[ ]^*[0-9]+$

## 2. Separa la información

a. Si es del tipo visto en 1.a se solicita el análisis para identificar qué es:

- Identificador
- Nombre
- Apellido paterno
- Todo

b. Si es tipo 1.b solicitar para cada uno de los elementos identificación:

- Identificador
- Nombre
- Apellido paterno
- Todo

En este caso si solicita “todo” como parte de los criterios inmediatamente se desconoce el resto y si hay más del máximo permitido solo se toma en cuenta a los primeros tres.

- c. Si es del tipo 1.c se extraen los tres tokens: “falta” “<,>=, >” y número de faltas. También se sustituye “falta” por el nombre del campo correspondiente en la BD.
  - d. Si es del tipo 1.d se extraen los tres tokens: “retardo” “<,>=, >” y número de retardos. También se sustituye “retardo” por el nombre del campo correspondiente en la BD.
3. Dar sentido a lo solicitado dependiendo de la tabla que se desea consultar y crear la instrucción SQL.

Observamos que se hace un análisis de la cadena, comprobamos su validez y damos significado de lo escrito, estos son los pasos hechos en un *Front End*. Dado el tamaño de opciones dadas al usuario podemos no seguir rigurosamente la secuencia de análisis de un compilador, es decir, primero el análisis léxico para continuar con el sintáctico y finalizar con el semántico. En este caso particular se efectúa el análisis léxico y parcialmente el sintáctico, este último se completa en el paso 2.

Hay dos puntos importantes que aclarar, el primero se refiere al como se sabe si es un nombre, apellido o identificador, para ellos analizamos como está constituido, por ejemplo:

Nombre: Juan  
Apellido Paterno: ap.Pérez  
Identificador: GEHJ8104090001

Como podemos ver el nombre esta constituido por letras y el apellido paterno también, es por eso que se solicita anteponer la etiqueta “ap.”, un identificador esta compuesto por letras y números. Esto tiene una desventaja, si ponemos una palabra como “tesis” esto es correcto para el sistema y se procederá a buscar en la BD, pero en general es algo tolerable. Se le dio preferencia al nombre ya que como parte de nuestra metodología de trabajo estamos en constante comunicación con el usuario y nos indico que hace mayor uso del nombre de los becarios para buscar su información, el apellido paterno pasa a segundo plano.

Si quisiéramos erradicar esta desventaja para así tener un sistema capaz de diferenciar entre un nombre y un apellido paterno sería necesario contar con una Base de Conocimiento, pero su desarrollo se aleja del propósito del Sistema.

El segundo y último punto es el referente a “dar sentido”, con el análisis hecho en los pasos 1 y 2 podemos empezar a deducir que es lo que se quiere buscar pero no en donde, esto está ligado a ofrecer resultados útiles al usuario. Podemos dar sentido a lo que se busca si sabemos desde qué ficha se esta solicitando información, por ejemplo, si se solicita información en la ficha de asistencia no tiene sentido informar al usuario del teléfono o dirección del becario. Entonces dependiendo desde donde se esta solicitando información se creará una instrucción SQL que obtenga resultados de la tabla correspondiente a la ficha que hizo la solicitud. A continuación se describe por ficha que tiene sentido buscar en las tablas correspondientes:

Ficha Becarios:

- a. Identificador
- b. Nombre
- c. Apellido paterno
- d. Varios criterios
- e. Todo

Ficha Asistencia

- a. Identificador
- b. Nombre
- c. Apellido paterno
- d. Varios criterios
- e. Todo
- f. Faltas
- g. Retardos

Ficha Sabana: Consultar Becarios Asignados a Sala

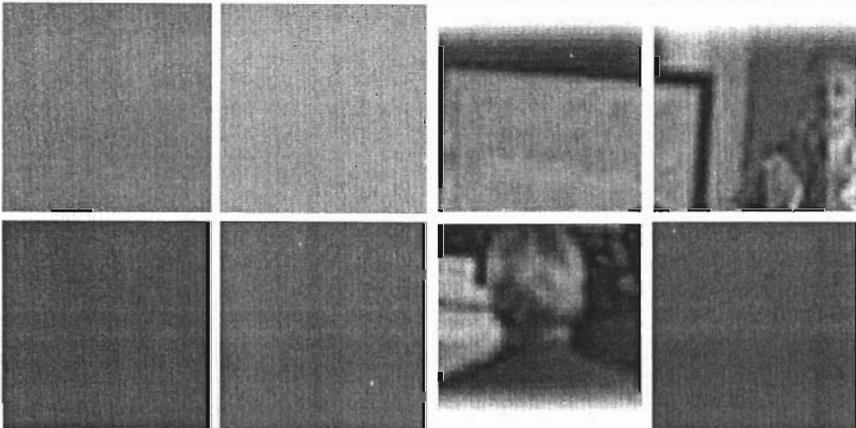
- a. Identificador
- b. Nombre
- c. Apellido paterno
- d. Varios criterios
- e. Todo

Para poder consultar el horario es necesario proporcionar su identificador.

Implantación  
del  
Sistema

Capítulo

6





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.

# CAPÍTULO 6

## IMPLANTACIÓN DEL SISTEMA

*En este capítulo realicemos la implantación del sistema así como la puesta a punta del mismo se verán los requerimientos de Hardware y Software que se tienen y los que se necesitan para llevar a cabo esta tarea tan importante y con la cual concluiremos todo el desarrollo del Sistema de Control de Becarios.*

## 6. Implantación del Sistema

Una vez terminada la etapa de desarrollo vamos a definir la etapa de implantación del Sistema de Control de Becarios para la DGDC, hemos podido realizar las pruebas del sistema conforme hemos desarrollado todas las aplicaciones, esto hace que nosotros tengamos la facilidad de prueba.

Para un ingeniero en computación (enfocado al software) un programa o aplicación se diseña con la finalidad de ser probado fácilmente, es decir, el ingeniero ya tiene la idea en la mente de hacer su sistema a prueba de todo, pero ¿que es la facilidad de prueba? James Bach<sup>1</sup> describe la facilidad de prueba de la siguiente manera:

“La facilidad de prueba del software es simplemente la facilidad con la que se puede probar un programa de computadora. Como la prueba es tan profundamente difícil merece la pena saber que se puede hacer para hacerlo mas sencillo. A veces los programadores están dispuestos a hacer cosas que faciliten el proceso de prueba y una lista de comprobación de los posibles puntos de diseño, características, etc., puede ser útil a la hora de negociar con ellos”

¿Cómo sabemos si un programa o sistema es fácil de probar?, cuando cumple con lo siguiente:

- **Operatividad:** cuando mejor funcione, más eficientemente se puede probar.
  - El sistema tiene pocos errores (los errores añaden sobrecarga de análisis o procesos).
  - Ningún error bloquea la ejecución de las pruebas.
  - El producto evoluciona en fases funcionales (permite simultanear el desarrollo y las pruebas).
  -
- **Observabilidad:** lo que ves es lo que pruebas.
  - Se genera una salida distinta para cada entrada.
  - Todos los factores que afectan a los resultados están visibles.
  - Un resultado incorrecto se identifica fácilmente.

---

<sup>1</sup> James Bach, 1994

- Los errores internos se detectan automáticamente a través de mecanismos de auto comprobación.
- Se informa automáticamente de los errores internos.
- **Controlabilidad:** cuando mejor podamos controlar el software, más se puede automatizar y optimizar.
  - Todos los resultados posibles se pueden generar a través de alguna combinación de entrada.
- **Capacidad de descomposición:** controlando el ámbito de las pruebas, podemos aislar más rápidamente los problemas y llevar a cabo mejores pruebas de regresión.
  - El sistema (software) esta construido con módulos independientes.
  - Los módulos del software se pueden probar independientemente.
- **Simplicidad:** cuando menos haya que probar, más rápidamente podemos probarlo.
  - Simplicidad funcional (por ejemplo, el conjunto de características es el mínimo necesario para cumplir los requisitos)
  - Simplicidad estructural (por ejemplo, la arquitectura es modularizada para limitar la propagación de fallos)
  - Simplicidad del código (por ejemplo, se adopta un estándar de código para facilitar la inspección y el mantenimiento).
- **Estabilidad:** cuando menos cambios, menos interrupciones a las pruebas.
  - Los cambios del software son infrecuentes.
  - Los cambios del software están controlados.
  - Los cambios del software no invalidan las pruebas existentes.
  - El software se recupera bien de los fallos.
- **Facilidad de comprensión:** cuanta más información tengamos, más inteligentes serán las pruebas.
  - El diseño se ha entendido perfectamente.
  - Las dependencias entre los componentes internos, externos y compartimos se han entendido perfectamente.
  - Se han comunicado los cambios del diseño.

- La documentación técnica es accesible.
- La documentación técnica esta bien organizada.
- La documentación técnica es específica y detallada.
- La documentación técnica es exacta.

Nuestro sistema cumple con todas estas características para la prueba del mismo.

Para la etapa de implantación se debe planear la disponibilidad de los recursos de Hardware, Software y de Comunicaciones para el arranque de las aplicaciones. En esta etapa también se debe de calendarizar actividades de capacitación de usuarios finales. Permaneceremos en el sitio de desarrollo para detectar fallas de la aplicación, así como inconvenientes que resulten en la operación del Sistema.

## **6.1 Requerimientos de Software**

Los requerimientos de software que hemos mencionado desde el capítulo 1 son aquellos con los que ya se cuenta en la dependencia tales como los sistemas operativos:

- Windows XP
- OS X
- Máquina Virtual Java

La máquina virtual es libre y se puede descargar de la página de Sun Microsystems. Se dispone de estos sistemas operativos además de la máquina virtual.

En servidores solicitamos lo siguiente:

- Java 2
- Servidor de aplicaciones JSP Tomcat
- Base de Datos Mysql

Estas aplicaciones también son libres están en los recursos de la página de Sun Microsystems, en la Página de Apache y en la de Mysql. Se cuenta con servidores capaces de soportar estas aplicaciones

## **6.2 Requerimientos de Hardware**

Debido a que en la dependencia, como ya se vio, se cuenta con diferentes plataformas pero no se cuenta con una cámara web para la toma de la fotografía del becario para realizar su credencial, además se requiere de una impresión de alta calidad se adquiere lo siguiente:

Los requerimientos de Hardware son los siguientes:

- Una computadora PC para alojar el sistema de credenciales
- Un scanner para la lectura de código de barras conectado a la PC
- Una Web Cam para la toma de la fotografía del Becario
- Una impresora Láser a Color para la impresión del gafete o credencial de acceso de los becarios.

Ya se cuenta con la PC en la cual estará el scanner conectado para la toma de la asistencia en el sitio. También se cuenta con el scanner, impresoras y cámara Web por lo que se realiza la instalación del software después de haber superado las pruebas y los requerimientos.

La instalación se llevó a cabo sin ningún problema y se realizaron todos los archivos ejecutables para el funcionamiento de todas las aplicaciones realizadas.

## **6.3 Capacitación de Los Usuarios**

Una vez instalado el Sistema de Control de Becarios la tarea siguiente es capacitar a los usuarios para el uso del sistema los usuarios son 3 el jefe de Becarios y dos de sus asistentes, para esto contamos con la disposición de tales usuarios así como con las herramientas necesarias y con los manuales del sistema (Apéndices de esta tesis).

El tiempo que se empleo para la capacitación del usuario fue realmente bajo, ya que el sistema está hecho de forma muy intuitiva y por lo tanto fácil de aprender, el tiempo de capacitación se ilustra en la siguiente tabla:

Sistema	Descripción	Tiempo en Hrs.
C2WCAM	<ul style="list-style-type: none"> <li>• Agregar a los Becarios a la Base de Datos Local</li> </ul>	2
	<ul style="list-style-type: none"> <li>• Como hacer una credencial para el Becario.</li> </ul>	2
	<hr style="width: 100%;"/>	
Sistema de Control de Becarios fase Red	<ul style="list-style-type: none"> <li>• Consultar Datos de Becarios</li> </ul>	1
	<ul style="list-style-type: none"> <li>• Consultar Horario de Becario</li> </ul>	1
	<ul style="list-style-type: none"> <li>• Modificar Datos de los Becarios</li> </ul>	2
	<ul style="list-style-type: none"> <li>• Imprimir Listas</li> </ul>	1
	<ul style="list-style-type: none"> <li>• Imprimir Sábanas</li> </ul>	1
	<ul style="list-style-type: none"> <li>• Crear usuarios para acceso al Sistema</li> </ul>	1
<hr style="width: 100%;"/>		Subtotal: 6
Sistema de Control de Becarios fase Red Local	<ul style="list-style-type: none"> <li>• Posibles Errores en el Arranque del Sistema</li> </ul>	1
	<ul style="list-style-type: none"> <li>• Migrar Base de datos Local al Servidor</li> </ul>	2
	<ul style="list-style-type: none"> <li>• Hacer Consultas Avanzadas y Rápidas</li> </ul>	1
	<ul style="list-style-type: none"> <li>• Modificar Datos de los Becarios</li> </ul>	1
	<ul style="list-style-type: none"> <li>• Asignación de Becarios a las Salas</li> </ul>	1
	<ul style="list-style-type: none"> <li>• Asignación de Becarios Suplentes</li> </ul>	2
	<ul style="list-style-type: none"> <li>• Mostrar Sabanas</li> </ul>	1
	<hr style="width: 100%;"/>	

Tabla 6.1. Tabla de descripción de actividades y tiempo de capacitación

Como se deja ver en la tabla anterior la capacitación se llevo a cabo con un total aproximado de 19 horas en donde el usuario pudo conocer el sistema y las principales opciones las cuales utilizará con frecuencia.

La liberación del Sistema de Control de Becarios se dio el día 1° de Junio de 2005 y vamos a permanecer en el sitio de desarrollo por dos semanas para cualquier posible comentario o pregunta a cerca del sistema, también se deja a los usuarios los manuales correspondientes (que son los Apéndices de esta tesis), para cualquier duda que resulte al utilizar este sistema.

## **6.4 Políticas de Seguridad**

Una vez realizada la capacitación de los Usuarios implantaremos políticas de Seguridad en el Sistema de control de Becarios Pero veamos primero en que consiste una política de seguridad.

El término política de seguridad se suele definir como el conjunto de requisitos definidos por los responsables directos o indirectos de un sistema que indica en términos generales qué está y qué no está permitido en el área de seguridad durante la operación general de dicho sistema. Al tratarse de 'términos generales', aplicables a situaciones o recursos muy diversos, suele ser necesario refinar los requisitos de la política para convertirlos en indicaciones precisas de qué es lo permitido y lo denegado en cierta parte de la operación del sistema, lo que se denomina política de aplicación específica.

Una política de seguridad puede ser **prohibitiva**, si todo lo que no está expresamente permitido está denegado, o **permisiva**, si todo lo que no está expresamente prohibido está permitido. Evidentemente la primera aproximación es mucho mejor que la segunda de cara a mantener la seguridad de un sistema; en este caso la política contemplaría todas las actividades que se pueden realizar en los sistemas, y el resto (las no contempladas) serían consideradas ilegales.

Cualquier política ha de contemplar seis elementos claves en la seguridad de un sistema informático:

- Disponibilidad: Es necesario garantizar que los recursos del sistema se encontrarán disponibles cuando se necesitan, especialmente la información crítica.
- Utilidad: Los recursos del sistema y la información manejada en el mismo ha de ser útil para alguna función.
- Integridad: La información del sistema ha de estar disponible tal y como se almacenó por un agente autorizado.
- Autenticidad: El sistema ha de ser capaz de verificar la identidad de sus usuarios, y los usuarios la del sistema.
- Confidencialidad: La información sólo ha de estar disponible para agentes autorizados, especialmente su propietario.
- Posesión: Los propietarios de un sistema han de ser capaces de controlarlo en todo momento; perder este control en favor de un usuario malicioso compromete la seguridad del sistema hacia el resto de usuarios.

#### **6.4.1 Políticas de Seguridad Para el Sistema de Control de Becarios**

Las políticas de seguridad que sugerimos para el sistema de control de Becarios se expresa en los siguientes puntos:

En cuanto a los Usuarios del Sistema:

1. El Jefe de Becarios y dos personas más que manejan el departamento de Becarios son los únicos que pueden tener acceso al SCBDGDC por lo que en ellos radica gran parte de la seguridad del sistema.
2. Los usuarios podrán manipular la información de los Becarios:
  - a. Podrán Asignar, Re-asignar, y modificar los datos personales de los Becarios, dar de alta horarios dentro del sistema, crear nuevas salas, dar de alta a becarios dentro del sistema. Generar un nuevo Gafete o una reposición del mismo a petición de los Becarios
  - b. No podrán dar a otra persona las contraseñas de acceso al Sistema de control de Becarios.
  - c. No podrán desinstalar las aplicaciones instaladas para el correcto funcionamiento del sistema.
3. El usuario al terminar de usar las aplicaciones del Sistema de Control de Becarios Red Local deberá salir de la sesión correctamente.
4. Los usuarios deberán de memorizar sus contraseñas de acceso.

En cuanto a los Becarios:

1. Los **Becarios** al momento de pasar lista mostrarán su gafete o credencial con su código de barras por el lector.
2. No podrá de ninguna forma jugar con el lector de códigos ni pasar cualquier otro código fuera del de su gafete o credencial.
3. Los Becarios podrán consultar su horario vía red dentro del museo
4. Los Becarios deberán pasar lista minutos antes de su hora de llegada formal y hasta antes de cumplir seis minutos después de su hora de entrada.
5. Cualquier falla en el lector de códigos deberán reportarla al encargado del área de Becarios
6. Los Becarios no podrán pasar lista por otros Becarios.

En cuanto al Encargado de Administrar la Base de Datos:

1. El encargado de administrar la Base de Datos del Sistema de Control de Becarios deberá asegurar que el servidor se encuentre en optimas condiciones de funcionamiento para que las transacciones y consultas de información sean rápidas y eficientes
2. El administrador de Servidores deberá realizar los respaldos de la información de la Base de Datos del Sistema de Control de Becarios.
3. Deberá entregar al jefe de Becarios el respaldo de la Base de datos del Sistema de Control de Becarios una vez cada semana en un CD con la fecha del respaldo.

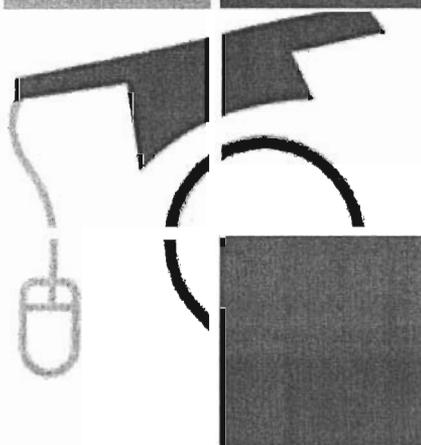
En cuanto al Área de Soporte:

1. El Área de Soporte técnico deberá asegurarse del correcto funcionamiento de las computadoras en las cuales reside el Sistema de Control de Becarios.
2. Soporte deberá actualizar el Antivirus periódicamente para evitar algún problema con las aplicaciones del Sistema de Control de Becarios.
3. El administrador de Redes debe de asegurar las conexiones adecuadas para la transmisión de datos.

# Conclusiones

Capítulo

7





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.

# **CAPÍTULO 7**

# **CONCLUSIONES**



## 7. Conclusiones

En un principio la necesidad básica del usuario era tener una forma de agilizar el procedimiento para tomar lista a los becarios de la DGDC así como tener una forma más ágil de manejar su sabana<sup>1</sup>. Sabían que un sistema, algún tipo de lector y una base de datos se podrían ver involucrados; el como no era claro, mucho menos todo lo que fuera necesario para lograr dicho objetivo.

Un software con conexión al lector y la base podía cumplir con los requerimientos básicos, pero gracias a la visión obtenida a lo largo de nuestra carrera nos facilitó ver el potencial de la problemática, es decir, crear un sistema que abarcara más procesos del área de becarios y agilizarlos, en otras palabras: proveer al usuario de una solución.

Hasta antes de este sistema, Universum no contaba con sistemas hechos a la medida, ahora cuentan con una solución que no solamente cumple con las perspectivas originales, las rebasa. El Sistema de Control de Becarios puede ser visto como la unión de tres subsistemas<sup>2</sup> que en conjunto forman esta solución, la cual cumple por los siguientes motivos:

- Originalmente cuando se aceptaba a un nuevo becario era necesario llenar algunas formas y solicitar una foto para poder crear su gafete en una máquina de escribir, ahora cuando se solicita estos datos se crea el gafete de forma automática. Además se tienen listos para ser migrados a la base ubicada en el servidor.
- En el procedimiento manual los becarios escribían su nombre en una lista, cuando se excedía el límite para ser considerado como una asistencia, alguien del área tenía que dibujar una raya para saber quien tendría retardo para delimitar asistencia de retardo en la lista, al final del día se efectuaba el vaciado de información, podían pasar varios días para que una becario fuera informado sobre acumulación de faltas o en el peor de los casos de una baja temporal. En estos momentos todo ocurre en segundos, el becario es informado y se actualiza la base de datos mientras los usuarios del sistema pueden ver en sus pantallas en

---

<sup>1</sup> Ver capítulos 1, 4 y 5.

<sup>2</sup> Ver capítulo 5

- tiempo real quien llega, además cuentan con una sabana del día cien por ciento actualizada, todo de forma automática.
- El crear una sabana representaba cotejar becario por becario su horario y sala de asignación, este procedimiento se efectuaba sala por sala y día por día, al final se tienen alrededor de 20. Un cambio de horario implicaba volver a hacer la sabana por día y sala que se vieran afectadas, además era necesario mandar una copia a todos los usuarios. Ahora solo basta modificar el horario del becario o la asignación de sala y las modificaciones se reflejarán en todas las sabanas afectadas de forma automática, también todos los usuarios pueden ver dichas modificaciones sin necesidad de ser avisados. Además pueden consultar y modificar la información de múltiples formas, con un solo clic en la sabana se puede ver, así como cambiar información personal, horario, sala asignada.
  - Si los usuarios no se encuentran en sus computadoras y necesitan información pueden consultarla, incluso modificarla en algunos casos desde otra que tenga conexión a Internet, todo de forma segura. Al final cuentan con múltiples posibilidades para acceder y efectuar cambios a la información.

Fue posible brindar las características antes mencionadas gracias a la creación de una forma de representar el horario del becario de forma compacta y fácil de almacenar en la base de datos. Ahora podemos ver que se cumplieron con los propósitos originales y no solo eso. Crear la credencial con una cámara web no estaba dentro de estos planes, pero fue considerado para mejorar los procesos y la calidad de la solución. En el mercado existen programas que pueden hacerlo, en general es poco común tener la posibilidad de tener una vista previa o conexión en tiempo real entre el programa y la cámara, una base de datos interna es otra opción, es posible tener el software con estas características pero es necesario solicitar los módulos necesarios, esto incrementa el costo total. En nuestro caso se tienen estas características como parte de la solución, usando una interfaz de usuario sencilla de usar.

Por otra parte, el tener información en tiempo real interconectando diferentes plataformas fue de gran importancia para tener la solución final. El Servidor de Información de Asistencia se encuentra en una PC con la plataforma Windows, la base de datos a la que se conecta esta sobre Linux y la mayoría de las terminales cuentan con la plataforma Mac OSX, pero es posible enviar

y recibir información dentro de la solución. Contar con información de la asistencia en tiempo real le da más armas al área de becarios para planificar y efectuar las actividades que sean requeridas por el museo con la certeza de saber con que becarios cuentan.

Son las partes que podemos considerar como ayudas extras que mejoran la experiencia de usuario con el sistema, por ejemplo, las terminales tienen la capacidad de diferenciar entre los criterios que se desean usar para una búsqueda. Cuando se requiere información el usuario puede teclear un identificador o nombre y el sistema lo identificará para efectuar una búsqueda más precisa. Estas características son inherentes de las terminales, recordemos la facilidad de poder consultar la información desde cualquier máquina usando Internet, esta característica se tiene como una herramienta de todo el conjunto que conforman la solución, la cuál también puede ser empleada por los becarios para consultar sólo sus datos.

Cuando se habla de Ingeniería de Software, Desarrollo, etc. en la bibliografía regularmente el usuario es un ente abstracto, pone los requerimientos y usa el software, no se hace mucha referencia en la gran importancia que tiene para desarrollo. Resulta de mucha relevancia tener una buena comunicación con él o ellos, a parte de facilitar un mejor entendimiento de la problemática obtendremos datos no relacionados con el motivo original por la cual solicitaron nuestros servicios, estos datos pueden ser muy valiosos, en particular, para mejorar la experiencia del usuario con el sistema, es decir, saber su nivel de conocimientos en cuanto a el área de computación se refiere. Por ejemplo, se puede llegar a crear una interfaz que no sea fácilmente manejable por el usuario, lo cual, lejos de facilitar su labor puede complicarla. Gracias a la buena comunicación con el personal del área de becarios se facilitó la comprensión de la problemática y ayudó para determinar que la interfaz gráfica de usuario tenía que ser simple. Además, gracias a esta buena comunicación logramos aumentar la cultura en tecnologías de la información (TI) de algunos usuarios, lo cual motivo por un lado a incrementar el entendimiento del sistema y las exigencias sobre el mismo, ello se ve reflejado en una solución con la cual el usuario está satisfecho.

Después de completar la solución el siguiente paso natural es pensar hacia el futuro, como es bien conocido en el mundo TI la evolución no ha parado de traer cambios y mejoras, por tal motivo se debe pensar como afectarán en le

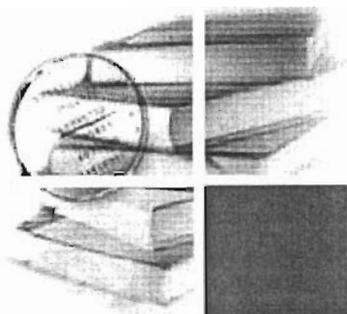
ciclo de vida del software. El primer punto fue pensar en el software para crear los gafetes o credenciales, en estos momentos la tecnología del .NET Framework ya se consolidó y la hemos tomado en cuenta para migrar este software, ya que, las herramientas que Microsoft provee para desarrollar con esta tecnología ayudan en gran medida para mejorar la conexión con bases de datos y tener componentes más robustos entre otras cosas. Con estos cambios se logrará mantener vigente el software por más tiempo, mejorar procesos y facilitar el manteniendo del código. Se piensa en mantener el código por otro motivo ligado a los planes a futuro, esta solución resuelve una problemática particular, pero podemos pensar en generalizar, es decir, pasar del caso particular al general, existen muchas instituciones con programas de becarios, poder abstraer la funcionalidad facilitará la creación de una solución que pueda ser adoptada en otras instituciones, así se le puede dar seguimiento y una continua serie de mejoras, los usuarios se verán beneficiados. Incluso podemos pensar en extrapolar a otros casos en los que interese la asistencia y la generación de credenciales, como pueden ser escuelas e instituciones.

El dar seguimiento a esta solución la ayudará a crecer, mejorar cada vez más la experiencia del usuario, su desempeño, calidad y a mantenerse en el camino de la evolución inherente al mundo que pertenece, el de las Tecnologías de la Información.

# Bibliografía

Capítulo

8





Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**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.

# **CAPÍTULO 8**

# **BIBLIOGRAFÍA**

## **Bibliografía**

Enciclopedia de Microsoft Visual Basic 6.0

Francisco Javier Ceballos

Ed. Alfa Omega

Ingeniería del Software

Roger S. Pressman

5° edición

Ed. McGraw Hill

Apuntes de Redes

MsC. Marco Antonio Vigueras Villaseñor

Prof. Facultad de Ingeniería

UNAM

Reengineering, The Corporation

Hammer and Champy

ED. Harper Bussiness.

Java 2 Manual de Referencia

Herbert Schildt

4° Edición

Ed. McGraw Hill

Cómo Programar en Java

Harvey and Paul M. Deitel

5° Edición

Ed. Prentice Hall

MySQL Edición Especial

Paul DuBois

Ed. Prentice Hall

SQL y Java

Guía para SQLJ, JDBC y tecnologías relacionadas

Jim Melton, Andrew Eisenberg

Ed. Alfaomega Ra-Ma

JSP Manual de Referencia

Phil Hanna

Ed. McGraw Hill

Manual de MySQL en Inglés

Traducción del tema 6.2 (*Column Types*)

Referencias Electrónicas

[http://fmc.axarnet.es/redes/tema\\_06.htm](http://fmc.axarnet.es/redes/tema_06.htm)

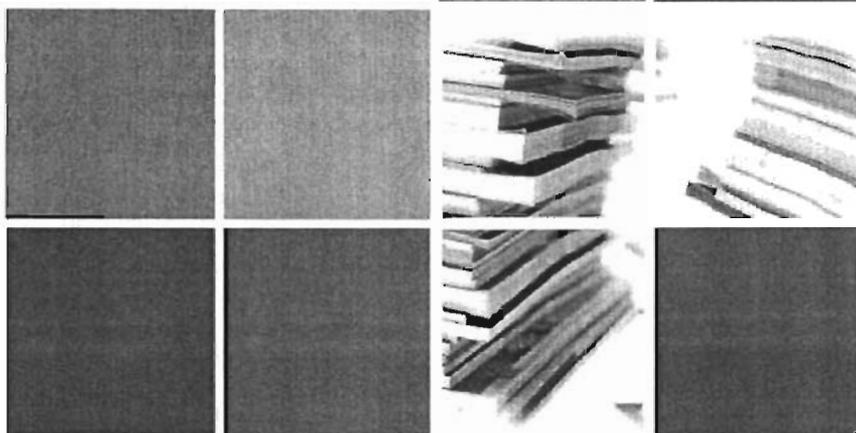
<http://www.mysql-hispano.org/>

<http://www.apache.org/>

# Apéndices

Capítulo

9





Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**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.

# **Apéndice A**

## **Manual de Usuario C<sup>2</sup>WCAM**



**MANUAL DE USUARIO**

**SISTEMA PARA LA DGDC**

**REALIZADO POR:  
YAIR GÓMEZ MENDEZ  
J. ROBERTO HERNÁNDEZ MOLINA**

Índice:	Página
 1. Guía para el usuario	04
 Nota Importante	
 2. ¿Cómo realizamos una credencial Para el Becario?	06
 Paso 1	06
 Paso 2	08
 Nota Importante	
 Paso 3	10
 Paso 4	11
 Nota Importante	
 3. Base de Datos	15
 Nota Importante	

 <b>4. Configuración</b>	<b>21</b>
 a) ¿Cómo agrego un usuario al sistema?	<b>22</b>
 b) ¿Cómo modifico los datos de un usuario?	<b>22</b>
 c) ¿Cómo elimino a un usuario registrado en el sistema?	<b>23</b>
 <b>5. Herramientas y Controles Especiales</b>	<b>24</b>
 <b>Nota Importante</b>	



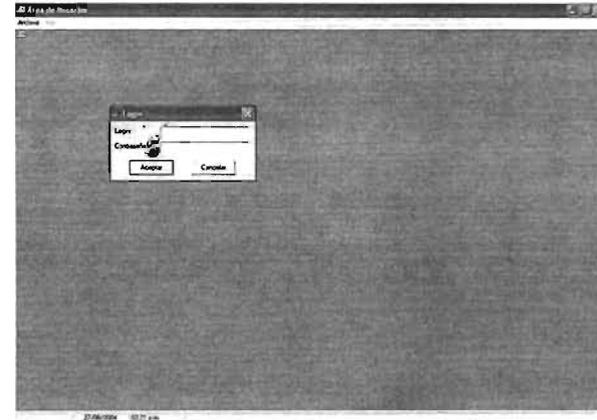
## 1. Guía para el usuario:

### Presentación

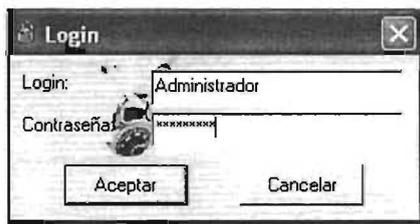


En la pantalla Principal Área de Becarios, la primera venta en aparecer es la denominada Login:

Aquí encontramos el modulo de seguridad.



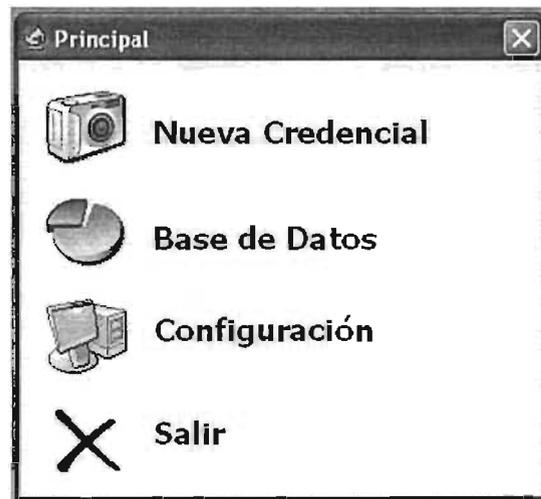
En esta ventana pondremos el Login o nombre de usuario, posteriormente la contraseña, después de dar Clic en Aceptar tendremos acceso al sistema. Ejemplo:



 **Nota importante:**

Cabe destacar que el ingreso al Login y a la Contraseña es sensible a las letras mayúsculas y minúsculas, es decir, reconoce tanto minúsculas como mayúsculas.

A continuación se despliega la pantalla Principal:  
En donde tenemos:



## ★ 2. ¿Cómo realizamos una credencial para el Becario?

### 👤 Paso 1:

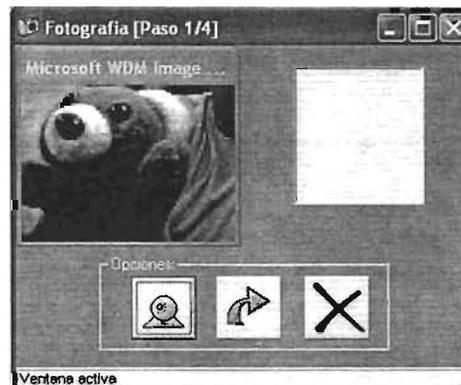
Comencemos por ver la Pantalla Nueva Credencial:

En este modulo veremos como es un proceso para realizar la credencial y la captura de los datos personales del Becario

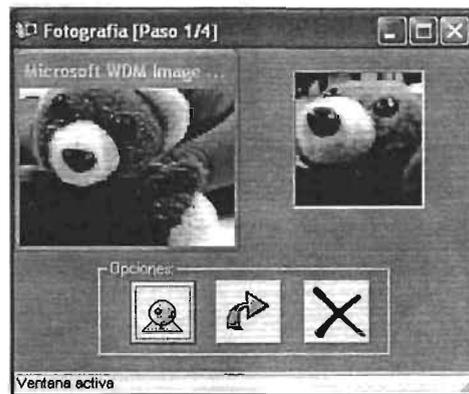
Aquí podremos realizar la captura de la fotografía del Becario, al dar clic en el botón



de la Cámara Web.

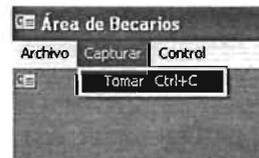


En la siguiente ventana se mostrara la foto que será la que tendremos para registrar los datos del becario. Tal como se muestra a continuación. Para tomar la foto de forma adecuada recomendamos que se ubique en la esquina superior izquierda de la pantalla Microsoft WDM Image.



Con el icono  cancelamos la captura y el registro del becario.

Con el icono  comenzamos con la captura de los datos Personales del Becario y continuamos con el paso 2.



Cabe señalar que la captura de la fotografía del becario también se puede realizar desde la barra de menú como se muestra. O con las teclas Ctrl+C.

Pasamos a la siguiente pantalla:

## Paso 2:

Aquí se capturan todos los datos personales de los Becarios empezando por el Apellido Paterno, Apellido Materno y Nombre(s), para pasar de una caja de Texto a otra hay dos formas:

1. Con la tecla Tab de nuestro teclado.
2. Con solo dar un clic sobre la caja de texto en la cual queremos agregar los datos.

Para verificar el RFC del Becario solo seleccionamos la Casilla "¿Verificar RFC?" esta casilla se encuentra en la parte inferior derecha de la pantalla. Aquí se mostrará el RFC actual, si deseamos cambiar el RFC solo seleccionamos la casilla **No** y en la caja de texto modificamos el RFC como se muestra a continuación.

## Nota importante:

En donde se seleccionan las listas de Fecha de nacimiento, Estado Civil y Delegación, al momento de seleccionar una deben aparecer sombreadas en Color Azul, de lo contrario no están seleccionadas. Como se muestra:

Fecha de Nacimiento:  Estado Civil:

Día  Mes  Año Edad:  Estado Civil:

**Domicilio**

C.P. Teléfono

Correo electrónico

Al dar clic en el texto **continuar**  Continuar pasamos a la siguiente pantalla que será el paso 3.

## Paso 3:

Aquí capturamos los datos de la preparación académica.

Al seleccionar la casilla **Idioma 1** podemos capturar el idioma y el porcentaje de habla, lectura y escritura del becario; en caso de que tenga conocimiento de otro idioma se selecciona la casilla **Idioma 2** y se realiza la misma operación de captura del siguiente idioma.

Al igual que en el paso anterior en Nota importante, la lista **Fecha de ingreso** se debe seleccionar y quedar sombreada en azul.

En **Horario Disponible:**

**Sala o Área** seleccionamos el área o sala donde se asignara al Becario.

Así mismo con las casillas (lunes, martes, miércoles, jueves, viernes, sábado, domingos), seleccionamos en que días se encontrara en esa sala el Becario.

En Otras Actividades se realiza la captura de cualquier otra actividad que realice el Becario. Como se muestra:

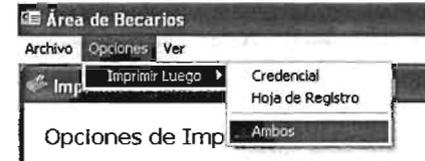
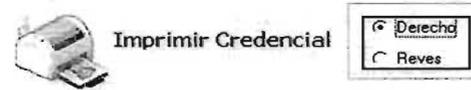
## Paso 4:

En este paso se imprime el **Registro de Ingreso, La Credencial del Becario**. Y se cuenta con las opciones **Continuar Agregando** y **Terminar**.



### En **Imprimir Credencial**

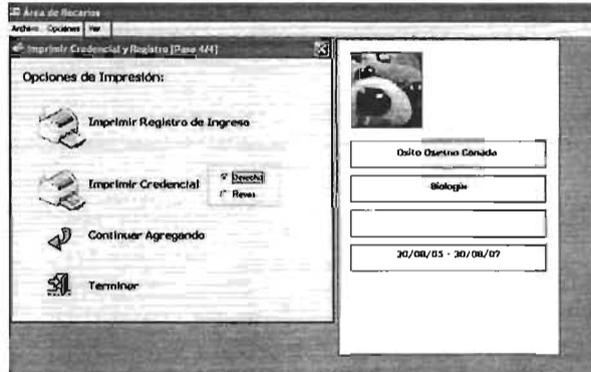
Se imprimirá la Credencial del Becario en las casillas de selección indicamos que lado de la credencial se imprimirá como se muestra en la figura. Derecho y Revés.



En la barra de Menú tenemos la opción de Imprimir luego si en ese momento

no contamos con una impresora para poder generar la Credencial del Becario, aquí podemos imprimir la **Credencial**, la **Hoja de Registro** o **Ambos** documentos.

En la barra de Menú tenemos la opción Ver esta opción genera la vista previa de nuestra Credencial, como se muestra.



El Reporte de impresión se da al terminar con el modulo de impresión en el paso 4.

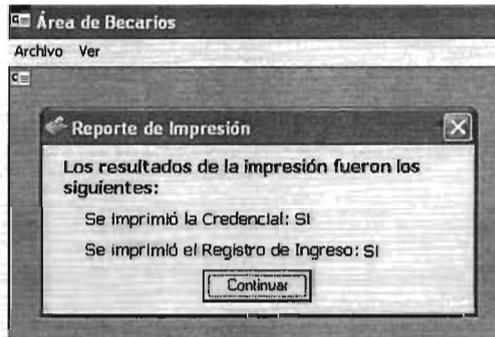
Así es como exitosamente podremos obtener la Credencial del Becario.

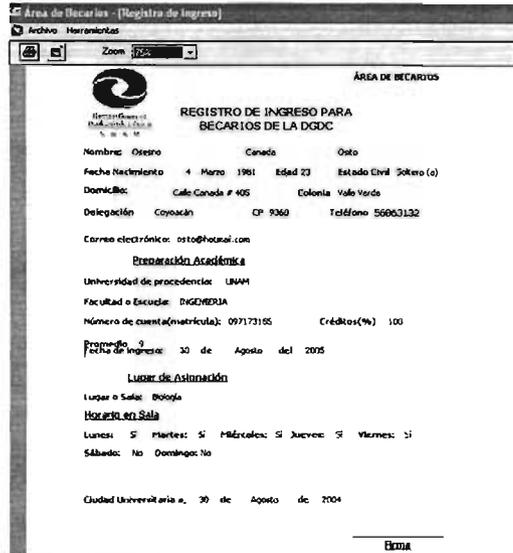
Para imprimir el Reporte **Registro de Ingreso**



Imprimir Registro de Ingreso

**Registro de Ingreso**, solo damos clic en el texto y aquí se genera el Reporte siguiente:





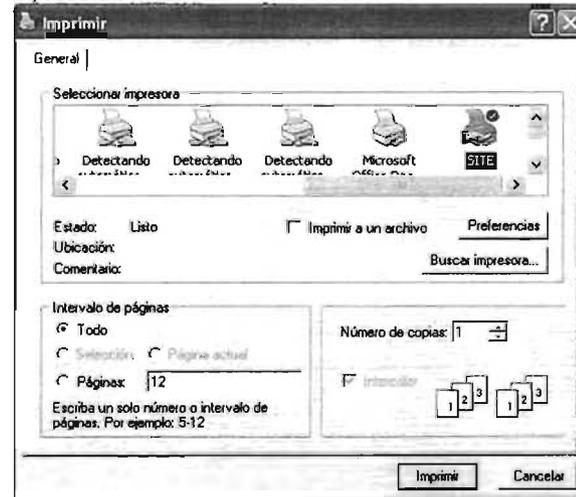
Para visualizar los datos del Becario que estamos agregando en ese momento solo hay que dar clic al botón **M** para llegar a la posición deseada este botón esta ubicado en la parte inferior izquierda del reporte:



Para realizar la impresión solo damos clic al botón , ubicado en la barra de herramientas.



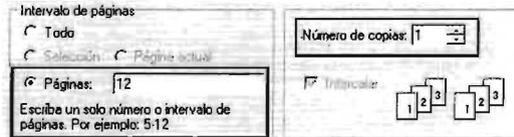
Una vez realizado esto se despliega la opción de las impresoras instaladas como se visualiza:



Damos clic en al botón **Imprimir** y así imprimimos el reporte **Registro de Ingreso**.

### **Nota importante:**

Es importante hacer notar que para que el reporte se imprima en las opciones de impresión debemos seleccionar las opciones adecuadas esto es, si queremos imprimir el reporte número 12 entonces seleccionaremos la casilla Páginas: y el número en este caso 12, así como el número de copias que requerimos.



Intervalo de páginas

Todo

Selección  Página actual

Páginas: 12

Escibe un solo número o intervalo de páginas. Por ejemplo: 5-12

Número de copias: 1

Impresión

1 2 3 1 2 3

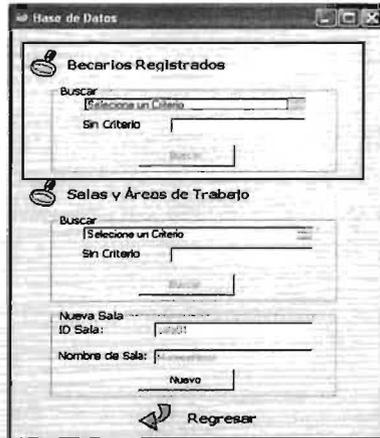


## 3. Bases de Datos



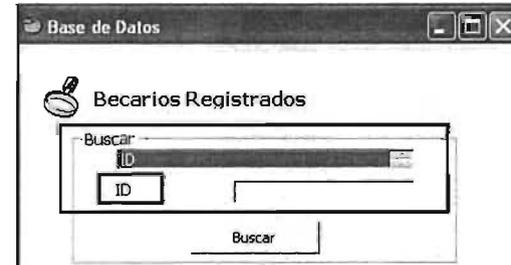
### Base de Datos

En este modulo Podremos consultar los datos del Becario y Realizar búsquedas por criterio para su localización dentro del sistema.



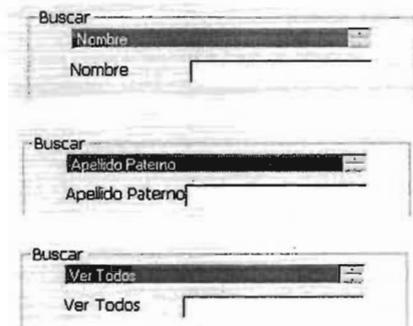
Veamos Primero el modulo de Becarios Registrados.

En primera instancia tenemos el tipo de búsqueda que deseamos realizar por el criterio, esto es por **ID**, **Nombre**, **Apellido Paterno**, y Ver Todos. Como se ilustra en la siguiente figura.



Nótese que al seleccionar de la lista el parámetro "ID" y sombreadarlo en color azul también cambia la etiqueta de abajo para así poder ingresar algún ID y que la búsqueda se realice lo mismo pasa con todos los siguientes.

Ejemplo:

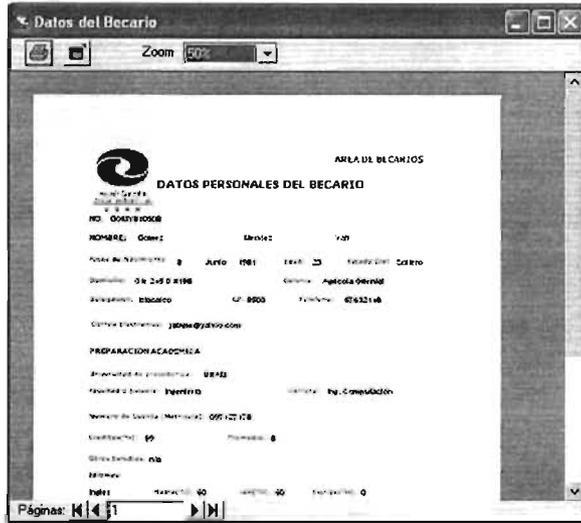


Y con solo dar clic en el botón **Buscar** tendremos los datos:

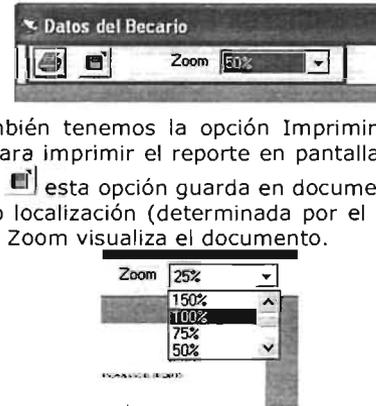


Con los botones  , podremos visualizar y tener acceso a todos los datos de los Becarios y ver su fotografía en pantalla.

Para visualizar los datos completos del Becario solo damos clic en el botón Datos completos  ubicado en el extremo inferior de derecho de la pantalla anterior. La siguiente pantalla que se visualiza será: el reporte Datos Personales del Becario.

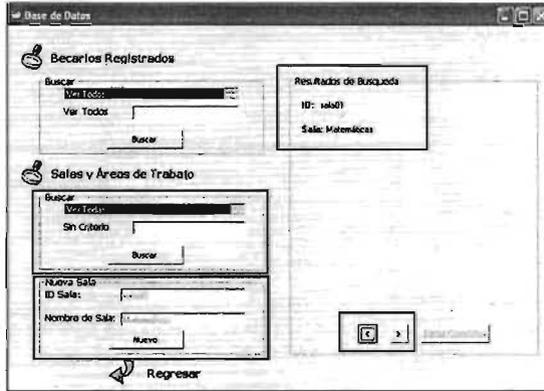


En esta pantalla tendremos acceso a todos los datos del Becario.  
 Nota: no olvide que para regresar a la pantalla Base de datos solo tenemos que cerrar la pantalla datos del Becario.



Aquí también tenemos la opción Imprimir , que servirá para imprimir el reporte en pantalla; la opción importar  esta opción guarda en documento en otro archivo o localización (determinada por el usuario); y la opción Zoom visualiza el documento.

Al igual que para realizar las búsquedas del Becario se cuenta con la Herramienta Salas y Áreas de Trabajo. Como se muestra a continuación en donde también se tiene que determinar el criterio de la búsqueda ya sea por **ID**, **Nombre de Sala** y **Ver todas**. Esto se realiza en el recuadro Buscar. Al seleccionar un Criterio en nuestra pantalla se ha seleccionado **Ver todas** y del lado derecho tenemos los resultados de la búsqueda. Las zonas en rojo son las que hemos operado.



En la parte inferior derecha podremos ver todas las salas que tenemos en ese momento en el sistema. En el recuadro azul tenemos la opción de Crear una **Nueva Sala**, dando el **ID Sala** y el **Nombre de Sala**, al dar clic al botón Nuevo se crea automáticamente la sala que podremos visualizar en el recuadro derecho seleccionado en rojo con el nombre de **Resultados de búsqueda**.

Para regresar a la Pantalla anterior solo damos clic en Regresar.

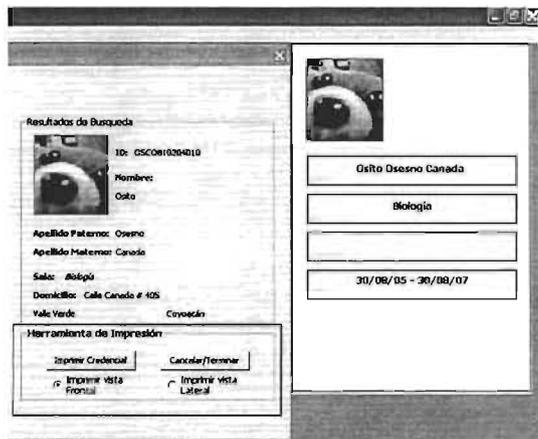
Otra de las herramientas importantes dentro de **Base de Datos** es la **Herramienta de impresión**.

Si queremos imprimir la credencial o el Reporte de algún Becario solo tenemos que llegar al menú **Herramientas** dentro del modulo **Base de Datos**, y seleccionar cualquiera de las opciones siguientes: **Imprimir Credencial**, **Imprimir Forma de Registro**.

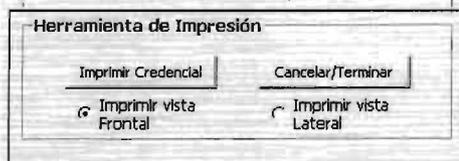
Como se muestra a continuación:



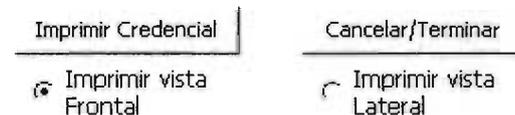
Supongamos que deseamos imprimir la Credencial, por tanto la siguiente pantalla a visualizar será:



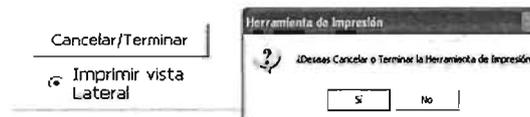
Aquí tenemos la **Herramienta de impresión** la cual en la figura se encuentra dentro de un recuadro rojo, también podremos visualizar el formato de la Credencial.



Aquí podremos **Imprimir la Credencial** con dar clic en el botón con el mismo nombre, podremos seleccionar entre **Vista Frontal** que es donde se encuentran la foto del becario y sus datos, y la **Vista lateral** que es donde se imprimirá el Código de Barras.

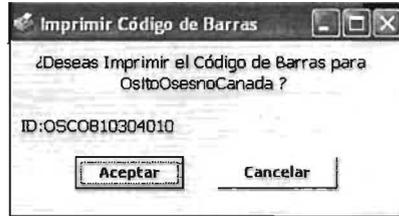


Si deseamos imprimir la **Vista Lateral** solo seleccionamos la casilla con el mismo nombre como se muestra:

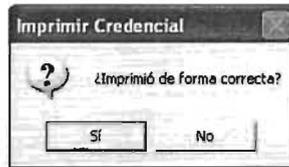


Para cancelar la herramienta de Impresión solo damos clic en el botón **Cancelar/Terminar**.

Se mostrara la siguiente pantalla de mensaje de Confirmación:



Aquí se imprimirá el Código de Barras único para el usuario seleccionado. Con solo dar clic en el botón Aceptar tendremos la impresión. Al mandar la impresión Se mostrara el siguiente mensaje:



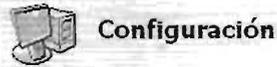
Si se imprimió de forma correcta damos clic al botón **Si** de lo contrario, daremos clic en el botón **No** y la impresión se pospondrá.



## Nota importante:

Posponer una impresión es una gran ventaja en este Sistema ya que se guarda la información para que este disponible en otro momento esto resulta muy útil cuando tenemos problemas con la impresora.

## ★ 4. Configuración



Al dar clic a configuración se muestra la siguiente pantalla.



En este modulo del sistema Tenemos a los Usuarios registrados con acceso al sistema. Aquí se pueden Agregar, Modificar y Eliminar a los usuarios.

## a) ¿Cómo agrego un usuario al sistema?

Damos clic al botón Nuevo  y realizamos la captura del nuevo usuario del sistema como se muestra:

Configuración

Usuarios Registrados con Acceso:

Nombre:

Password:

Confirmar Password:

Al finalizar la captura damos clic en el botón guardar.

Con esto ya tenemos un nuevo usuario en el sistema. Después aparecerá la pantalla usuarios registrados con acceso y visualizaremos a nuestro nuevo usuario.

Usuarios Registrados con Acceso:

Nombre:

## b) ¿Cómo modifico los datos de un usuario?

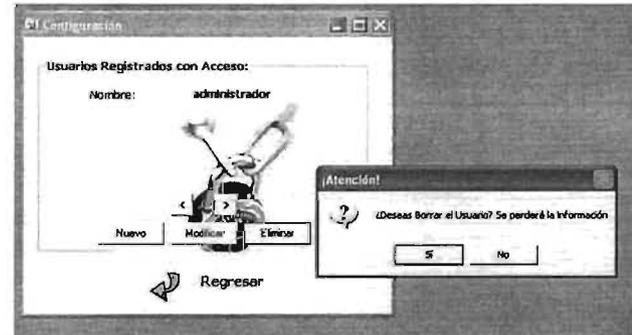
Supongamos que queremos cambiar la contraseña del usuario Administrador; entonces primero lo tenemos que tener en el recuadro **Usuarios registrados con Acceso** el nombre que vamos a modificar para poder realizar el cambio, después damos clic en **Modificar**  y se muestra la siguiente pantalla.



Aquí realizamos los cambios convenientes y damos clic en el botón Guardar.

### c) ¿Cómo elimino a un usuario registrado en el sistema?

Para eliminar a un usuario dentro del sistema primero tenemos que visualizarlo y después damos clic al botón Eliminar  y aparece un mensaje como se muestra.



Para borrarlo solo damos clic en el botón **Si**  del mensaje. Y se borrará al usuario.

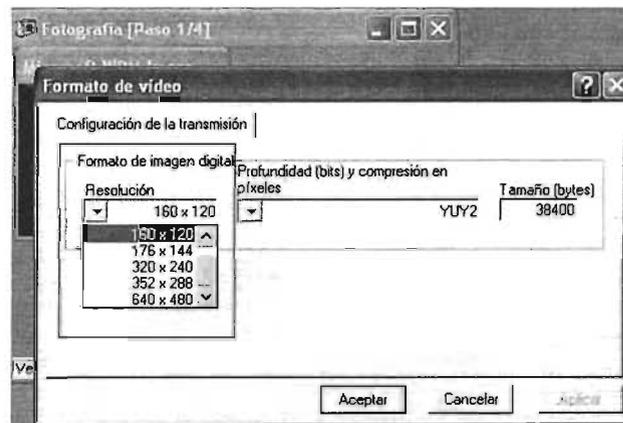
## 5. Herramientas y Controles Especiales

Uno de los controles especiales es la toma de la fotografía:



En la barra de menú tenemos la opción **Control**, aquí se encuentran las opciones siguientes: **Formato, Fuente de Captura, Compresión, Vista Previa, siempre Visible.**

Comencemos por ver la opción **Formato**:



Se abre la ventana **Formato de Video**. En esta opción tenemos la posibilidad de cambiar la **Resolución** de la imagen digital desde 128 x 96 hasta

640 x 480, Esto define el tamaño de la foto. La opción que se encuentra por omisión es la de 160 x 120 esta medida es la que se recomienda.

En la opción **Profundidad (bits) y compresión en píxeles**, se tienen varias configuraciones como se muestra a continuación esto cambia el tamaño en cantidad de bits de la imagen que se visualiza. La que se utiliza por omisión es el formato YUY2 es el que se recomienda ya que tiene una buena resolución y no ocupa demasiado espacio para el archivo de la fotografía.



En los recuadros rojos se muestra el formato de píxeles y el tamaño de bytes

En la opción **Fuente de Captura** tenemos la ventana **Origen de Video** aquí podemos seleccionar el **Origen de la captura**, La **Configuración de dispositivo**, **Controles de cámara**.



### Origen de Video

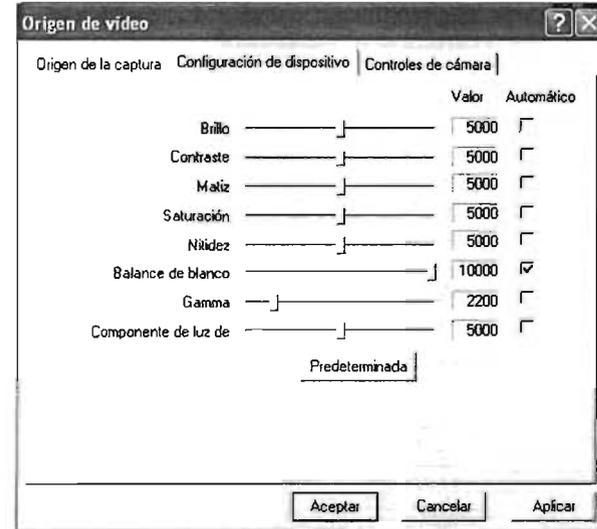
Aquí encontramos los dispositivos que tenemos instalados en la Computadora en nuestro caso es una cámara Web de marca Logitech como se muestra a continuación.



Donde podemos seleccionar un dispositivo de Video. En dado caso de que se tuviera más de un dispositivo se seleccionaría el dispositivo adecuado.

En la opción **Configuración de dispositivo** tenemos las opciones para modificar los colores del video de la cámara, esto es, modificar el brillo, el contraste, el matiz, la nitidez, etc. Como se muestra a continuación, el botón

predeterminada Predeterminada reestablece las opciones por omisión.





## Nota importante:

Para que cualquier cambio en la configuración tenga efecto solo hay que dar clic al botón

**Aplicar**  y después al botón **Aceptar** .



# **Apéndice B**

## **Manual de Usuario Ms Access a Mysql**

# MANUAL DE USUARIO DE ACCESS A MYSQL



***Como migrar la base de datos del sistema C2WCAM a la Base de Datos del Sistema de Control del Becarios a través del programa MS Access to MySQL.***

## Índice:

## Página

- |  |    |
|--|----|
|  1. Guía para el usuario                                | 03 |
|  2. ¿Cómo transferir la información de una base a otra? | 05 |



## 1. Guía para el usuario:

### Introducción

MS Access to Mysql es un pequeño programa que puede convertir la Base de datos de Microsoft Access a Mysql.

- Contiene una interfaz grafica
- Transferir datos directamente de una base de datos a otra
- Seleccionar las tablas a transferir
- Transferencia opcional de índices.
- Transferencia opcional de expedientes.
- Transferencia opcional de los valores predefinidos de los campos.
- Identifica y transfiere tipos de campo.
- Interfaz de línea de comando.
- Fácil de usar

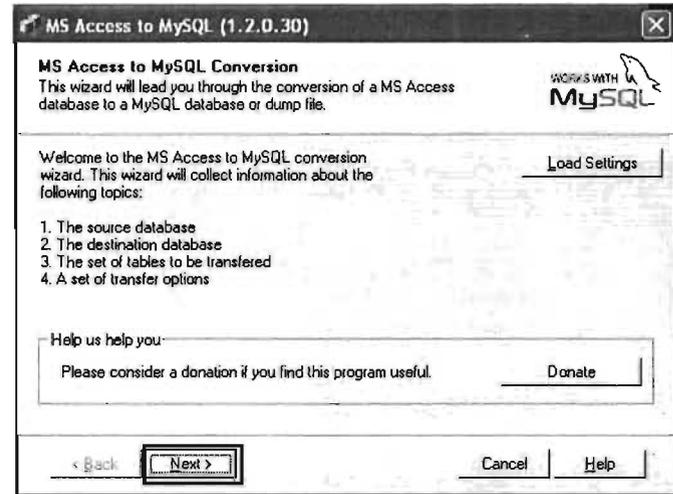
## 2. ¿Cómo transferir la información de una base a otra?

Primero en nuestro Escritorio tendremos el acceso directo **MS Access to MySQL** el icono tiene la apariencia siguiente:



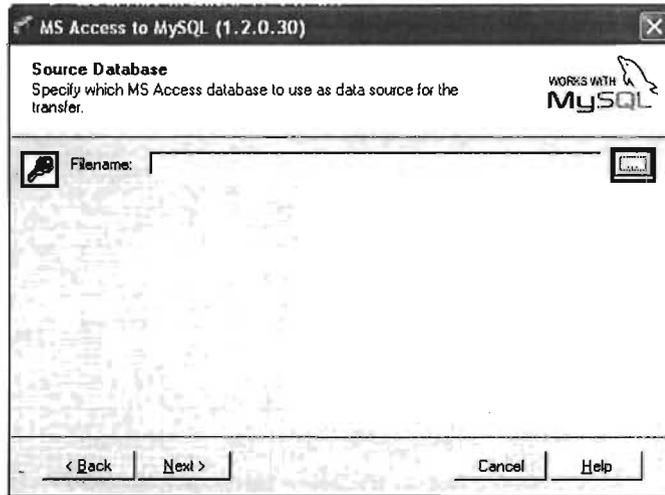
MS Access to  
MySQL

A este icono le daremos doble clic y se mostrara la siguiente pantalla.

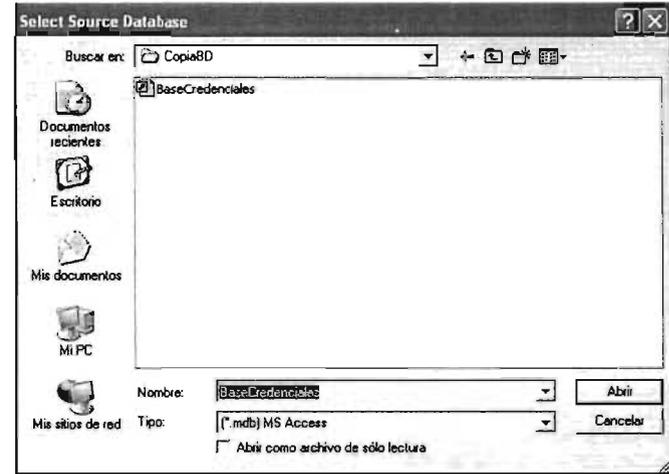


Aquí se nos da la bienvenida al software daremos clic en el botón **Next>**  o siguiente para comenzar el asistente este botón se ilustra encerrado en un recuadro rojo.

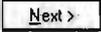
Al dar clic en el boton **Next >** Next > inmediatamente se muestra la siguiente forma:

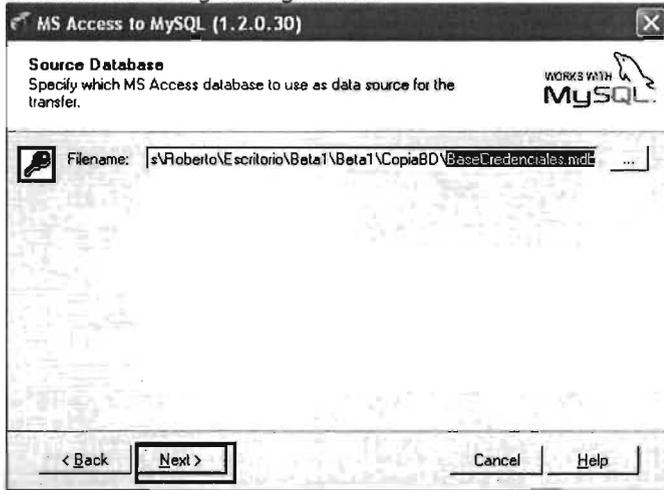


Donde se nos dice que especifiquemos que base de datos de Access es la que se va a transferir a Mysql Para buscarla daremos clic en el botón con los tres puntitos que en esta pantalla se encuentra en un recuadro rojo, al hacer esto se mostrará el siguiente cuadro **Select Source Database** (Seleccione el origen de la base de datos):

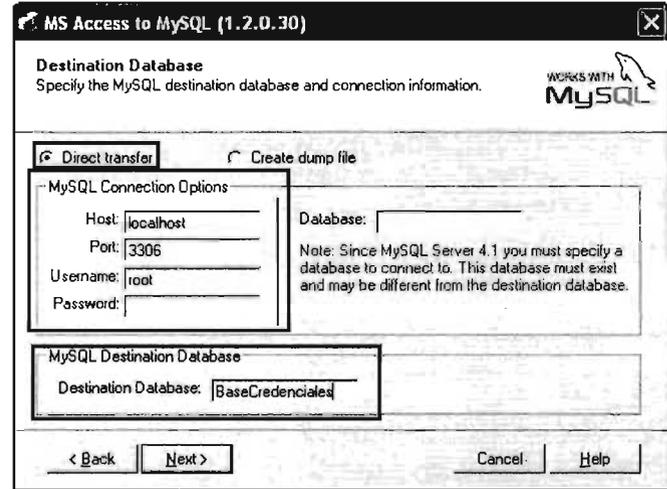


Seleccionamos la base de datos; en nuestro caso será **BaseCredenciales.mdb** damos clic en **Abrir.** Abrir

Entonces se muestra toda la ruta en la caja de texto y damos clic en el botón **Next>**  tal como se muestra en la figura siguiente:



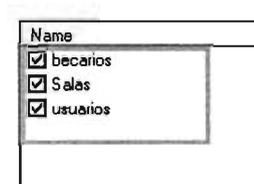
El siguiente cuadro es donde especificamos el destino de nuestra Base de datos así como la información del servidor al cual se le entregará dicha base de datos:

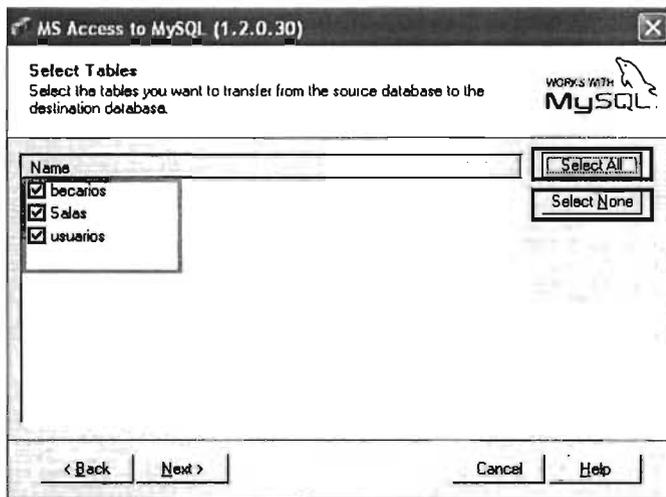


Se selecciona la opción **Direct Transfer** (Transferencia Directa) la cual se encuentra en la imagen encerrada en un recuadro azul. Después en el siguiente recuadro rojo Tenemos el frame MySQL Connection Options (Opciones de Conexión de Mysql) en la caja de Texto tenemos que poner la IP del servidor Ej. 132.248.66.225 en el cuadro de Texto **Host**; **Port**: o puerto es el mismo 3306 el **Username**: y **Password**: es el asignado por el Administrador de Servidores.

El siguiente frame también encerrado en un recuadro rojo **MySQL Destination Database** pondremos el nombre de la Base de datos de destino, en nuestro caso la llamaremos igual **BaseCredenciales** como se mostró en la figura anterior; damos clic en **Next>**  para continuar con el procedimiento.

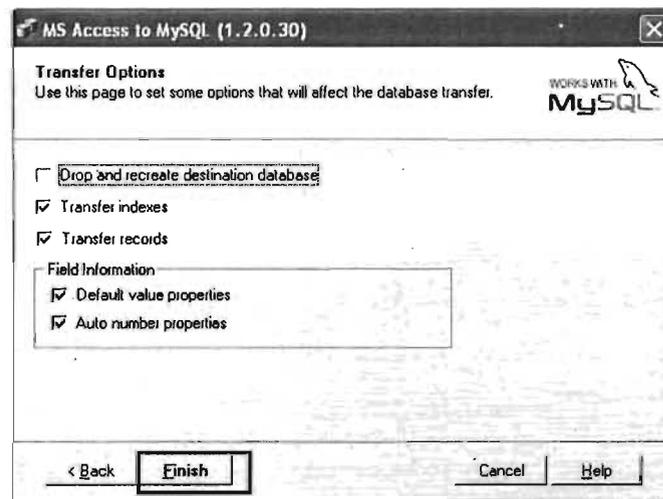
La siguiente pantalla muestra la selección de las tablas que vamos a elegir y que va a contener nuestra base de datos, tenemos la opción de seleccionar todas las tablas con el botón **Select All**  encerrado en un recuadro azul, de no seleccionar ninguna con **Select None**  encerrada en un recuadro rojo o manualmente las tablas que se necesiten mediante las casillas que se encuentran palomeadas encerradas en un recuadro verde en donde tenemos el nombre de las tablas **Name**.





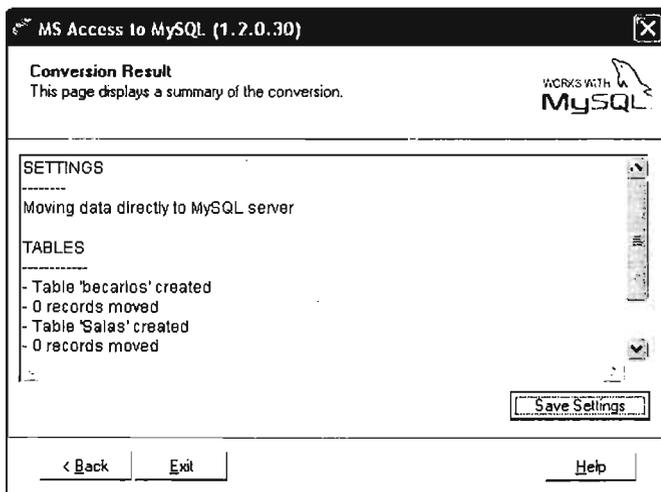
Se recomienda tener seleccionada la primera opción **Select All** una vez realizado esto damos clic en el botón **Next >**

La siguiente pantalla a visualizar será **Transfer Options** u Opciones de Transferencia. Aquí pondremos seleccionaremos todas excepto la primera **Drop and recreate destination database** ya que si lo hacemos borraremos los datos de la base de datos tal como se muestra en la figura debemos seleccionar nuestras opciones.



Finalmente damos clic en el botón **Finish**

La pantalla final **Conversión Result** nos da un Reporte de los resultados obtenidos donde nos dice que se han movido las tablas al servidor de Mysql y las tablas que se crearon en el servidor como muestra la figura siguiente:



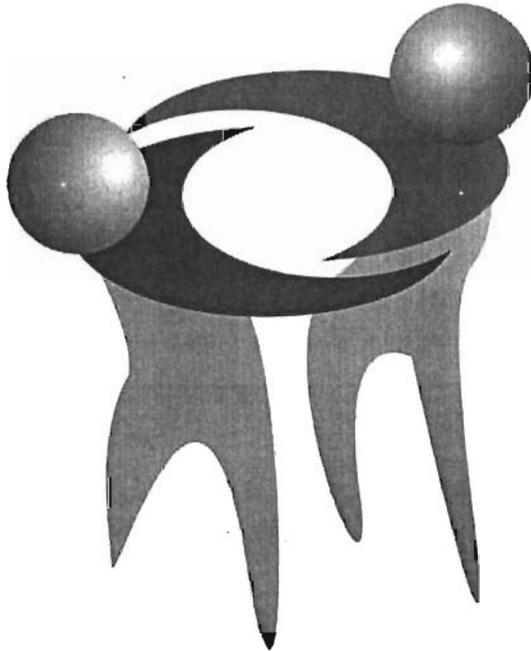
Para salir del asistente damos clic en **Exit**

**Y el proceso de migrar la base de datos se ha realizado.**

## **Apéndice C**

# **Manual Sistema de Control de Becarios Red**

# Manual de Usuario



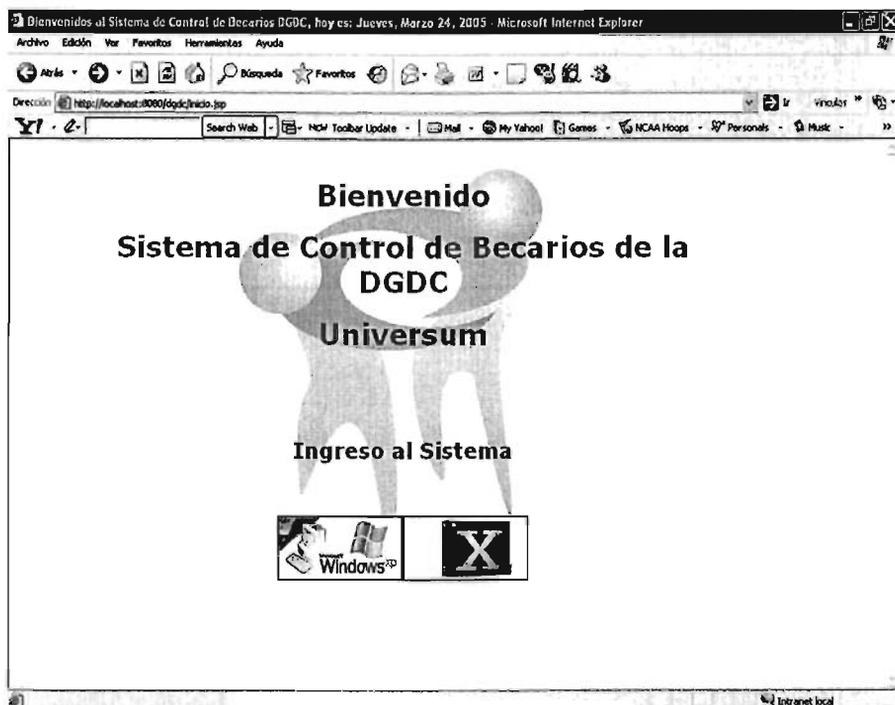
## Sistema de Control de Becarios Para la DGDC

Realizado por:  
Yair Gómez Méndez  
J. Roberto Hernández Molina

Índice	Página
I. Introducción Sistema de Control de Becarios	
Versión PC	03
II. Opciones del Menú del Sistema PC	
II. a) Información de Becarios	07
II. b) Consulta Horario	11
II. c) Modificar Datos de un Becario	15
II. d) Información de Salas	20
II. e) Alta de una Sala	22
II. f) Modificar Datos de una Sala	24
II. g) Agrega Horarios	26
II. h) Información Horarios	29
II. i) Manuales	30
II. j) Listas de Becarios	33
II. k) Generar Sabana por Sala	37
II. l) Generar Sábana por Día	41
II. m) Usuarios del Sistema	46
II. n) Salir	51
Versión Mac	53
III. Opciones del Menú del Sistema Mac	
III. a) Información de Becarios	57
III. b) Consulta Horario	61
III. c) Modificar Datos de un Becario	64
III. d) Información de Salas	70
III. e) Alta de una Sala	72
III. f) Modificar Datos de una Sala	75
III. g) Agrega Horarios	78
III. h) Información Horarios	81
III. i) Manuales	83
III. j) Listas de Becarios	87
III. k) Generar Sabana por Sala	90
III. l) Generar Sábana por Día	94
III. m) Usuarios del Sistema	98
III. n) Salir	103
IV. Modulo Para Becarios.	105
IV. a) Consulta de Asistencias	108
IV b) Datos Personales	109
IV. c) Consulta tu Horario	110

## I. Introducción Sistema de Control de Becarios (Internet).

El sistema nos da la bienvenida al poner la dirección en el explorador <http://servidor/dgdc/inicio.jsp>



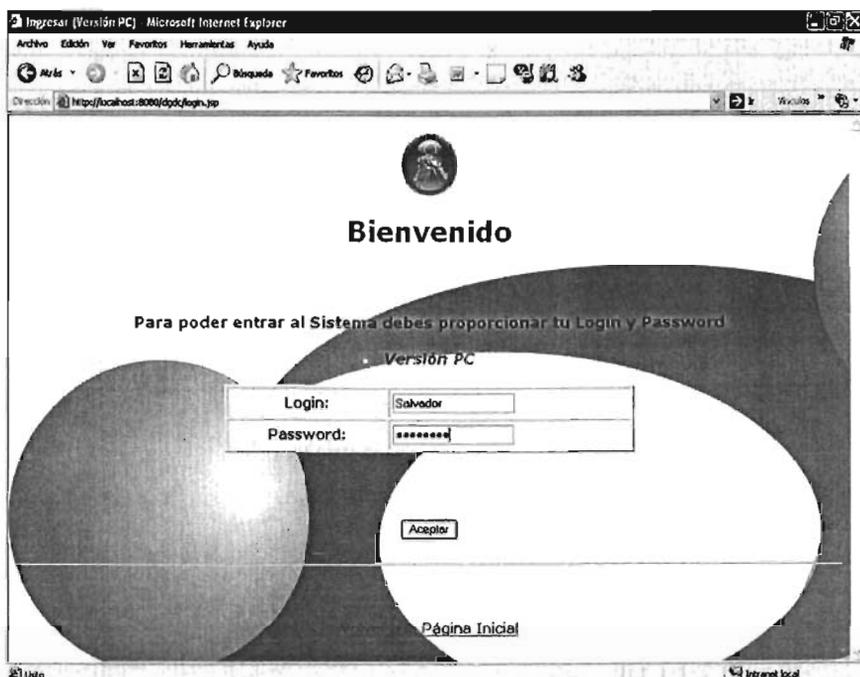
En esta pantalla tendremos dos opciones para acceder si nos encontramos en una PC que será nuestro primer caso daremos clic en

el icono **Versión PC**  el siguiente navegador mostrara lo siguiente:

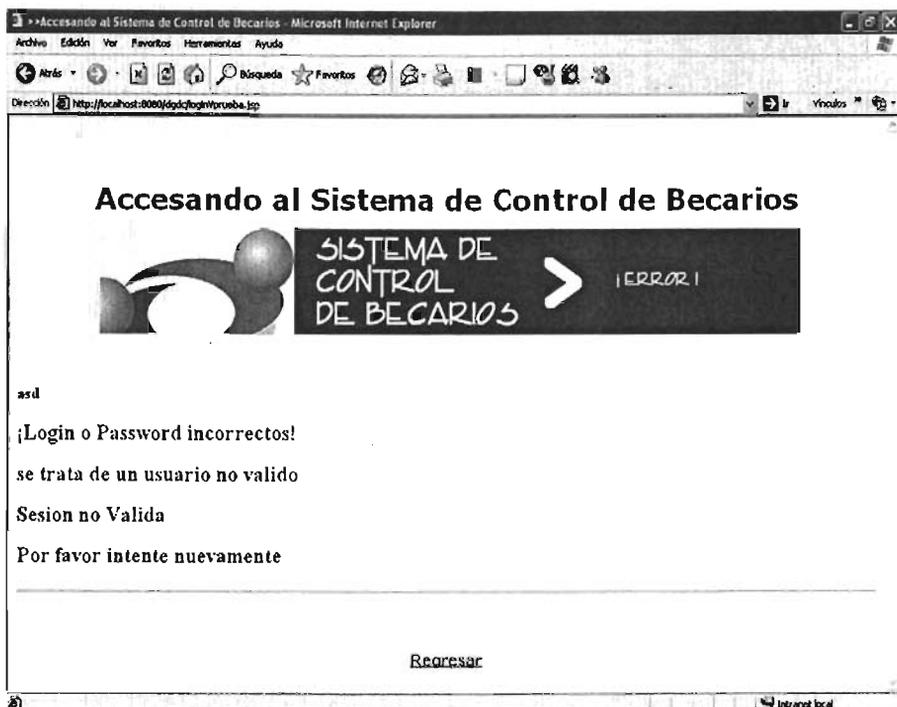
La pantalla de ingreso al sistema en donde tendremos que proporcionar nuestro Login o nombre de usuario y nuestra password o

contraseña para ser autorizados y dar **Enter** o un clic en el botón **Aceptar**

Login:	Salvador
Password:	*****



Si escribimos mal nuestro Login o Password o no estamos autorizados por el sistema al dar **Enter** o **Aceptar** nos mostrara el siguiente mensaje como se ilustra a continuación.

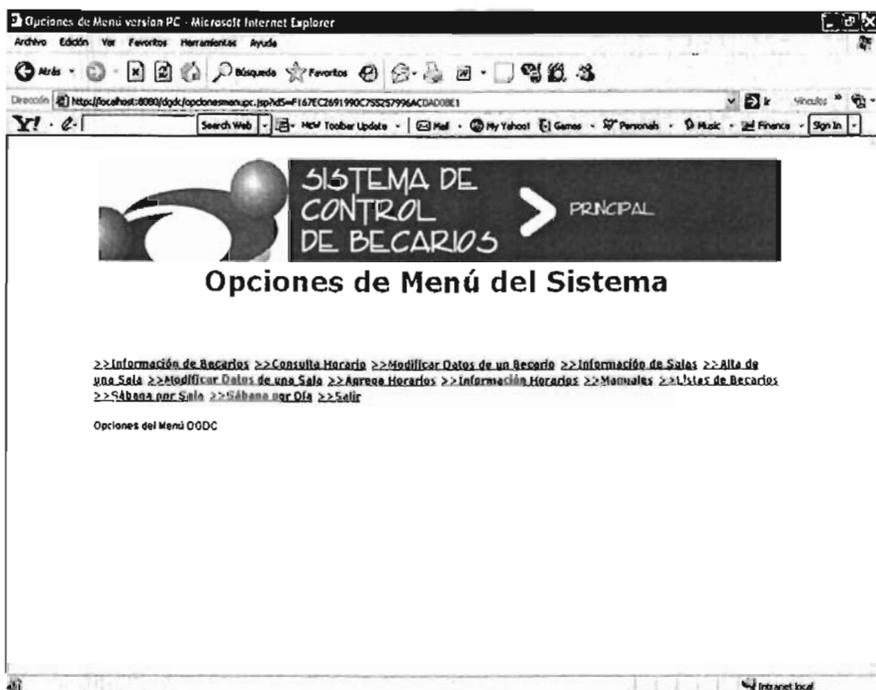


Al aparecer este mensaje solo tenemos que dar clic en la parte inferior en el centro donde esta la referencia de **Regresar** Regresar

Ahora volvemos a la página de acceso y escribimos correctamente el Login y el Password si lo hacemos bien y estamos autorizados entraremos con éxito al sistema:

## II. Opciones del Menú del Sistema (Versión PC)

Ya autorizados por el sistema tenemos todas las opciones siguientes:



- a) Información de Becarios
- b) Consulta Horario
- c) Modificar Datos de un Becario
- d) Información de Salas
- e) Alta de una Sala
- f) Modificar Datos de una Sala
- g) Agrega Horarios
- h) Información Horarios
- i) Manuales
- j) Listas de Becarios
- k) Generar Sabana por Sala
- l) Generar Sábana por Día
- m) Salir

Ahora veamos una por una estas opciones:

## II. a) Información de Becarios

Al hacer clic sobre Información de Becarios >> **Información de Becarios** (Ver la siguiente figura) encerrada se encuentra esta opción en un recuadro rojo y en el segundo recuadro tenemos una breve descripción para ayuda al usuario de lo que hace esta referencia.



Muestra a todos los Becarios contenidos en el Sistema

Opciones de Menú version PC - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Atrás Búsqueda Favoritos

Dirección http://localhost:8080/dgdz/opcionesmenu.pc.jsp?idS=2C3A0B8A1B6F69937782F46D7DA6880E

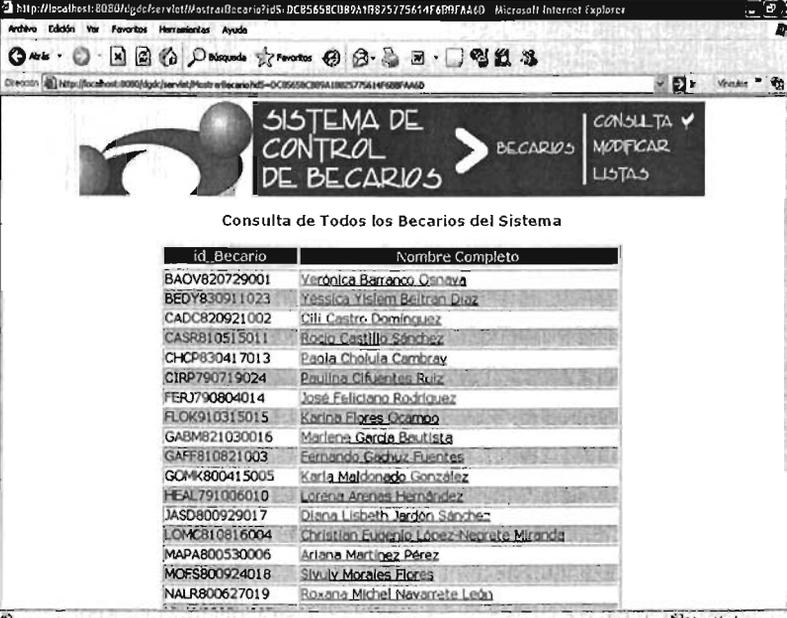
**SISTEMA DE CONTROL DE BECARIOS** > PRINCIPAL

**Opciones de Menú del Sistema**

>> **Información de Becarios** >> Consulta Horario >> Alta de un Becario >> Modificar Datos de un Becario >> Información de Salas >> Alta de una Sala >> Modificar Datos de una Sala >> Anular Horarios >> Información Horarios >> Manuales >> Listas de Becarios >> Salir

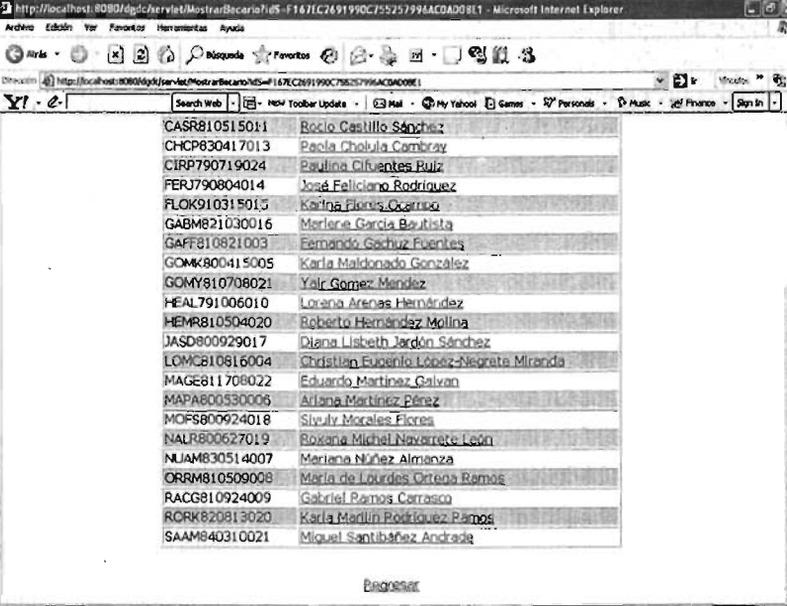
Muestra a todos los Becarios contenidos en el Sistema

Al seleccionar esta opción se desplegará una pantalla en donde nosotros podremos ver a todos los Becarios que se encuentran en nuestra Base de Datos de forma dinámica se mostrará su Id de Becario así como su nombre completo.



Consulta de Todos los Becarios del Sistema

id_Becario	Nombre Completo
BACV820729001	Verónica Barranco Canava
BEDY830911023	Yessica Yislem Beltrán Díaz
CADC820921002	Cili Castro Domínguez
CASR810515011	Rocio Castillo Sánchez
CHCP830417013	Paola Cholula Cambay
CIRP790719024	Paulina Cifuentes Ruiz
FERJ790804014	José Feliciano Rodríguez
FLOK910315015	Karina Flores Ocambo
GABM821030016	Merlene García Bautista
GAFF810821003	Fernando Gachuz Fuentes
GOMK800415005	Karla Maldonado González
HEAL791006010	Lorena Arenas Hernández
JASD800929017	Diana Lisbeth Jardón Sánchez
LWCB810816004	Christian Eusebio López-Negrata Miranda
MAPA800530006	Ariana Martínez Pérez
MOFS800924018	Sivily Morales Flores
NALR800627019	Roxana Michel Navarrete León



Mostrar Becario

CASR810515011	Rocio Castillo Sánchez
CHCP830417013	Paola Cholula Cambay
CIRP790719024	Paulina Cifuentes Ruiz
FERJ790804014	José Feliciano Rodríguez
FLOK910315015	Karina Flores Ocambo
GABM821030016	Merlene García Bautista
GAFF810821003	Fernando Gachuz Fuentes
GOMK800415005	Karla Maldonado González
GOMY810708021	Yair Gómez Méndez
HEAL791006010	Lorena Arenas Hernández
HEMR810504020	Roberto Hernández Molina
JASD800929017	Diana Lisbeth Jardón Sánchez
LWCB810816004	Christian Eusebio López-Negrata Miranda
MAGE811708022	Eduardo Martínez Galvan
MAPA800530006	Ariana Martínez Pérez
MOFS800924018	Sivily Morales Flores
NALR800627019	Roxana Michel Navarrete León
NIJAM830514007	Merlene Núñez Almanza
ORRM810509008	María de Lourdes Ortao Ramos
RACG810924009	Gabriel Ramos Carrasco
RORR820813020	Karla Madin Rodríguez Pamos
SAAM840310021	Miguel Santibáñez Andrade

[Regresar](#)

Si deseamos regresar al menú de opciones solo tenemos que dar clic en la referencia [Regresar](#).

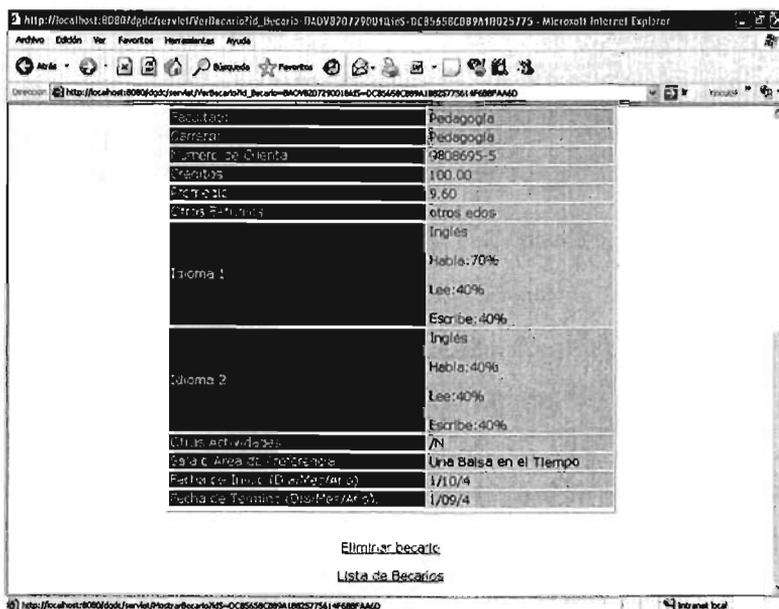
Si nosotros queremos saber la información de los Becarios solo basta dar un clic sobre su Nombre Completo (que se encuentra subrayado y de color azul), para que nos muestre sus datos personales.

id_Becario	Nombre Completo
BAOV820729001	<u>Verónica Barranco Osnaya</u>

Esto se ilustra a continuación:

The screenshot shows a web browser window with the URL [http://localhost:8080/egdc/ServletVerBecario?id\\_Becario=BAOV820729001&id5-DC6565BC089A18925775](http://localhost:8080/egdc/ServletVerBecario?id_Becario=BAOV820729001&id5-DC6565BC089A18925775). The page title is "SISTEMA DE CONTROL DE BECARIOS". The navigation menu includes "CONSULTA" (checked), "MODIFICAR", and "LISTAS". The main content area is titled "Becario Seleccionado" and displays the following personal data:

Id de Becario:	BAOV820729001
Nombre:	Verónica
Apellido Paterno:	Barranco
Apellido Materno:	Osnaya
Fecha de Nacimiento (Día/Mes/Año):	29/7/82
Ejército:	22
Ejército Civil:	soltero
Dirección:	Veracruzanos 26
Colonia:	Bonanza
Delegación:	Álvaro Obregón
Código Postal:	1220
Teléfono:	52-73-72-82
Correo Electrónico:	verooosnaya@hotmail.com
Universidad:	UNAM
Facultad:	Pedagogía
Carrera:	Pedagogía
Número de Cuenta:	9808695-5
Creditos:	100.00
Promedio:	9.60



Si deseamos regresar a la página anterior para ver los datos de otro Becario solo damos clic en [Lista de Becarios](#), nos vamos a la parte inferior de esta pagina donde tenemos las opciones Eliminar becario y Lista de Becarios como se muestra en la siguiente pantalla dentro de un recuadro rojo y uno negro respectivamente: Opciones de esta página:



## Nota importante

La opción **Eliminar Becario** Eliminar becario no es de uso común, es decir, no la utilices a menos que se este seguro de que ese becario se quiere eliminar. Si queremos Borrar un Becario definitivamente del sistema entonces podemos utilizar esta opción.

## II. b) Consulta Horario

Regresamos a las Opciones del Menú PC

En esta parte del menú de opciones tenemos Consulta Horario

>>**Consulta Horario** el cual nos da una breve información de lo que contiene.



Consulta Horarios de los Becarios

Opciones de Menú version PC - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Inicio Alrés Búsqueda Favoritos

Generado en http://localhost:8080/dgdc/opcionemenuopc.jsp?MS=F167EC2691990C795257996AC0AD08E1

Search Web New Toolbar Update Mail My Yahoo! Games Personal Music Finance Sign In

**SISTEMA DE CONTROL DE BECARIOS** > PRINCIPAL

**Opciones de Menú del Sistema**

>> Información de Becarios >> Consulta Horario >> Modificar Datos de un Becario >> Información de Salas >> Alta de una Sala >> Modificar Datos de una Sala >> Agrega Horarios >> Información Horarios >> Manuales >> Listas de Becarios >> Sábana por Sala >> Sábana por Día >> Salir

Consulta Horarios de los Becarios

http://localhost:8080/dgdc/opcionemenuopc.jsp?MS=F167EC2691990C795257996AC0AD08E1 Internet local

Aquí podemos consultar el horario para cada Becario que se encuentra en el Sistema vamos a ver un ejemplo de cómo se puede ver un horario. Lo que tenemos que hacer para Mostrar el Horario del Becario es poner en la caja de Texto su Id de Becario como ejemplo tenemos lo siguiente:

Supongamos que deseamos ver el Horario del Becario que tiene el siguiente Id: BAOV820729001 este id lo escribiremos en la caja de texto como se ilustra a continuación.

Consultar Horario de Becarlo

Para realizar la consulta del horario solo tenemos que dar clic en el botón

Por otra parte si lo que queremos es regresar de nuevo al menú de opciones solo tenemos que dar clic en la referencia [Regresar al menú](#)



Al dar clic en el botón  tendremos el siguiente resultado:

SISTEMA DE CONTROL DE BECARIOS

CONSULTA  
MODIFICAR  
LISTAS

### Horario Completo Becario: Verónica

Id\_Becario:BAOV820729001    Nombre:Verónica

Hora	Lunes	Martes	Miercoles	Jueves	Viernes	Sabado	Domingo
9 a 10	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>				
10 a 11	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
11 a 12	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
12 a 13	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
13 a 14							
14 a 15					<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

12 a 13	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
13 a 14							
14 a 15					<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
15 a 16					<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
16 a 17					<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
17 a 18					<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Imprime el Horario pulsando en la imagen

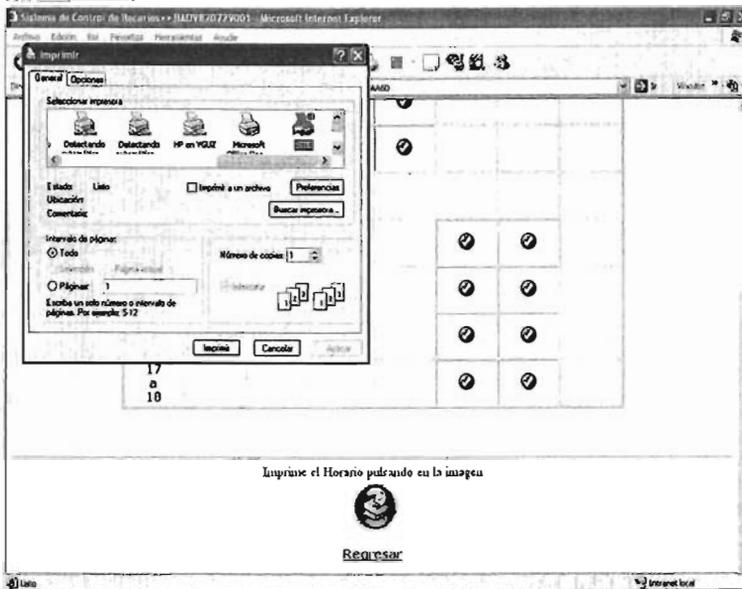
[Regresar](#)

De forma dinámica se genera un Horario del Becario seleccionado donde se muestra su Nombre, el Id de Becario, los días y horario que cubre dicho Becario. Si se requiere imprimir esta información solo damos clic en la figura de la impresora.

Imprime el Horario pulsando en la imagen



Al hacerlo se abrirá el siguiente mensaje de opciones de impresión seleccione la que mas se ajuste a sus preferencias y de clic en el botón imprimir [Imprimir](#)



Para regresar a buscar el horario de otro Becario solo basta con dar clic en la referencia regresar que se encuentra en la parte inferior de esta página. [Regresar](#)

## II. c) Modificar Datos de un Becario

Al seleccionar del menú de opciones [>>Modificar Datos de un Becario](#) lo que podremos hacer es modificar toda la información personal del Becario tal como lo dice la pequeña ayuda que tenemos con el Menú



Modifica la información de un Becario

Opciones de Menú versión PC - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Dirección <http://localhost:8080/dgdc/opcionesmenu.pc.jsp?ds=F167EC2691990C755257996AC0A00BE1>

SISTEMA DE CONTROL DE BECARIOS. > PRINCIPAL

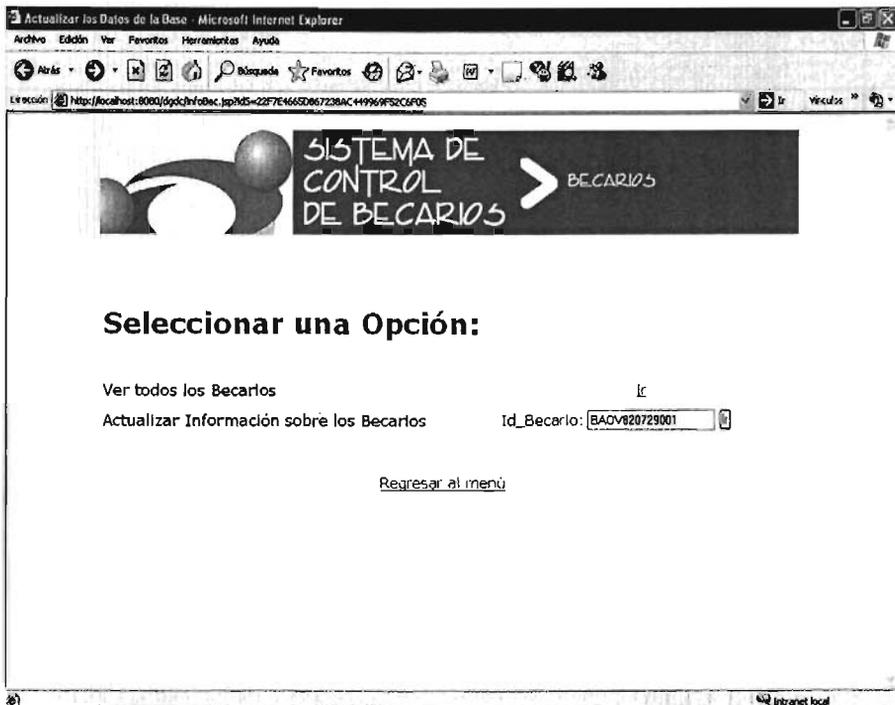
### Opciones de Menú del Sistema

>> Información de Becarios >> Consulta Horario >> Modificar Datos de un Becario >> Información de Salas >> Alta de una Sala >> Modificar Datos de una Sala >> Ausencia Horarios >> Información Horarios >> Manuales >> Listas de Becarios >> Sábana por Sala >> Sábana por Día >> Salir

Modifica la información de un Becario

<http://localhost:8080/dgdc/infoBec.jsp?ds=F167EC2691990C755257996AC0A00BE1>

Damos clic en la referencia [>>Modificar Datos de un Becario](#) y tenemos lo siguiente:



Podemos seleccionar una opción:

En la opción Ver todos los Becarios nos mostrara lo que se vio en el Inciso a), es decir, nos llevara a ver a todos los Becarios contenidos dentro del Sistema solo con dar clic en la referencia ir como se muestra

Ver todos los Becarios

Ir

En la opción actualizar Información sobre los Becarios pondremos en la caja de texto el id del Becario:

Supongamos que queremos modificar los datos personales del Becario que tiene el id siguiente BAOV820729001. Como se ilustra a continuación:

Actualizar Información sobre los Becarios

Id\_Becario:  Ir

Después de insertar el id de Becario damos clic en el botón  el cual nos mostrara lo siguiente:

Sistema de Control de Becarios -> Modificando la Información del Becario: BAOV820729001 Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

http://localhost:8080/jsp/frmModificarBecario.jsp?MS=22F7E1465D067238AC449969F52CAF05

**SISTEMA DE CONTROL DE BECARIOS** > BECARIOS | CONSULTA MODIFICAR LISTAS ✓

### Modificando la Información del Becario: BAOV820729001

Id_Becario:	BAOV820729001
Nombre:	Verónica
Apellido Paterno:	Barranco
Apellido Materno:	Osaya
Fecha de Nacimiento:	Día/Mes/Año: 29 / 7 / 82
Edad:	22
Edo. Civil:	soltero
Dirección:	Veracruzanos 25
Colonia:	Bonanza
Delegación:	Ávaro Obregón
C.P.:	1220
Teléfono:	52-73-72-82
Correo Electrónico:	verosnaya@hotmail.com

Sistema de Control de Becarios -> Modificando la Información del Becario: BAOV820729001 Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

http://localhost:8080/jsp/frmModificarBecario.jsp?MS=22F7E1465D067238AC449969F52CAF05

Teléfono:	52-73-72-82
Correo Electrónico:	verosnaya@hotmail.com
Universidad:	UNAM
Facultad:	Pedagogía
Carrera:	Pedagogía
Número de Cuentas:	900695-5
Creditos:	100.00 Promedio: 8.60
Otros Estudios:	otros estudios
Idioma 1	Inglés Habla: 70 (%) Lee: 40 (%) Escribe: 40 (%)
Idioma 2	Inglés Habla: 40 (%) Lee: 40 (%) Escribe: 40 (%)
Otras Actividades:	NI
Sala o Área de preferencia:	Una Balsa en el Tiempo
Fecha de Inicio (Día/Mes/Año):	1/10/04
Fecha de Termina (Día/Mes/Año):	1/09/04

Modificar la Información del Becario BAOV820729001

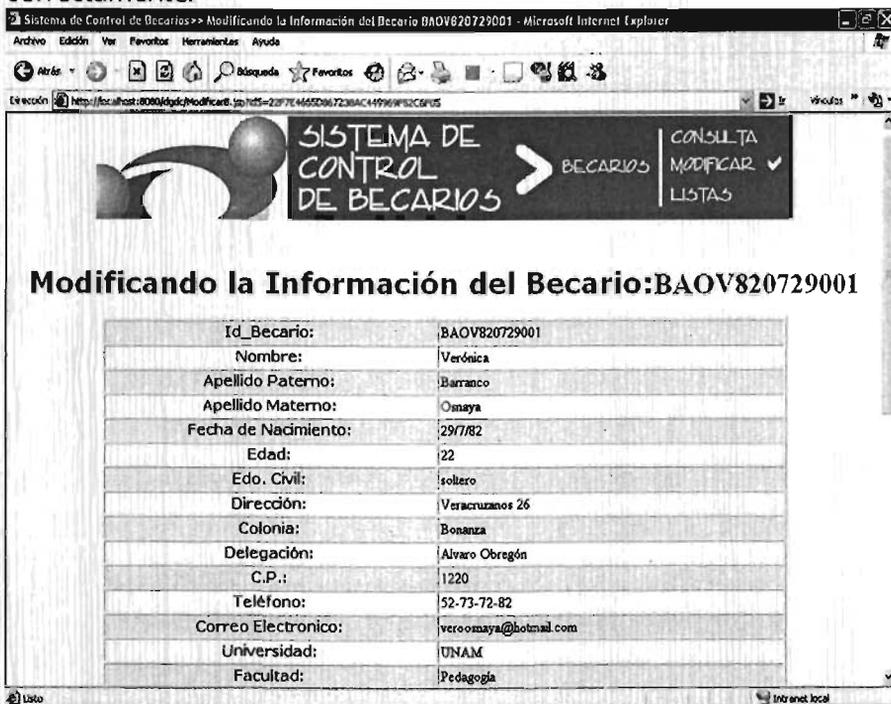
[Regresar](#)

Aquí podemos modificar cualquier campo excepto el ID de Becario, una vez terminando de modificar lo que deseamos solo tenemos que dar clic en el botón

Modificar la Información del Becario BAOV820729001

Si lo que deseamos es regresar a la pantalla anterior solo damos clic en Regresar

Una vez que hemos realizado nuestras modificaciones y dado clic en el botón . Aparecerá un formulario como el siguiente indicando si la información se actualizo correctamente:



**Modificando la Información del Becario:BAOV820729001**

Id_Becario:	BAOV820729001
Nombre:	Verónica
Apellido Paterno:	Barranco
Apellido Materno:	Omayá
Fecha de Nacimiento:	29/7/82
Edad:	22
Edo. Civil:	soltero
Dirección:	Veracruzanos 26
Colonia:	Bonanza
Delegación:	Alvaro Obregón
C.P.:	1220
Teléfono:	52-73-72-82
Correo Electronico:	veroomaya@hotmail.com
Universidad:	UNAM
Facultad:	Pedagogía

Sistema de Control de Becarios >> Modificando la Información del Becario BAOV820729001 - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Abre Busqueda Favoritos

Ubicación: http://localhost:8080/Modific...?id=2377E4665D6723BAC44996AF52CAF05

Universidad:	UNAM
Facultad:	Pedagogía
Carrera:	Pedagogía
Número de Cuenta:	9808695-5
Creditos:	100.00 Promedio: 9.60
Otros Estudios:	otros edos
Idioma1:	Inglés Habla:70(%) Lee: 40(%) Escribe: 40(%)
Idioma2:	Inglés Habla:40(%) Lee: 40(%) Escribe: 40(%)
Otras Actividades:	/N
Sala o Área de Preferencia:	Una Balsa en el Tiempo
Fecha de Inicio (Dia/Mes/Año):	1/10/04
Fecha de Termina (Dia/Mes/Año):	1/09/04



**SISTEMA DE CONTROL DE BECARIOS** > ¡ GUARDADO !

**Se ha modificado la Información del Becario: BAOV820729001**

[Volver a la Página Inicial](#)

Lista Intranet local

Lo que indica que se actualizo la información de forma correcta, de lo contrario mostrara un mensaje como este:



**SISTEMA DE CONTROL DE BECARIOS** > ¡ ERROR !

### No se pudo modificar la Información:

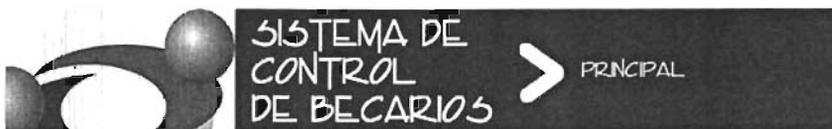
Este mensaje puede deberse a un error en la Base de Datos, es decir, que la base de datos no se encuentre disponible o este mal configurada, si este es su caso favor de ponerse en contacto con el administrador de servidores.

Para regresar a modificar la información de otro Becario damos clic en:

[Volver a la Página Inicial](#)

## II. d) Información de Salas

En esta opción **>>Información de Salas** del menú se muestran todas las salas contenidas dentro del sistema



### Opciones de Menú del Sistema

[>>Información de Becarios](#) [>>Consulta Horario](#) [>>Modificar Datos de un Becario](#) [>>Información de Salas](#) [>>Alta de una Sala](#) [>>Modificar Datos de una Sala](#) [>>Agregar Horarios](#) [>>Información Horarios](#) [>>Manuales](#) [>>Listas de Becarios](#) [>>Sábana por Sala](#) [>>Sábana por Día](#) [>>Salir](#)



Muestra todas las Salas contenidas en el Sistema

Damos clic en esta opción y se mostrara lo siguiente:

Lista de Salas

id_sala	Nombre de Sala
sala1	Matemáticas
sala10	Infraestructura de Nuestra Nación
sala11	Química
sala12	Tecnología Satelital
sala13	Una Balse en el Tiempo
sala2	Biodiversidad
sala3	Universo
sala4	Agricultura y Alimentación
sala5	Biología Humana y Salud
sala6	Conciencia de Nuestra Nación
sala7	Energía
sala8	Espacio Infantil
sala9	Estructura de la Materia

Agregar Sala  
Regresar

La opción Agregar Sala [Agregar Sala](#) se explicará mas adelante en el inciso f) si queremos regresar a al menú de opciones solo damos clic en Regresar [Regresar](#)

Podemos observar todas las salas del sistema; si queremos saber su contenido solo hacemos clic en el Nombre de la sala:

id_sala	Nombre de Sala
sal01	<u>Matemáticas</u>

Al hacer clic se muestra la siguiente pantalla:

Id de Sala: sal01

Nombre de Sala: Matemáticas

[Eliminar Sala](#)

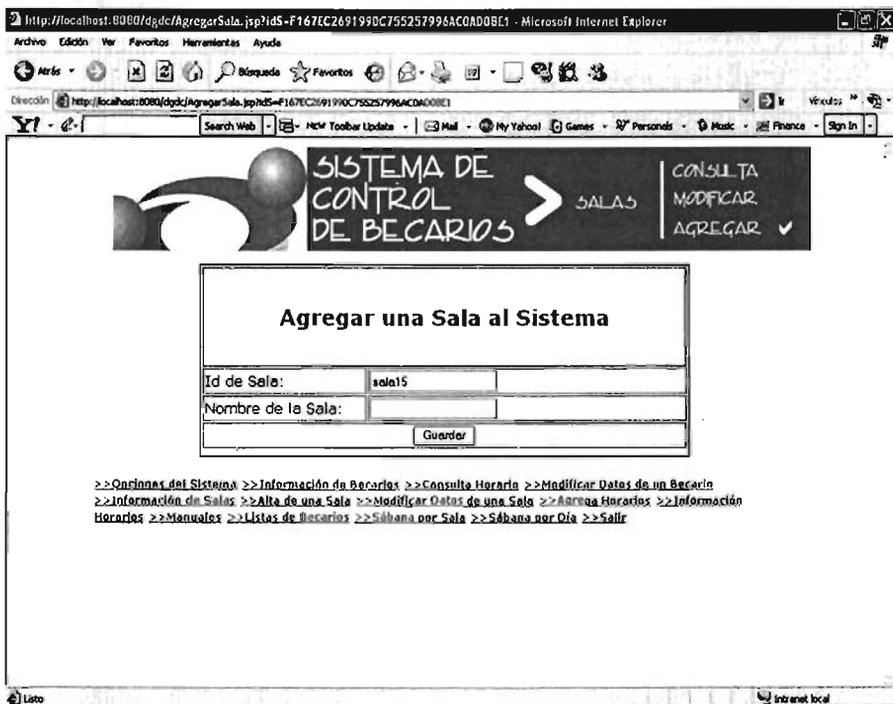
[Lista de Salas](#)

Si queremos eliminar la sala solo tenemos que dar clic en Eliminar Sala, esta opción es recomendable solo si se esta seguro de querer eliminar la sala definitivamente del sistema.

Para regresar a ver las Salas contenidas en el Sistema damos clic en Lista de Salas

## II. e) Alta de una Sala

En esta parte del sistema podemos dar de alta una sala si se requiere, cabe mencionar que el ID de Sala se genera automáticamente por lo que se necesita cambiar solo hay que agregar el Nombre de la Sala.



Supongamos que queremos agregar una Sala que llamaremos Sala X como ejemplo se ilustra lo siguiente:

<b>Agregar una Sala al Sistema</b>	
Id de Sala:	sala14
Nombre de la Sala:	Sala X
<input type="button" value="Guardar"/>	

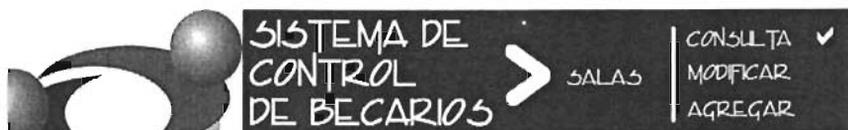
Damos clic en el botón guardar  y el registro de la sala será agregado al sistema. Si se realizo de forma correcta tendremos un mensaje como este:



El registro ha sido agregado

[Lista de Salas](#)

Damos clic en Lista de Salas para ver que se agrego efectivamente nuestra Sala X con el ID sala14

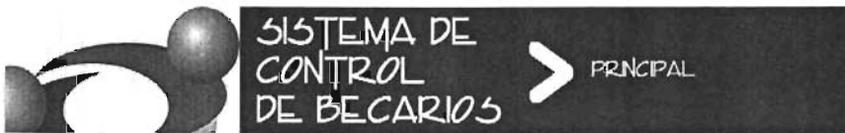


[Lista de Salas](#)

id_sala	Nombre de Sala
sala1	Matemáticas
sala10	Infraestructura de Nuestra Nación
sala11	Química
sala12	Tecnología Satelital
sala13	Una Balse en el Tiempo
sala14	Sala X
sala2	Biodiversidad
sala3	Universo
sala4	Agricultura y Alimentación

Así es como agregamos una Sala al Sistema.

## II. f) Modificar Datos de una Sala



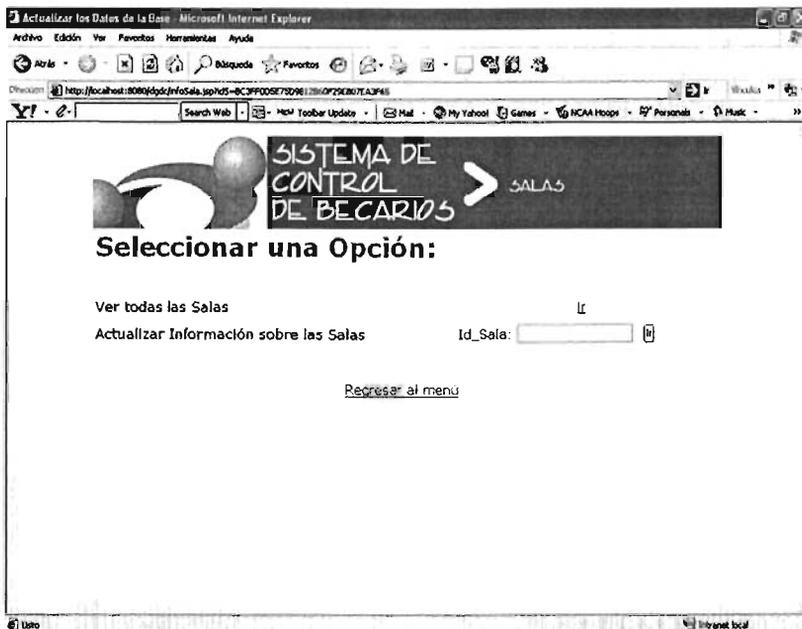
### Opciones de Menú del Sistema

>> [Información de Becarios](#) >> [Consulta Horario](#) >> [Modificar Datos de un Becario](#) >> [Información de Salas](#) >> [Alta de una Sala](#) >> [Modificar Datos de una Sala](#) >> [Agrega Horarios](#) >> [Información Horarios](#) >> [Manuales](#) >> [Listas de Becarios](#) >> [Sábano por Sala](#) >> [Sábano por Día](#) >> [Salir](#)



Modifica la Información de una Sala

Para modificar el nombre de una Sala solo tenemos que seleccionar la opción [>>Modificar Datos de una Sala](#) del menú, aparecerá la siguiente pantalla en donde solo tenemos que dar el id de la Sala:



Si deseamos ver todas las salas contenidas en el sistema solo hay que dar un clic en la referencia ir como se muestra:

Ver todas las Salas

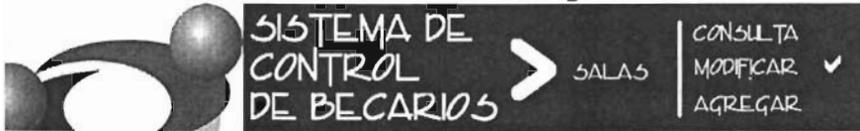
[ir](#)

Como ejemplo vamos a cambiar el nombre de la sala que tiene el id: sala 14 su nombre es Sala X, por Sala X32, por lo que tecleamos en la caja de texto el id de la sala como se muestra:

Actualizar Información sobre las Salas

Id\_Sala:  [ir](#)

Y damos clic en el botón [ir](#) se mostrará lo siguiente:



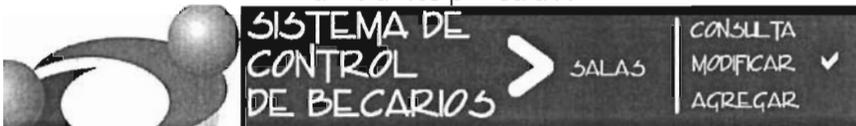
## Modificando la Información de la Sala: sala14

Id_Sala:	sala14
Nombre de la Sala:	Sala X

Modificar la Información de la Sala: sala14

[Regresar](#)

Como se observa en el Recuadro marcado en rojo tenemos el nombre de la sala Sala X ahora lo cambiamos por Sala X32

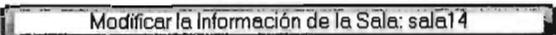


## Modificando la Información de la Sala: sala14

Id_Sala:	sala14
Nombre de la Sala:	Sala X32

Modificar la Información de la Sala: sala14

[Regresar](#)

damos clic en el botón  para que la información se actualice.

Si la información se actualizo de forma correcta tendremos el siguiente mensaje en pantalla;

## Modificando la Información de la Sala: sala14

Id_Sala:	sala14
Nombre de la Sala:	Sala X32



**Se ha modificado la Información de la Sala: sala14**

[Volver a la Página Inicial](#)

Para regresar a seleccionar otro id de sala para modificar damos clic en la referencia [Volver a la Página Inicial](#)

## II. g) Agrega Horarios

Aquí podemos agregar horarios al sistema como se desee.



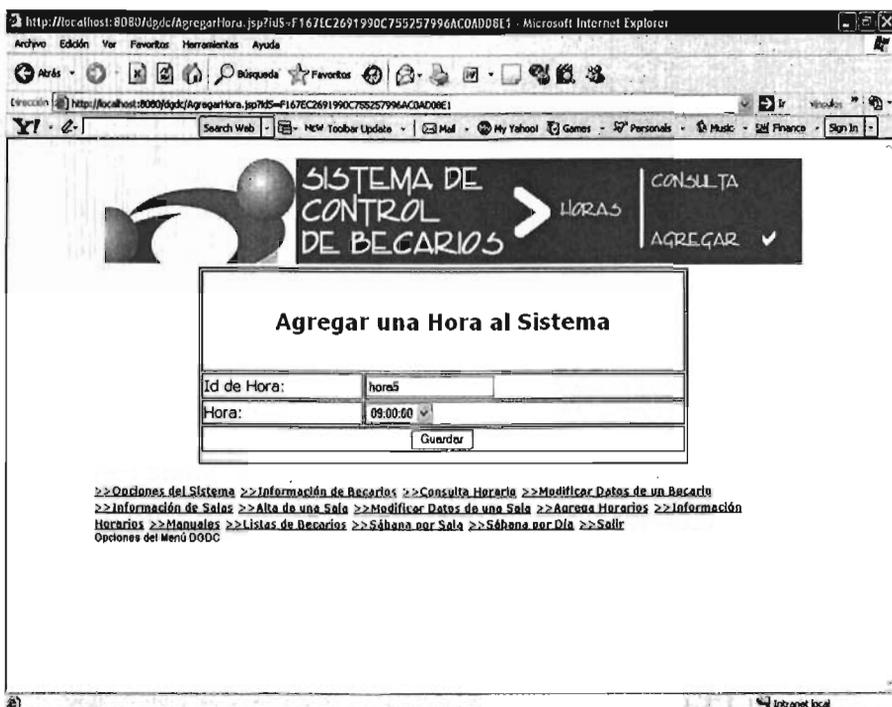
## Opciones de Menú del Sistema

[>>Información de Becarios](#) >>[Consulta Horario](#) >>[Modificar Datos de un Becario](#) >>[Información de Salas](#) >>[Alta de una Sala](#) >>[Modificar Datos de una Sala](#) >>[Agrega Horarios](#) >>[Información Horarios](#) >>[Manuales](#) >>[Listas de Becarios](#) >>[Sábana por Sala](#) >>[Sábana por Día](#) >>[Salir](#)

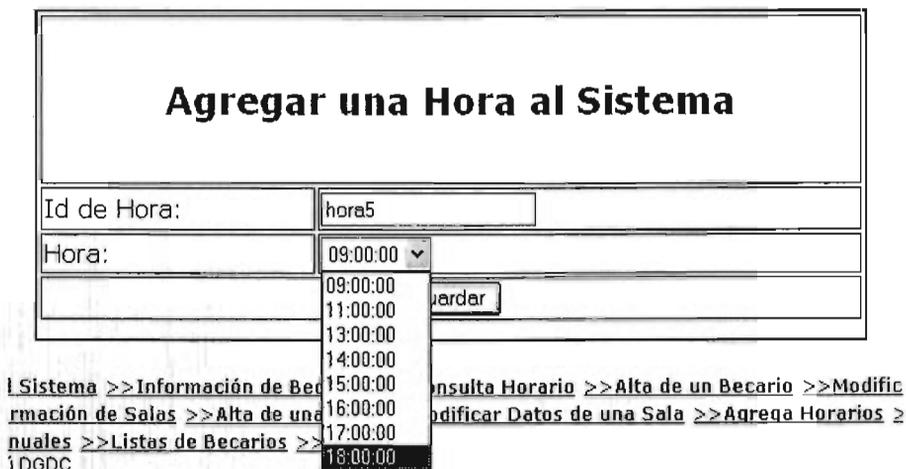


Agrega Horarios al Sistema

Damos clic en la opción [>>Agrega Horarios](#) la siguiente pantalla será:



Supongamos que necesitamos agregar un horario, que sea a las 18:00 hrs., el id de hora se genera automáticamente por lo que no hay que cambiarlo, entonces seleccionaremos de la lista de Hora: las 18:00 hrs. como se ilustra a continuación:



Damos clic en el botón guardar  para agregar el horario al sistema.

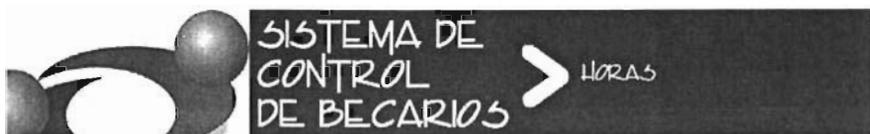
Si se ha guardado correctamente tendremos un mensaje como este:



El registro ha sido agregado

[Lista de Horas](#)

Para regresar a ver que efectivamente se agrego nuestro horario damos clic en la referencia Lista de Horas [Lista de Horas](#)



Lista de Horas

id_llegada	Hora de Llegada
hora1	09:00:00
hora2	10:00:00
hora3	14:00:00
hora4	15:00:00
hora5	18:00:00

[Agregar Hora](#)

[Regresar](#)

Podemos observar que se agrego con éxito nuestro horario id\_llegada hora5 18:00 hrs.

Para regresar al menú de opciones damos clic en la referencia [Regresar](#)

## II. h) Información Horarios

Se muestran los horarios contenidos dentro del sistema en esta opción:

[>>Información Horarios](#)



### Opciones de Menú del Sistema

[>>Información de Becarios](#) [>>Consulta Horario](#) [>>Modificar Datos de un Becario](#) [>>Información de Salas](#) [>>Alta de una Sala](#) [>>Modificar Datos de una Sala](#) [>>Agrega Horarios](#) [>>Información Horarios](#) [>>Manuales](#) [>>Listas de Becarios](#) [>>Sábana por Sala](#) [>>Sábana por Día](#) [>>Salir](#)



Agrega Horarios al Sistema

Al dar clic en [>>Información Horarios](#) se muestra la siguiente pantalla:

Lista de Horas

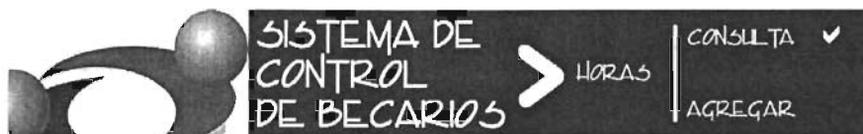
id_llegada	Hora de Llegada
hora1	09:00:00
hora2	10:00:00
hora3	14:00:00
hora4	15:00:00
hora5	18:00:00

Agregar Hora  
Regresar

En donde queremos podemos ver la Lista de Horas por id\_llegada y por la Hora de Llegada.

id_llegada	Hora de Llegada
hora1	09:00:00
hora2	10:00:00

Al dar clic sobre la columna de Hora de Llegada tendremos la información de esa hora seleccionada como se deja ver a continuación.



Hora Seleccionada

Id de Llegada	hora5
Hora de Llegada	18:00:00

[Eliminar Hora](#)

[Lista de Horas](#)

(Si queremos regresar a ver la lista damos clic sobre la referencia [Lista de Horas](#) ).

Supongamos como ejemplo que estamos seguros de borrar la hora5 que fue la que se creó en el inciso anterior h) damos clic sobre la referencia [Eliminar Hora](#). Si el registro se eliminó con éxito tendremos el siguiente mensaje en la pantalla.

El registro ha sido eliminado

[Lista de Horas](#)

y la referencia para regresar a la lista.

## II. i) Manuales

En esta parte de las opciones tenemos todos los manuales que se necesitan para manejar correctamente los sistemas entre ellos se encuentra este manual.



## Opciones de Menú del Sistema

>>Información de Becarios >>Consulta Horario >>Modificar Datos de un Becario >>Información de Salas >>Alta de una Sala >>Modificar Datos de una Sala >>Agrega Horarios >>Información Horarios >>Manuales >>Listas de Becarios >>Sábana por Sala >>Sábana por Día >>Salir



Guías de usuario o manuales del sistema

Hacemos clic en la liga [>>Manuales](#) y la pantalla que aparece es la siguiente:

**Manuales de Usuario del Sistema de Control de Becarios DGDC**

Seleccione uno de los Manuales:

Manuales	HTML	PDF
C2WCAM (credenciales)	C2WCAM	Ir
Migrar Base de Datos de Access a Mysql (De la base local al Servidor)	Mysql-Access	Ir

[Regresar al menú](#)

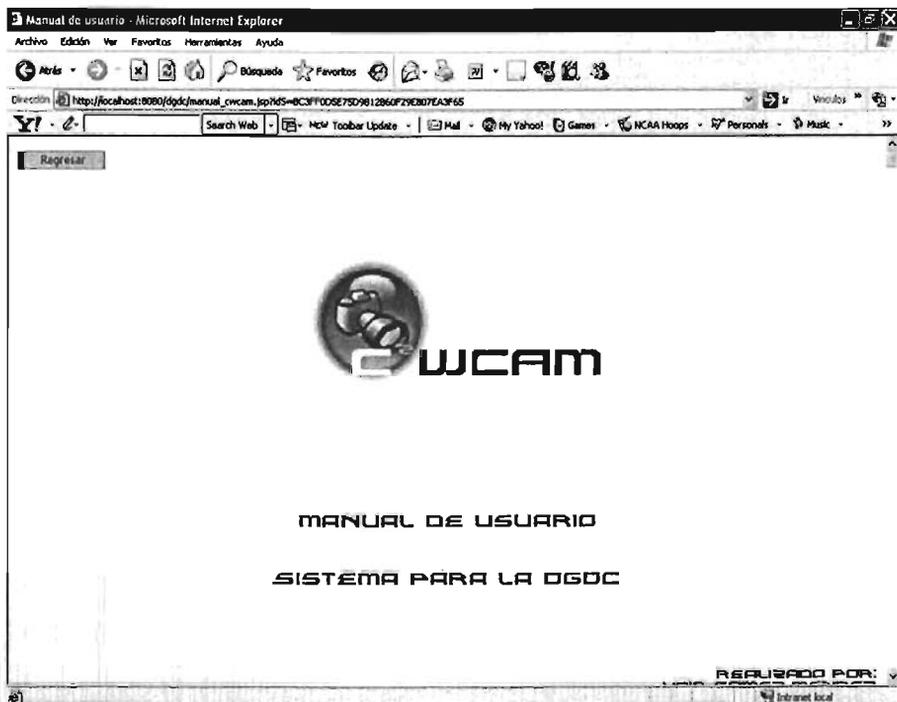
Para regresar al menú de opciones damos clic en [Regresar al menú](#)

El formato de los manuales es en HTML y en PDF esto es para facilitar la visualización de los mismos. Si damos clic en las referencias aparecerán los manuales en sus respectivos formatos:

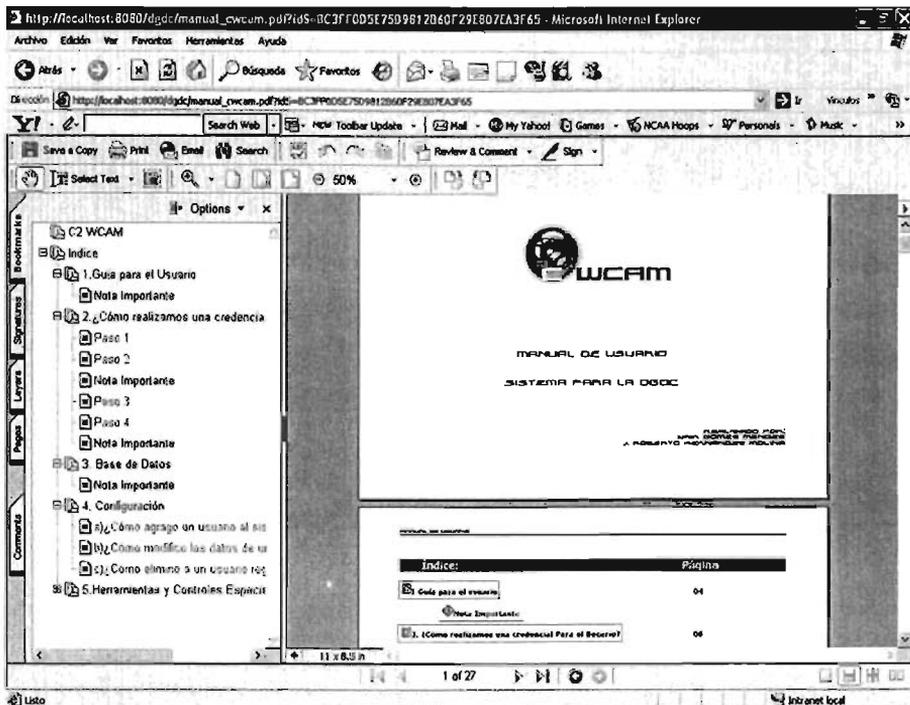
**Seleccione uno de los Manuales:**

Manuales	HTML	PDF
C2WCAM (credenciales)	<a href="#">C2WCAM</a>	<a href="#">Ir</a>
Migrar Base de Datos de Access a Mysql (De la base local al Servidor)	<a href="#">Mysql-Access</a>	<a href="#">Ir</a>

Supongamos que deseamos ver el manual C2WCAM en formato HTML solo damos clic en la referencia [C2WCAM](#) de la columna HTML, se mostrara lo siguiente:



Para regresar a los manuales solo damos clic en el botón [Regresar](#). Si lo que deseamos es ver el manual en formato de Acrobat o .pdf damos clic en la referencia [Ir](#) de la columna PDF; se mostrara lo siguiente:

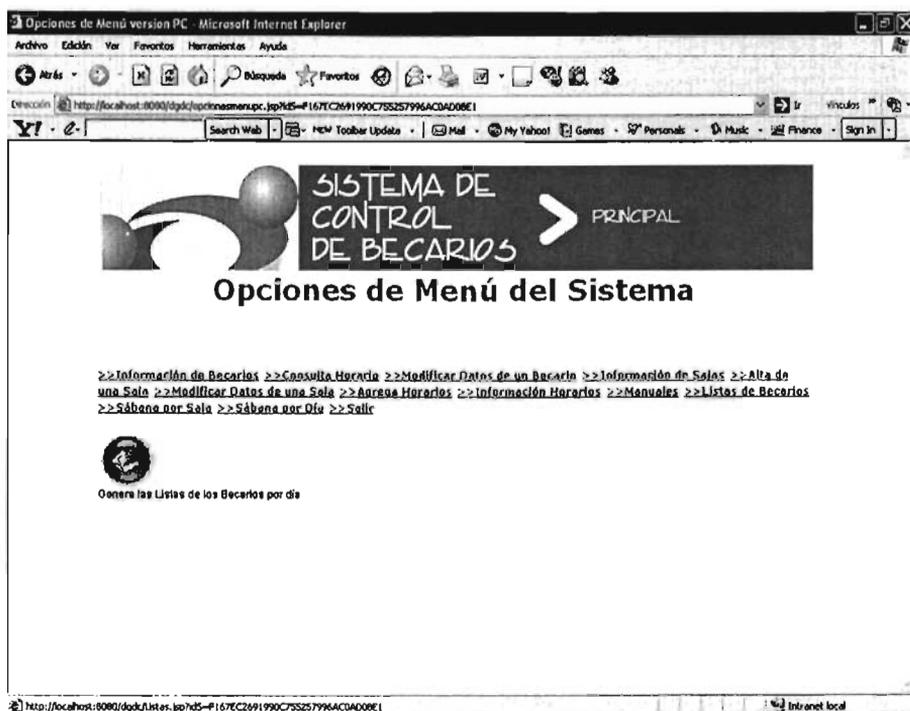


Para regresar a los manuales damos clic en  Atrás en la barra del explorador.

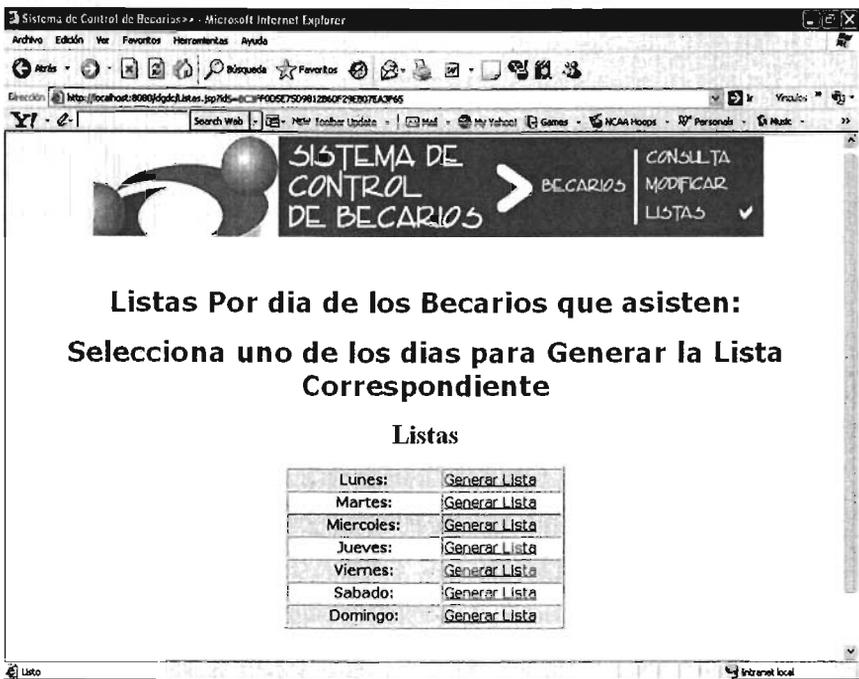
El mismo ejemplo se aplica para todos los manuales contenidos en el apartado Manuales de este sistema.

## II. j) Listas de Becarios

En listas de Becarios podremos acceder e imprimir la lista de asistencia de los becarios que asisten por día, es decir, una lista de Lunes a Domingo de los Becarios.



Hacemos clic en la referencia [>>Listas de Becarios](#) la pantalla siguiente muestra las opciones por día para visualizar las listas.



**SISTEMA DE CONTROL DE BECARIOS**

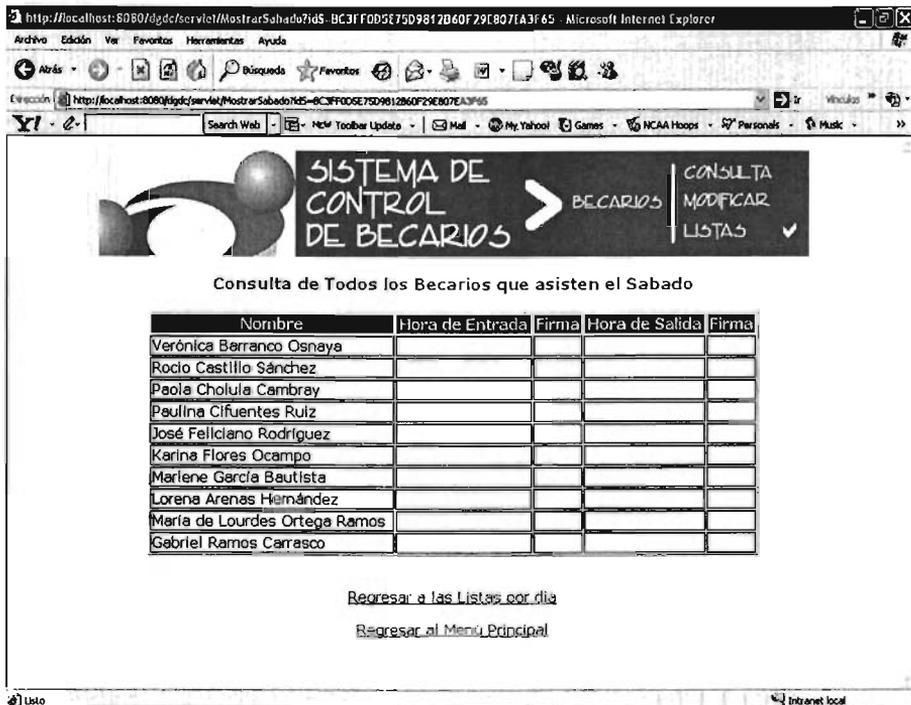
CONSULTA  
MODIFICAR  
LISTAS ✓

**Listas Por día de los Becarios que asisten:  
Selecciona uno de los días para Generar la Lista  
Correspondiente**

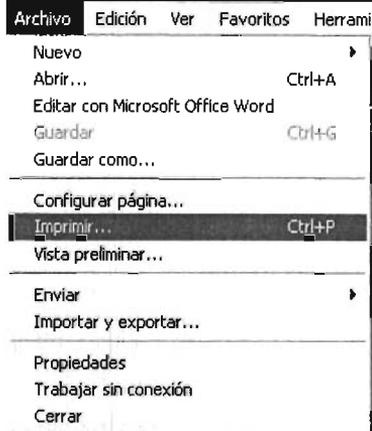
**Listas**

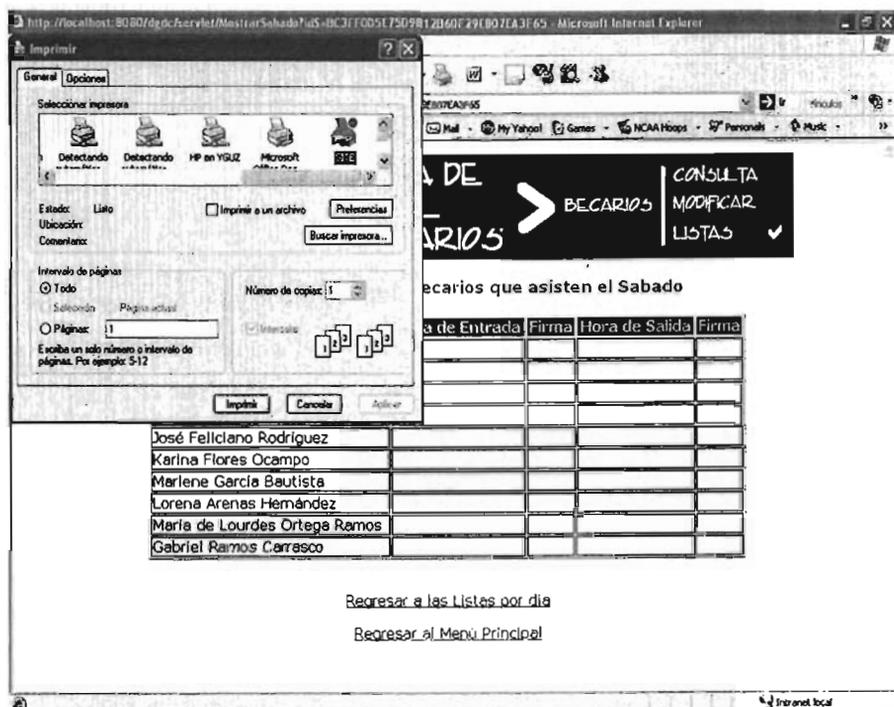
Lunes:	<a href="#">Generar Lista</a>
Martes:	<a href="#">Generar Lista</a>
Miércoles:	<a href="#">Generar Lista</a>
Jueves:	<a href="#">Generar Lista</a>
Viernes:	<a href="#">Generar Lista</a>
Sábado:	<a href="#">Generar Lista</a>
Domingo:	<a href="#">Generar Lista</a>

Por ejemplo: nosotros queremos ver la lista de los Sábados (que Becarios asisten el Sábado), entonces solo damos clic sobre la referencia [Generar Lista](#)



Para imprimir esta lista damos clic en el icono (parte superior de la barra del explorador)  o Archivo y seleccionamos la opción Imprimir tal como se ilustra:





Seleccionamos nuestras preferencias de impresión y damos clic en el botón  y Listo.

Para regresar a ver otra lista damos clic en la referencia

[Regresar a las Listas por día](#)

Para regresar al menú de opciones damos clic en la referencia

[Regresar al Menú Principal](#)

## II. k) Generar Sábana por Sala

Para generar la Sábana por sala solo tenemos que ir a la siguiente opción [>>Sábana por Sala](#) como se muestra en la pantalla siguiente



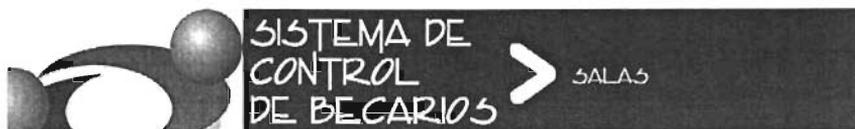
## Opciones de Menú del Sistema

>> [Información de Becarios](#) >> [Consulta Horario](#) >> [Modificar Datos de un Becario](#) >> [Información de Salas](#) >> [Alta de una Sala](#) >> [Modificar Datos de una Sala](#) >> [Agrupar Horarios](#) >> [Información Horarios](#) >> [Manuales](#) >> [Listas de Becarios](#) >> [Sábana por Sala](#) >> [Sábana por Día](#) >> [Salir](#)



Genera la Sábana por Sala

Damos clic en esta referencia >> [Sabana por Sala](#) se muestra lo siguiente:



## Consulta de la Sábana por Sala

### Selecciona una Sala de la Lista

Sala o Área de preferencia:	Biología Humana y Salud	Generar Sábana
-----------------------------	-------------------------	----------------

[Regresar al Menú](#)

Para ver la Sábana de la sala que nosotros queremos solo hay que seleccionarla de la lista como se ilustra a continuación:

## Consulta de la Sábana por Sala

### Selecciona una Sala de la Lista

Sala o Área de preferencia:	Biología Humana y Salud	Generar Sábana
	<ul style="list-style-type: none"> <li>Biología Humana y Salud</li> <li>Agricultura y Alimentación</li> <li>Universo</li> <li>Matemáticas</li> <li>Biodiversidad</li> <li>Conciencia de Nuestra Nación</li> <li>Energía</li> <li>Espacio Infantil</li> <li>Estructura de la Materia</li> <li><b>Infraestructura de Nuestra Nación</b></li> <li>Química</li> <li>Tecnología Satelital</li> <li>Una Bolsa en el Tiempo</li> <li>Sala X32</li> </ul>	

Supongamos que deseamos ver la Sábana de la Sala: Infraestructura de Nuestra Nación, entonces seleccionamos esta sala y damos clic en el botón **Generar Sábana**. Se mostrara lo siguiente:

Sistema de Control de Becarios >> Generar la Sábana por Sala sala10 - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Inicio | Buscar | Favoritos | Herramientas | Ayuda

http://localhost:8080/ggdc/GeneraSábana.jsp?IDS=167EC2691990C75525799AC0A0400E1

Search Web | New Toolbar Update | Mail | My Yahoo! | Games | Personals | Music | Finance | Sign In



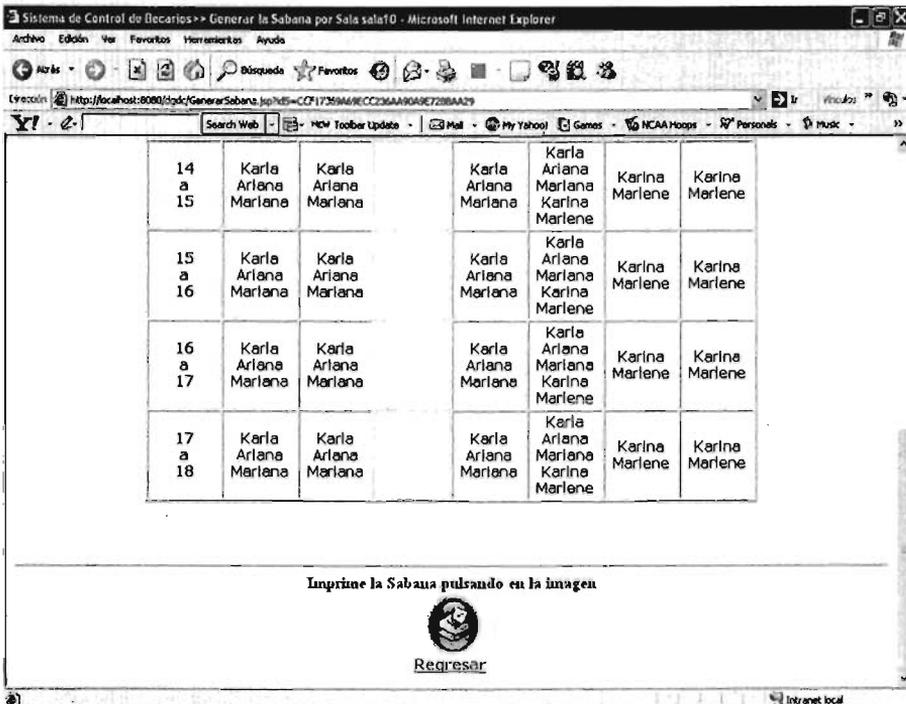
# SISTEMA DE CONTROL DE BECARIOS

> SALAS

## Sábana de la Sala: Infraestructura de Nuestra Nación

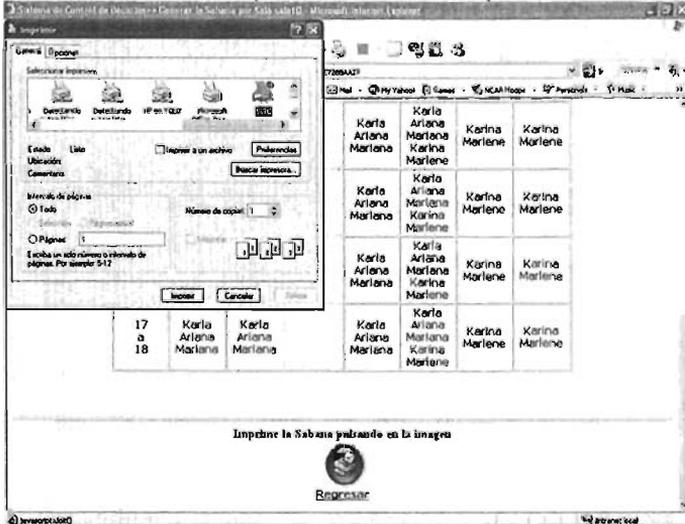
Hora	Lunes	Martes	Miercoles	Jueves	Viernes	Sabado	Domingo
9 a 10	Fernando Christian Eugenio	Fernando Christian Eugenio	Fernando Christian Eugenio		Fernando Christian Eugenio		
10 a 11	Fernando Christian Eugenio	Fernando Christian Eugenio	Fernando Christian Eugenio		Fernando Christian Eugenio		
11 a 12	Fernando Christian Eugenio	Fernando Christian Eugenio	Fernando Christian Eugenio		Fernando Christian Eugenio		
12 a 13	Fernando Christian Eugenio	Fernando Christian Eugenio	Fernando Christian Eugenio		Fernando Christian Eugenio		
13 a 14							
14 a 15	Karla Ariana Marlana	Karla Ariana Marlana		Karla Ariana Marlana	Karla Ariana Marlana	Karina Marlene	Karina Marlene

Inicio | Intranet local



Aquí podemos ver la Sabana de la Sala seleccionada para imprimir la

Sabana damos clic en la imagen  se mostrará una nueva ventana



Seleccionamos la impresora y nuestras opciones y damos clic en el botón **Imprimir** Para regresar a seleccionar otra sala damos clic en la referencia **Regresar**

## II. I) Generar Sábana por Día

Para generar la Sábana por Día damos clic en la opción **>>Sábana por Día** del menú como se muestra en la pantalla siguiente:



### Opciones de Menú del Sistema

[>>Información de Becarios](#) [>>Consulta Horario](#) [>>Modificar Datos de un Becario](#) [>>Información de Salas](#) [>>Alta de una Sala](#) [>>Modificar Datos de una Sala](#) [>>Agrega Horarios](#) [>>Información Horarios](#) [>>Manuales](#) [>>Listas de Becarios](#) [>>Sábana por Sala](#) [>>Sábana por Día](#) [>>Salir](#)



Genera la Sábana por Día

La pantalla que se abrirá será:



### Consulta de la Sábana por Día

Selecciona un Día de la Lista:

[Regresar al Menú](#)

Supongamos que deseamos ver la Sabana del día lunes; seleccionamos el día de la lista y damos clic en el botón **Generar Sábana por Día**

## Consulta de la Sábana por Día

Selecciona un Día de la Lista: Lunes Generar Sábana por Día

Lunes

Martes

Miercoles

Jueves

Viernes

Sábado

Domingo

Reg: enú

Lo que se muestra a continuación es la Sábana generada para el día lunes con la información de los nombres de los becarios y las salas correspondientes:

Sistema de Control de Becarios >> lunes - Microsoft Internet Explorer

http://localhost:8080/GeneraSábanaDi...?id=1670C261990C79225799ACD400E1



### SISTEMA DE CONTROL DE BECARIOS > BECARIOS

## Sábana del Día: lunes

Hora	Matemáticas	Infraestructura de Nuestra Nación	Química	Tecnología Satelital	Luna Balsa en el Tiempo	Sala X32	Biodiversidad	Universo	Agricultura y Alimentación	Biología Humana y Salud	Geociencia de Nuestra Nación	Ener
9:00		Fernando Christian Eugenio		Diana Liebeth Shyly Rebecca Michol				Héctor Carla Marlén		Verónica		CS
10:00		Fernando Christian Eugenio		Diana Liebeth Shyly Rebecca Michol			Paulina	Héctor Carla Marlén		Verónica		CS
11:00		Fernando Christian Eugenio		Diana Liebeth Shyly Rebecca Michol			Paulina	Héctor Carla Marlén		Verónica		CS
12:00		Fernando Christian Eugenio		Diana Liebeth Shyly Rebecca Michol			Paulina	Héctor Carla Marlén		Verónica		CS
13:00							Paulina					
14:00		Karla Ariana Mariana										
15:00												

The screenshot shows a web browser window titled 'Sistema de Control de Becarios'. The address bar shows a local URL. The main content area displays a calendar grid for Monday. The grid has columns for dates and names. The names listed are: Fernando Olivares Eugenio, Diana Lizabeth Bpuly Rosana Michal, Karla Arlene Hartana, and Miguel Karla Harlin. Below the grid, there are two buttons: 'Imprimir la Sábana día: lunes' and 'Regresar'.

11									
12		Fernando Olivares Eugenio	Diana Lizabeth Bpuly Rosana Michal		Paulina	Miguel Karla Harlin		Verónica	CS
13		Fernando Olivares Eugenio	Diana Lizabeth Bpuly Rosana Michal		Paulina	Miguel Karla Harlin		Verónica	CS
14					Paulina				
15		Karla Arlene Hartana							
16		Karla Arlene Hartana							
17		Karla Arlene Hartana							
18		Karla Arlene Hartana							

Imprimir la Sábana día: lunes

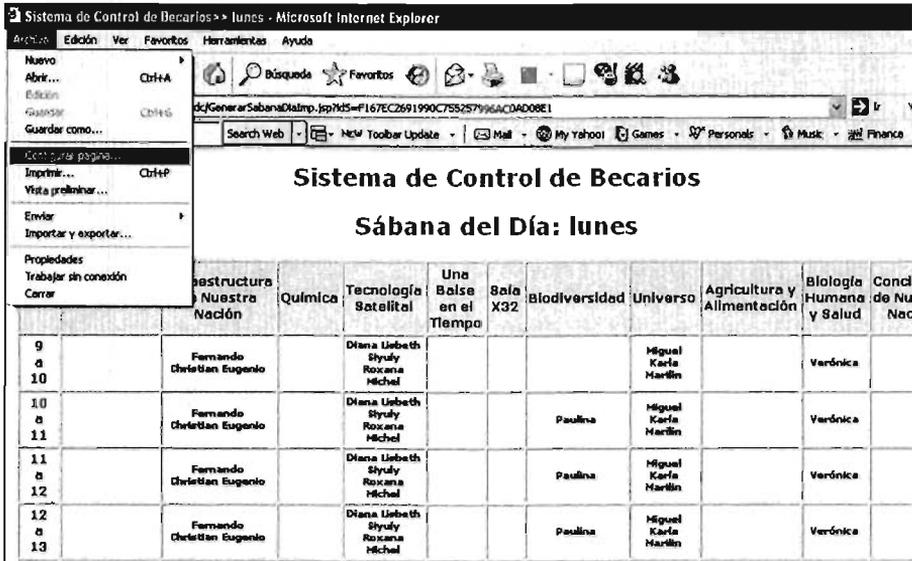
Regresar

Para imprimir esta Sábana damos clic en el botón

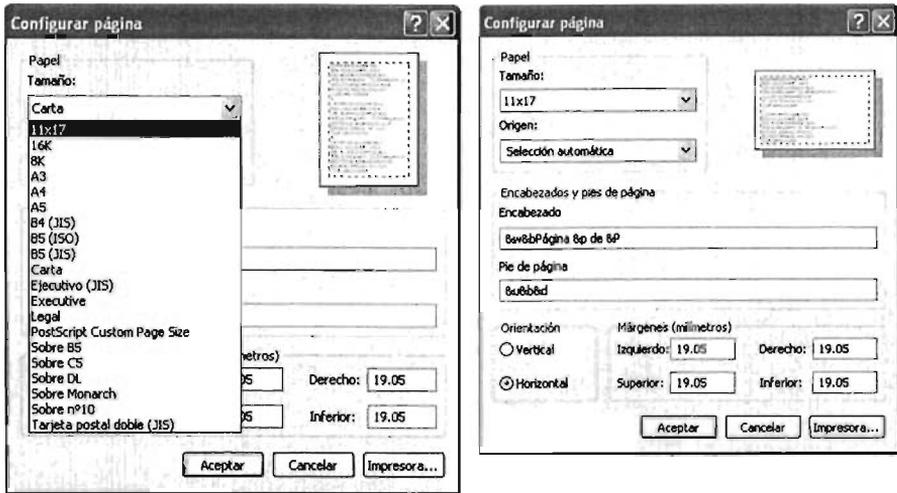
si queremos regresar a seleccionar otro

día para generar su Sábana damos clic sobre la referencia Regresar

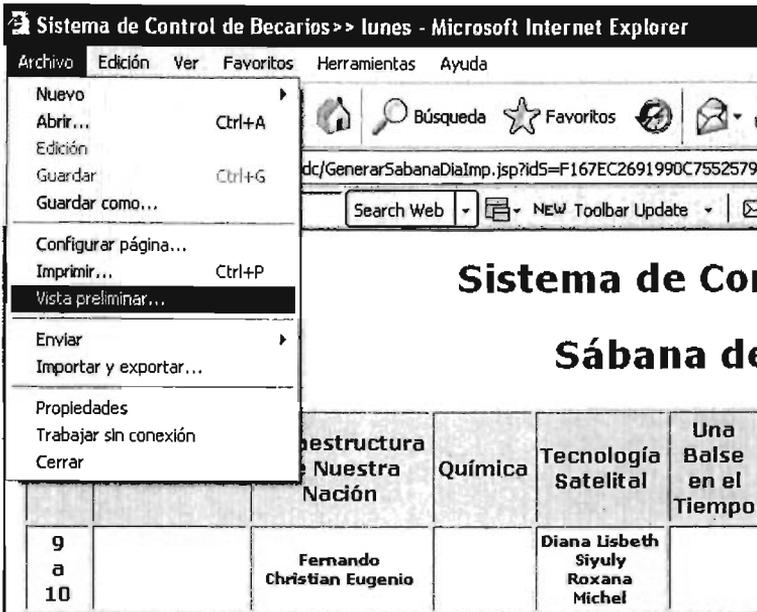
Cabe señalar que esta sábana se imprime en una hora de 11" X 16" por tanto debemos configurar la página así como la impresora en la cual se va a imprimir nuestra sábana. En la barra de menús de nuestro explorador nos vamos al menú Archivo después a Configurar página como se muestra y damos clic.



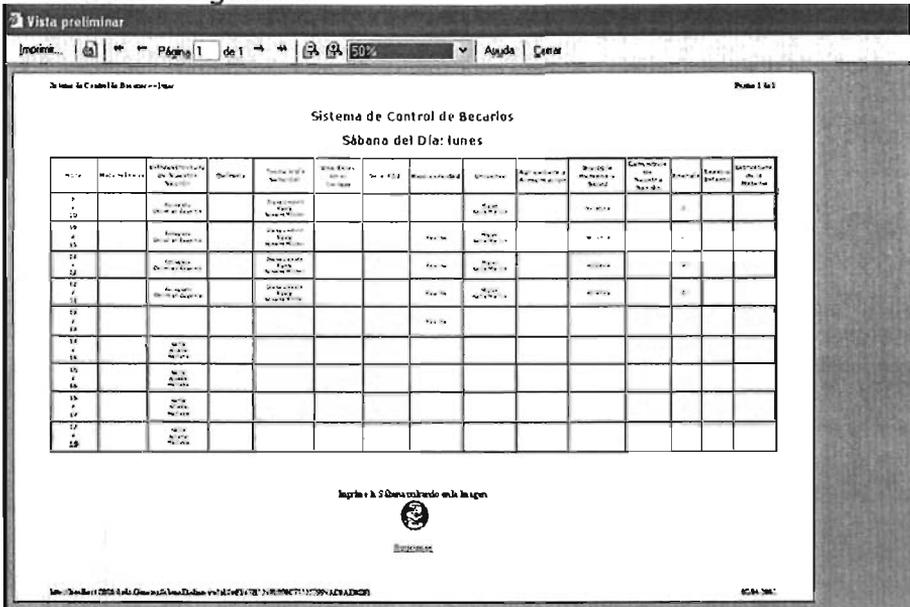
Se abrirá una pantalla de selección donde escogemos el tamaño indicado y la hoja en Horizontal. Como se muestra:



Damos clic en el botón **Aceptar**  
Ahora para ver que realmente se imprimirá toda nuestra Sábana vamos de nueva cuenta al menú Archivo y seleccionamos Vista preeliminar...



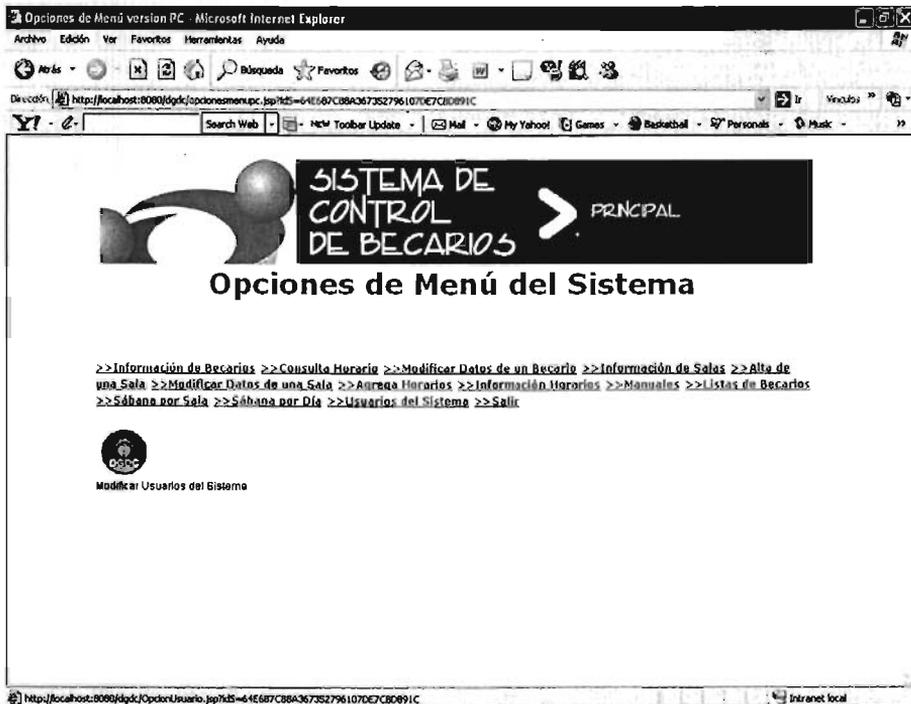
Se mostrará lo siguiente:



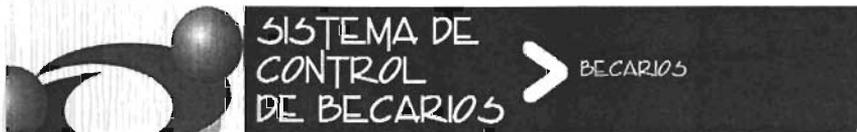
Cerramos la Vista preeliminar y damos clic en la imagen para imprimir, seleccionamos nuestra impresora y tendremos nuestra Sábana.

## II. m) Usuarios del Sistema

En este modulo el usuario del sistema podrá dar de alta a otro usuario de confianza o cambiar si así lo desea su login y su password, en la siguiente referencia [>>Usuarios del Sistema](#) tenemos esta opción:



Al dar clic en esta liga [>>Usuarios del Sistema](#) tenemos lo siguiente:



### Opciones Menú Usuario

Seleccióna una de las Opciones siguientes:

Modificar Cuenta de Usuario:	<input type="button" value="Modificar"/>
Crear un Nuevo Usuario:	<input type="button" value="Crear"/>

[Regresar el menú Principal](#)

Las opciones del Menú de Usuario: donde podemos seleccionar:

A) Modificar la cuenta de Usuario y B) Crear un nuevo Usuario.

Vamos a ver la primera opción:

A) Modificar cuenta de usuario:

Modificar Cuenta de Usuario:  Hacemos clic en el botón modificar y tenemos lo siguiente:



### Modificar Login y Password:

Por Favor intoduzca su Login y Password Actuales

Login:	<input type="text" value="Salvador"/>	<input type="button" value="Modificar"/>
Password:	<input type="password" value="*****"/>	<input type="button" value="Modificar"/>

[Regresar al menú Usuario](#)

Si el login o el password son incorrectos nos aparecerá un mensaje como el siguiente al momento de dar clic en el botón

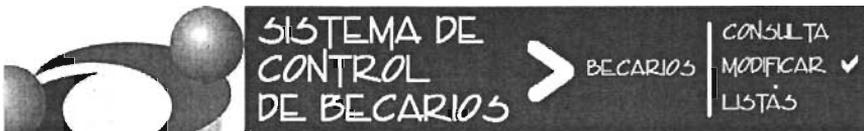


## Modificando la Información del Usuario

No se encontró el dato ingresado

[Regresar](#)

Damos clic en regresar para volver a identificarnos. De lo contrario si introducimos el login y el password de forma correcta aparecerá una pantalla como la siguiente:



## Modificando la Información del Usuario

Ingresa tu Nuevo Login Y Password:

Login:	<input type="text" value="Salvador"/>
Password:	<input type="password" value="*****"/>
Confirmar Password:	<input type="password"/>

Aquí podremos modificar el login y el password, se pide que se confirme el password para asegurarnos de que el password es el que realmente queremos.

## Modificando la Información del Usuario

Ingresas tu Nuevo Login Y Password:

Login:	<input type="text" value="Salvador_M"/>
Password:	<input type="password" value="*****"/>
Confirmar Password:	<input type="password" value="*****"/>

Damos clic en el botón  y la información se actualizará tal como se muestra.

Sistema de Control de Becarios >> Modificando la Información del Usuario Salvador - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Archivos Recientes

http://localhost:8000/dpdt/ModificarU.jsp?ID=64E687C8A3673527961070E7CB0991C

Search Web - NEW Toolbar Update - Mail - My Yahoo! - Games - Basketball - Personal - Music

SISTEMA DE CONTROL DE BECARIOS > BECARIOS | CONSULTA MODIFICAR ✓ LISTAS

### Modificando la Información del Usuario

Login: Salvador\_M

SISTEMA DE CONTROL DE BECARIOS > ¡ GUARDADO !

Se ha modificado la Información del Usuario

[Volver a la Página Inicial](#)

Listo Intranet local

Damos clic en la referencia [Volver a la Página Inicial](#) para regresar al Menú de opciones.

Si los passwords nos son iguales tendremos un mensaje como el siguiente:

Login: |Salvador\_M



**No se pudo modificar la Información del Usuario  
Los passwords no son iguales Intentalo de Nuevo**

[Volver a la Página Inicial](#)

Por lo que tendremos que regresar al Menú de Opciones.

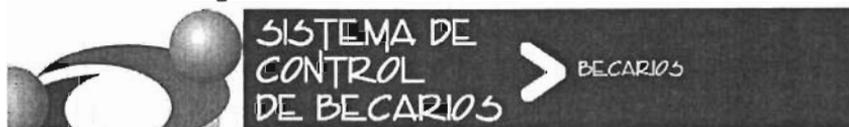
B) Crear un nuevo usuario

Crear un Nuevo Usuario:

Crear

Damos clic en Crear

Y se mostrará lo siguiente:



## Agregar un Nuevo Usuario al Sistema

Introduce el Nuevo Login y Password

Login:	<input type="text"/>
Password:	<input type="password"/>
<input type="button" value="Agregar Usuario"/>	

[Regresar a Opciones de Usuario](#)

Pondremos el login y el password del nuevo usuario damos clic en el botón  y Listo tenemos un nuevo usuario en el sistema, esto se muestra a continuación:



El registro ha sido agregado

[Regresar](#)

Damos clic sobre la referencia [Regresar](#) para ir al menú de Opciones del sistema.

## II. n) Salir

Para asegurarnos de Salir de la Sesión correctamente damos clic en la referencia [>>Salir](#) del menú de opciones.



## Opciones de Menú del Sistema

[>>Información de Becarios](#) [>>Consulta Horario](#) [>>Modificar Datos de un Becario](#) [>>Información de Salas](#) [>>Alta de una Sala](#) [>>Modificar Datos de una Sala](#) [>>Agenda Horarios](#) [>>Información Horarios](#) [>>Manuales](#) [>>Listas de Becarios](#) [>>Sábana por Sala](#) [>>Sábana por Día](#) [>>Salir](#)



Cerrar la Sesión y Salir del Sistema

Hacemos clic sobre nuestra referencia [>>Salir](#) la siguiente pantalla será:



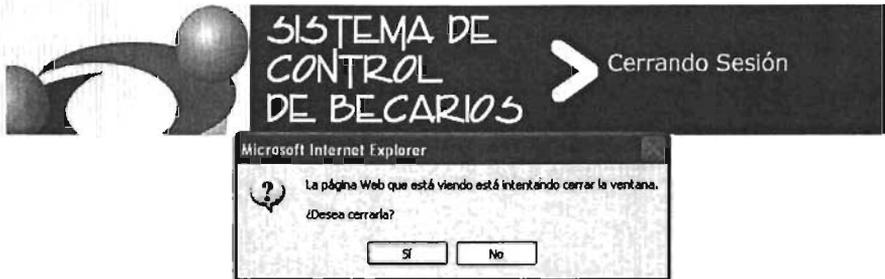
[Salir del Sistema](#)

[Regresar al Sistema](#)

Si deseamos regresar al Sistema damos clic sobre la referencia:

Regresar al Sistema Para volver al menú de opciones.

Si lo que deseamos es salir definitivamente del sistema elegimos la opción Salir del Sistema, ya que es la forma correcta de cerrar el sistema con seguridad, también se cerrara nuestro explorador, pero antes enviara un mensaje en una ventana emergente.

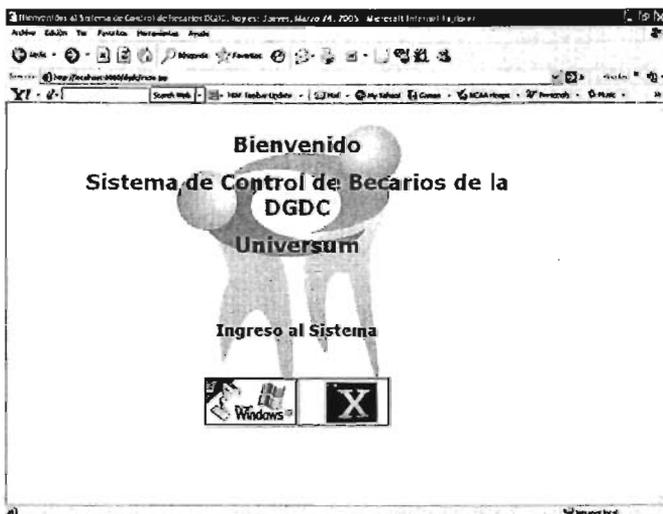


Como deseamos salir aceptamos y damos clic en el botón  y se cerrara nuestro navegador de Internet.

### III. Opciones del Menú del Sistema Mac (Versión Mac)

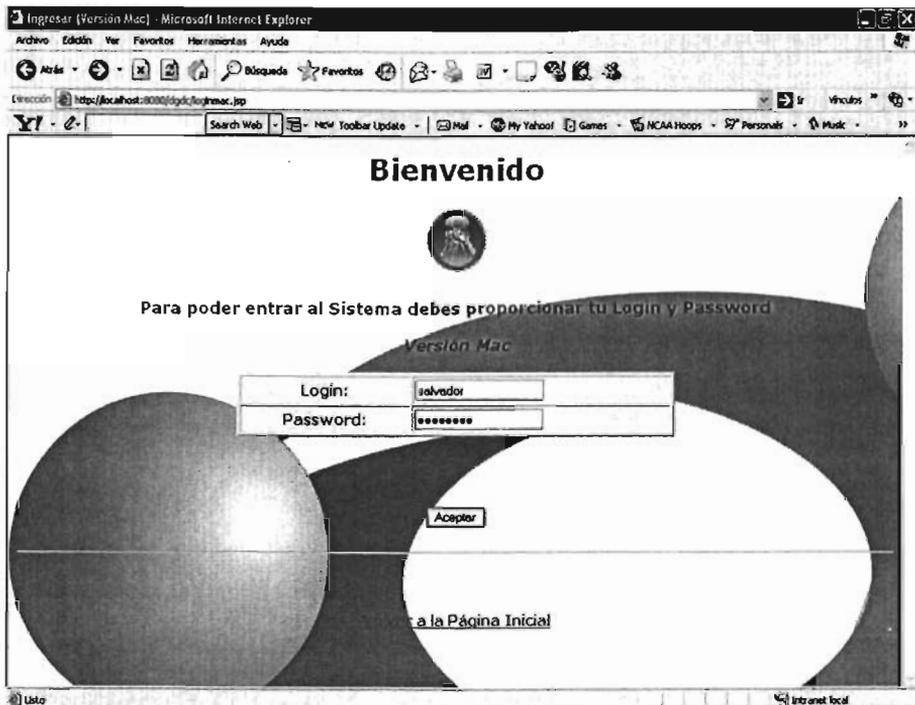
Las opciones del Menú en Mac como en PC son las mismas lo único que cambia es el menú de opciones, tal como se muestra a continuación.

El sistema nos da la bienvenida al poner la dirección en el explorador <http://servidor/dgdc/inicio.jsp>



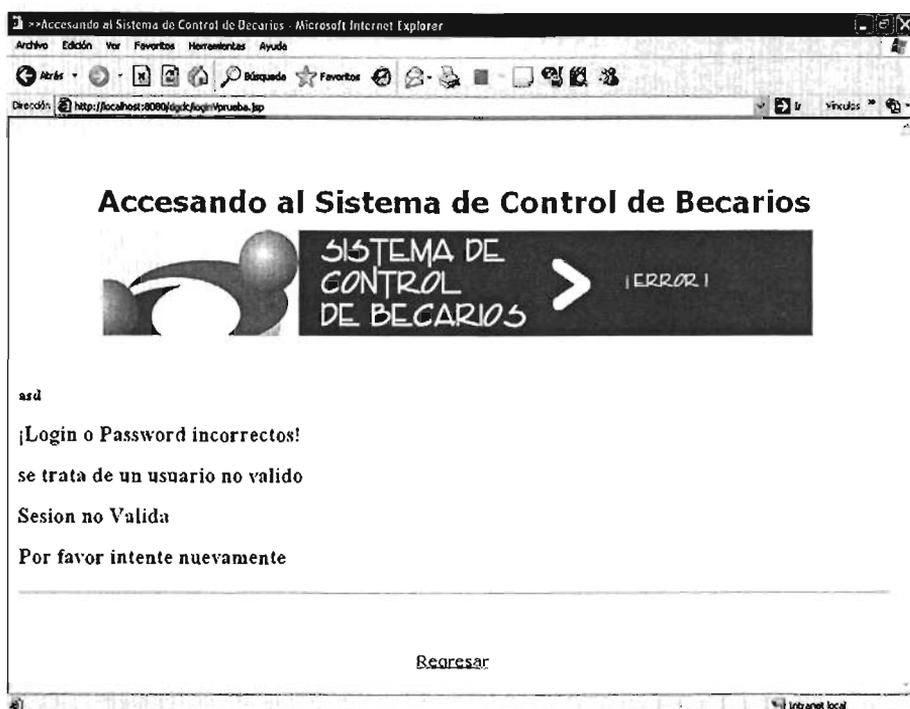
Ahora seleccionamos el icono  para la Versión Mac si nos encontramos en una computadora Macintosh al hacer clic en el icono tendremos la pantalla de ingreso al sistema en donde tendremos que proporcionar nuestro Login o nombre de usuario y nuestra password o contraseña para ser autorizados y dar **Enter** o un clic en el botón **Aceptar**

Login:	<input type="text" value="Salvador"/>
Password:	<input type="password" value="••••••"/>



Nótese que esta es la versión Mac.

Si escribimos mal nuestro Login o Password o no estamos autorizados por el sistema al dar **Enter** o **Aceptar** nos mostrara el siguiente mensaje como se ilustra a continuación.



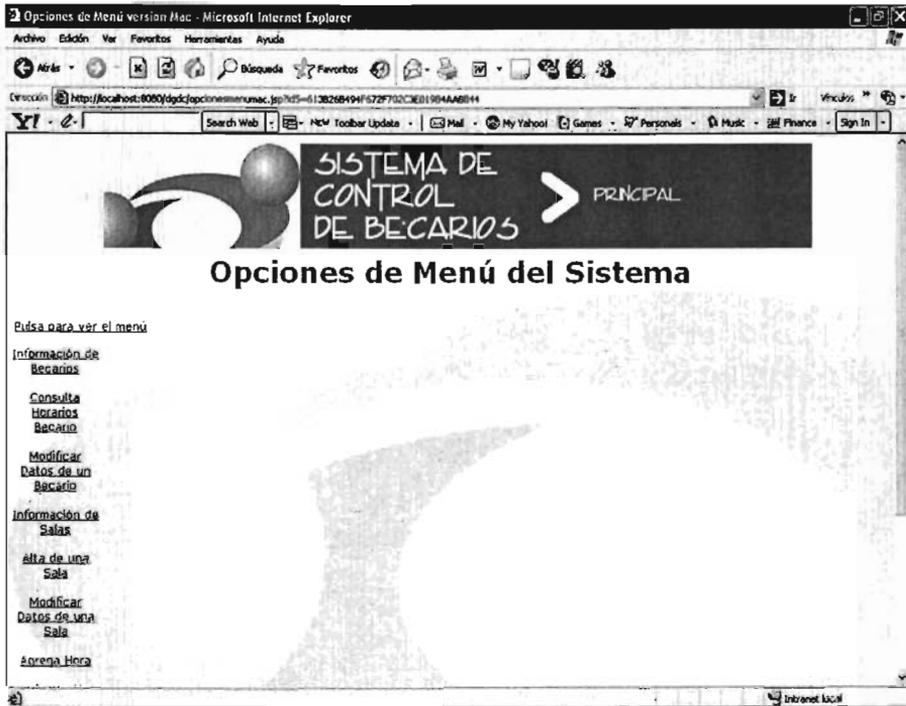
Al aparecer este mensaje solo tenemos que dar clic en la parte inferior en el centro donde esta la referencia de **Regresar** [Regresar](#)

Ahora volvemos a la página de acceso y escribimos correctamente el Login y el Password si lo hacemos bien y estamos autorizados entraremos con éxito al sistema:

Ya autorizados por el sistema tenemos todas las opciones siguientes:

Para ver el Menú de opciones solo hay que dar clic en la opción que aparece en el extremo superior izquierdo de la pantalla:

[Pulsa para ver el menú](#)

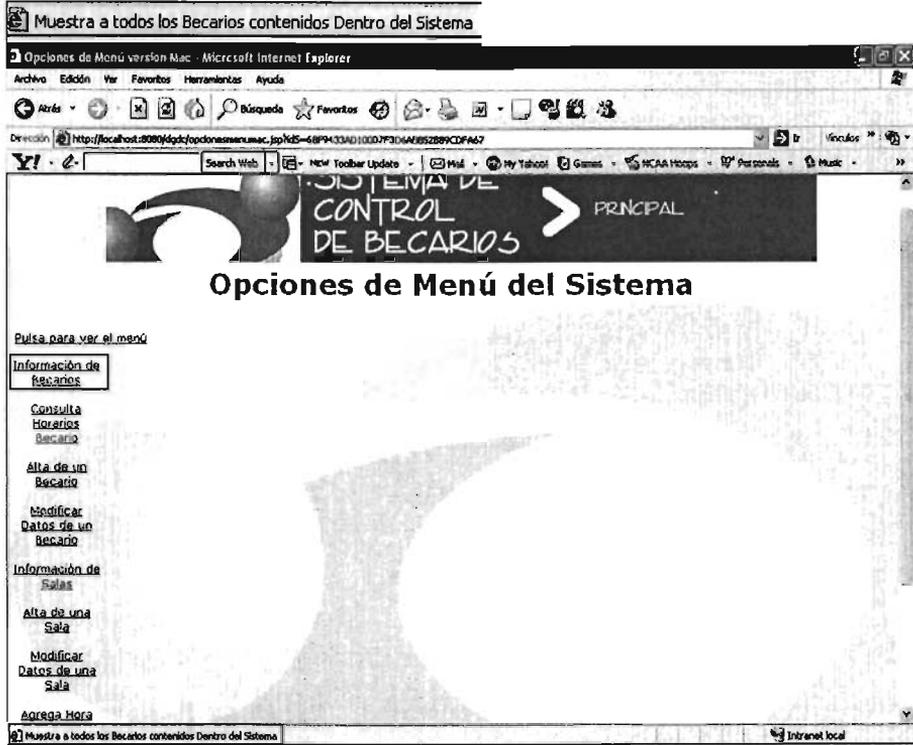


- a) Información de Becarios
- b) Consulta Horario
- c) Modificar Datos de un Becario
- d) Información de Salas
- e) Alta de una Sala
- f) Modificar Datos de una Sala
- g) Agrega Horarios
- h) Información Horarios
- i) Manuales
- j) Listas de Becarios
- k) Generar Sábana por Sala
- l) Generar Sábana por Día
- m) Salir

Ahora veamos una por una estas opciones:

### III. a) Información de Becarios

Al hacer clic sobre Información de Becarios Información de Becarios (Ver la siguiente figura) encerrada se encuentra esta opción en un recuadro rojo y en el segundo recuadro (ver la Barra de Estado del explorador) tenemos una breve descripción para ayuda al usuario de lo que hace esta referencia.



Al seleccionar esta opción se desplegará una pantalla en donde nosotros podremos ver a todos los Becarios que se encuentran en nuestra Base de Datos de forma dinámica se mostrará su Id de Becario así como su nombre completo.

http://localhost:8080/dgdc/servlet/MostrarBecario?idS=4819433AD10007F3D6A00570B9C0FA67 Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Inicio Buscar Favoritos

http://localhost:8080/dgdc/servlet/MostrarBecario?idS=4819433AD10007F3D6A00570B9C0FA67

Search Web

**SISTEMA DE CONTROL DE BECARIOS** > BECARIOS CONSULTA MODIFICAR LISTAS

Consulta de Todos los Becarios del Sistema

id Becario	Nombre Completo
BAOV820729001	Verónica Barranco Osrieva
BEDY830911023	Yasica Yislem Beltrán Díaz
CADC820921002	Celli Castro Domínguez
CASR810515011	Rocio Castilla Sánchez
CHCP830417013	Paola Cholula Carrizosa
CIRP790719024	Paulina Cifuentes Ruiz
FERJ790804014	José Feliciano Rodríguez
FLOK910315015	Karina Flores Ocampo
GABM821030016	Mariela García Bautista
GAPP810821003	Fernando Gachuz Fuentes
GOMK800415005	Karla Maldonado González
HEAL791006010	Lorena Arenas Hernández
JASD800929017	Diana Lisbeth Jardón Sánchez
LOMC810816004	Christian Eugenio López-Negrete Miranda
MAPA800530006	Ariana Martínez Pérez
MOFS800924018	Sivuly Morales Flores

http://localhost:8080/dgdc/servlet/MostrarBecario?idS=F167EC2691990C755257996AC0A08E1 Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Inicio Buscar Favoritos

http://localhost:8080/dgdc/servlet/MostrarBecario?idS=F167EC2691990C755257996AC0A08E1

Search Web

CASR810515011	Rocio Castilla Sánchez
CHCP830417013	Paola Cholula Carrizosa
CIRP790719024	Paulina Cifuentes Ruiz
FERJ790804014	José Feliciano Rodríguez
FLOK910315015	Karina Flores Ocampo
GABM821030016	Mariela García Bautista
GAPP810821003	Fernando Gachuz Fuentes
GOMK800415005	Karla Maldonado González
GOMY810708021	Yair Gómez Méndez
HEAL791006010	Lorena Arenas Hernández
HEMR810504020	Roberto Hernández Molina
JASD800929017	Diana Lisbeth Jardón Sánchez
LOMC810816004	Christian Eugenio López-Negrete Miranda
MAGE811708022	Eduardo Martínez Galván
MAPA800530006	Ariana Martínez Pérez
MOFS800924018	Sivuly Morales Flores
NALR800627019	Roxana Michel Navarrete León
NIJAM830514007	Mariana Núñez Almanza
ORRM810509008	María de Lourdes Ortega Ramos
RACG810924009	Gabriel Ramos Carrasco
RORK820813020	Karla Marilín Rodríguez Ramos
SAAM840310021	Miguel Santibáñez Andrade

Buscar

Si deseamos regresar al menú de opciones solo tenemos que dar clic en la referencia [Regresar](#).

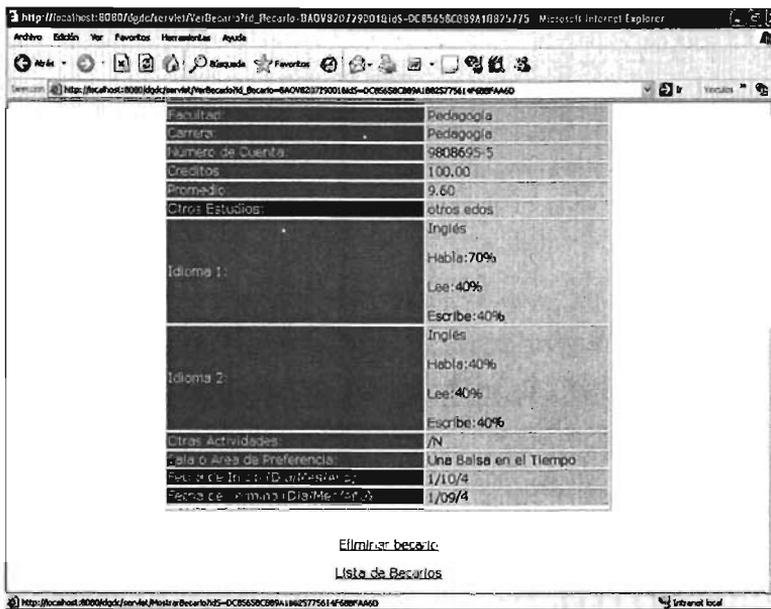
Si nosotros queremos saber la información de los Becarios solo basta dar un clic sobre su Nombre Completo (que se encuentra subrayado y de color azul), para que nos muestre sus datos personales.

id_Becario	Nombre Completo
BAOV820729001	<a href="#">Verónica Barranco Osnaya</a>

Esto se ilustra a continuación:

The screenshot shows a web browser window with the URL [http://localhost:8080/dp2cfserver/VerBecario?id\\_Becario=BAOV820729001&id\\_DCS=658C0B9A11875775](http://localhost:8080/dp2cfserver/VerBecario?id_Becario=BAOV820729001&id_DCS=658C0B9A11875775). The page title is 'SISTEMA DE CONTROL DE BECARIOS'. The main content area is titled 'Becario Seleccionado' and contains a table with the following data:

Id de Becario:	BAOV820729001
Nombre:	Verónica
Apellido Paterno:	Barranco
Apellido Materno:	Osnaya
Fecha de Nacimiento(Dia/Mes/Año):	29/7/82
Edad:	22
Eda. Civil:	soltero
Dirección:	Veracruzanos 26
Colonia:	Bonanza
Delegación:	Álvaro Obregón
Código Postal:	1220
Teléfono:	52-73-72-82
Correo Electrónico:	veroosnaya@hotmail.com
Universidad:	UNAM
Facultad:	Pedagogía
Carrera:	Pedagogía
Número de Cuenta:	9808695-5
Creditos:	100.00
Promedio:	9.60



Si deseamos regresar a la página anterior para ver los datos de otro Becario solo damos clic en Lista de Becarios Lista de Becarios, nos vamos a la parte inferior de esta pagina donde tenemos las opciones Eliminar becaric y Lista de Becarios como se muestra en la siguiente pantalla dentro de un recuadro rojo y uno negro respectivamente: Opciones de esta página:



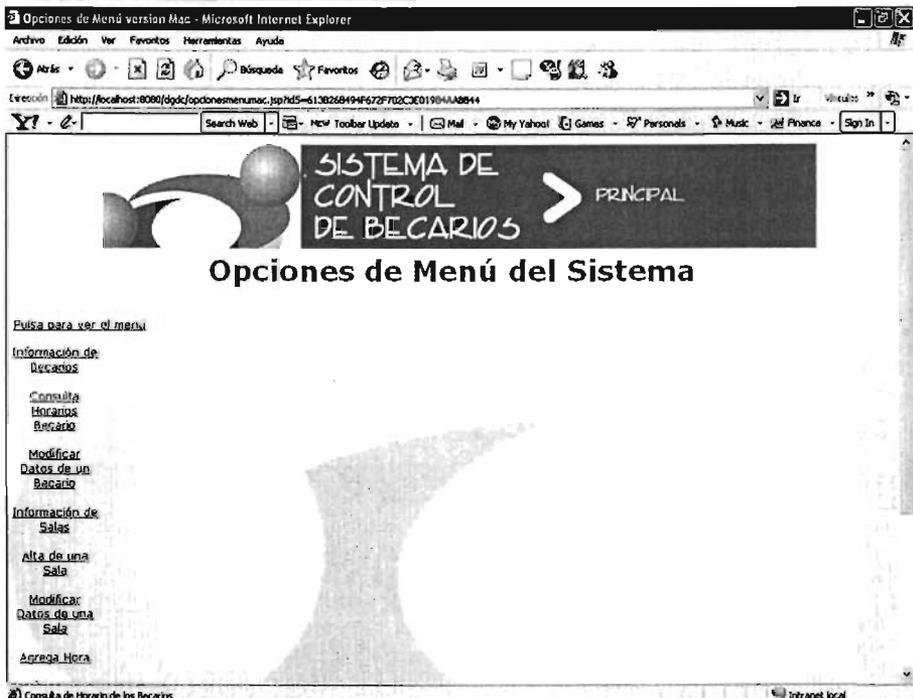
## Nota importante

La opción **Eliminar Becario** Eliminar becario no es de uso común, es decir, no la utilices a menos que se este seguro de que ese becario se quiere eliminar. Si queremos Borrar un Becario definitivamente del sistema entonces podemos utilizar esta opción.

## III. b) Consulta Horario

Regresamos a las Opciones del Menú Mac.

En esta parte del menú de opciones tenemos [Consulta Horarios](#) [Becario](#) el cual nos da una breve información de lo que contiene en la Barra de Estado. [Consulta de Horario de los Becarios](#)



Aquí podemos consultar el horario para cada Becario que se encuentra en el Sistema vamos a ver un ejemplo de cómo se puede ver un

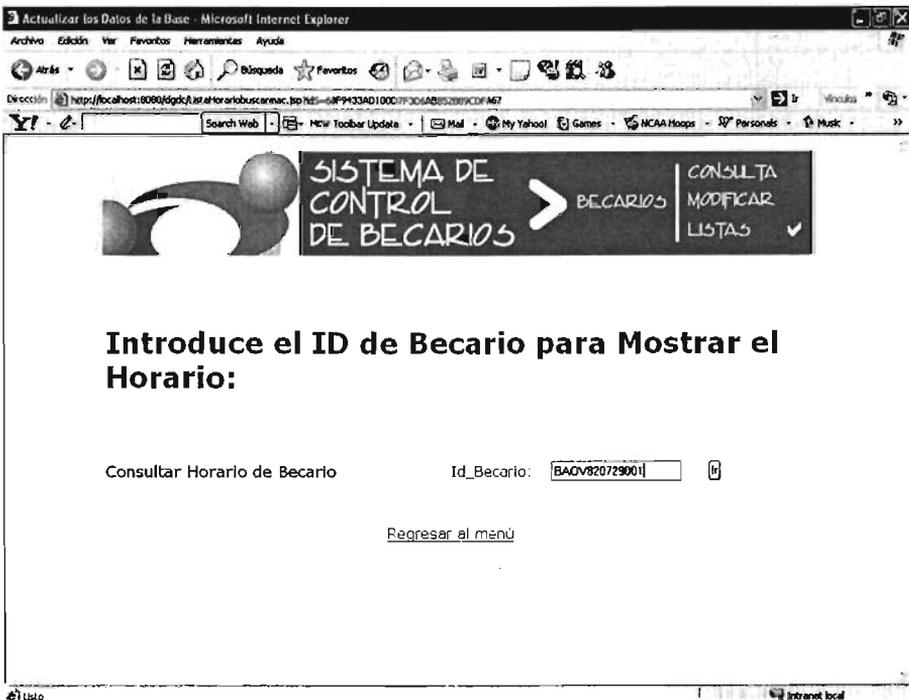
horario. Lo que tenemos que hacer para Mostrar el Horario del Becario es poner en la caja de Texto su Id de Becario como ejemplo tenemos lo siguiente:

Supongamos que deseamos ver el Horario del Becario que tiene el siguiente Id: BAOV820729001 este id lo escribiremos en la caja de texto como se ilustra a continuación.

Consultar Horario de Becario                      Id\_Becario:  

Para realizar la consulta del horario solo tenemos que dar clic en el botón Ir 

Por otra parte si lo que queremos es regresar de nuevo al menú de opciones solo tenemos que dar clic en la referencia [Regresar al menú](#) .



Al dar clic en el botón ir  tendremos el siguiente resultado:

Sistema de Control de Becarios - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

http://localhost:8080/dyc/InormeHorario.jsp?ID=DC85659C389A18825775614F688FAA6D



**SISTEMA DE CONTROL DE BECARIOS**

BECAARIOS | CONSULTA MODIFICAR LISTAS ✓

### Horario Completo Becario: Verónica

Id\_Becario:BAOV820729001 Nombre:Verónica

Hora	Lunes	Martes	Miercoles	Jueves	Viernes	Sabado	Domingo
9 a 10	✓		✓				
10 a 11	✓	✓	✓	✓			
11 a 12	✓	✓	✓	✓			
12 a 13	✓	✓	✓	✓			
13 a 14							
14 a 15					✓	✓	

Sistema de Control de Becarios - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

http://localhost:8080/dyc/InormeHorario.jsp?ID=DC85659C389A18825775614F688FAA6D

12 a 13	✓	✓	✓	✓			
13 a 14							
14 a 15					✓	✓	
15 a 16					✓	✓	
16 a 17					✓	✓	
17 a 18					✓	✓	

Imprime el Horario pulsando en la imagen



[Regresar](#)

De forma dinámica se genera un Horario del Becario seleccionado donde se muestra su Nombre, el Id de Becario, los días y horario que cubre dicho Becario. Si se requiere imprimir esta información solo damos clic en la figura de la impresora.

Imprime el Horario pulsando en la imagen



En Mac nos saldrá un mensaje como este:

Esta función no se puede realizar en una Macintosh.

Utiliza la siguiente combinación de teclas "Manzanita-P"

Aceptamos el Mensaje y para imprimir tecleamos Manzanita + P así saldrá nuestra pantalla de impresión

Al hacerlo se abrirá el siguiente mensaje de opciones de impresión seleccione la que mas se ajuste a sus preferencias y de clic en el botón imprimir .

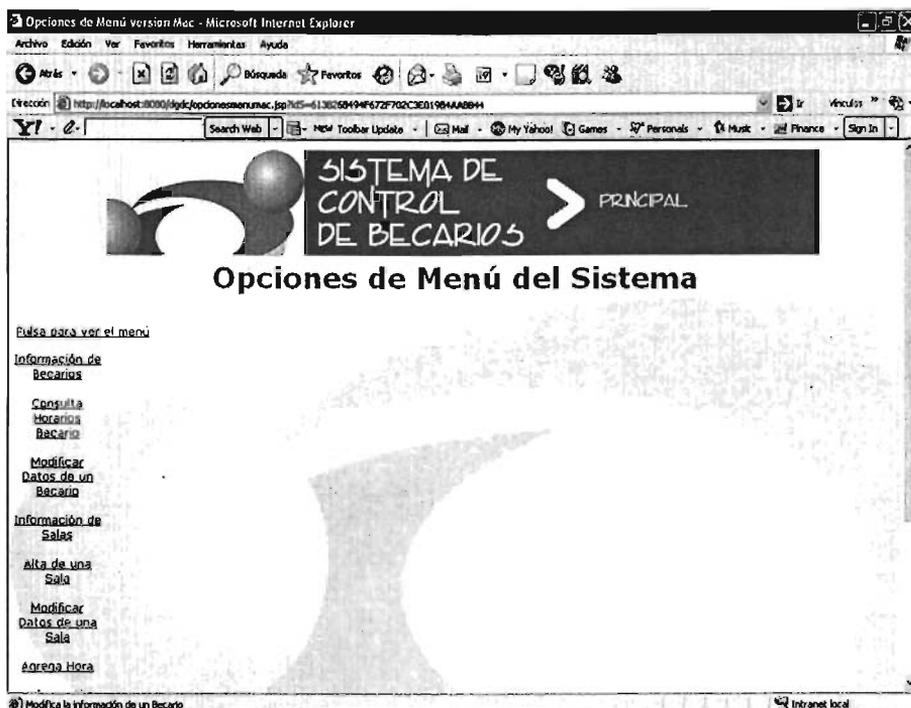
Para regresar a buscar el horario de otro Becario solo basta con dar clic en la referencia regresar que se encuentra en la parte inferior de esta página. [Regresar](#)

### III. c) Modificar Datos de un Becario

[Modificar  
Datos de un  
Becario](#)

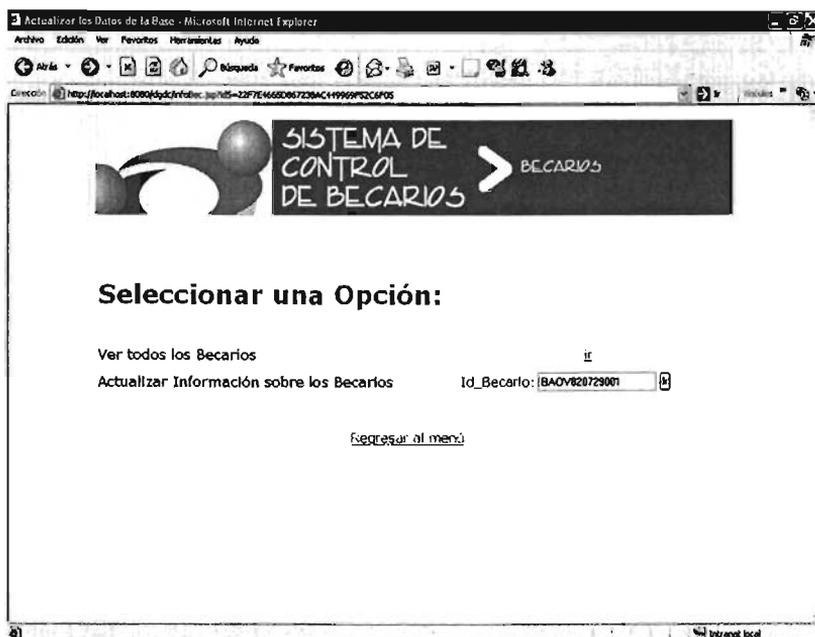
Al seleccionar del menú de opciones [Modificar Datos de un Becario](#) lo que podremos hacer es modificar toda la información personal del Becario tal como lo dice la pequeña ayuda que tenemos en la barra de estado.

 [Modifica la Información de un Becario](#)



Modificar  
Datos de un  
Becario

Damos clic en la referencia Modificar Datos de un Becario y tenemos lo siguiente:



Podemos seleccionar una opción:

En la opción Ver todos los Becarios nos mostrara lo que se vio en el Inciso a), es decir, nos llevara a ver a todos los Becarios contenidos dentro del Sistema solo con dar clic en la referencia ir como se muestra

Ver todos los Becarios

ir

En la opción actualizar Información sobre los Becarios pondremos en la caja de texto el id del Becario:

Supongamos que queremos modificar los datos personales del Becario que tiene el id siguiente BAOV820729001. Como se ilustra a continuación:

Actualizar Información sobre los Becarios

Id\_Becario:  ir

Después de insertar el id de Becario damos clic en el botón  el cual nos mostrara lo siguiente:

Sistema de Control de Becarios - Modificando la Información del Becario: BAOV820729001 - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Abre Busqueda Favoritos

http://localhost:8080/sgc/frmModifBec.jsp?ES=2277E46650867238AC449969F52C6F05

**SISTEMA DE CONTROL DE BECARIOS** > BECARIOS | CONSULTA MODIFICAR LISTAS

### Modificando la Información del Becario: BAOV820729001

Id_Becario:	BAOV820729001
Nombre:	Verónica
Apellido Paterno:	Barranco
Apellido Materno:	Osanya
Fecha de Nacimiento:	Día/Mes/Año: 29 / 7 / 82
Edad:	22
Edo. Civil:	soltero
Dirección:	Veracruzanos 26
Colonia:	Bonanza
Delegación:	Alvaro Obregón
C.P.:	1220
Teléfono:	52-73-72-82
Correo Electrónico:	verosanya@hotmail.com

Lista Intranet local

---

Sistema de Control de Becarios - Modificando la Información del Becario: BAOV820729001 - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Abre Busqueda Favoritos

http://localhost:8080/sgc/frmModifBec.jsp?ES=2277E46650867238AC449969F52C6F05

Teléfono:	52-73-72-82
Correo Electrónico:	verosanya@hotmail.com
Universidad:	UNAM
Facultad:	Pedagogía
Carrera:	Pedagogía
Número de Cuenta:	980635-6
Creditos:	100.00 Promedio: 8.50
Otros Estudios:	otros estudios
Idioma 1	Inglés Habla: 70 (%) Lee: 40 (%) Escribe: 40 (%)
Idioma 2	Inglés Habla: 40 (%) Lee: 40 (%) Escribe: 40 (%)
Otras Actividades:	NI
Sala o Área de preferencia:	Una Balsa en el Tiempo
Fecha de Inicio (Día/Mes/Año):	1/10/04
Fecha de Terminó (Día/Mes/Año):	1/09/04

Modificar la información del Becario BAOV820729001

[Regresar](#)

Lista Intranet local

Aquí podemos modificar cualquier campo excepto el ID de Becario, una vez terminando de modificar lo que deseamos solo tenemos que dar clic en el botón

Modificar la Información del Becario BAOV820729001

Si lo que deseamos es regresar a la pantalla anterior solo damos clic en Regresar

Una vez que hemos realizado nuestras modificaciones y dado clic en el botón 

Modificar la Información del Becario BAOV820729001

 .Aparecerá un formulario como el siguiente indicando si la información se actualizo correctamente:

Id_Becario:	BAOV820729001
Nombre:	Verónica
Apellido Paterno:	Barranco
Apellido Materno:	Ornaya
Fecha de Nacimiento:	29/7/82
Edad:	22
Edo. Civil:	soltero
Dirección:	Veracruzanos 26
Colonia:	Bonanza
Delegación:	Alvaro Obregón
C.P.:	1220
Teléfono:	52-73-72-82
Correo Electronico:	veroonmaya@hotmail.com
Universidad:	UNAM
Facultad:	Pedagogía

Sistema de Control de Becarios >> Modificando la Información del Becario BAOV820729001 - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Inicio | Buscar | Favoritos | Historial | Imprimir | ...

Dirección: http://localhost:8080/jsp/ModificB.jsp?ID=22F7E4665D0657230AC449909F52C0F05

Universidad:	UNAM
Facultad:	Pedagogía
Carrera:	Pedagogía
Número de Cuenta:	9808695-5
Creditos:	100.00 Promedio: 9.60
Otros Estudios:	otros edos
Idioma 1:	Inglés Habla:70(%) Lee: 40(%) Escribe: 40(%)
Idioma 2:	Inglés Habla:40(%) Lee: 40(%) Escribe: 40(%)
Otras Actividades:	AT
Sala o Área de Preferencia:	Una Balsa en el Tiempo
Fecha de Inicio (Dia/Mes/Año):	1/1/04
Fecha de Termin (Dia/Mes/Año):	1/09/4



**SISTEMA DE CONTROL DE BECARIOS** > | GUARDADO |

**Se ha modificado la Información del Becario: BAOV820729001**

[Volver a la Página Inicial](#)

Lista Intranet local

Lo que indica que se actualizo la información de forma correcta, de lo contrario mostrara un mensaje como este:



**SISTEMA DE CONTROL DE BECARIOS** > | ERROR |

### No se pudo modificar la Información:

Este mensaje puede deberse a un error en la Base de Datos, es decir, que la base de datos no se encuentre disponible o este mal configurada, si este es su caso favor de ponerse en contacto con el administrador de servidores.

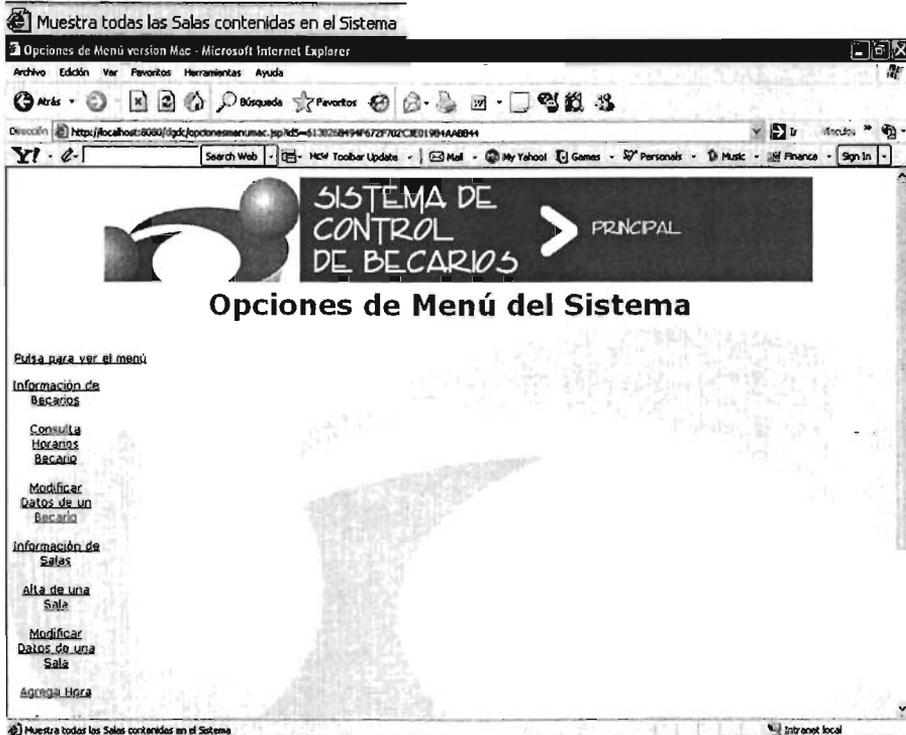
Para regresar a modificar la información de otro Becario damos clic en: [Volver a la Página Inicial](#)

### III. d) Información de Salas

#### Información de

#### Salas

En esta opción del menú se muestran todas las salas contenidas dentro del sistema como lo dice la barra de estado del Explorer.



Damos clic en esta opción y se mostrara lo siguiente:

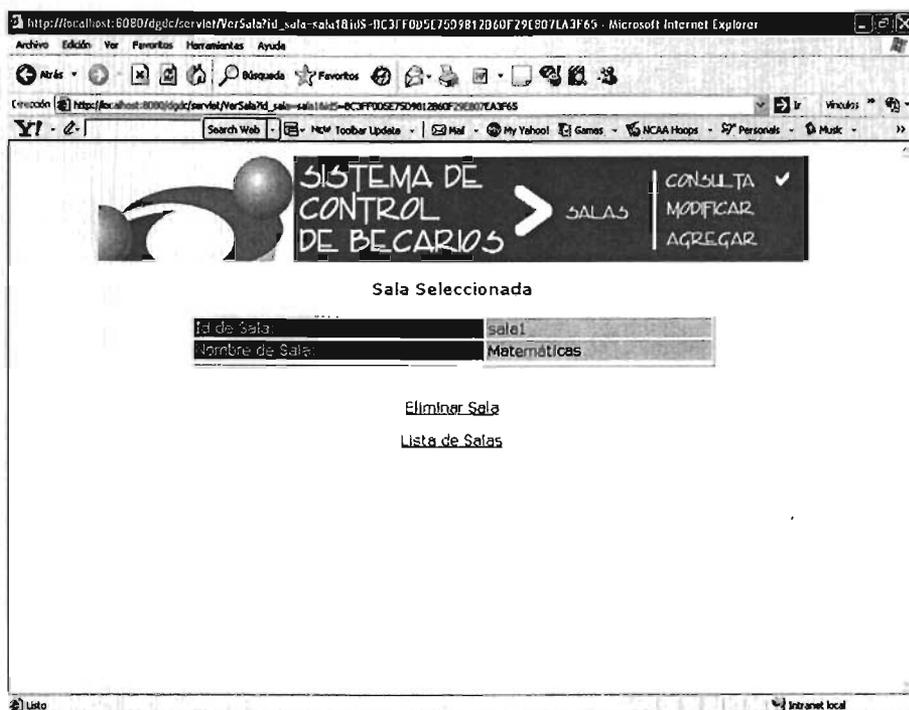


La opción Agregar Sala Agregar Sala se explicará mas adelante en el inciso f) si queremos regresar a al menú de opciones solo damos clic en Regresar Regresar

Podemos observar todas las salas del sistema; si queremos saber su contenido solo hacemos clic en el Nombre de la sala:

id_sala	Nombre de Sala
sala1	Matemáticas

Al hacer clic se muestra la siguiente pantalla:

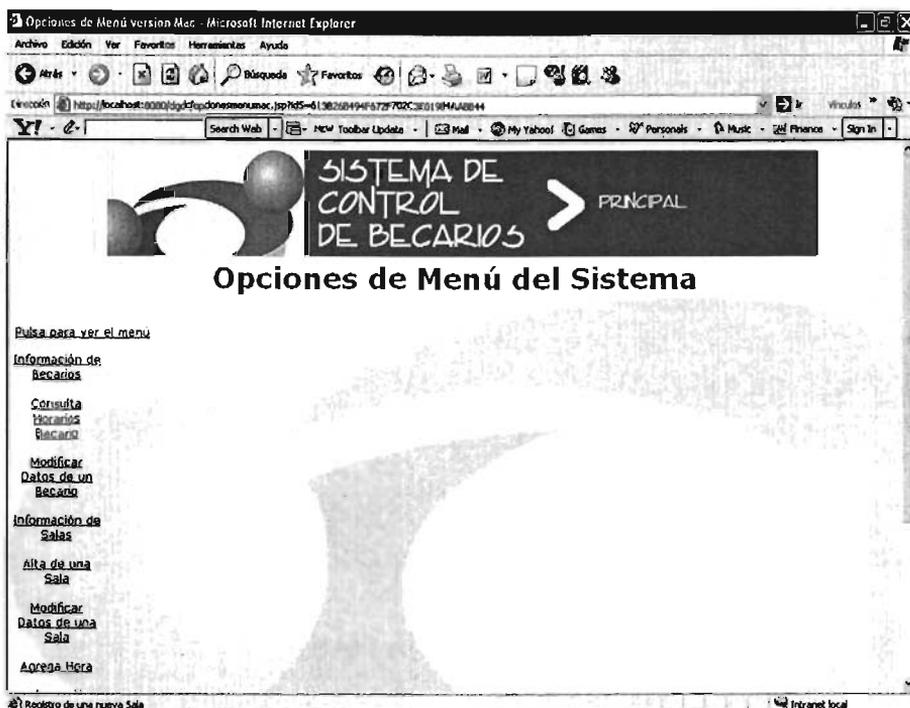


Si queremos eliminar la sala solo tenemos que dar clic en Eliminar Sala, esta opción es recomendable solo si se esta seguro de querer eliminar la sala definitivamente del sistema.

Para regresar a ver las Salas contenidas en el Sistema damos clic en Lista de Salas

### III. e) Alta de una Sala

En esta parte del sistema podemos dar de alta una sala si se requiere, cabe mencionar que el ID de Sala se genera automáticamente por lo que se necesita cambiar solo hay que agregar el Nombre de la Sala.



Supongamos que queremos agregar una Sala que llamaremos Sala X como ejemplo se ilustra lo siguiente:

<b>Agregar una Sala al Sistema</b>	
Id de Sala:	<input type="text" value="sala14"/>
Nombre de la Sala:	<input type="text" value="Sala X"/>
<input type="button" value="Guardar"/>	

Damos clic en el botón guardar  y el registro de la sala será agregado al sistema. Si se realizo de forma correcta tendremos un mensaje como este:



[Lista de Salas](#)

Damos clic en Lista de Salas para ver que se agrego efectivamente nuestra Sala X con el ID sala14

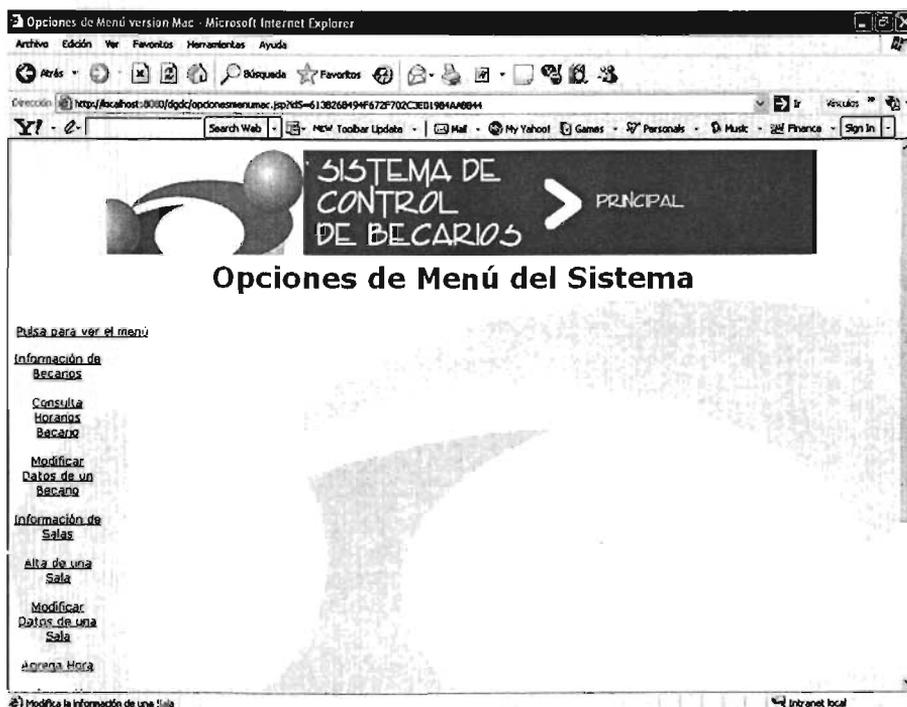


Lista de Salas

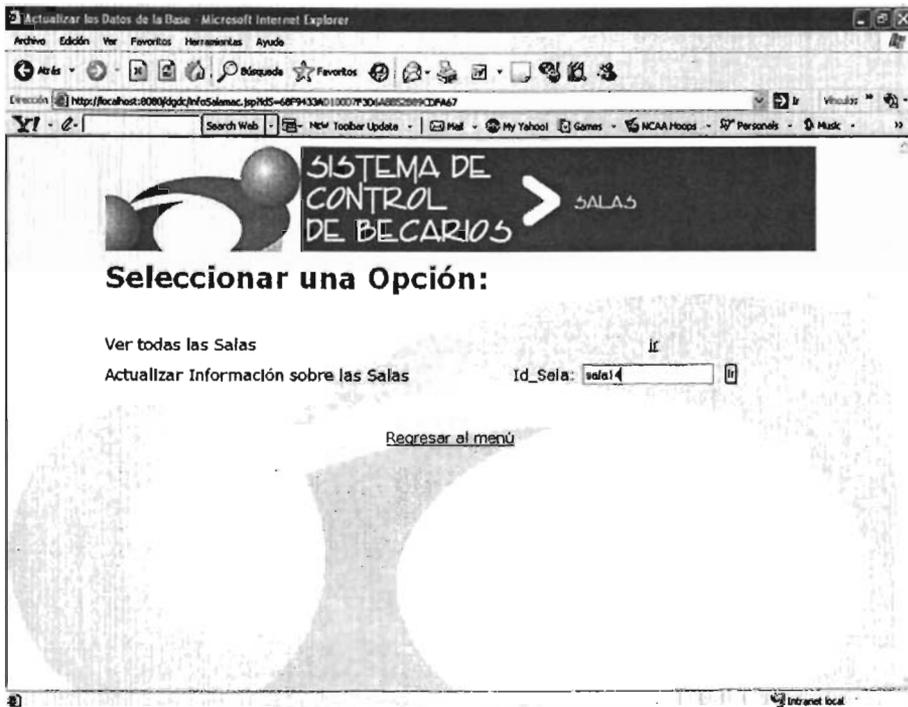
id_sala	Nombre de Sala
sala1	Matemáticas
sala10	Infraestructura de Nuestra Nación
sala11	Química
sala12	Tecnología Satelital
sala13	Una Balse en el Tiempo
sala14	Sala X
sala2	Biodiversidad
sala3	Universo
sala4	Agricultura y Alimentación

Así es como agregamos una Sala al Sistema.

### III. f) Modificar Datos de una Sala



Para modificar el nombre de una Sala solo tenemos que seleccionar la Modificar Datos de una Sala opción del menú, aparecerá la siguiente pantalla en donde solo tenemos que dar el id de la Sala:



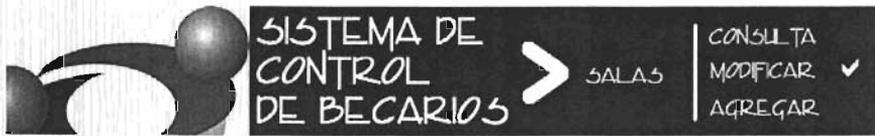
Si deseamos ver todas las salas contenidas en el sistema solo hay que dar un clic en la referencia ir como se muestra:

Ver todas las Salas [Ir](#)

Como ejemplo vamos a cambiar el nombre de la sala que tiene el id: sala14 su nombre es Sala X, por Sala X32, por lo que tecleamos en la caja de texto el id de la sala como se muestra:

Actualizar Información sobre las Salas Id\_Sala:  [Ir](#)

Y damos clic en el botón [Ir](#) se mostrará lo siguiente:



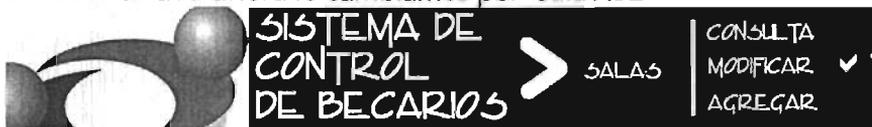
## Modificando la Información de la Sala: sala14

Id_Sala:	sala14
Nombre de la Sala:	Sala X

Modificar la información de la Sala: sala14

[Regresar](#)

Como se observa en el Recuadro marcado en rojo tenemos el nombre de la sala "Sala X" ahora lo cambiamos por "Sala X32"



## Modificando la Información de la Sala: sala14

Id_Sala:	sala14
Nombre de la Sala:	Sala X32

Modificar la información de la Sala: sala14

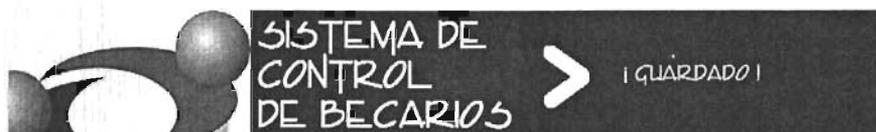
[Regresar](#)

Damos clic en el botón  para que la información se actualice.

Si la información se actualizo de forma correcta tendremos el siguiente mensaje en pantalla;

## Modificando la Información de la Sala: sala14

Id_Sala:	sala14
Nombre de la Sala:	Sala X32



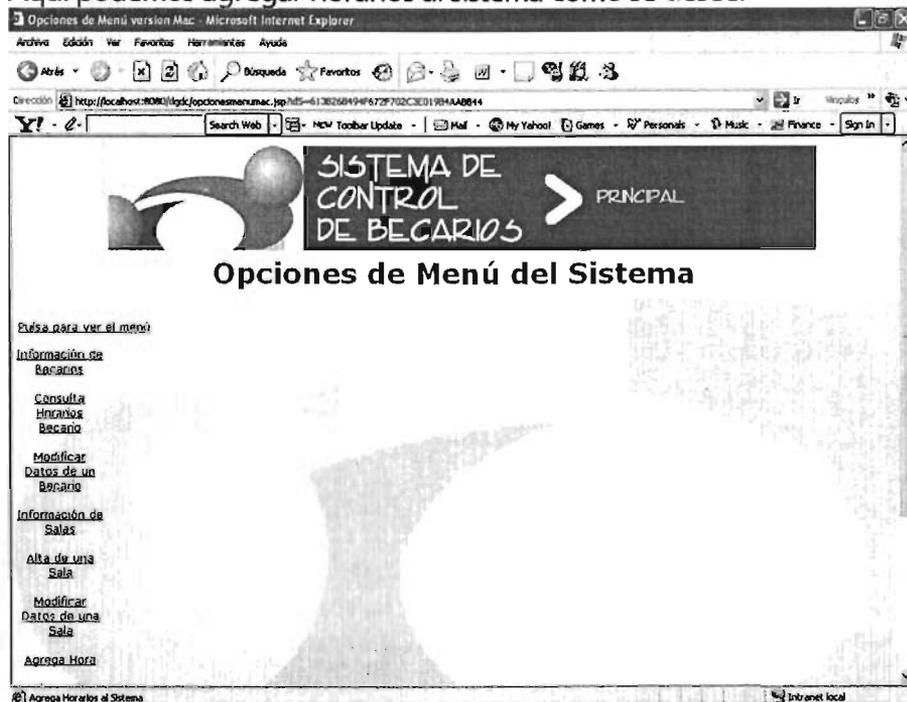
Se ha modificado la Información de la Sala: sala14

[Volver a la Página Inicial](#)

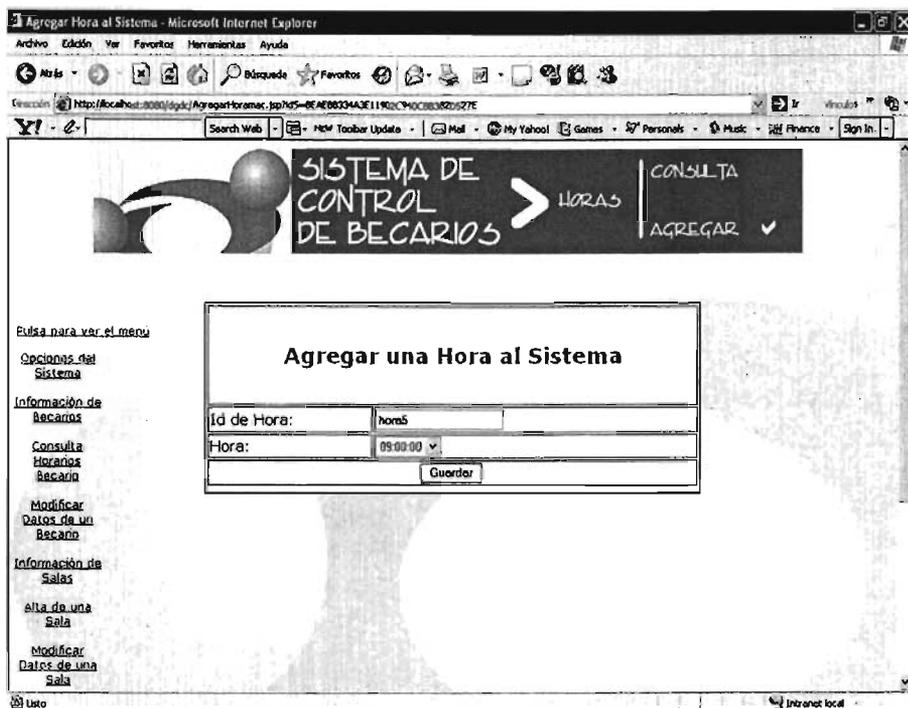
Para regresar a seleccionar otro id de sala para modificar damos clic en la referencia [Volver a la Página Inicial](#)

### III. g) Agrega Horarios

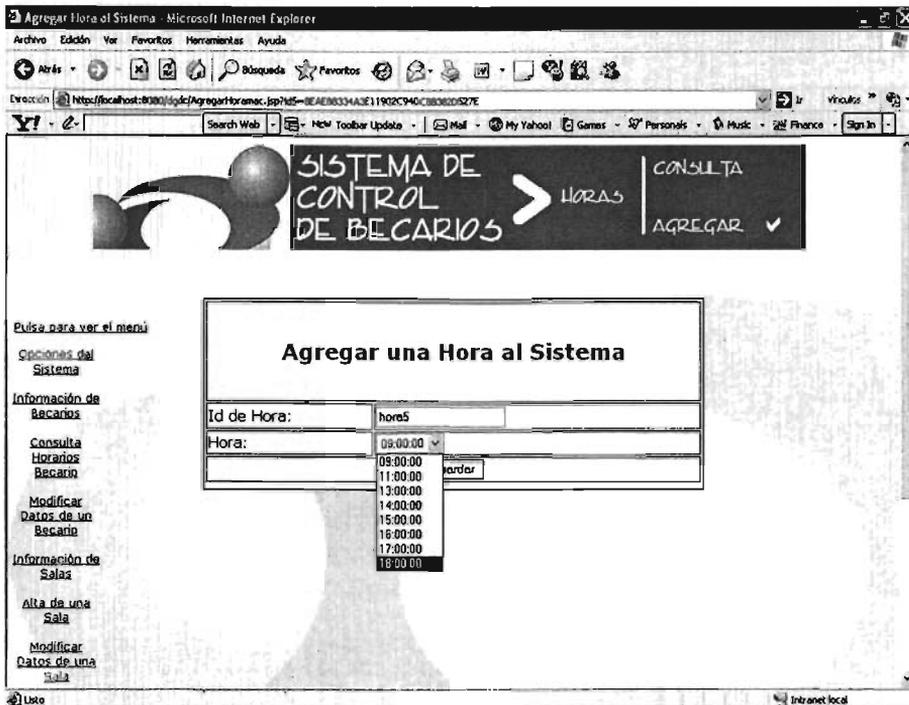
Aquí podemos agregar horarios al sistema como se desee.



Damos clic en la opción [Agrega Hora](#) la siguiente pantalla será:



Supongamos que necesitamos agregar un horario, que sea a las 18:00 hrs., el id de hora se genera automáticamente por lo que no hay que cambiarlo, entonces seleccionaremos de la lista de Hora: las 18:00 hrs. como se ilustra a continuación:



Damos clic en el botón guardar  para agregar el horario al sistema.

Si se ha guardado correctamente tendremos un mensaje como este:



Para regresar a ver que efectivamente se agrego nuestro horario damos clic en la referencia [Lista de Horas](#) [Lista de Horas](#)



Lista de Horas

id_llegada	Hora de Llegada
hora1	09:00:00
hora2	10:00:00
hora3	14:00:00
hora4	15:00:00
hora5	18:00:00

[Agregar Hora](#)

[Regresar](#)

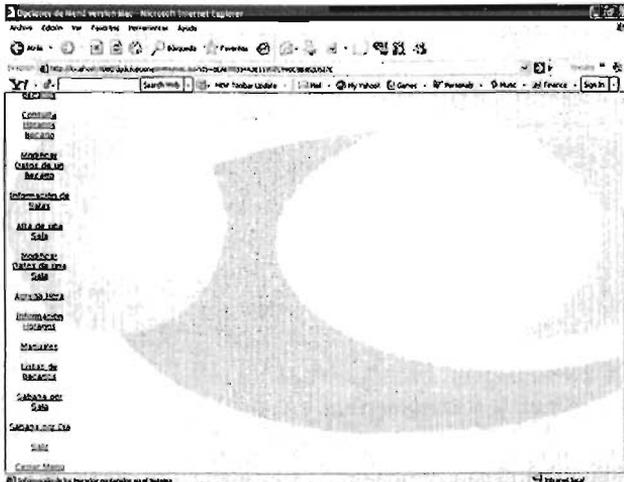
Podemos observar que se agrego con éxito nuestro horario id\_llegada hora5 18:00 hrs.

Para regresar al menú de opciones damos clic en la referencia [Regresar](#)

### III. h) Información Horarios

Se muestran los horarios contenidos dentro del sistema en esta opción:

[Información Horarios](#)



Al dar clic en [Información](#) [Horarios](#) se muestra la siguiente pantalla:

Lista de Horas

id_llegada	Hora de Llegada
hora1	09:00:00
hora2	10:00:00
hora3	14:00:00
hora4	15:00:00
hora5	18:00:00

[Agregar Hora](#)  
[Regresar](#)

En donde queremos podemos ver la Lista de Horas por `id_llegada` y por la Hora de Llegada.

id_llegada	Hora de Llegada
hora1	09:00:00
hora2	10:00:00

Al dar clic sobre la columna de Hora de Llegada tendremos la información de esa hora seleccionada como se deja ver a continuación.



#### Hora Seleccionada

Id de Llegada	hora5
Hora de Llegada	18:00:00

[Eliminar Hora](#)

[Lista de Horas](#)

(Si queremos regresar a ver la lista damos clic sobre la referencia [Lista de Horas](#) ).

Supongamos como ejemplo que estamos seguros de borrar la hora5 que fue la que se creo en el inciso anterior h) damos clic sobre la referencia [Eliminar Hora](#) . Si el registro se elimino con éxito tendremos el siguiente mensaje en la pantalla.

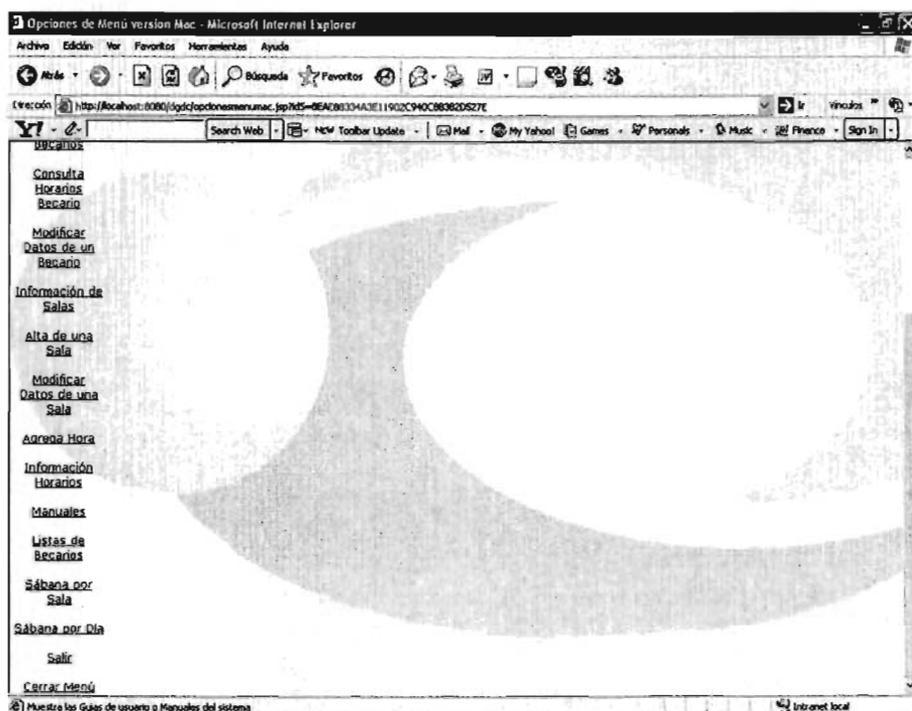
El registro ha sido eliminado

[Lista de Horas](#)

Y la referencia para regresar a la lista.

### III. i) Manuales

En esta parte de las opciones tenemos todos los manuales que se necesitan para manejar correctamente los sistemas entre ellos se encuentra este manual.



Hacemos clic en la liga [Manuales](#) y la pantalla que aparece es la siguiente:

The screenshot shows a Microsoft Internet Explorer window with the title 'Manuales del Sistema - Microsoft Internet Explorer'. The address bar contains a local URL. The page header features a logo and the text 'SISTEMA DE CONTROL DE BECARIOS' with a right-pointing arrow and the word 'MANUALES'. Below this, the main heading reads 'Manuales de Usuario del Sistema de Control de Becarios DGDC'. A sub-heading asks the user to 'Seleccione uno de los Manuales:'. A table lists two manual titles with links for HTML and PDF formats. At the bottom, there is a link to 'Regresar al menú'.

Manuales	HTML	PDF
C2WCAM (credenciales)	<a href="#">C2WCAM</a>	<a href="#">Ir</a>
Migrar Base de Datos de Access a Mysql (De la base local al Servidor)	<a href="#">Mysql-Access</a>	<a href="#">Ir</a>

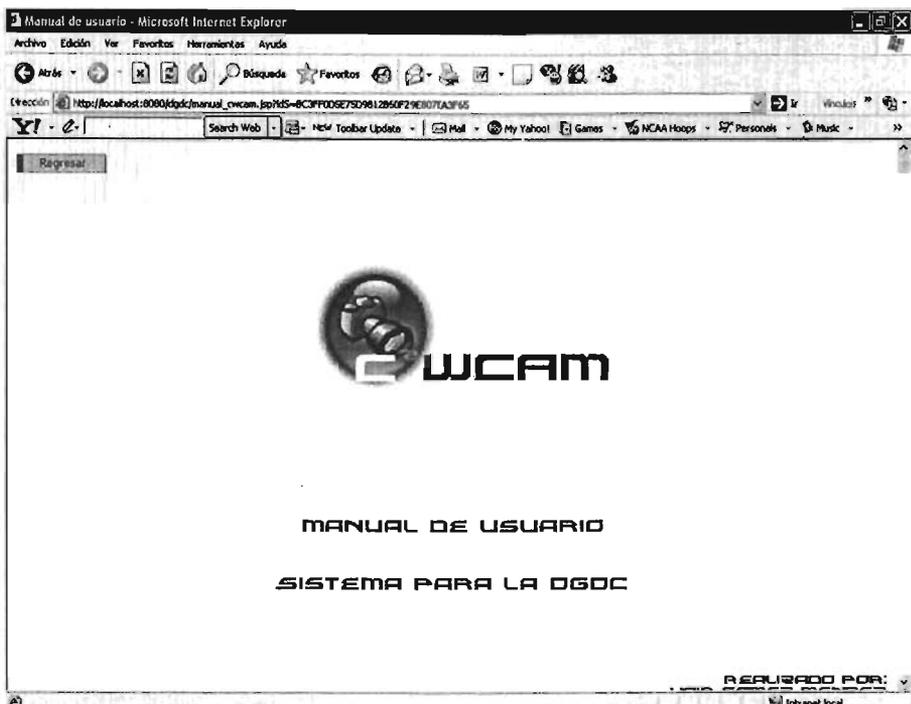
[Regresar al menú](#)

Para regresar al menú de opciones damos clic en [Regresar al menú](#). El formato de los manuales es en HTML y en PDF esto es para facilitar la visualización de los mismos. Si damos clic en las referencias aparecerán los manuales en sus respectivos formatos:

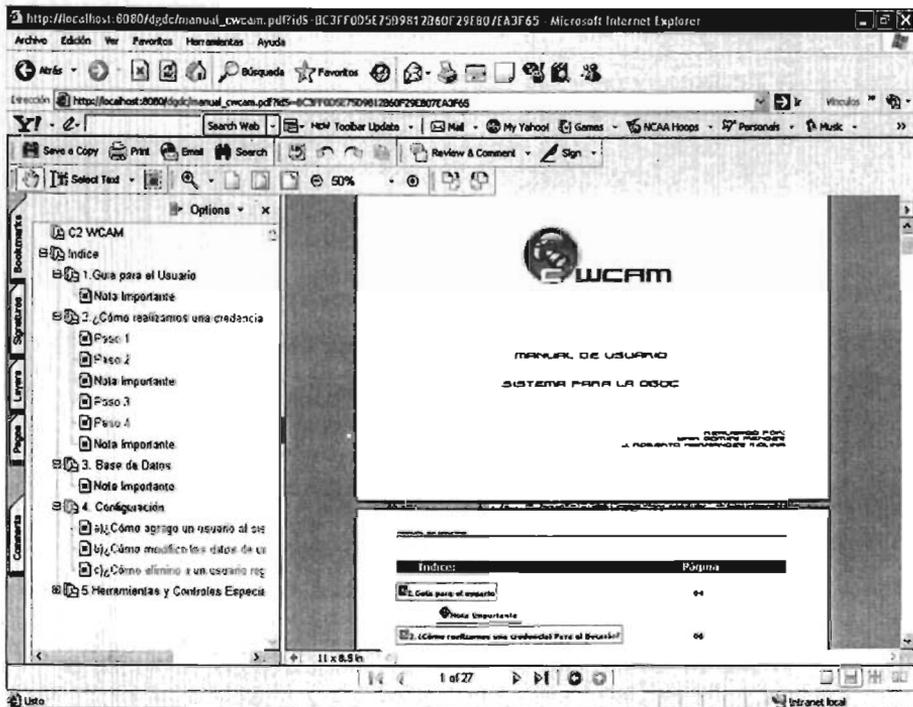
**Seleccione uno de los Manuales:**

Manuales	HTML	PDF
C2WCAM (credenciales)	<a href="#">C2WCAM</a>	<a href="#">Ir</a>
Migrar Base de Datos de Access a Mysql (De la base local al Servidor)	<a href="#">Mysql-Access</a>	<a href="#">Ir</a>

Supongamos que deseamos ver el manual C2WCAM en formato HTML solo damos clic en la referencia [C2WCAM](#) de la columna HTML, se mostrara lo siguiente:



Para regresar a los manuales solo damos clic en el botón **Regresar**. Si lo que deseamos es ver el manual en formato de Acrobat o .pdf damos clic en la referencia I de la columna PDF; se mostrara lo siguiente:

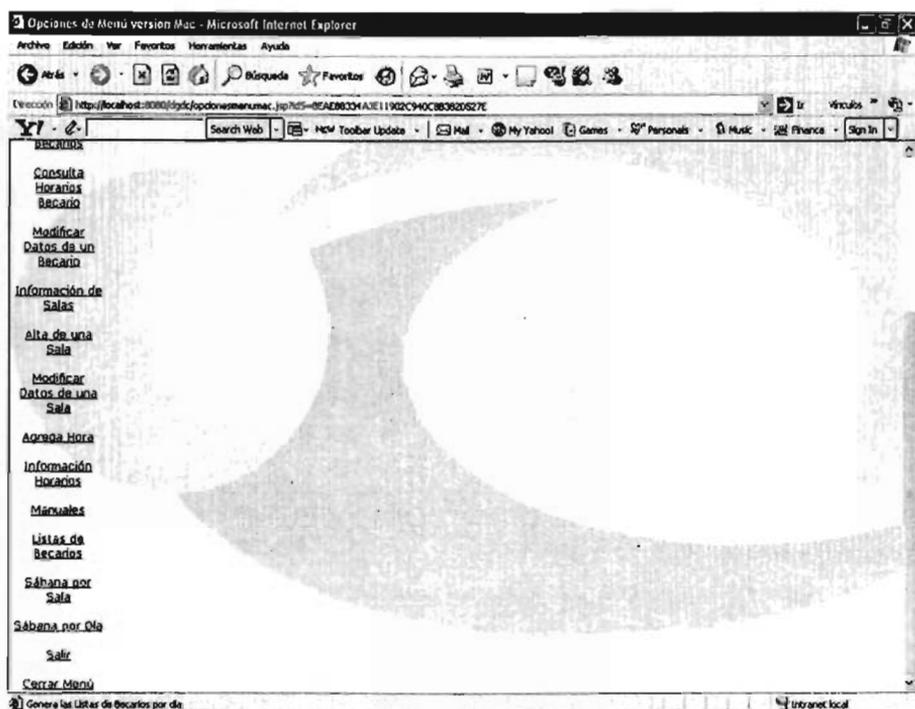


Para regresar a los manuales damos clic en **Atrás** en la barra del explorador.

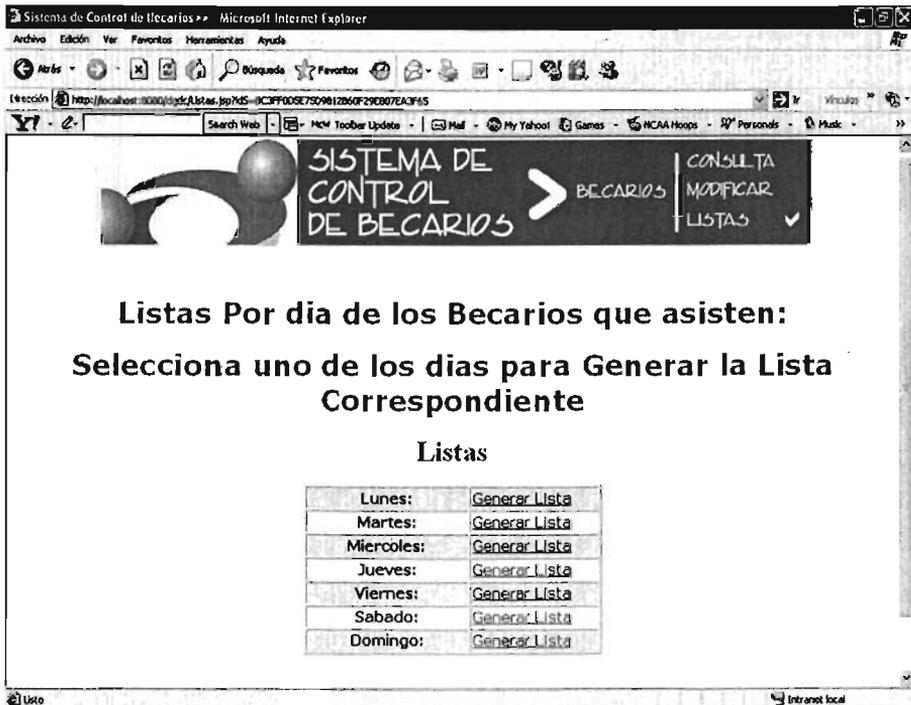
El mismo ejemplo se aplica para todos los manuales contenidos en el apartado Manuales de este sistema.

### III. j) Listas de Becarios

En listas de Becarios podremos acceder e imprimir la lista de asistencia de los becarios que asisten por día, es decir, una lista de Lunes a Domingo de los Becarios.



Hacemos clic en la referencia [Listas de Becarios](#) la pantalla siguiente muestra las opciones por día para visualizar las listas.



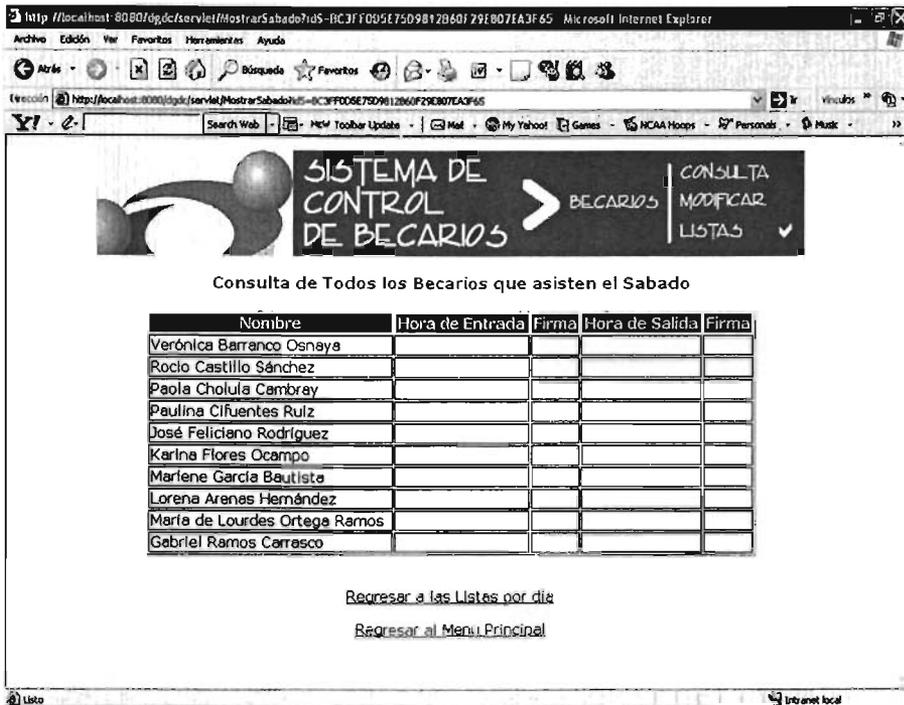
**SISTEMA DE CONTROL DE BECARIOS** > BECARIOS | CONSULTA MODIFICAR LISTAS ✓

**Listas Por día de los Becarios que asisten:  
Selecciona uno de los días para Generar la Lista Correspondiente**

**Listas**

Lunes:	<a href="#">Generar Lista</a>
Martes:	<a href="#">Generar Lista</a>
Miércoles:	<a href="#">Generar Lista</a>
Jueves:	<a href="#">Generar Lista</a>
Viernes:	<a href="#">Generar Lista</a>
Sábado:	<a href="#">Generar Lista</a>
Domingo:	<a href="#">Generar Lista</a>

Por ejemplo: nosotros queremos ver la lista de los sábados (que Becarios asisten el sábado), entonces solo damos clic sobre la referencia [Generar Lista](#)



Consulta de Todos los Becarios que asisten el Sabado

Nombre	Hora de Entrada	Firma	Hora de Salida	Firma
Verónica Barranco Osnaya				
Rocio Castillo Sánchez				
Paola Choiula Cambrey				
Paulina Cifuentes Ruiz				
José Feliciano Rodríguez				
Karina Flores Ocampo				
Marlene García Bautista				
Lorena Arenas Hernández				
María de Lourdes Ortega Ramos				
Gabriel Ramos Carrasco				

[Regresar a las Listas por día](#)

[Regresar al Menú Principal](#)

Para imprimir esta lista damos clic en el Icono (parte superior de la barra del explorador)  o Archivo y seleccionamos la opción Imprimir y la otra opción que tenemos en Mac es dar los comandos Manzanita + P.

Para regresar a ver otra lista damos clic en la referencia

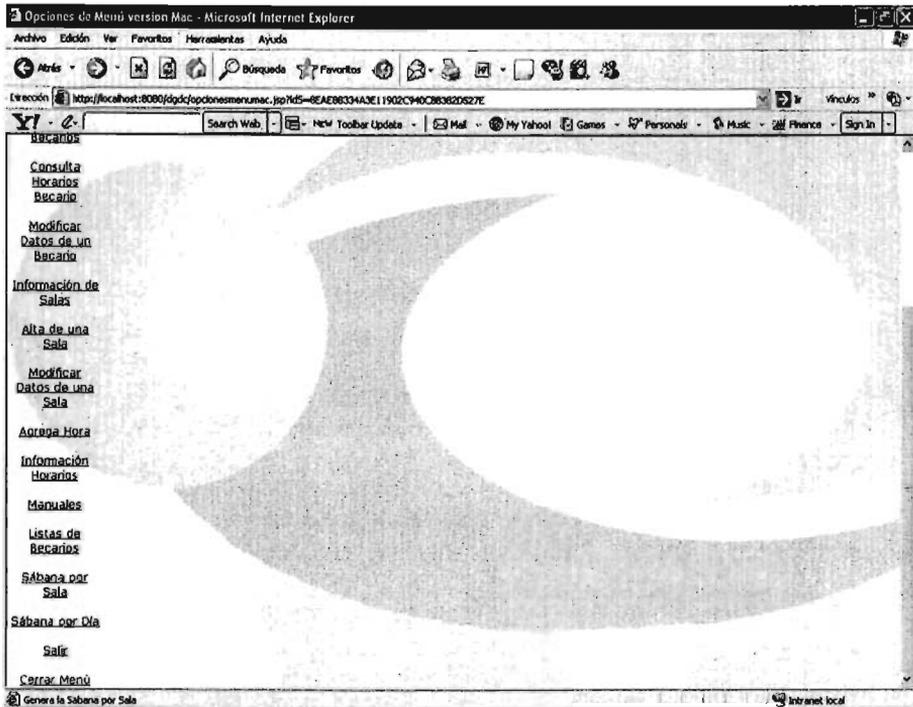
[Regresar a las Listas por día](#)

Para regresar al menú de opciones damos clic en la referencia

[Regresar al Menú Principal](#)

### III. k) Generar Sabana por Sala

Para generar la Sabana por sala solo tenemos que ir a la siguiente opción [Sábana por Sala](#) como se muestra en la pantalla siguiente:



Damos clic en esta referencia **Sábana de Sala** se muestra lo siguiente:



## Consulta de la Sábana por Sala

### Selecciona una Sala de la Lista

Sala o Área de preferencia:

[Regresar al Menú](#)

Para ver la Sabana de la sala que nosotros queremos solo hay que seleccionarla de la lista como se ilustra a continuación:

## Consulta de la Sábana por Sala

### Selecciona una Sala de la Lista

Sala o Área de preferencia: Biología Humana y Salud Generar Sábana

- Biología Humana y Salud
- Agricultura y Alimentación
- Universo
- Matemáticas
- Biodiversidad
- Conciencia de Nuestra Nación
- Energía
- Espacio Infantil
- Estructura de la Materia
- Infraestructura de Nuestra Nación**
- Química
- Tecnología Satelital
- Una Balsa en el Tiempo
- Sala X32

Supongamos que deseamos ver la Sabana de la Sala Infraestructura de Nuestra Nación, entonces seleccionamos esta sala y damos clic en el botón **Generar Sabana**. Se mostrara lo siguiente:

Sistema de Control de Becarios >> Generar la Sabana por Sala sala10 - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Abre Búsqueda Favoritos

Dirección http://localhost:8080/igdc/GenerarSabana.jsp?C=1167EC2691990C755257996AC0AD0BE1

Search Web NEW Toolbar Update Mail My Yahoo Games Personals Music Finance Sign In



**SISTEMA DE CONTROL DE BECARIOS** > SALAS

### Sábana de la Sala:Infraestructura de Nuestra Nación

Hora	Lunes	Martes	Miercoles	Jueves	Viernes	Sabado	Domingo
9 a 10	Fernando Christian Eugenio	Fernando Christian Eugenio	Fernando Christian Eugenio		Fernando Christian Eugenio		
10 a 11	Fernando Christian Eugenio	Fernando Christian Eugenio	Fernando Christian Eugenio		Fernando Christian Eugenio		
11 a 12	Fernando Christian Eugenio	Fernando Christian Eugenio	Fernando Christian Eugenio		Fernando Christian Eugenio		
12 a 13	Fernando Christian Eugenio	Fernando Christian Eugenio	Fernando Christian Eugenio		Fernando Christian Eugenio		
13 a 14							
14 a 15	Karla Ariana Mariana	Karla Ariana Mariana		Karla Ariana Mariana	Karla Ariana Mariana Karina	Karina Marlene	Karina Marlene

Lista Intranet local

Sistema de Control de Becarios -> Generar la Sabana por Sala sala10 - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Abres - - - - - Búsqueda Favoritos - - - - -

http://localhost:8080/tydc/GenerarSabana.jsp?IS=CF17369A9ECC23AA90A9E72BBA29

Search Web - - - - - Mail - - - - - My Yahoo! - - - - - Games - - - - - NCAA Hoops - - - - - Personal - - - - - Music - - - - -

14 a 15	Karla Ariana Marlana	Karla Ariana Marlana		Karla Ariana Marlana	Karla Ariana Marlana Karina Marlene	Karina Marlene	Karina Marlene
15 a 16	Karla Ariana Marlana	Karla Ariana Marlana		Karla Ariana Marlana	Karla Ariana Marlene Karina Marlene	Karina Marlene	Karina Marlene
16 a 17	Karla Ariana Marlana	Karla Ariana Marlana		Karla Ariana Marlana	Karla Ariana Marlana Karina Marlene	Karina Marlene	Karina Marlene
17 a 18	Karla Ariana Marlana	Karla Ariana Marlana		Karla Ariana Marlana	Karla Ariana Marlana Karina Marlene	Karina Marlene	Karina Marlene

Imprime la Sabana pulsando en la imagen

  
Regresar

Internet local

Aquí podemos ver la Sabana de la Sala seleccionada para imprimir la

Sabana damos clic en la imagen  se mostrará una nueva ventana indicando nuestras preferencias de impresión

En Mac nos saldrá un mensaje como este:

Esta función no se puede realizar en una Macintosh.

Utiliza la siguiente combinación de teclas "Manzanita-P"

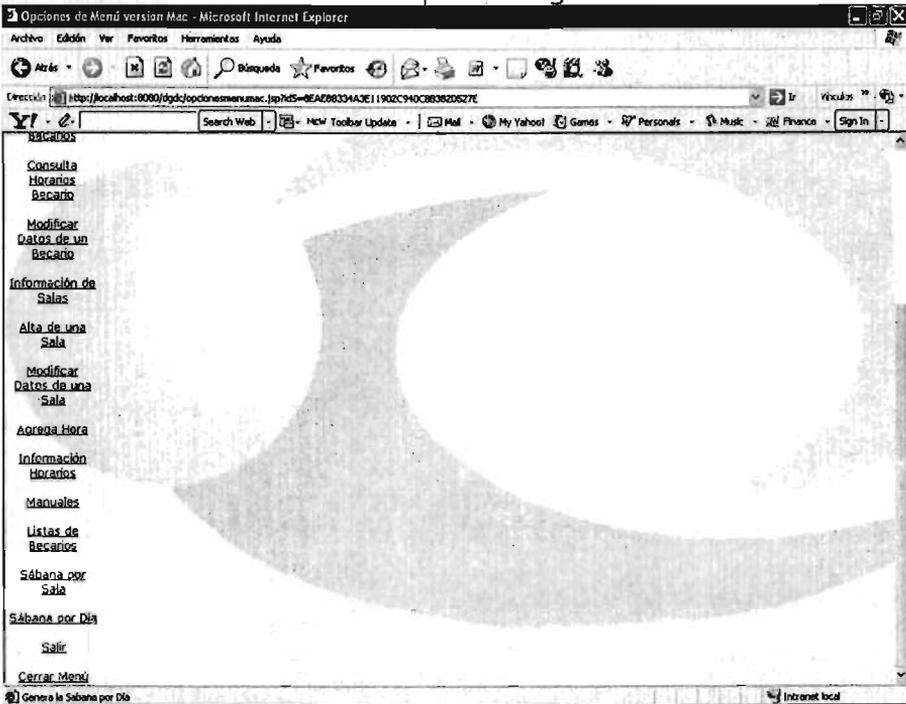
Aceptamos el Mensaje y para imprimir tecleamos Manzanita + P así saldrá nuestra pantalla de impresión

Al hacerlo se abrirá un mensaje de opciones de impresión seleccione la que mas se ajuste a sus preferencias y de clic en el botón imprimir 

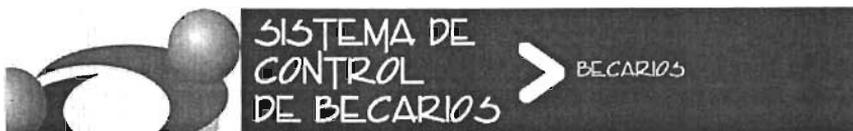
Para regresar a seleccionar otra sala damos clic en la referencia Regresar

### III. I) Generar Sábana por Día

Para generar la Sábana por Día damos clic en la opción Sábana por Día del menú como se muestra en la pantalla siguiente:



La pantalla que se abrirá será:



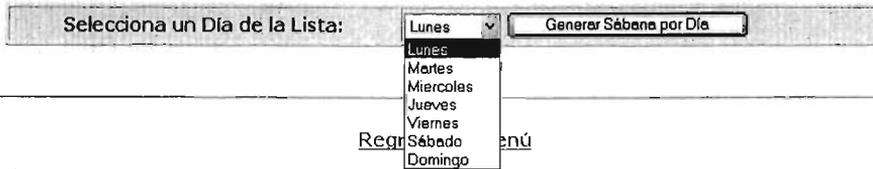
## Consulta de la Sábana por Día

Selecciona un Día de la Lista:

[Regresar al Menú](#)

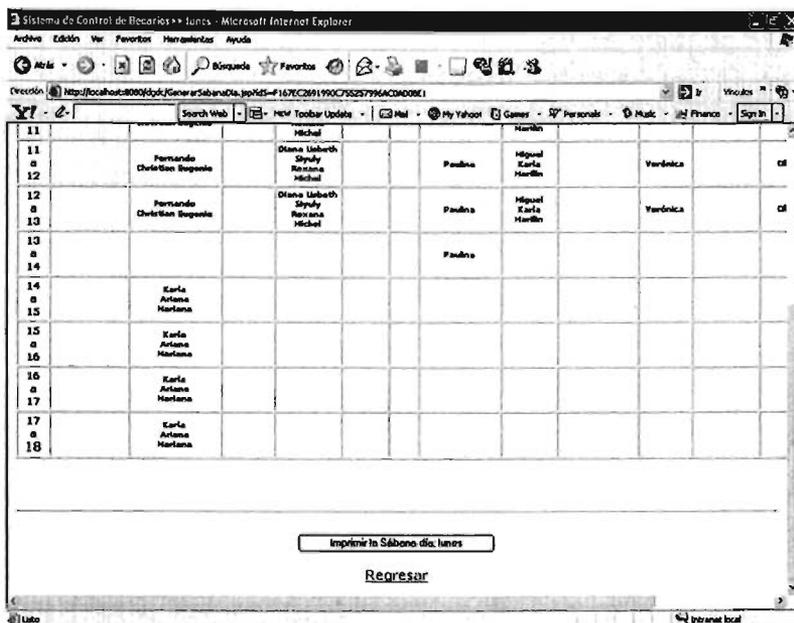
Supongamos que deseamos ver la Sabana del día lunes; seleccionamos el día de la lista y damos clic en el botón **Generar Sábana por Día**

## Consulta de la Sábana por Día



Lo que se muestra a continuación es la Sábana generada para el día lunes con la información de los nombres de los becarios y las salas correspondientes:

Hora	Matemáticas	Infraestructura de Nuestra Nación	Química	Tecnología Satelital	Una Balsa en el Tiempo	Bala X32	Biodiversidad	Universo	Agricultura y Alimentación	Biología Humana y Salud	Conciencia de Nuestra Nación	Ener
9 a 10		Fernando Christian Eugenio		Diana Liebeth Shydy Rencana Michal				Miguel Karla Karlin		Verónica		CI
10 a 11		Fernando Christian Eugenio		Diana Liebeth Shydy Rencana Michal			Paulina	Miguel Karla Karlin		Verónica		CI
11 a 12		Fernando Christian Eugenio		Diana Liebeth Shydy Rencana Michal			Paulina	Miguel Karla Karlin		Verónica		CI
12 a 13		Fernando Christian Eugenio		Diana Liebeth Shydy Rencana Michal			Paulina	Miguel Karla Karlin		Verónica		CI
13 a 14							Paulina					
14 a 15		Karla Ariana Mariana										

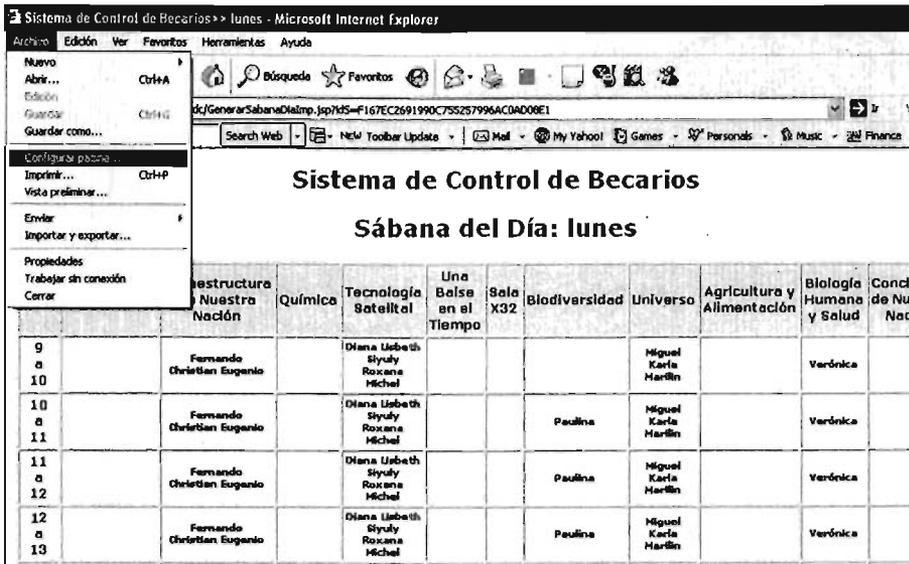


Para imprimir esta Sábana damos clic en el botón

si queremos regresar a seleccionar otro

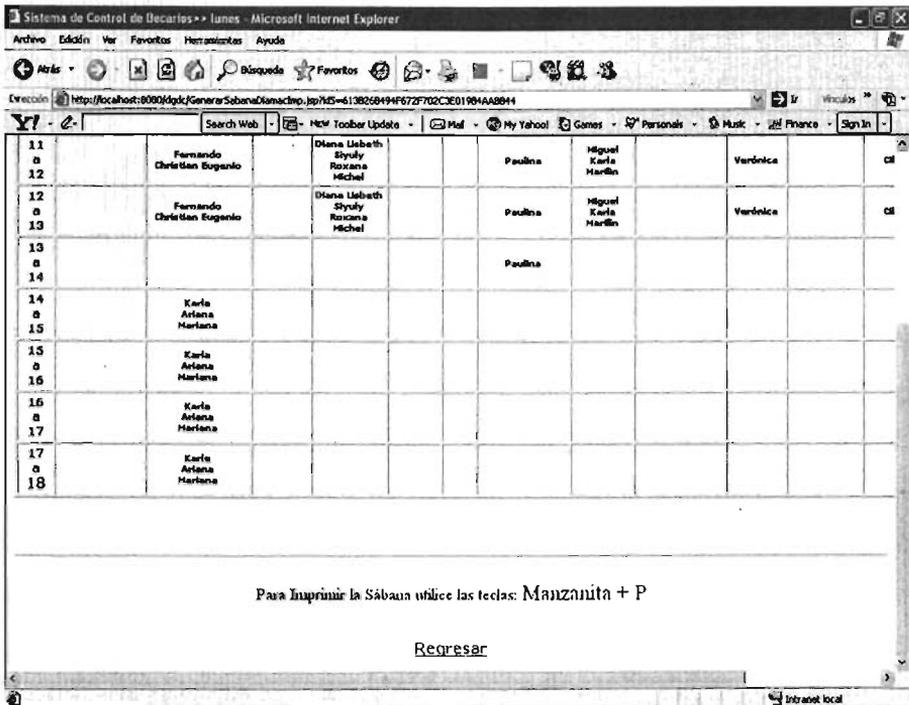
día para generar su Sábana damos clic sobre la referencia Regresar

Cabe señalar que esta sábana se imprime en una hora de 11" X 17" por tanto debemos configurar la página así como la impresora en la cual se va a imprimir nuestra sábana. En la barra de menús de nuestro explorador nos vamos al menú Archivo después a Configurar página como se muestra y damos clic.



Se abrirá una pantalla de selección donde escogemos el tamaño indicado y la hoja en Horizontal.

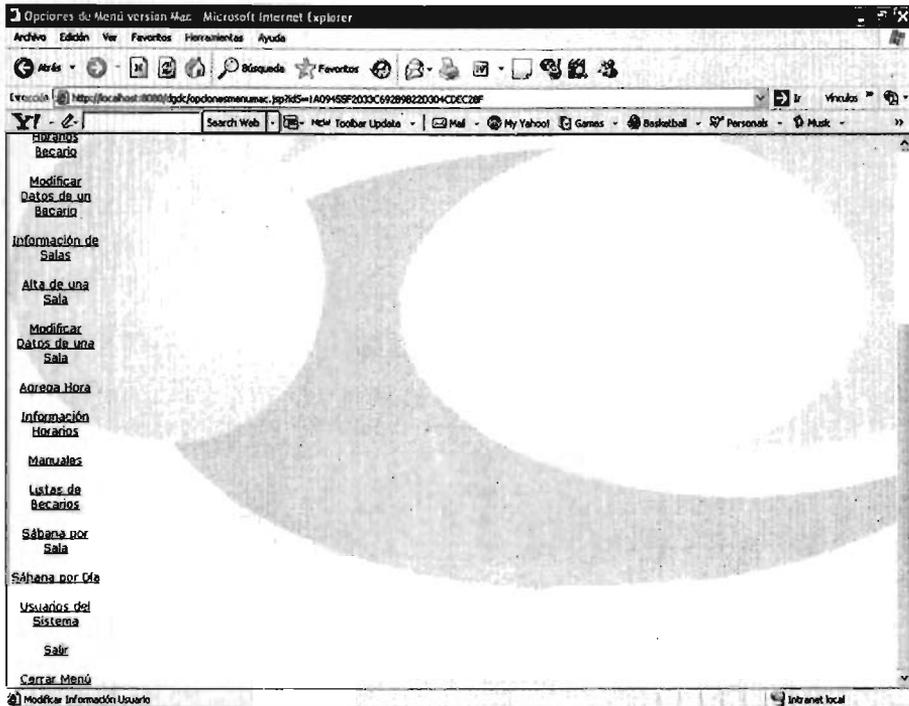
Para imprimir nuestra Sabana tenemos que dar Manzanita + P para imprimir. Como se ilustra a continuación: [Regresar](#)



Para regresar a seleccionar otro día para generar la Sábana damos clic sobre la referencia

**III. m) Usuarios del Sistema**

En este modulo el usuario del sistema podrá dar de alta a otro usuario de confianza o cambiar si así lo desea su login y su password, en la siguiente referencia Usuarios del Sistema tenemos esta opción:



Al dar clic en esta liga [Usuarios del Sistema](#) tenemos lo siguiente:



### Opciones Menú Usuario

Seleccióna una de las Opciones siguientes:

Modificar Cuenta de Usuario:	<input type="button" value="Modificar"/>
Crear un Nuevo Usuario:	<input type="button" value="Crear"/>

[Regresar el menú Principal](#)

Las opciones del Menú de Usuario: donde podemos seleccionar:  
B) Modificar la cuenta de Usuario y B) Crear un nuevo Usuario.  
Vamos a ver la primera opción:

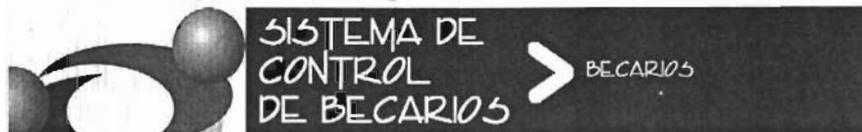
C) Modificar cuenta de usuario:

Modificar Cuenta de Usuario:

Modificar

Hacemos clic en el

botón modificar y tenemos lo siguiente:



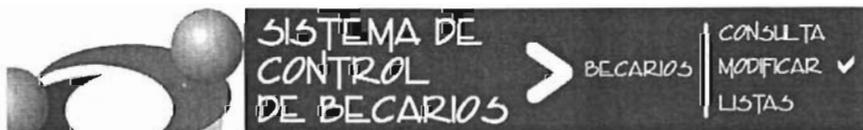
## Modificar Login y Password:

Por Favor Intoduzca su Login y Password Actuales

Login:	<input type="text" value="Salvador"/>	
Password:	<input type="password" value="*****"/>	<input type="button" value="Modificar"/>

[Regresar al menú Usuario](#)

Si el login o el password son incorrectos nos aparecerá un mensaje como el siguiente al momento de dar clic en el botón



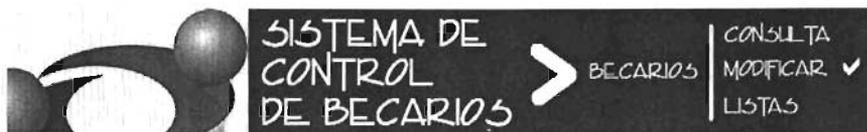
## Modificando la Información del Usuario

No se encontró el dato ingresado

[Regresar](#)

Damos clic en regresar para volver a identificarnos.

De lo contrario si introducimos el login y el password de forma correcta aparecerá una pantalla como la siguiente:



## Modificando la Información del Usuario

Ingresas tu Nuevo Login Y Password:

Login:	<input type="text" value="Salvador"/>
Password:	<input type="password" value="*****"/>
Confirmar Password:	<input type="password"/>

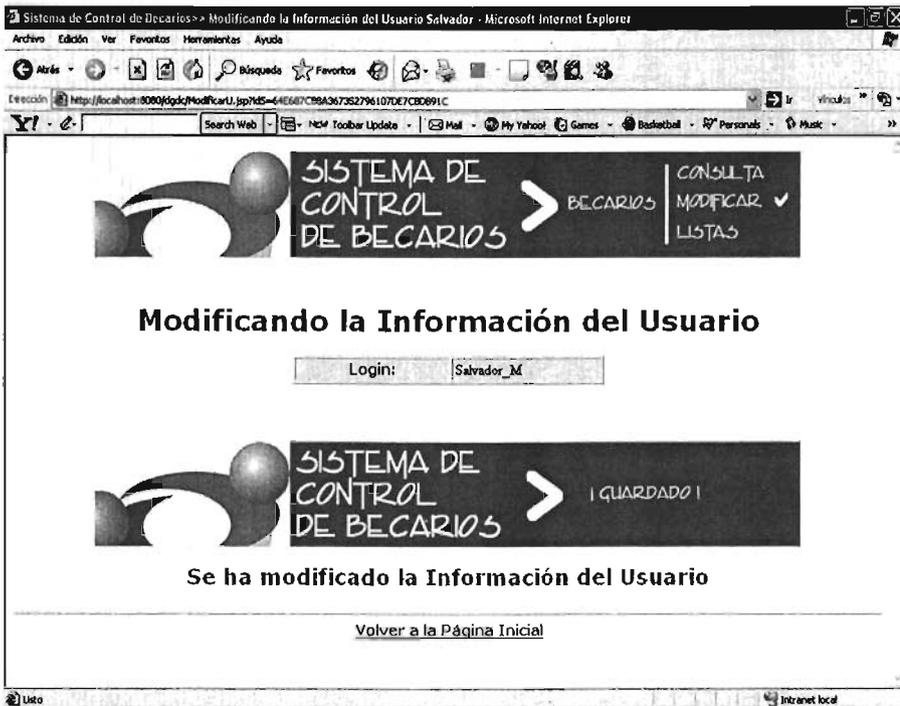
Aquí podremos modificar el login y el password, se pide que se confirme el password para asegurarnos de que el password es el que realmente queremos.

## Modificando la Información del Usuario

Ingresas tu Nuevo Login Y Password:

Login:	<input type="text" value="Salvador_M"/>
Password:	<input type="password" value="*****"/>
Confirmar Password:	<input type="password" value="*****"/>

Damos clic en el botón  y la información se actualizará tal como se muestra.



Damos clic en la referencia Volver a la Página Inicial para regresar al Menú de opciones.

Si los passwords nos son iguales tendremos un mensaje como el siguiente:



**No se pudo modificar la Información del Usuario**  
**Los passwords no son iguales Intentalo de Nuevo**

[Volver a la Página Inicial](#)

Por lo que tendremos que regresar al Menú de Opciones.

D) Crear un nuevo usuario

Crear un Nuevo Usuario:

Crear

Damos clic en Crear

Y se mostrará lo siguiente:



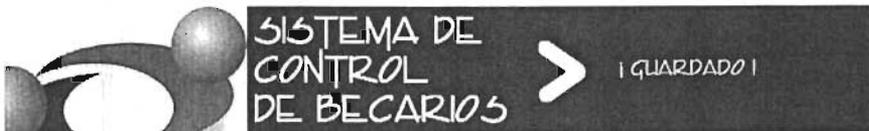
## Agregar un Nuevo Usuario al Sistema

Introduce el Nuevo Login y Password

Login:	<input type="text"/>
Password:	<input type="password"/>
<input type="button" value="Agregar Usuario"/>	

[Regresar a Opciones de Usuario](#)

Pondremos el login y el password del nuevo usuario damos clic en el botón [Agregar Usuario](#) y Listo tenemos un nuevo usuario en el sistema, esto se muestra a continuación:



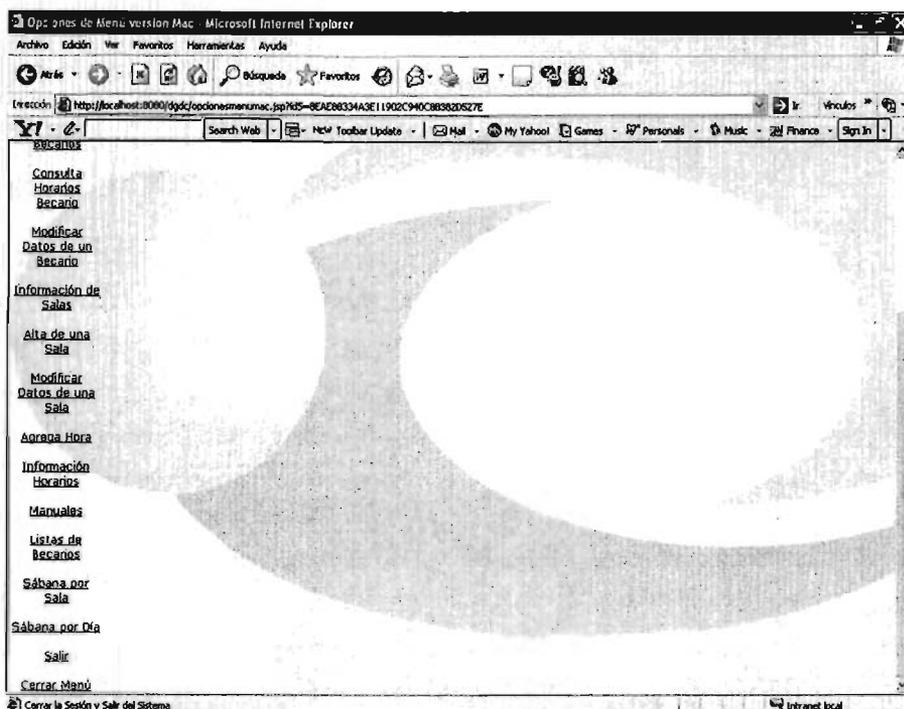
El registro ha sido agregado

[Regresar](#)

Damos clic sobre la referencia [Regresar](#) para ir al menú de Opciones del sistema.

### III. n) Salir

Para asegurarnos de Salir de la Sesión correctamente damos clic en la referencia [Salir](#) del menú de opciones:



Hacemos clic sobre nuestra referencia [Salir](#) la siguiente pantalla será:



[Salir del Sistema](#)

[Regresar al Sistema](#)

Si deseamos regresar al Sistema damos clic sobre la referencia:

[Regresar al Sistema](#) Para volver al menú de opciones.

Si lo que deseamos es salir definitivamente del sistema elegimos la opción [Salir del Sistema](#), ya que es la forma correcta de cerrar el sistema con seguridad, también se cerrara nuestro explorador, pero en el caso de Mac no mandara ningún mensaje y solo cerrara en explorador.

La opción [Cerrar Menú](#) solo oculta el menú de opciones Mac.

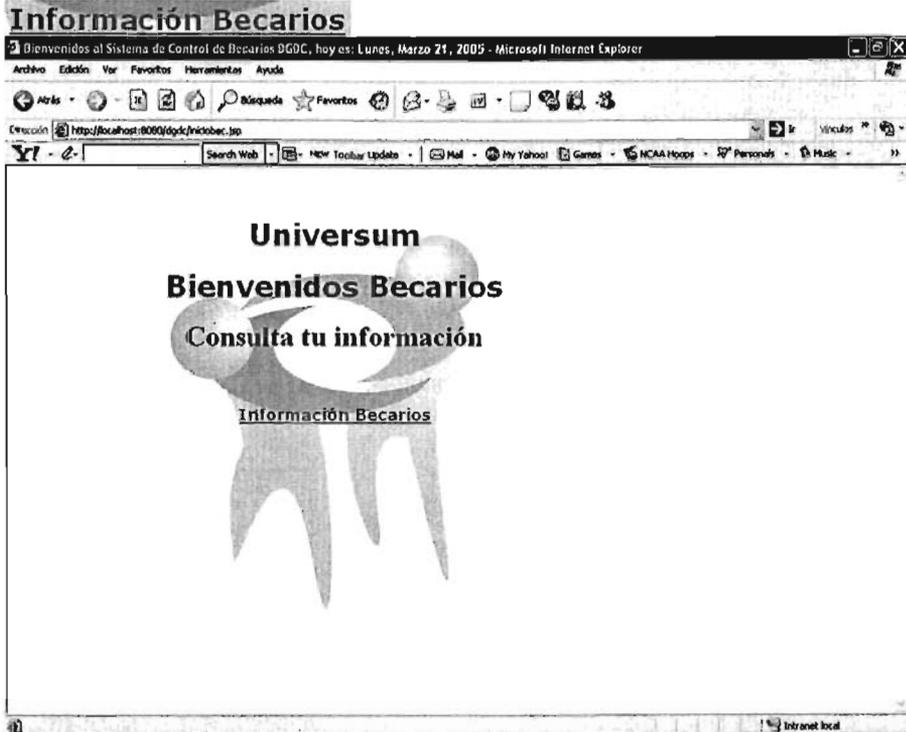
#### IV. Modulo Para Becarios.

La página que se les dará a los becarios para que realicen esta consulta será: <http://servidor/dgdc/iniciobec.jsp>

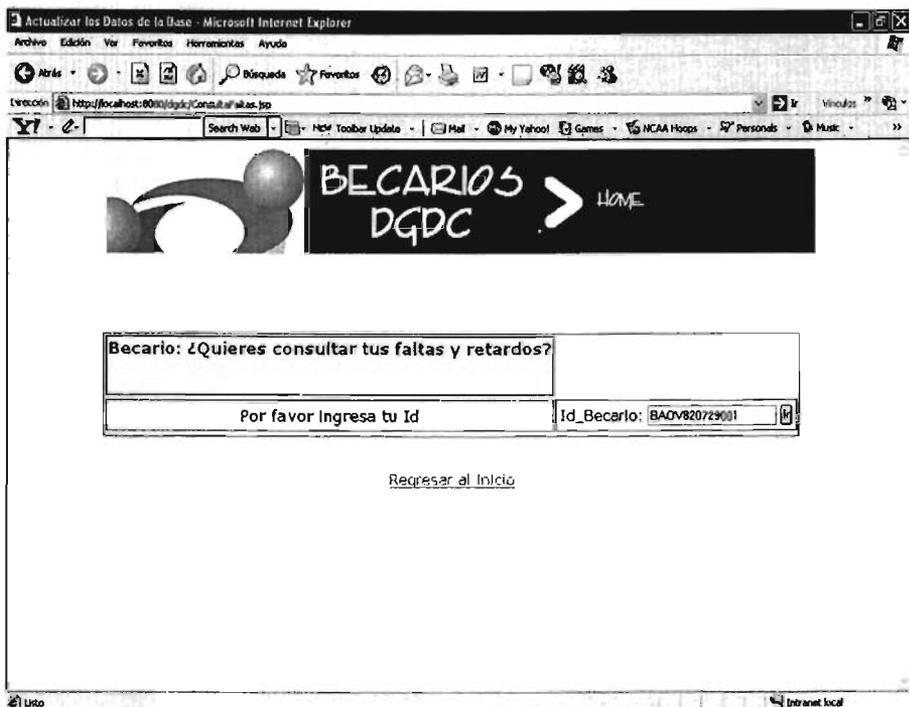
Aquí los becarios podrán consultar su horario, sus asistencias, retardos y faltas que tengan así como consultar sus datos personales.

La pantalla de bienvenida es la siguiente:

Para ingresar a este modulo solo tenemos que dar clic en

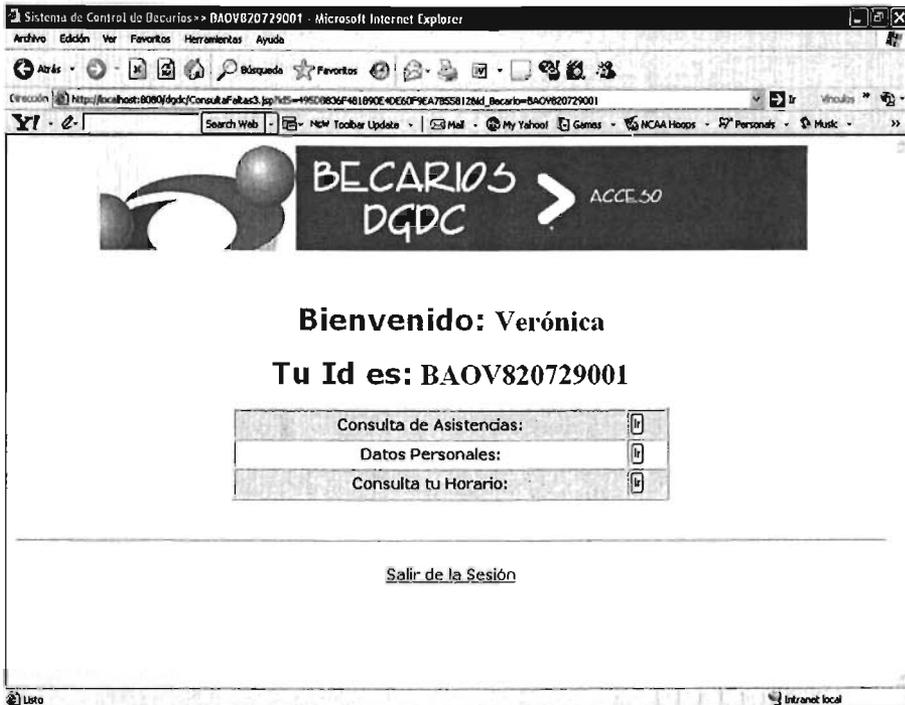


La siguiente pantalla mostrará un formulario donde el becario tiene que poner su ID.



Como ejemplo vamos a poner un id de uno de los Becarios y damos clic en el botón [Ir](#)

Si en cualquier momento el becario desea salir de la sesión solo tiene que dar clic en la referencia: [Salir de la Sesión](#)



En esta pantalla se muestra El nombre del Becario, su id y la opción que se desea saber:

Consulta de Asistencias:	<input type="button" value="Ir"/>
Datos Personales:	<input type="button" value="Ir"/>
Consulta tu Horario:	<input type="button" value="Ir"/>

Para ver la información damos clic en el botón  en la opción deseada.

## IV. a) Consulta de asistencias

**Consulta de Faltas y Retardos Becario: Verónica**

Id_Becario:	BAOV820729001
Número de Faltas:	2
Número de Asistencias:	50
Número de Retardos:	1
<input type="button" value="Regresar"/>	

Para regresar a seleccionar otra opción solo damos clic en el botón

## IV. b) Datos Personales

**Información del Becario: Verónica**

Id_Becario:	BAOV820729001
Nombre:	Verónica
Apellido Paterno:	Barranco
Apellido Materno:	Osaya
Fecha de Nacimiento:	29/7/82
Edad:	22
Edo. Civil:	soltero
Dirección:	Veracruzanos 26
Colonia:	Bonanza
Delegación:	Alvaro Obregón
C.P.:	1220
Teléfono:	52-73-72-82
Correo Electronico:	verosmaya@hotmail.com
Universidad:	UNAM
Facultad:	Pedagogía

Edo. Civil:	soltero
Dirección:	Veracruzanos 26
Colonia:	Bonanza
Delegación:	Alvaro Obregón
C.P.:	1220
Teléfono:	52-73-72-82
Correo Electronico:	verosmaya@hotmail.com
Universidad:	UNAM
Facultad:	Pedagogía
Carrera:	Pedagogía
Número de Cuenta:	9808695-5
Creditos:	100.00 Promedio: 9.60
Otros Estudios:	otros edos
Idioma1:	Inglés Habla:70(%) Lee: 60(%) Escribe: 40(%)
Idioma2:	Inglés Habla:40(%) Lee: 40(%) Escribe: 40(%)
Otras Actividades:	/N/
Sala o Área de Preferencia:	Una Sala en el Tiempo
Fecha de Inicio (Día/Mes/Año):	1/1/04
Fecha de Terminación (Día/Mes/Año):	1/09/04

Salir de la Sesión

Para regresar a seleccionar otra opción solo damos clic en el botón

## IV. c) Consulta tu Horario

Sistema de Control de Becarios - BAOV820729001 - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Inicio

http://localhost:8080/dgdc/ConsultaHorario.jsp?HS=192048364-481890E4DE609EA785581284\_Becario=BAOV820729001

Search Web

MSN Toolbar Update Mail My Yahoo! Games NCAA Hoops Personals Music

ACCESO

**Tu Horario Es:**

Id\_Becario:BAOV820729001      Nombre: Verónica Barranco Osnaya

Hora	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
9 a 10	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>				
10 a 11	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
11 a 12	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
12 a 13	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
13 a 14							
14 a 15					<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
15 a 16					<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
16 a 17					<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
17 a 18					<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Regresar

Imprimir el Horario pulsando en la imagen

Salir de la Sesión

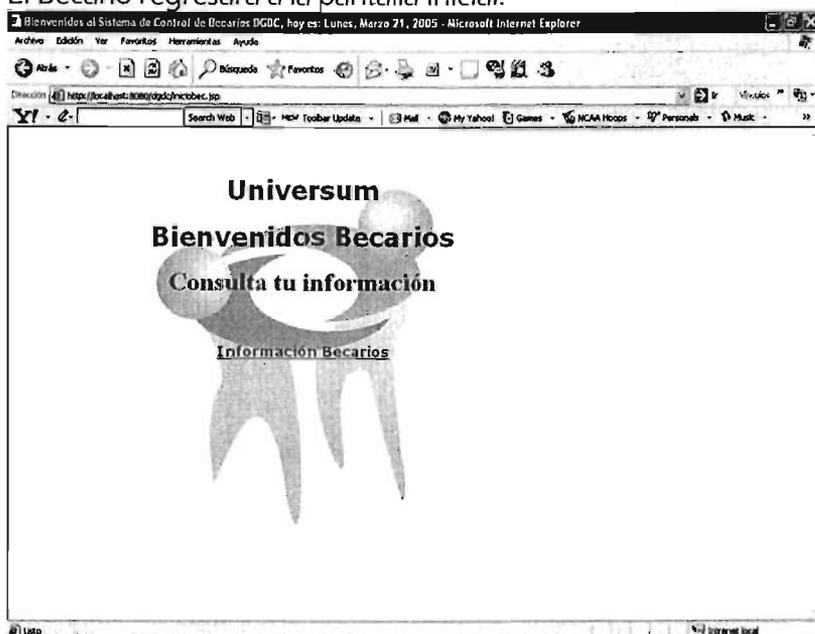
Inicio

Aquí tenemos la opción de imprimir el Horario al dar clic en la figura de la impresora.

Imprime el Horario pulsando en la imagen



Para salir de la Sesión damos clic en Salir de la Sesión.  
El Becario regresará a la pantalla inicial.



## **Apéndice D**

# **Manual Sistema de Control de Becarios Red Local**

## Manual de Usuario

### Sistema de Control de Información de Becarios



Realizado por:  
Yair Gómez Méndez  
J. Roberto Hernández Molina

Índice	Página
<b>I. Manual Sistema de control de Becarios Fase Red Local</b>	<b>03</b>
I. a) Iniciando el Sistema (Despertando el Sistema)	03
I. b) Inicio y Pantalla Principal	03
I. c) Iconos de Acceso Directos	05
Icono Sábana por Día	06
Icono Sábana por Sala	06
Icono Consultar Horario o Asignación a Salas	07
<b>II. Fichas del Sistema de Control de Información de Becarios</b>	<b>07</b>
II. a) Ficha Inicio	07
Icono Configuración del Sistema	07
1. IP del Servidor de Base de Datos	08
2. IP del Servidor de JSP	09
3. Usuario Local	09
Icono Herramientas	10
Icono Salir	13
II. b) Ficha Asistencia	13
i. Consulta en Tiempo Real	14
ii. Justificar	15
iii. Histórico	16
Modo Rápido	16
Versión Completa	18
II. c) Ficha Becario	25
i. Consulta Rápida	25
ii. Consulta Detallada	26
iii. Modificar Información	28
II. d) Ficha Salas	30
i. Agregar una Sala o Área	31
ii. Consultar/Modificar Información	32
II. e) Horas Entrada	34
i. Agregar Nueva Hora	35
ii. Consultar/Modificar Información	36
II. f) Sábana	36
i. Asignar un Becario a una Sala	36
ii. Asignar el Horario de un Becario	45
iii. Consultar la Sábana Por Día	50
iv. Consultar Sábana Por Sala	55

## **I. Manual Sistema de control de Becarios Fase Red Local**

Este manual es la referencia al usuario para usar el sistema de Control de Información de Becarios en su fase Red Local. Es una guía de referencia para el usuario destinado al manejo del sistema.

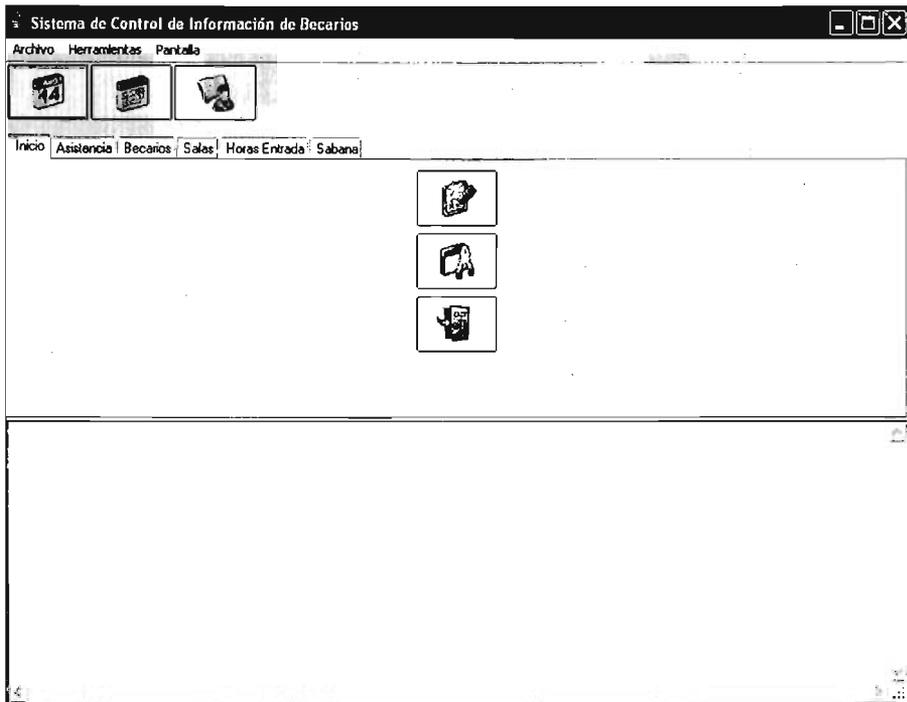
### **I. a) Iniciando el Sistema (Despertando el Sistema)**

Vamos a ver como es que abrimos este programa para su uso:

- Si estamos en una Mac tendremos un icono en el escritorio al cual daremos doble clic.
- Si se accesa al Sistema a través de una PC tendremos en el escritorio un acceso directo al cual también daremos doble clic.

### **I. b) Inicio y Pantalla Principal**

Una vez iniciado el programa la pantalla que veremos a continuación será la que se muestra:



Veamos primero las opciones de la Barra de Menús:



Aquí tenemos tres opciones:

1. Archivo: En esta opción podemos salir del Sistema seleccionando el menú Archivo > Salir, tal como se muestra en la figura:



## 2. Herramientas:

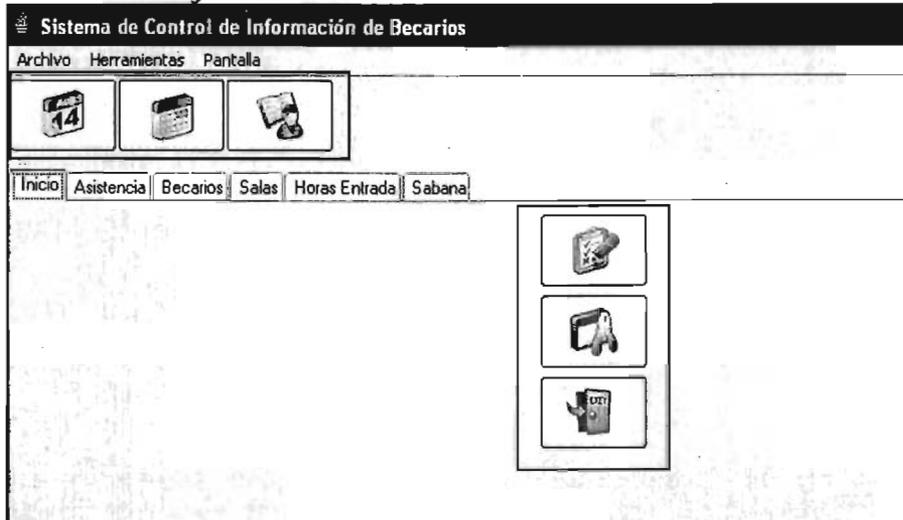
### 3. Pantalla:

Aquí tenemos la opción de limpiar la pantalla, seleccionando Pantalla>Limpiar como se muestra:



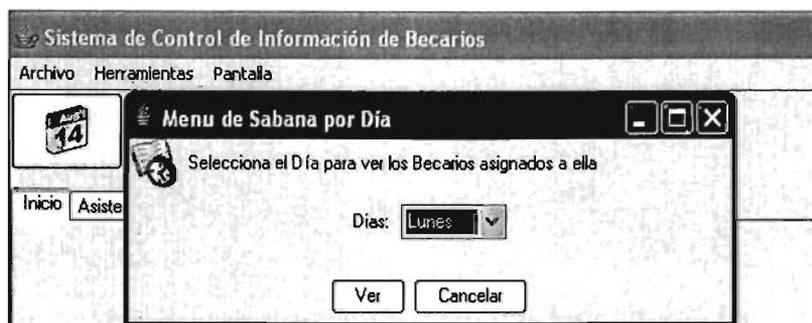
### I. c) Iconos de Acceso Directos

Los iconos de acceso directo son los que se encuentran encerrados en un recuadro rojo tal como se muestra:





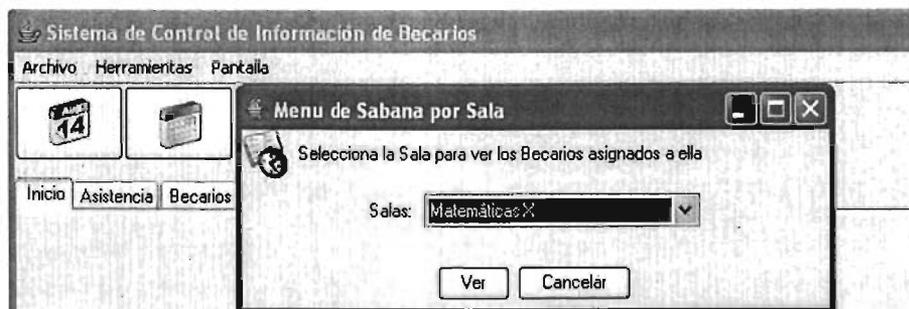
Al dar clic en el primer Icono  nos mostrará una pantalla emergente la cual nos indica generar la Sábana por Día como se ilustra a continuación:



La descripción se vera mas a detalladamente en la ficha Sabana.



El siguiente icono es  el cual al dar clic en el mostrará como generar la Sábana por Sala tal como muestra la siguiente figura:

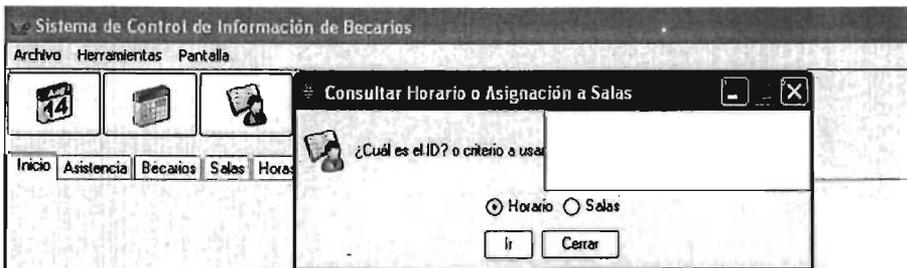


El funcionamiento se vera mas detalladamente en la ficha Sabana.



### Icono Consultar Horario o Asignación a Salas

El último icono de la parte superior es  donde podemos consultar Horario o la asignación de los becarios a las salas tal como se muestra aparece una pantalla emergente donde podemos dar el id del Becario o el criterio a usarse.



## II. Fichas del Sistema de Control de Información de Becarios

### II. a) Ficha Inicio

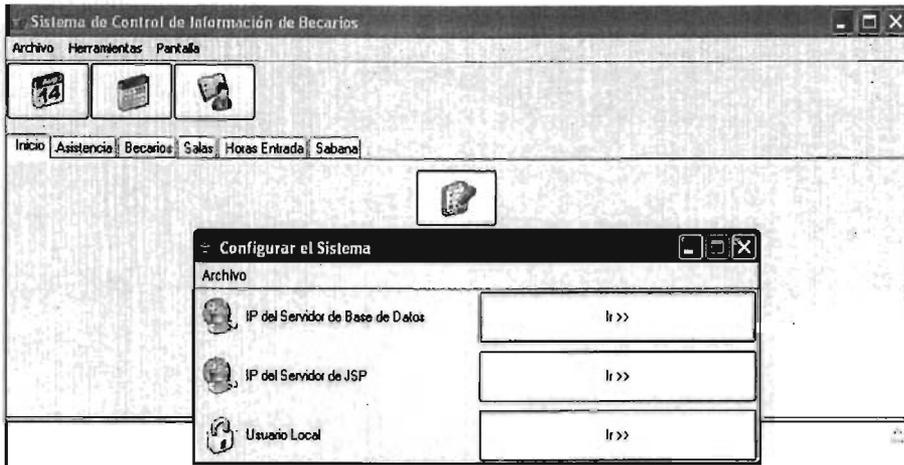


### Icono Configuración del Sistema

El siguiente icono de la parte inferior es Configuración del Sistema



en donde podemos configurar el Sistema, aquí al dar clic sobre el icono mencionado aparece una pantalla tal como se ilustra:



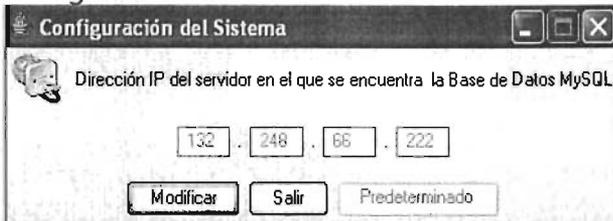
En esta pantalla hay tres opciones:

1. IP del Servidor de Base de Datos
2. IP del Servidor de JSP
3. Usuario Local

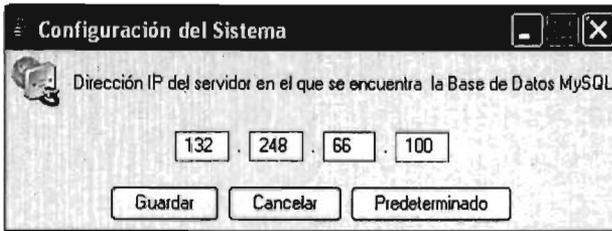
### 1. IP del Servidor de Base de Datos



Al dar clic en el botón Ir>> veremos las siguientes opciones de configuración:



Aquí tenemos la dirección IP del servidor de la Base de Datos que esta en este momento funcionando, en caso de que nuestra Base de Datos se encuentre en otra dirección IP y necesitemos cambiarla solo damos clic en el botón **Modificar** y cambiamos la dirección IP, por ejemplo; supongamos que se cambia la base de datos a la dirección 132.248.66.100 entonces tenemos que cambiar la dirección tal como se muestra:

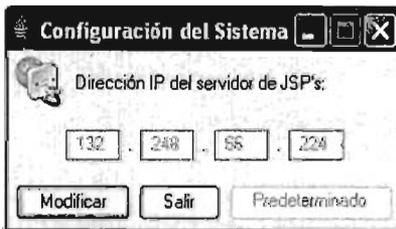


Para conservar esta configuración damos clic en el botón **Guardar** de lo contrario damos clic en el botón **Cancelar**, si queremos volver a la configuración anterior solo damos clic en el botón **Predeterminado**.

## 2. IP del Servidor de JSP

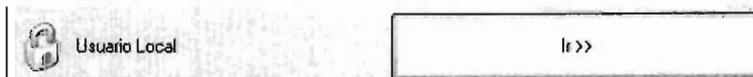


Damos clic en Ir>> veremos las siguientes opciones:

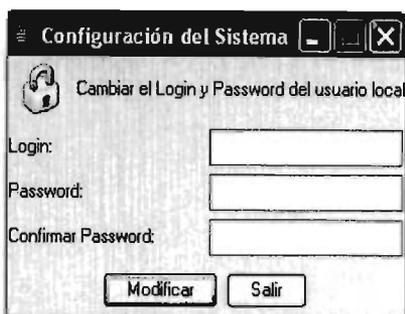


Al igual que en el punto numero 1 aquí podremos cambiar la Dirección de la IP del servidor de JSP's, si deseamos modificar esta dirección damos clic en el botón **Modificar**, de lo contrario damos clic en **Salir**.

## 3. Usuario Local



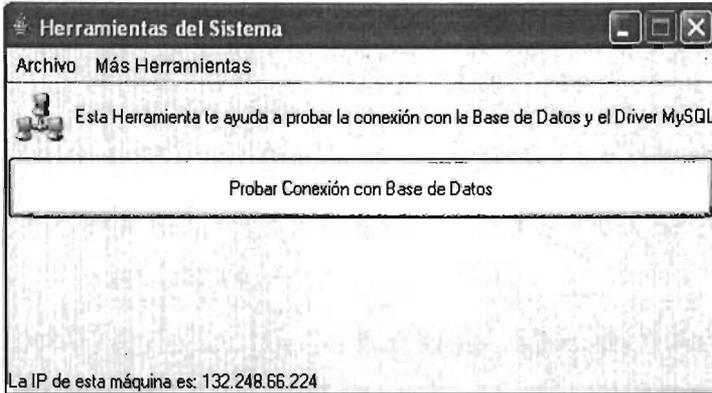
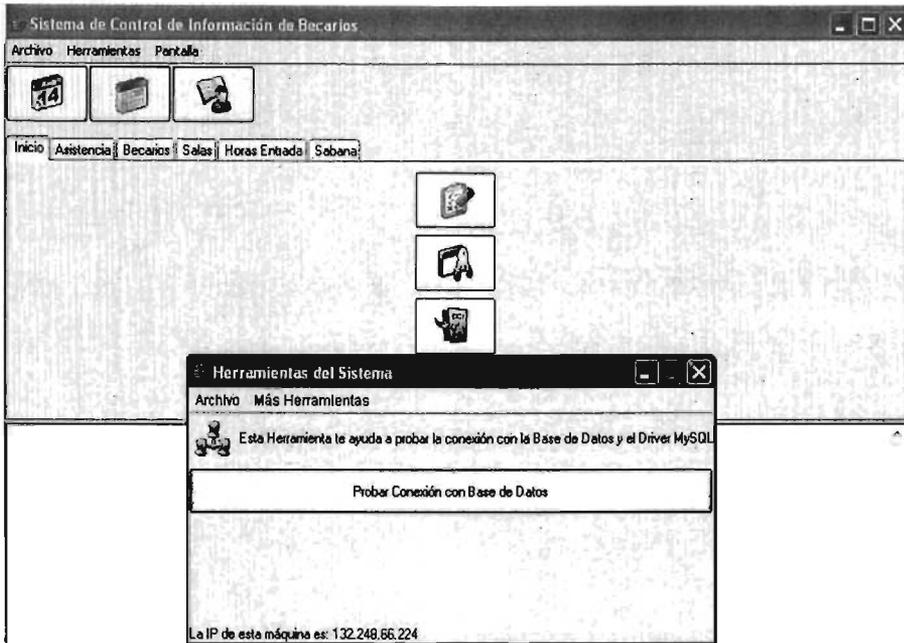
Aquí podremos cambiar el nombre de usuario y el password del Usuario Local tal como se muestra:



Para modificar el Login y password solo tenemos que llenar los campos solicitados y dar clic en el botón **Modificar**, de lo contrario damos clic en **Salir**.

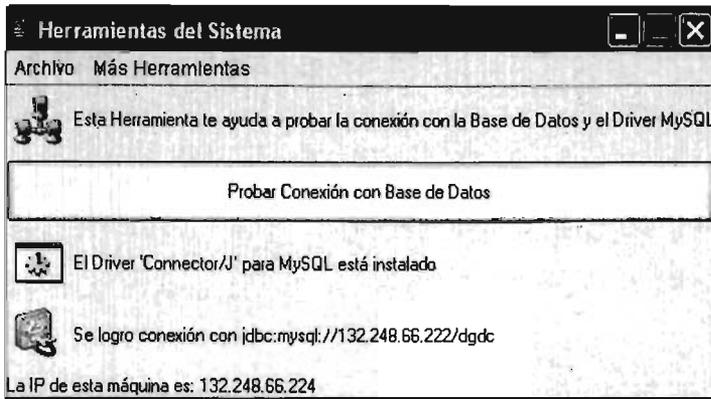


El siguiente icono de la parte inferior es el de Herramientas , aquí podremos saber si nuestro sistema esta debidamente configurado, al dar clic en el icono se muestra la siguiente pantalla:



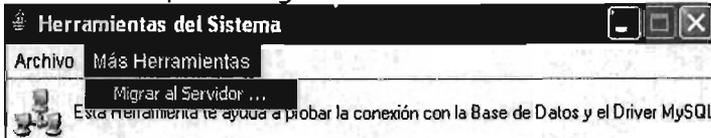
Como se muestra Esta herramienta ayuda a probar la conexión con la base de datos y el Driver Mysql en la parte inferior se indica la IP de la maquina de donde estamos accedendo al sistema.

En esta pantalla al dar clic en Probar Conexión con la Base de Datos nos mostrara mensajes a cerca de si tenemos bien configurado el Conector Java, así al estar bien configurado los mensajes serán los siguientes:

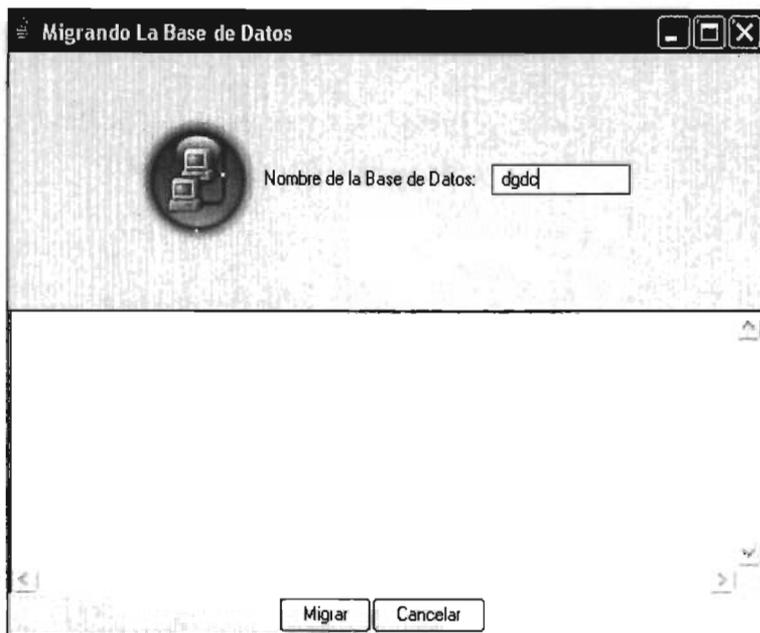


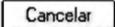
Esto quiere decir el sistema se encuentra en optimas condiciones para su funcionamiento, si los mensajes son de error el problema se encuentra en la base de datos.

En la Barra de Tareas se encuentra el Menú Mas Herramientas donde tenemos la opción Migrar al Servidor... tal como se muestra:



Aquí lo que hacemos es Migrar Al servidor la Base de datos Temporal que la cual se capturo al momento de realizar la credencial véase Manual C2wCAM o manual de credenciales.



Aquí escribimos el nombre de la base de Datos en donde se indica y damos clic en el botón  para Salir Damos clic en el botón 



Al dar clic en el icono  podremos salir del sistema.

## II. b) Ficha Asistencia

En la ficha Asistencia tenemos tres fichas las cuales llamaremos subfichas:

- i. Consultas en Tiempo Real
- ii. Justificar
- iii. Histórico

Inicio | Asistencia | Becarios | Salas | Horas Entrada | Sabana |  
Consultas en Tiempo Real | Justificar | Histórico |



Para hacer una consulta en Tiempo Real selecciona que tipo de consulta quieres:

Modo Texto  Modo Gráfico

Aceptar

### i. Consulta en Tiempo Real

En la subficha Consulta en Tiempo Real

Podremos hacer una consulta de los becarios que ya han llegado las opciones son:

Modo texto y Modo Gráfico

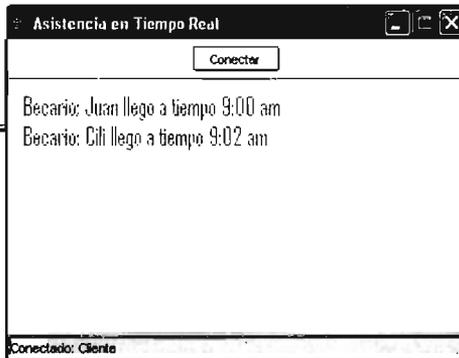
En el modo texto nos indica que becario ha llegado a tiempo a su sala como se deja ver en la siguiente figura:

Inicio | Asistencia | Becarios | Salas | Horas Entrada | Sabana |  
Consultas en Tiempo Real | Justificar | Histórico |



Para hacer una consulta en Tiempo Real selecciona que tipo de consulta quieres:

Modo Texto  Modo Gráfico



El Modo gráfico solo se le agregan iconos para saber si el becario ha llegado a tiempo.

## ii. Justificar

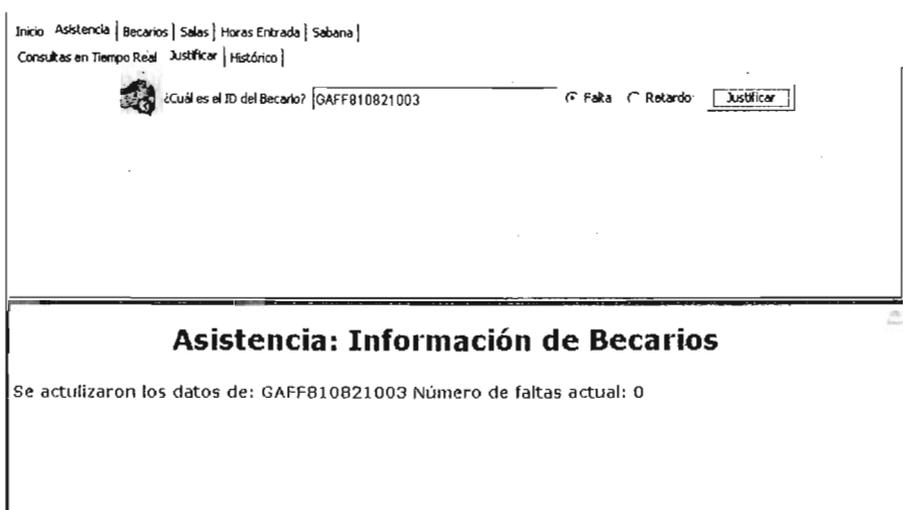
Para Justificar alguna falta de los Becarios tenemos esta subficha a la cual simplemente tendremos que teclear en la caja de texto el Id del Becario y seleccionar que es lo que se va a justificar si una Falta o un Retardo, enseguida damos clic en el botón Justificar y la falta estará Justificada.



Inicio | Asistencia | Becarios | Salas | Horas Entrada | Sabana |  
Consultas en Tiempo Real | Justificar | Histórico |

¿Cuál es el ID del Becario?   Falta  Retardo

Ejemplo deseamos justificar la falta del becario con ID GAFF810821003, este Id lo tecleamos, seleccionamos la opción Falta y damos clic en el botón Justificar como se muestra:



Inicio | Asistencia | Becarios | Salas | Horas Entrada | Sabana |  
Consultas en Tiempo Real | Justificar | Histórico |

¿Cuál es el ID del Becario?   Falta  Retardo

---

**Asistencia: Información de Becarios**

Se actualizaron los datos de: GAFF810821003 Número de faltas actual: 0

El mensaje que nos proporciona el sistema se observa y el número de faltas de este Becario ya es 0.

Lo mismo sucede si deseamos justificar los retardos en este caso seleccionaremos la opción Retardo y las instrucciones son las mismas: Veamos esto con el mismo Becario:

Inicio | Asistencia | Becarios | Salas | Horas Entrada | Sabana |  
 Consultas en Tiempo Real | Justificar | Histórico |

¿Cuál es el ID del Becario? GAFF810821003  Falta  Retardo

### Asistencia: Información de Becarios

Se actualizaron los datos de: GAFF810821003 Número de retardos actual: 0

### iii. Histórico

En la subficha histórico tenemos dos opciones Modo Rápido y Versión Completa tal como se muestra:

Inicio | Asistencia | Becarios | Salas | Horas Entrada | Sabana |  
 Consultas en Tiempo Real | Justificar | Histórico |  
 Modo Rápido | Versión Completa |

Selecciona el Criterio de Búsqueda

¿Cuál es el ID del Becario?

Faltas

### II. c) Modo Rápido

Podemos seleccionar entre faltas, retardos y asistencias de la lista y teclear el Id de Becario una vez hecho esto damos clic en Buscar, veamos un ejemplo:

Supongamos que queremos ver las faltas de un Becario con el ID GAFF810821003 esta consulta se muestra a continuación:

Inicio | Asistencia | Becarios | Salas | Horas Entrada | Sabana |  
 Consultas en Tiempo Real | Justificar | Histórico |  
 Modo Rápido | Versión Completa |

Selecciona el Criterio de Búsqueda

Faltas

¿Cuál es el ID del Becario?

GAFF810821003

Buscar

### Asistencia: Información de Becarios

Resultados de Consulta	
Nombre	Fernando Gachuz Fuentes
Faltas	1

Como se observa se genera una consulta para este Id de Becario mostrando su numero de faltas en este caso este Becario tiene 1 falta.

Lo mismo sucede si deseamos conocer el número de asistencias o Retardos que tiene este Becario con solo seleccionar otro criterio de Búsqueda como se muestra:

Para Retardos:

Inicio | Asistencia | Becarios | Salas | Horas Entrada | Sabana |  
 Consultas en Tiempo Real | Justificar | Histórico |  
 Modo Rápido | Versión Completa |

Selecciona el Criterio de Búsqueda

Retardos

¿Cuál es el ID del Becario?

GAFF810821003

Buscar

### Asistencia: Información de Becarios

Resultados de Consulta	
Nombre	Fernando Gachuz Fuentes
Retardos	1

Para Asistencias:

Inicio | Asistencia | Becarios | Salas | Horas Entrada | Sabana |

Consultas en Tiempo Real | Justificar | Histórico |

Modo Rápido | Versión Completa |

Selecciona el Criterio de Búsqueda

Asistencia

¿Cuál es el ID del Becario?

QAFF810821003

Buscar

---

### Asistencia: Información de Becarios

Resultados de Consulta	
Nombre	Fernando Gachuz Fuentes
Asistencia	30

## II. d) Versión Completa

En esta parte podremos seleccionar el criterio que deseamos, esto es muy útil cuando queremos ver a todos los becarios o saber quienes tienen faltas, asistencias o retardos:

Supongamos que deseamos ver las asistencias, los retardos o las faltas de todos los becarios contenidos en el sistema, entonces tendremos que poner en la caja de texto la instrucción "todo" en ¿Cuál es el Criterio? Tal como se muestra:

Inicio | Asistencia | Becarios | Salas | Horas Entrada | Sabana |

Consultas en Tiempo Real | Justificar | Histórico |

Modo Rápido | Versión Completa |

¿Cuál es el Criterio?

Todo

Buscar

Aparecerá una ventana en la cual se muestran a todos los Becarios contenidos en el sistema como se puede observar:

Resultados de La Consulta					
id_Becario	nombre	appat	num_faltas	num_retardos	num_asistencias
BAOV820729001	Verónica	Barranco	0	1	50
CADC820921002	Cili	Castro	0	1	40
GAFF810821003	Fernando	Gachuz	1	1	30
LOMC810816004	Christian Eugenio	López-Negrete	0	2	20
GOMK800415005	Karla	Maldonado	1	1	15
MAPA800530006	Ariana	Martínez	0	0	60
NUAM830514007	Mariana	Núñez	0	1	20
ORRM810509008	María de Lourdes	Ortega	1	2	40
RACG810924009	Gabriel	Ramos	0	1	21
HEAL791006010	Lorena	Arenas	1	1	25
ORRM810509008	María de Lourdes	Ortega	0	1	5
RACG810924009	Gabriel	Ramos	1	1	21
NUAM830514007	Mariana	Núñez	0	0	0
BEDY830911023	Yéssica Yislem	Bekrán	0	0	0
CASR810515011	Rocio	Castillo	0	0	0
CHCP830417013	Paola	Cholula	0	0	0
CIRP790719024	Paulina	Cifuentes	3	3	6
FERJ790804014	José	Feliciano	0	0	0
FLOK910315015	Karina	Flores	0	0	0
CAGE810102012	Elsie	Castro	0	0	0
GOMY810708021	Yair	Gomez	0	0	0
HEMR810504020	Roberto	Hernández	0	0	0
NALR800627019	Roxana Michel	Navarrete	0	0	1
RORK820813020	Karla Marilin	Rodríguez	0	0	0
SAAMR840310021	Mimiel	Santhiáñez	0	0	0

Probablemente deseamos conocer quienes de los becarios no tienen ningún retardo o ninguna falta para esto utilizamos otros criterios los cuales serán:

Falta=0, Falta=1, retardo=0, retardo=1, etc., o Falta >2, Falta >3, retardo >2, retardo>3, etc., o Falta <1, retardo<1, Falta <2, retardo<2, etc. Tal como se muestra en los ejemplos siguientes:

Escribimos falta < 2 en la caja de texto y damos clic en el botón Buscar:

Modo Rápido | Versión Completa

¿Cual es el Criterio?

falta < 2

Buscar

Nos despliega una ventana donde tenemos a todos los becarios que tienen menos de dos faltas como se muestra:

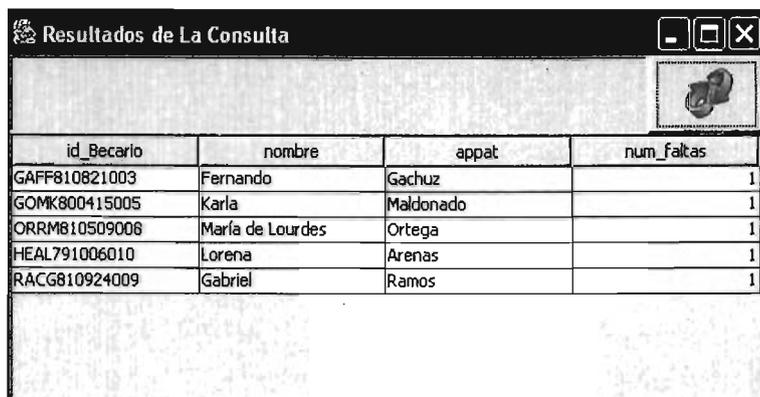
Resultados de La Consulta			
id_Becario	nombre	appat	num_faltas
BAOV820729001	Verónica	Barranco	0
CADC820921002	Cili	Castro	0
GAFF810821003	Fernando	Gachuz	1
LOMC810816004	Christian Eugenio	López-Negrete	0
GOMK800415005	Karla	Maldonado	1
MAPA800530006	Ariana	Martínez	0
NUAM830514007	Mariana	Núñez	0
ORRM810509008	María de Lourdes	Ortega	1
RACG810924009	Gabriel	Ramos	0
HEAL791006010	Lorena	Arenas	1
ORRM810509008	María de Lourdes	Ortega	0
RACG810924009	Gabriel	Ramos	1
NUAM830514007	Mariana	Núñez	0
BEDY830911023	Yéssica Yislem	Beltrán	0
CASR810515011	Rocio	Castillo	0
CHCP8304117013	Paola	Cholula	0
FERJ790804014	José	Feliciano	0
FLOK910315015	Karina	Flores	0
CAGE810102012	Elsie	Castro	0
GOMY810708021	Yair	Gomez	0
HEMR810504020	Roberto	Hernández	0
NALR800627019	Roxana Michel	Navarrete	0
RORK820813020	Karla Marilin	Rodríguez	0
SAAM840310021	Miguel	Santibáñez	0

Si queremos saber cuantos son los que tienen más de dos faltas, escribimos en la caja de texto falta > 2 tendremos lo siguiente:

Resultados de La Consulta			
id_Becario	nombre	appat	num_faltas
CIRP790719024	Paulina	Cifuentes	3

Nos muestra una ventana donde solo tenemos un becario que tiene faltas mayores que dos.

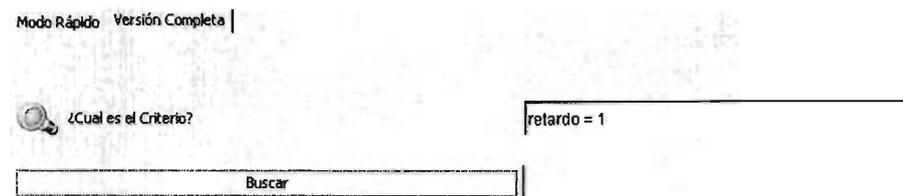
Tal vez deseamos saber todos los becarios que solo tienen una falta, por tanto tecleamos falta = 1, se muestra lo siguiente:



The screenshot shows a window titled "Resultados de La Consulta" with a search icon in the top right corner. Below the title bar is a table with the following data:

id_Becario	nombre	appat	num_faltas
GAFF810821003	Fernando	Gachuz	1
GOMK800415005	Karla	Maldonado	1
ORRM810509008	María de Lourdes	Ortega	1
HEAL791006010	Lorena	Arenas	1
RACG810924009	Gabriel	Ramos	1

El mismo criterio se utiliza para saber cual es el número de retardos que tienen los Becarios registrados en el sistema, por ejemplo deseamos saber los becarios que tienen solo un retardo, escribimos en la caja de texto retardo = 1, damos clic en el botón Buscar y se muestra lo siguiente:



The screenshot shows the search interface with the following elements:

- Mode: **Modo Rápido** | **Versión Completa**
- Search criteria: **¿Cual es el Criterio?** [retardo = 1]
- Search button: **Buscar**

Resultados de La Consulta

id_Becario	nombre	appat	num_retardos
BAOV820729001	Verónica	Barranco	1
CADC820921002	Cili	Castro	1
GAFF810821003	Fernando	Gachuz	1
GOMK800415005	Karla	Maldonado	1
NUAM830514007	Mariana	Núñez	1
RACG810924009	Gabriel	Ramos	1
HEAL791006010	Lorena	Arenas	1
ORRM810509008	María de Lourdes	Ortega	1
RACG810924009	Gabriel	Ramos	1

Pero no solo esto, si deseamos buscar por nombre del becario solo lo ponemos en la caja de texto y damos clic en el botón Buscar tal como se muestra:

Ejemplo buscar al becario cuyo nombre es Cili:

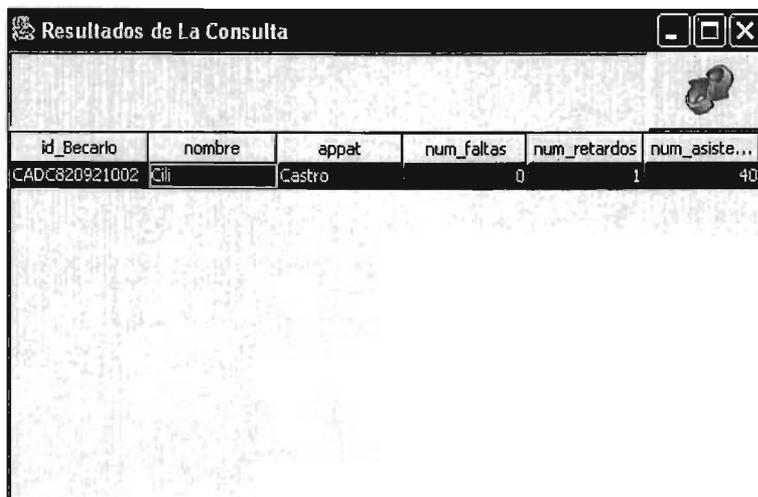
Inicio | Asistencia | Becarios | Salas | Horas Entrada | Sabana |

Consultas en Tiempo Real | Justificar | Histórico |

Modo Rápido | Versión Completa |

¿Cual es el Criterio?

Aparece la siguiente ventana:



id_Becario	nombre	appat	num_faltas	num_retardos	num_asiste...
CAD0620921002	Cili	Castro	0	1	40

Aquí se muestra el resultado de nuestra búsqueda la cual era encontrar al becario cuyo nombre es Cili.

De forma similar podemos buscar por apellido paterno anteponiendo un (ap.) al apellido paterno del becario.

Tal como se muestra:

Supongamos que deseamos conocer a los Becarios que tengan el apellido paterno Castillo entonces tecleamos ap.Castillo y damos clic en el botón Buscar, esto se muestra a continuación:

Inicio | Asistencia | Becarios | Salas | Horas Entrada | Sabana |  
Consultas en Tiempo Real | Justificar | Histórico |  
Modo Rápido | Versión Completa |



¿Cual es el Criterio?

ap.Castillo

Buscar

Aparece la siguiente ventana que es el resultado de nuestra búsqueda:

id_Becario	nombre	appat	num_faltas	num_retardos	num_asiste...
CASR810515011	Rocio	Castillo	0	0	0

Se observa que el becario efectivamente tiene el apellido Castillo

Si lo deseamos buscar por Id solo tecleamos el ID del Becario damos clic en Buscar y aparece la ventana donde nos indica los datos del Becario en cuanto a la asistencia como se puede ver:

Inicio | Asistencia | Becarios | Salas | Horas Entrada | Sabana |

Consultas en Tiempo Real | Justificar | Histórico |

Modo Rápido | Versión Completa |

¿Cual es el Criterio?

Buscar

id_Becario	nombre	appat	num_faltas	num_retardos	num_asisten...
BAOV820729...	Verónica	Barranco	0	1	50

El criterio no

Como se observa podemos utilizar cualquiera de estas palabras con sus respectivos valores:

- Todo.
- Falta >número, <número, =número.
- Retardo >número, <número, =número.
- Nombre del Becario
- Apellido paterno con la siguiente instrucción:
  - ap. Apellido paterno
- Id de Becario

## II. c) Ficha Becarios

En la ficha Becarios tenemos tres fichas más a las cuales llamaremos subfichas, dos tipos de Consultas y modificar los datos de un Becario

- i. Consulta Rápida
- ii. Consulta Detallada
- iii. Modificar Información

### i. Consulta Rápida

En consulta Rápida podremos insertar el Id de un Becario para mostrar algunos de sus datos Personales, por ejemplo supongamos que deseamos saber los datos personales (nombre, universidad, teléfono, dirección, etc.) del Becario cuyo Id es BAOV820729001, entonces lo tecleamos como se muestra a continuación:

Inicio Asistencia Becarios Salas Horas Entrada Sabana

Consulta Rápida Consulta Detallada Modificar Información

¿Cuál es el ID del Becario?

BAOV820729001

Buscar

Después de teclear el Id mencionado damos clic en el botón Buscar a continuación en el área inferior aparecerán los datos de este Becario tal como se ilustra:

Inicio	Asistencia	Becarios	Salas	Horas Entrada	Sabana
Consulta Rápida	Consulta Detallada	Modificar Información			
 ¿Cuál es el ID del Becario?		<input type="text" value="BAOV820729001"/>			
<input type="button" value="Buscar"/>					

Becarios: Información de Becarios	
Resultados de Consulta	
Nombre	Verónica Barranco Osnaya
ID	BAOV820729001
Calle y número	Veracruzanos 26
Colonia	Bonanza
Delegación	Alvaro Obregón
Teléfono	52-73-72-82
e-mail	veroosnaya@hotmail.com
Universidad	UNAM
Carrera	Pedagogía

## ii. Consulta Detallada

En consulta detallada podremos buscar, por los siguientes criterios:  
 Por Id de Becario, por Nombre del Becario, por Apellido paterno:

Supongamos que deseamos buscar a diferentes Becarios a la vez, es decir, necesitamos buscar a un becario que tiene el id: BAOV820729001; a un Becario cuyo nombre es: Cili; y a un Becario que tiene el siguiente apellido paterno: Castillo; ap.Castillo para buscar por apellido primero se tiene que teclear las siglas (ap.) y después del punto el apellido Paterno, tal como se muestra en la figura:

Inicio	Asistencia	Becarios	Salas	Horas Entrada	Sabana
Consulta Rápida	Consulta Detallada	Modificar Información			
 ¿Cuál es criterio a usar?		<input type="text" value="BAOV820729001,Cil,ap.Castillo"/>			
<input type="button" value="Buscar"/>					



Resultados de La Consulta												
ID Becario	nombre	apell	apmat	fechaNB	mesNB	ano...						
BAOV820729001	Verónica	Basanco	Domínguez	29/7	1982	22						
CADC820921002	Cla	Castro	Domínguez	21/9	82	22						
GAFF810821003	Fernando	Gachuz	Fuentes	21/8	81	23						
LDWC810818004	Orvilson Eugenio	López-Magrelá	Miranda	16/08	81	23						
GOMX800415005	Karla	Maldonado	González	15/4	80	24						
MAPA800530006	Astoria	Martínez	Pérez	30/5	80	24						
NIUA830514007	Mariana	Ruíz	Alvarez	14/05	83	21						
ORRM810509008	Marta de Lourdes	Díaz	Ramos	9/05	81	23						
RACB810924009	Gabriel	Ramos	Castro	24/9	81	23						
HEAL791006010	Lorena	Arenas	Hernández	6/10	79	25						
CASR810515011	Rocio	Castillo	Sánchez	15/5	81	23						
GOMY810708021	Yar	Gómez	Mendez	7/08	81	23						
MAGE811708022	Eduardo	Martínez	Galvan	17/08	81	23						
ONCP830417013	Paco	Cholula	Cambray	17/4	83	22						
FEJF790804014	Luis	Falciano	Rodríguez	4/8	79	25						
FLDK810315015	Karna	Flore	Quintero	15/3	81	23						
GABM821030016	Melrose	García	Bautista	30/10	82	22						
JASD800829017	Diana Lebeth	Jardón	Sánchez	6/8	80	24						
MDFS800924018	Sandy	Morales	Flores	24/09	80	24						
NALR800627019	Rosana Michel	Navarrete	León	27/6	80	24						
RDRK820813020	Karla Marín	Rodríguez	Ramos	13/8	82	22						
SAAM840316021	Miguel	Santibáñez	Andrade	10/3	84	21						
CAGE810102012	Elise	Castro	Galicia	2/1	81	23						
BEDY830811023	Ynésica Yslim	Balboa	Díaz	11/8	83	22						
CHPP80719024	Rafaela	Clemente	Ruiz	19/07	79	25						
HEMR810504028	Roberto	Hernández	Molina	4/05	81	23						

### iii. Modificar Información

En la subficha modificar información, podremos actualizar los datos personales de los Becarios, o corregir algún error en los Datos del Becario.

Lo único que tenemos que hacer para modificar la información de los Becarios es conocer su id y escribirlo en la caja de texto:

Inicio Asistencia Becarios Salas Horas Entrada Sabana

Consulta Rápida Consulta Detallada **Modificar Información**


 Modificar Información Personal de los Becarios. ¿Cuál es el ID?:

Por ejemplo supongamos que deseamos modificar los datos del becario cuyo id es BAOV820729001, entonces lo escribimos y damos clic en el botón  como se muestra:

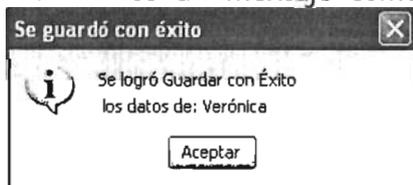
Inicio	Asistencia	Becarios	Salas	Horas Entrada	Sabana
Consulta Rápida	Consulta Detallada	Modificar Información			
		Modificar Información Personal de los Becarios. ¿Cuál es el ID?:	<input type="text" value="BADV820729001"/>	<input type="button" value="Ir a Modificar"/>	

Aparecerá la siguiente pantalla:

Modificar Información de Becarios	
Modificar Información del Becarios	
Nombre	<input type="text" value="Yanivica"/>
Apellido Paterno	<input type="text" value="Barranco"/>
Apellido Materno	<input type="text" value="Ornaga"/>
Fecha de Nacimiento	Día/Mes/Año <input type="text" value="23"/> <input type="text" value="7"/> <input type="text" value="1982"/>
Edad	<input type="text" value="22"/>
Estado Civil	<input type="text" value="Soltero(a)"/>
Dirección	<input type="text" value="Maracanitas 20"/>
Colonia	<input type="text" value="Bonánica"/>
Delegación	<input type="text" value="Alvaro Obregón"/>
C.P.	<input type="text" value="1220"/>
Teléfono	<input type="text" value="52-73-72-62"/>
Correo Electrónico	<input type="text" value="yvecohnaya@hotmail.com"/>
Universidad	<input type="text" value="UNAM"/>
Facultad	<input type="text" value="Pedagogía"/>
<input type="button" value="Modificar"/> <input type="button" value="Salir"/>	

C.P.	1220
Teléfono	52-73-72-82
Correo Electrónico	veroonnaye@hotmail.com
Universidad	UNAM
Facultad	Pedagogía
Carrera	Pedagogía
Número de Cuenta	9908635-5
Créditos	100.00 Promedio 9.60
Otros Estudios	otros edos
Idioma 1	Inglés Hable 50 Lee 50 Escribe 40
Idioma 2	Inglés Hable 40 Lee 50 Escribe 40
Otras Actividades	/N
Sexo o Área	Biología Humana y Salud

Para modificar cualquier Dato damos clic en el botón **Modificar** y realizamos las correcciones que deseamos. Una vez modificados estos datos damos clic en el botón **Guardar** para actualizar los cambios. Tendremos un mensaje como este si se logro guardar con éxito



Para regresar damos clic en el botón Aceptar.

Si no queremos guardar la información solo damos clic en **Cancelar** para regresar

## II. d) Ficha Salas

En la ficha Salas tenemos las siguientes subfichas:

- i. Agregar una Sala o área
- ii. Consultar/Modificar Información.

Tal como se observa en la figura siguiente:

The screenshot shows a web application interface with a menu bar at the top containing 'Inicio', 'Asistencia', 'Becarios', 'Salas', 'Horas Entrada', and 'Sabana'. Below the menu, there are two tabs: 'Agregar una Sala o Área' (selected) and 'Consultar/Modificar Información'. The form contains a small icon of a building, a text input field with the label '¿Cuál es el ID de la Sala?', and a 'Crear un ID' button. Below this is another text input field labeled 'Nombre de la Sala'. At the bottom of the form are two buttons: 'Cancelar' and 'Nuevo'.

### i. Agregar una Sala o Área

Si lo que deseamos es agregar una Sala al sistema damos clic en el botón Nuevo para agregar la nueva sala e inmediatamente nos deshabilitará los botones y nuevo cambiara su nombre por Guardar tal como se observa:

This screenshot shows the same form as above, but after clicking the 'Nuevo' button. The 'Crear un ID' button is now disabled (grayed out). The 'Nombre de la Sala' input field is now empty. The 'Cancelar' button remains, but the 'Nuevo' button has been replaced by a 'Guardar' button.

Supongamos que vamos a crear una Sala nueva que llamaremos "Sala Nueva1", entonces primero creamos su Id y damos clic en el botón  automáticamente se creará el Id de Sala, después pondremos el nombre de la sala y damos clic en el botón Guardar, tal como se muestra:

The screenshot shows a web application interface with a menu bar at the top containing 'Inicio', 'Asistencia', 'Becarios', 'Salas', 'Horas Entrada', and 'Sabana'. Below the menu, there are two tabs: 'Agregar una Sala o Área' and 'Consultar/Modificar Información'. The 'Agregar una Sala o Área' tab is active, displaying a form with the following elements: a printer icon and the text '¿Cuál es el ID de la Sala?' next to a text input field containing 'sala17'; a 'Crear un ID' button; the text 'Nombre de la Sala' next to a text input field containing 'Sala Nueva1'; and two buttons at the bottom: 'Cancelar' and 'Guardar'.

Si se guardo con éxito tendremos el siguiente mensaje:

### Salas: Guardar una Nueva Sala

Se guardó con Éxito

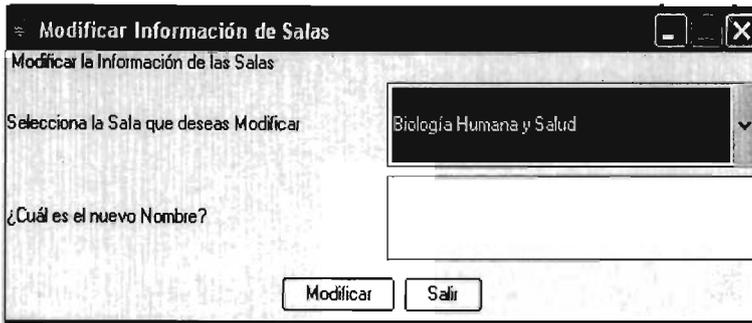
Si no queremos crear ninguna sala damos clic en Cancelar.

### ii. Consultar/Modificar Información.

Al dar clic en la subficha Consultar/Modificar Información tenemos lo siguiente:

The screenshot shows the 'Consultar/Modificar Información' sub-tab selected. It contains two instructions with corresponding buttons: 'Haz click en el Botón para hacer MODIFICACIONES o ACTUALIZACIONES en la Información' with an 'Ir a Modificar' button, and 'Haz click en el Botón para CONSULTAR la Información' with a 'Consultar' button.

Si deseamos hacer modificaciones o actualizaciones a alguna maquina damos clic en el botón **Ir a Modificar**. Se vera una pantalla como esta:



Modificar Información de Salas

Modificar la Información de las Salas

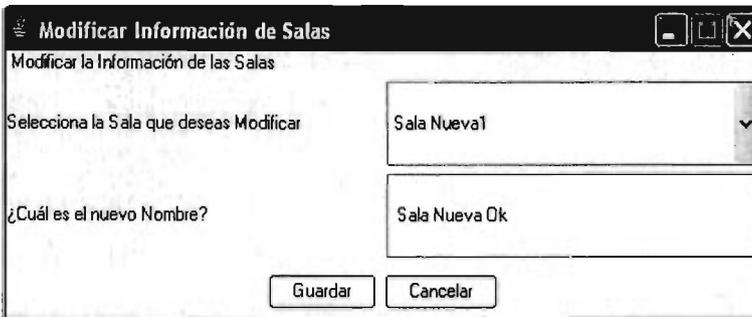
Selecciona la Sala que deseas Modificar

Biología Humana y Salud

¿Cuál es el nuevo Nombre?

Modificar Salir

Supongamos que queremos modificar el nombre de la Sala que creamos con el nombre "Sala Nueva 1" por "Sala Nueva OK". Buscamos el nombre de la sala inicial de la lista una vez seleccionada damos clic en el botón Modificar y cambiamos el nuevo nombre en la caja de texto tal como se muestra:



Modificar Información de Salas

Modificar la Información de las Salas

Selecciona la Sala que deseas Modificar

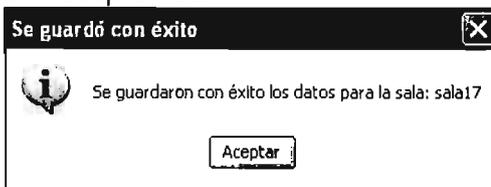
Sala Nueva1

¿Cuál es el nuevo Nombre?

Sala Nueva Ok

Guardar Cancelar

Damos clic en el botón Guardar y la Sala ha cambiado de nombre. Si no hubo ningún problema tiene que aparecer el siguiente mensaje el cual aceptamos.

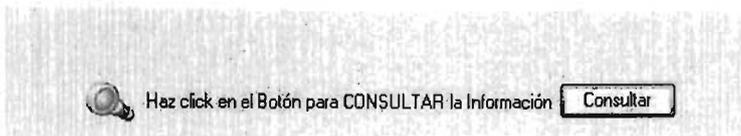


Se guardó con éxito

Se guardaron con éxito los datos para la sala: sala17

Aceptar

Regresaremos a la subficha Consultar/Modificar Información. Ahora si solo queremos ver la información de todas las salas contenidas en el sistema con su id de sala y su nombre damos clic en el botón Consultar



Aparecerá una nueva pantalla con los datos tal como se muestra:

Resultados de La Consulta	
id_sala	nombre_de_sala
sala5	Biología Humana y Salud
sala4	Agricultura y Alimentación
sala3	Universo
sala1	Matemáticas X
sala2	Biodiversidad
sala6	Conciencia de Nuestra Nación
sala7	Energía
sala8	Espacio Infantil
sala9	Estructura de la Materia
sala10	Infraestructura de Nuestra Nación
sala11	Química
sala12	Tecnología Satelital
sala13	Una Balse en el Tiempo
sala15	x34
sala14	X32
sala16	Sala 3D
sala17	Sala Nueva Ok

## II. e) Horas Entrada

En esta ficha (Horas de Entrada) tenemos las subfichas:

- i. Agregar Nueva Hora.
- ii. Consultar/Modificar Información.

### i. Agregar Nueva Hora

Al igual que cuando agregamos las salas dentro del sistema podemos agregar las horas

The screenshot shows the 'Agregar Nueva Hora' form. At the top, there are navigation tabs: Inicio, Asistencia, Becarios, Salas, Horas Entrada, and Sabana. Below the tabs, there are two sub-tabs: 'Agregar Nueva Hora' (selected) and 'Consultar / Modificar Información'. The form contains a text input field labeled '¿Cuál es el ID del Horario?' with a 'Crear un ID' button to its right. Below this is a dropdown menu labeled 'Selecciona una Hora' with '13:00' selected. At the bottom, there are two buttons: 'Cancelar' and 'Nuevo'.

Para agregar una hora solo damos clic en Nuevo, se liberan los botones de esta pantalla entre ellos el botón Crear un ID como se muestra:

The screenshot shows the 'Agregar Nueva Hora' form. The 'Crear un ID' button is now disabled. The 'Selecciona una Hora' dropdown menu now shows '9:00'. The 'Nuevo' button is highlighted, indicating it is the active action.

Damos clic en el Botón crear un ID y después seleccionamos una hora de la lista en nuestro caso serán las 13:00 hrs., el Id se genera automáticamente Tal como se ilustra a continuación:

The screenshot shows the 'Agregar Nueva Hora' form. The 'Crear un ID' button is now active. The 'Selecciona una Hora' dropdown menu now shows '13:00'. The 'Guardar' button is highlighted, indicating it is the active action.

Damos clic en el botón Guardar y listo se ha guardado nuestra Nueva Hora.

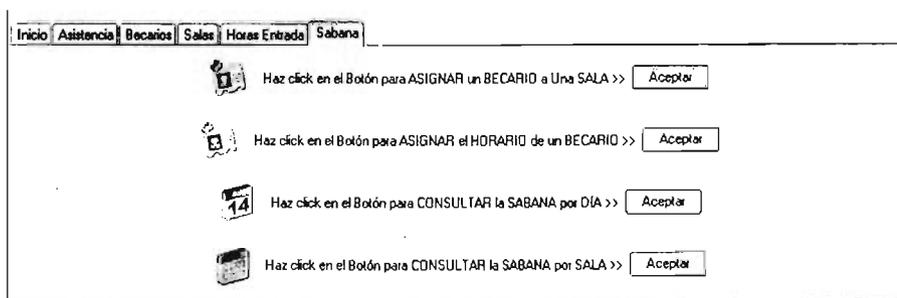
## ii. Consultar/ Modificar Información

Realiza las mismas funciones que en la ficha sala, puedes consultar los horarios así como modificar la información.

## II. f) Sabana

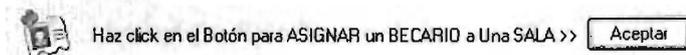
En la ficha Sabana se realiza la tarea principal de este sistema. Tenemos las siguientes opciones:

- i. Asignar un Becario a una Sala.
- ii. Asignar el Horario a un Becario.
- iii. Consultar la Sábana por Día.
- iv. Consultar la Sábana por Sala.

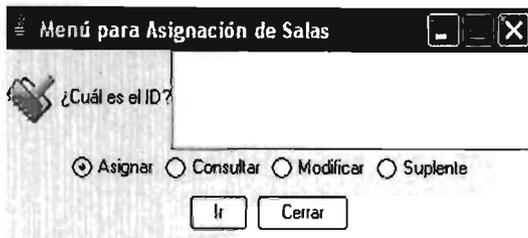


### i. Asignar un Becario a una Sala.

Para asignar un Becario a una sala damos clic en el botón Aceptar de esta opción:



Saldrá una nueva pantalla como la siguiente:

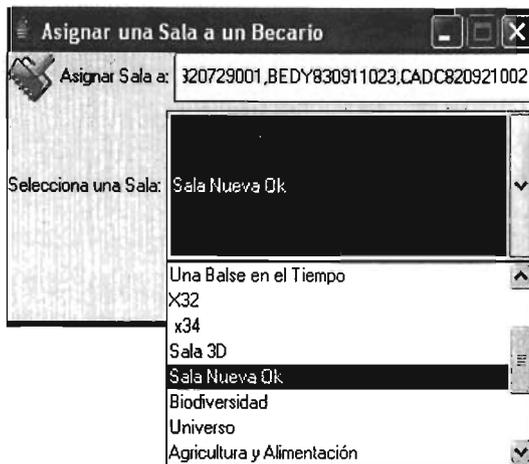


En donde podremos teclear el id del Becario para Asignar, consultar, modificar, o tenerlo como suplente en otra sala; esto gracias a las opciones que tenemos.

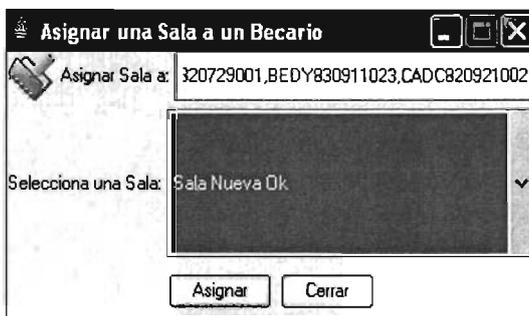
Si lo que queremos es Asignar un Becario o mas a alguna sala solo tecleamos los id de los Becarios separados por comas como se ilustra.



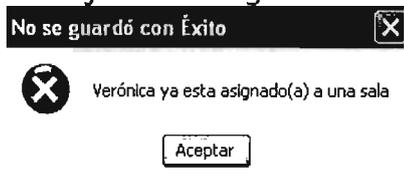
Damos clic en Ir para realizar la Asignación, aparece una pantalla con una lista de las salas contenidas en el sistema para seleccionar la que deseamos en nuestro caso será la sala "Sala Nueva OK" en esta sala quedarán asignados estos tres Becarios. Esto se muestra a continuación:



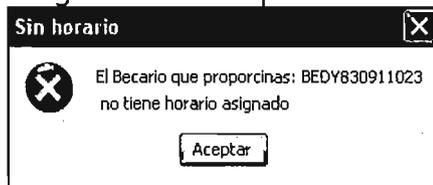
Seleccionamos la Sala y damos clic en el botón Asignar.



Debido a que el becario ya esta asignado a una Sala nos mostrará un mensaje como el siguiente:



El siguiente becario presenta otro error:



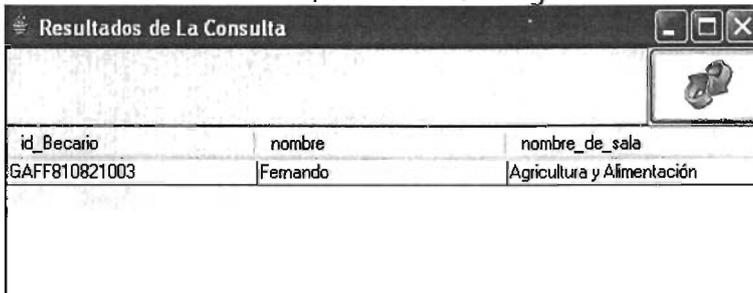
Este becario no tiene un horario asignado.

Si los Id de los becarios no están asignados a ninguna sala y tienen un horario asignado entonces la asignación se guardará.

Si lo que queremos es realizar una consulta, damos la opción de consulta en la misma ventana tal como se ilustra e introducimos el Id del Becario que deseamos buscar para saber a que sala esta asignado:



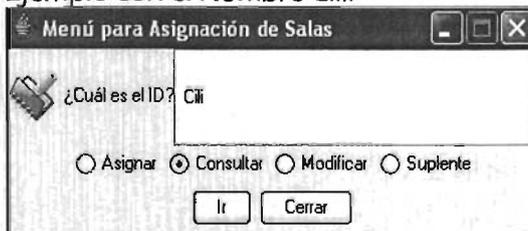
Ponemos el Id en la caja de texto con la Opción de Consultar y luego damos clic en el botón Ir, se mostrará lo siguiente:

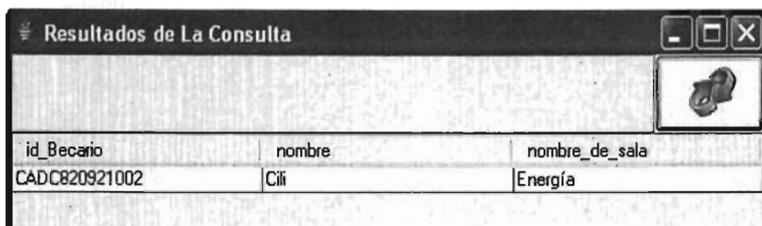


id_Becario	nombre	nombre_de_sala
GAFF810821003	Fernando	Agricultura y Alimentación

También podemos escribir el nombre del Becario o el Apellido Paterno y dar clic en Ir tal como se muestra, realiza la consulta y nos dice a que sala esta asignado este Becario. Esto se ilustra a continuación:

Ejemplo con el Nombre Cili:

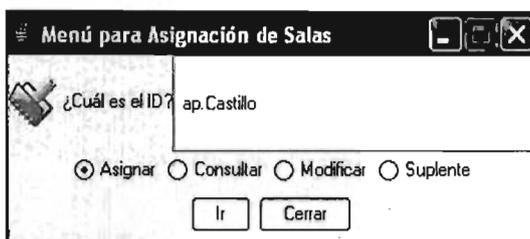




id_Becario	nombre	nombre_de_sala
CADC820921002	Cili	Energía

Ejemplo con el apellido Paterno Castillo:

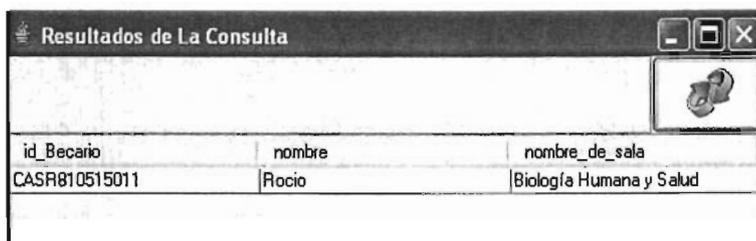
Para que se distinga de un nombre a un apellido solo tenemos que teclear antes ap. Y después el apellido del Becario tal como se muestra:



Menú para Asignación de Salas

¿Cuál es el ID? ap.Castillo

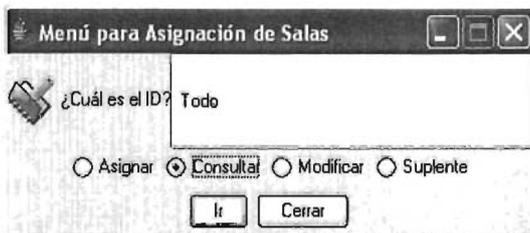
Asignar  Consultar  Modificar  Suplente



id_Becario	nombre	nombre_de_sala
CASR810515011	Rocio	Biología Humana y Salud

Si deseamos ver a todos los becarios que estén asignados a una sala simplemente tecleamos en la caja de texto la instrucción TODO se verá lo siguiente:

Ejemplo con la Instrucción Todo:



Menú para Asignación de Salas

¿Cuál es el ID? Todo

Asignar  Consultar  Modificar  Suplente

**Resultados de La Consulta**



id_Becario	nombre	nombre_de_sala
CIRP790719024	Paulina	Biodiversidad
BAOV820729001	Verónica	Biología Humana y Salud
SAAM840310021	Miguel	Universo
CADC820921002	Ciñ	Energía
GAFF810821003	Fernando	Agricultura y Alimentación
LDMC810816004	Christian Eugenio	Matemáticas X
RORR820813020	Karla Marlin	Universo
GOMK800415005	Karla	Infraestructura de Nuestra Nación
MAPA800530006	Ariana	Matemáticas X
NUAM830514007	Mariana	Infraestructura de Nuestra Nación
RACG810924009	Gabriel	Conciencia de Nuestra Nación
HEAL791006010	Lorena	Conciencia de Nuestra Nación
CASR810515011	Rocio	Biología Humana y Salud
CHCP830417013	Paola	Matemáticas X
FERJ790804014	José	Biología Humana y Salud
FLOK910315015	Karina	Infraestructura de Nuestra Nación
GABM821030016	Marlene	Infraestructura de Nuestra Nación
JASD800929017	Diana Lisbeth	Tecnología Satelital
MOFS800924018	Siyuly	Tecnología Satelital
NALR800627019	Roxana Michel	Tecnología Satelital
ORRM810509008	María de Lourdes	Matemáticas X

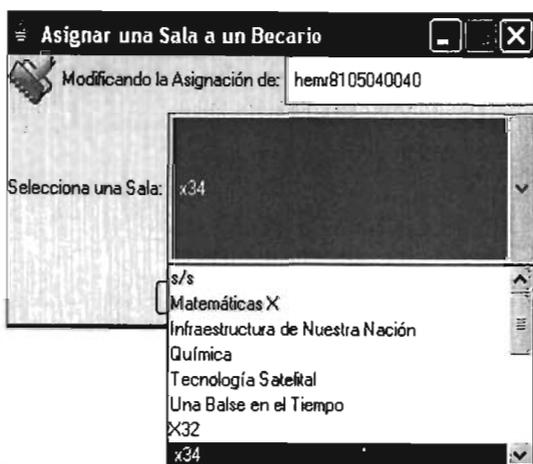
Si lo que queremos es modificar La asignación de un becario ponemos su Id en la caja de texto y seleccionamos la Opción Modificar después damos clic en el botón Ir; como se muestra:

**Menú para Asignación de Salas**

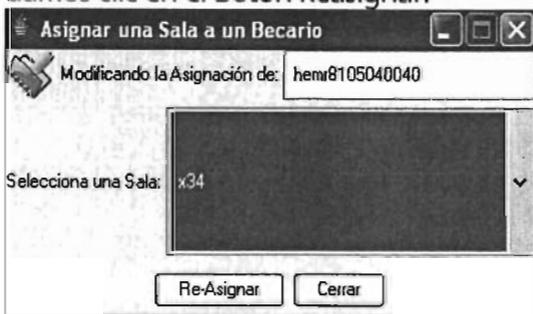
¿Cuál es el ID?

Asignar
  Consultar
  Modificar
  Suplente

Se despliega la siguiente pantalla:



En donde seleccionamos la Sala a la cual se cambiara el Becario. Y damos clic en el Botón Reasignar:



Si el becario esta registrado nos mostrará un mensaje en donde el becario fue Re-Asignado, de lo contrario nos dirá que el Becario no esta registrado en la Base de datos.

Supongamos que deseamos reasignar al Becario con Id:

CHCP830417013 Paola Matemáticas X

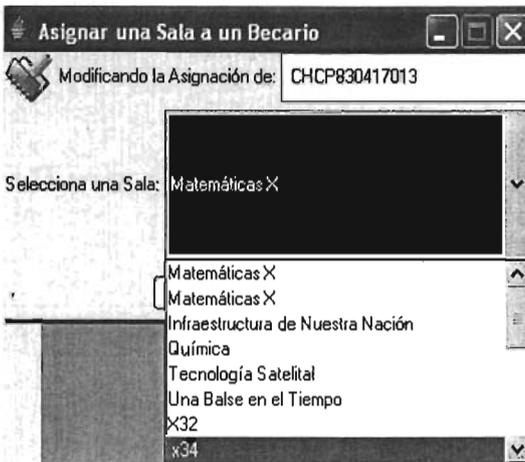
Que esta asignada a la Sala de Matemáticas, queremos asignarla a la Sala x34, el procedimiento es el siguiente:

1. Tecleamos el Id del Becario
2. Seleccionamos la Opción Modificar
3. Damos clic en el botón Ir

Como se muestra a continuación:

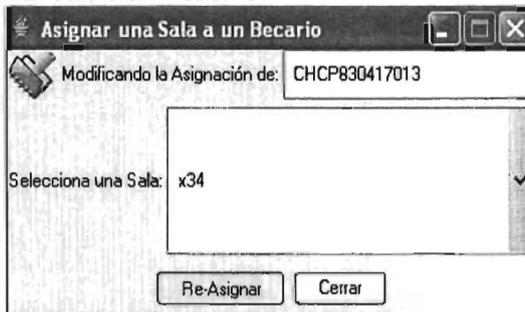


Se despliega la siguiente pantalla:

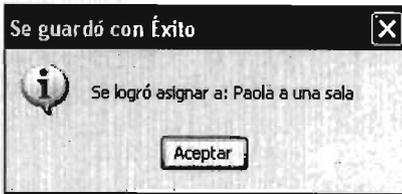


4. Seleccionamos de la lista de Salas, la sala a donde será reasignado este Becario, en nuestro caso es x34, y damos clic en el botón Re-Asignar.

Tal como se observa.



Después nos mostrará un mensaje como el siguiente:



Damos clic en Aceptar

Si deseamos Asignar un Becario como Becario suplente lo que tenemos que hacer es lo siguiente:

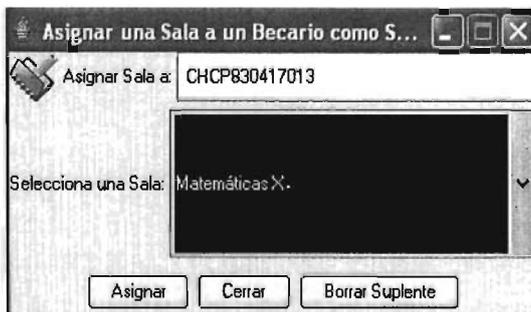
1. Teclar el Id del Becario
2. Seleccionar la opción Suplente
3. Dar clic en Ir

Tal como se muestra:

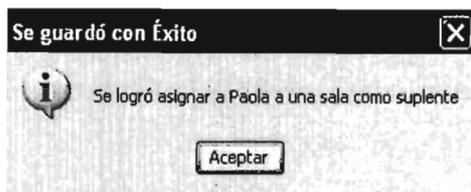


Supongamos que se quiere asignar como suplente a la Sala de Matemáticas X entonces:

4. Seleccionamos la Sala
5. Damos clic en el Botón Asignar

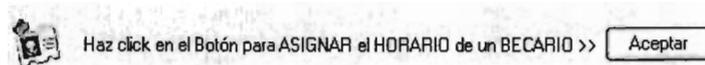


Si se asigno correctamente tendremos un mensaje como el siguiente:



Damos clic en Aceptar.

## ii. Asignar el Horario de un Becario



Para asignar el Horario de un Becario damos clic en el botón Aceptar. Se muestra la siguiente pantalla en donde tecleamos el Id del Becario para Asignar el Horario.



Una vez teclado el Id de Becario y Con la opción de Asignar damos clic en el botón Ir, aparece la siguiente pantalla.

Asignar Horario a un Becario

Asignar el Horario a: HEMIR810504040 Cambiar

HORA	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES	SABADO	DOMINGO
9:00 a 10:00	<input checked="" type="checkbox"/> L	<input checked="" type="checkbox"/> Ma	<input checked="" type="checkbox"/> Mi	<input type="checkbox"/> J	<input type="checkbox"/> V	<input type="checkbox"/> S	<input type="checkbox"/> D
10:00 a 11:00	<input checked="" type="checkbox"/> L	<input checked="" type="checkbox"/> Ma	<input checked="" type="checkbox"/> Mi	<input type="checkbox"/> J	<input type="checkbox"/> V	<input type="checkbox"/> S	<input type="checkbox"/> D
11:00 a 12:00	<input checked="" type="checkbox"/> L	<input type="checkbox"/> Ma	<input checked="" type="checkbox"/> Mi	<input type="checkbox"/> J	<input type="checkbox"/> V	<input type="checkbox"/> S	<input type="checkbox"/> D
12:00 a 13:00	<input checked="" type="checkbox"/> L	<input checked="" type="checkbox"/> Ma	<input checked="" type="checkbox"/> Mi	<input type="checkbox"/> J	<input type="checkbox"/> V	<input type="checkbox"/> S	<input type="checkbox"/> D
13:00 a 14:00	<input type="checkbox"/> L	<input type="checkbox"/> Ma	<input type="checkbox"/> Mi	<input type="checkbox"/> J	<input type="checkbox"/> V	<input type="checkbox"/> S	<input type="checkbox"/> D
14:00 a 15:00	<input type="checkbox"/> L	<input type="checkbox"/> Ma	<input type="checkbox"/> Mi	<input type="checkbox"/> J	<input type="checkbox"/> V	<input type="checkbox"/> S	<input type="checkbox"/> D
15:00 a 16:00	<input type="checkbox"/> L	<input type="checkbox"/> Ma	<input type="checkbox"/> Mi	<input type="checkbox"/> J	<input type="checkbox"/> V	<input type="checkbox"/> S	<input type="checkbox"/> D
16:00 a 17:00	<input type="checkbox"/> L	<input type="checkbox"/> Ma	<input type="checkbox"/> Mi	<input type="checkbox"/> J	<input type="checkbox"/> V	<input type="checkbox"/> S	<input type="checkbox"/> D
17:00 a 18:00	<input type="checkbox"/> L	<input type="checkbox"/> Ma	<input type="checkbox"/> Mi	<input type="checkbox"/> J	<input type="checkbox"/> V	<input type="checkbox"/> S	<input type="checkbox"/> D

Con solo dar clic en las casillas de verificación Asignamos el horario que este Becario va a cubrir. Después damos clic en Asignar. El Becario será asignado a este horario.

Si queremos consultar la información del Horario de un Becario hacemos lo siguiente:

1. Tecleamos el Id de Becario
2. Seleccionamos la opción Consultar
3. Damos clic en Ir

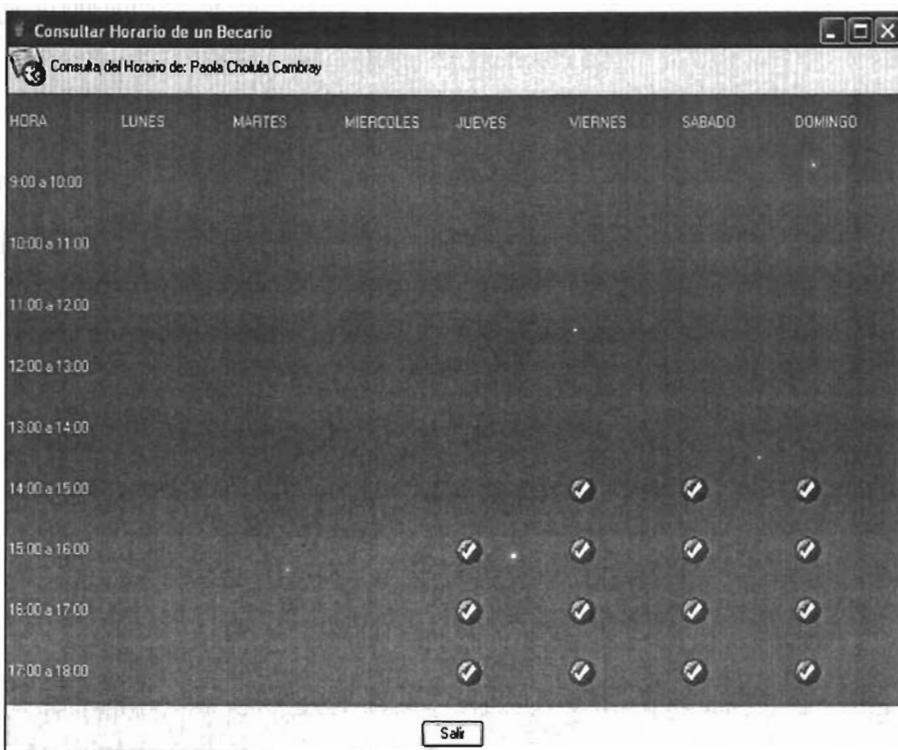
Tal como se ilustra

Menú para el Horario de un Becario

¿Cuál es el ID? CHCP830417013

Asignar
  Consultar
  Modificar
  Suplente

Se mostrará lo siguiente:



Consulta del Horario de: Paola Cholula Cambay

HORA	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES	SABADO	DOMINGO
9:00 a 10:00							
10:00 a 11:00							
11:00 a 12:00							
12:00 a 13:00							
13:00 a 14:00							
14:00 a 15:00					<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
15:00 a 16:00				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
16:00 a 17:00				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
17:00 a 18:00				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

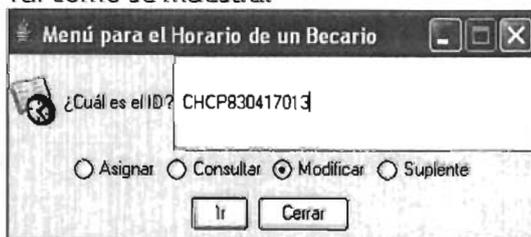
Salir

Como se observa se muestra el Horario que este Becario cumple. Para Regresar Damos clic en el botón Salir.

Para Modificar el Horario de un Becario hacemos lo siguiente:

1. Tecleamos el Id del Becario
2. Seleccionamos la opción Modificar
3. Damos clic en Ir

Tal como se muestra:



Menú para el Horario de un Becario

¿Cuál es el ID?

Asignar  Consultar  Modificar  Suplente

Se despliega la siguiente pantalla:  
En donde tenemos el horario del Becario

Modificar Horario a un Becario

Modificando el Horario de: Paola Cholula Cambray

HORA	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES	SABADO	DOMINGO
9:00 a 10:00	<input type="checkbox"/> L	<input type="checkbox"/> Ma	<input type="checkbox"/> Mi	<input type="checkbox"/> J	<input type="checkbox"/> V	<input type="checkbox"/> S	<input type="checkbox"/> D
10:00 a 11:00	<input type="checkbox"/> L	<input type="checkbox"/> Ma	<input type="checkbox"/> Mi	<input type="checkbox"/> J	<input type="checkbox"/> V	<input type="checkbox"/> S	<input type="checkbox"/> D
11:00 a 12:00	<input type="checkbox"/> L	<input type="checkbox"/> Ma	<input type="checkbox"/> Mi	<input type="checkbox"/> J	<input type="checkbox"/> V	<input type="checkbox"/> S	<input type="checkbox"/> D
12:00 a 13:00	<input type="checkbox"/> L	<input type="checkbox"/> Ma	<input type="checkbox"/> Mi	<input type="checkbox"/> J	<input type="checkbox"/> V	<input type="checkbox"/> S	<input type="checkbox"/> D
13:00 a 14:00	<input type="checkbox"/> L	<input type="checkbox"/> Ma	<input type="checkbox"/> Mi	<input type="checkbox"/> J	<input type="checkbox"/> V	<input type="checkbox"/> S	<input type="checkbox"/> D
14:00 a 15:00	<input type="checkbox"/> L	<input type="checkbox"/> Ma	<input type="checkbox"/> Mi	<input type="checkbox"/> J	<input checked="" type="checkbox"/> V	<input checked="" type="checkbox"/> S	<input checked="" type="checkbox"/> D
15:00 a 16:00	<input type="checkbox"/> L	<input type="checkbox"/> Ma	<input type="checkbox"/> Mi	<input checked="" type="checkbox"/> J	<input checked="" type="checkbox"/> V	<input checked="" type="checkbox"/> S	<input checked="" type="checkbox"/> D
16:00 a 17:00	<input type="checkbox"/> L	<input type="checkbox"/> Ma	<input type="checkbox"/> Mi	<input checked="" type="checkbox"/> J	<input checked="" type="checkbox"/> V	<input checked="" type="checkbox"/> S	<input checked="" type="checkbox"/> D
17:00 a 18:00	<input type="checkbox"/> L	<input type="checkbox"/> Ma	<input type="checkbox"/> Mi	<input checked="" type="checkbox"/> J	<input checked="" type="checkbox"/> V	<input checked="" type="checkbox"/> S	<input checked="" type="checkbox"/> D

Modificar Cancelar

Para modificar el horario seleccionamos la hora nueva y el día y hacemos clic en el botón Modificar. El nuevo horario del Becario se guarda.

Para cambiar el horario de un Becario suplente hacemos lo siguiente:

1. Teclamos el Id del Becario
2. Seleccionamos la Opción Suplente
3. Damos clic en el botón Ir

Tal como se muestra

Menú para el Horario de un Becario

¿Cuál es el ID? CHCP930417013

Asignar
  Consultar
  Modificar
  Suplente

Se despliega la siguiente pantalla.

Asignar Horario de Suplente a un Becario

Asignar el Horario de Suplente a: CHCP930417013

HORA	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES	SABADO	DOMINGO
9:00 a 10:00	<input type="checkbox"/> L	<input type="checkbox"/> Ma	<input type="checkbox"/> Mi	<input type="checkbox"/> J	<input type="checkbox"/> V	<input type="checkbox"/> S	<input type="checkbox"/> D
10:00 a 11:00	<input type="checkbox"/> L	<input type="checkbox"/> Ma	<input type="checkbox"/> Mi	<input type="checkbox"/> J	<input type="checkbox"/> V	<input type="checkbox"/> S	<input type="checkbox"/> D
11:00 a 12:00	<input type="checkbox"/> L	<input type="checkbox"/> Ma	<input type="checkbox"/> Mi	<input type="checkbox"/> J	<input type="checkbox"/> V	<input type="checkbox"/> S	<input type="checkbox"/> D
12:00 a 13:00	<input type="checkbox"/> L	<input type="checkbox"/> Ma	<input type="checkbox"/> Mi	<input type="checkbox"/> J	<input type="checkbox"/> V	<input type="checkbox"/> S	<input type="checkbox"/> D
13:00 a 14:00	<input type="checkbox"/> L	<input type="checkbox"/> Ma	<input type="checkbox"/> Mi	<input type="checkbox"/> J	<input type="checkbox"/> V	<input type="checkbox"/> S	<input type="checkbox"/> D
14:00 a 15:00	<input type="checkbox"/> L	<input type="checkbox"/> Ma	<input type="checkbox"/> Mi	<input type="checkbox"/> J	<input type="checkbox"/> V	<input type="checkbox"/> S	<input type="checkbox"/> D
15:00 a 16:00	<input type="checkbox"/> L	<input type="checkbox"/> Ma	<input type="checkbox"/> Mi	<input type="checkbox"/> J	<input type="checkbox"/> V	<input type="checkbox"/> S	<input type="checkbox"/> D
16:00 a 17:00	<input type="checkbox"/> L	<input type="checkbox"/> Ma	<input type="checkbox"/> Mi	<input type="checkbox"/> J	<input type="checkbox"/> V	<input type="checkbox"/> S	<input type="checkbox"/> D
17:00 a 18:00	<input type="checkbox"/> L	<input type="checkbox"/> Ma	<input type="checkbox"/> Mi	<input type="checkbox"/> J	<input type="checkbox"/> V	<input type="checkbox"/> S	<input type="checkbox"/> D

Aquí seleccionamos el horario que deseamos para el Becario Suplente haciendo clic en cada casilla que necesitamos, después damos clic en el botón Asignar. La información se guarda.

Para borrar el Horario de Suplente damos clic en el botón con el mismo nombre, la pantalla se limpia.

### iii. Consultar la Sábana Por Día



Haz click en el Botón para CONSULTAR la SABANA por DÍA >>

Para realizar esta consulta damos clic en el botón Aceptar  
Se despliega la siguiente ventana:

Aquí seleccionamos de la lista el Día para generar la Sábana por Día.

Por ejemplo:

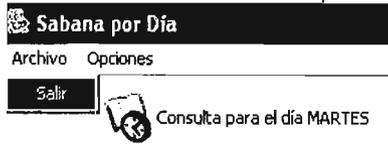
Vamos a seleccionar el día Martes de la lista, enseguida damos clic en el botón Ver. Se genera la Sábana:

HORA / DÍA	Matemáticas X	Infraestructura de Nuestra Nación	Química	Tecnología Satelital	Una Base en el Tiempo
9:00 a 10:00	Christian Eugenio				
10:00 a 11:00	Christian Eugenio				
11:00 a 12:00	Christian Eugenio				
12:00 a 13:00	Christian Eugenio				
13:00 a 14:00					
14:00 a 15:00	Ariana	Karla Mariana		Diana Lisbeth Syruy Rosana Michel	
15:00 a 16:00	Ariana	Karla Mariana		Diana Lisbeth Syruy Rosana Michel	

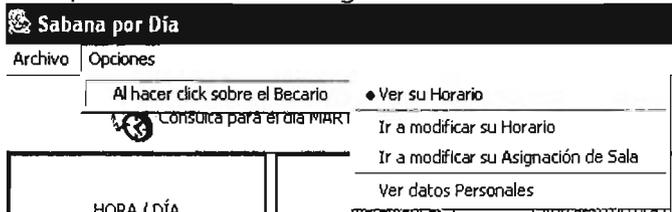
En cada nombre de los Becarios tenemos un botón  
Como se observa en la ventana anterior tenemos en la barra de menús dos opciones:

1. Archivo
2. Opciones

En Archivo tenemos la Opción Salir la cual cierra esta ventana

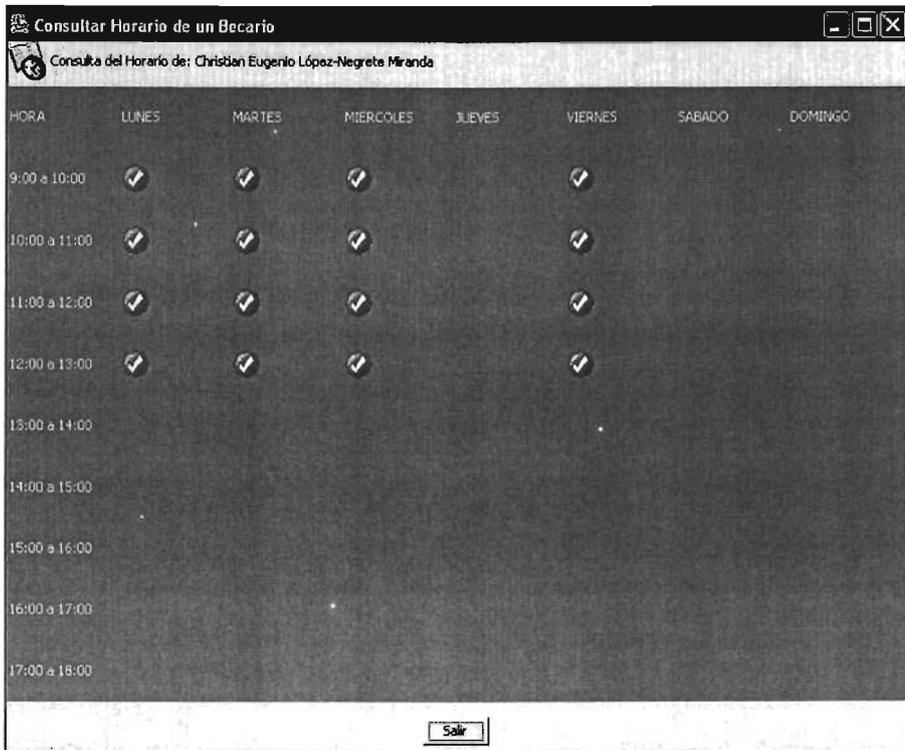


En Opciones tenemos lo siguiente:



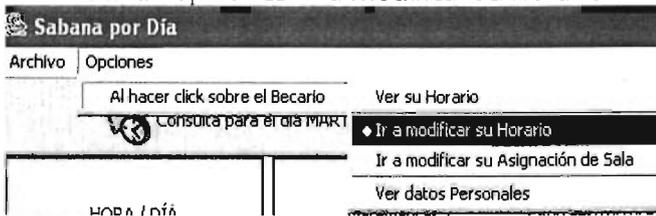
Cada nombre de Becario aparece como un botón esto es por si queremos Ver su horario, Modificar su horario, Modificar su Asignación a la Sala o si queremos Ver los datos Personales del Becario.

Supongamos que deseamos ver el horario de un Becario en el Menú de opciones seleccionamos Ver su Horario y damos clic en el botón del becario, esto se muestra a continuación:



Damos clic en el botón Salir para regresar.

Si lo que queremos es modificar el horario del Becario seleccionamos del menú de opciones "Ir a modificar su horario".



Después damos clic sobre el botón con el nombre del Becario.

Modificar Horario a un Becario

Modificando el Horario de: Christian Eugenio López-Negrete Miranda

HORA	LUNES	MARTES	MIERCOLES	JUEVES	VIERNES	SABADO	DOMINGO
9:00 a 10:00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Ma	<input checked="" type="checkbox"/> Mi	<input type="checkbox"/> J	<input checked="" type="checkbox"/> V	<input type="checkbox"/> S	<input type="checkbox"/> D
10:00 a 11:00	<input checked="" type="checkbox"/> L	<input checked="" type="checkbox"/> Ma	<input checked="" type="checkbox"/> Mi	<input type="checkbox"/> J	<input checked="" type="checkbox"/> V	<input type="checkbox"/> S	<input type="checkbox"/> D
11:00 a 12:00	<input checked="" type="checkbox"/> L	<input checked="" type="checkbox"/> Ma	<input checked="" type="checkbox"/> Mi	<input type="checkbox"/> J	<input checked="" type="checkbox"/> V	<input type="checkbox"/> S	<input type="checkbox"/> D
12:00 a 13:00	<input checked="" type="checkbox"/> L	<input checked="" type="checkbox"/> Ma	<input checked="" type="checkbox"/> Mi	<input type="checkbox"/> J	<input checked="" type="checkbox"/> V	<input type="checkbox"/> S	<input type="checkbox"/> D
13:00 a 14:00	<input type="checkbox"/> L	<input type="checkbox"/> Ma	<input type="checkbox"/> Mi	<input type="checkbox"/> J	<input type="checkbox"/> V	<input type="checkbox"/> S	<input type="checkbox"/> D
14:00 a 15:00	<input type="checkbox"/> L	<input type="checkbox"/> Ma	<input type="checkbox"/> Mi	<input type="checkbox"/> J	<input type="checkbox"/> V	<input type="checkbox"/> S	<input type="checkbox"/> D
15:00 a 16:00	<input type="checkbox"/> L	<input type="checkbox"/> Ma	<input type="checkbox"/> Mi	<input type="checkbox"/> J	<input type="checkbox"/> V	<input type="checkbox"/> S	<input type="checkbox"/> D
16:00 a 17:00	<input type="checkbox"/> L	<input type="checkbox"/> Ma	<input type="checkbox"/> Mi	<input type="checkbox"/> J	<input type="checkbox"/> V	<input type="checkbox"/> S	<input type="checkbox"/> D
17:00 a 18:00	<input type="checkbox"/> L	<input type="checkbox"/> Ma	<input type="checkbox"/> Mi	<input type="checkbox"/> J	<input type="checkbox"/> V	<input type="checkbox"/> S	<input type="checkbox"/> D

Modificar Cancelar

Damos clic en el botón Modificar y Listo.

Si lo que deseamos es modificar la asignación de Sala de un Becario tenemos que seleccionar del menú de Opciones "Ir a Modificar su asignación de Sala".

Sabana por Día

Archivo Opciones

Al hacer click sobre el Becario  Ver su Horario

Consulta para el día MARTES  Ir a modificar su Horario

**Ir a modificar su Asignación de Sala**

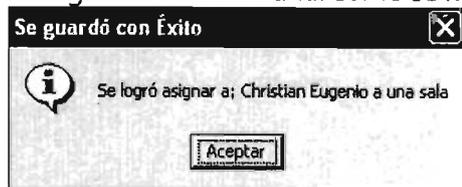
Ver datos Personales

HORA / DÍA

Después damos clic sobre el botón con el nombre del Becario. Aparecerá la siguiente pantalla:

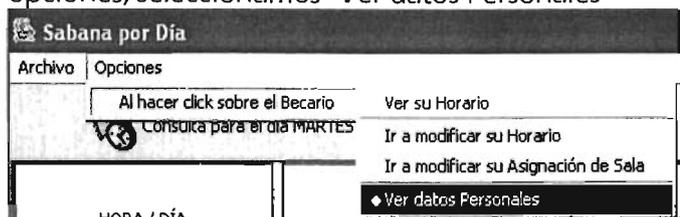


Aquí tenemos el id del Becario seleccionamos la sala a la cual será reasignado el Becario damos clic en el botón Re-Asignar y el Becario es reasignado a otra sala tal como se muestra:

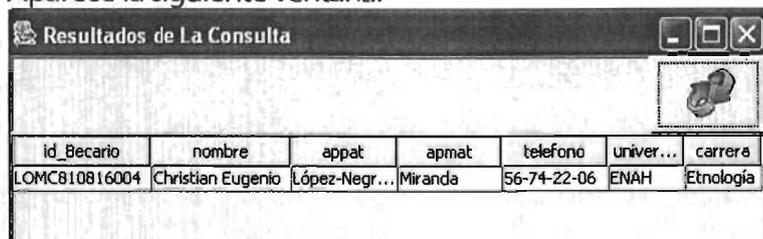


Damos clic en Aceptar.

Si queremos ver los datos personales del Becario, vamos al menú de opciones, seleccionamos "Ver datos Personales"



Después damos clic en el botón con el nombre del Becario. Aparece la siguiente ventana:



Aquí tenemos los datos personales del Becario seleccionado, esto se puede hacer con cualquier Becario que este en la Sábana Seleccionada.

## iv. Consultar Sabana por Sala



Haz click en el Botón para CONSULTAR la SABANA por SALA >>

Aceptar

Solo damos clic en la Ficha Salas y elegimos la opción Consultar la Sábana por Sala y damos clic en el botón Aceptar. Se despliega la siguiente ventana:

Aquí seleccionamos una de las salas de la lista y damos clic en el botón Ver para que nos muestre la Sábana por Sala. Tal como se ilustra a continuación:

Sabana por Sala

Archivo Opciones

Consulta para la Sala Matemáticas X

HORA / DÍA	LUNES	MARTES	MIERCOLES	JUEVES	VIERNES	SABADO	DOMINGO
9:00 a 10:00	Christian Eugenio	Christian Eugenio	Christian Eugenio María de Lourdes		Christian Eugenio	María de Lourdes	María de Lourdes
10:00 a 11:00	Christian Eugenio	Christian Eugenio	Christian Eugenio María de Lourdes		Christian Eugenio	María de Lourdes	María de Lourdes
11:00 a 12:00	Christian Eugenio	Christian Eugenio	Christian Eugenio María de Lourdes		Christian Eugenio	María de Lourdes	María de Lourdes
12:00 a 13:00	Christian Eugenio	Christian Eugenio	Christian Eugenio María de Lourdes		Christian Eugenio	María de Lourdes	María de Lourdes
13:00 a 14:00						María de Lourdes	María de Lourdes
14:00 a 15:00	Ariana	Ariana		Ariana María de Lourdes	Ariana		
15:00 a 16:00	Ariana	Ariana		Ariana María de Lourdes	Ariana		
16:00 a 17:00	Ariana	Ariana		Ariana María de Lourdes	Ariana		
17:00 a 18:00	Ariana	Ariana		Ariana María de Lourdes	Ariana		

Terminar

Aquí se genera toda la Sábana para la sala seleccionada, además tenemos las mismas opciones que en Sabana por Día en el menú de opciones como se muestra en la parte superior de la siguiente ventana:

The screenshot shows a window titled "Sabana por Sala" with a menu bar containing "Archivo" and "Opciones". A dropdown menu is open under "Opciones", listing: "Al hacer clic sobre el Becario", "Consultar para la sala marcada", "Ver su Horario", "Ir a modificar su Horario", "Ir a modificar su Adscripción de Sala", and "Ver datos Personales".

HORA / DÍA	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES	SABADO	DOMINGO
9:00 a 10:00	Christian Eugenio	Christian Eugenio	Christian Eugenio María de Lourdes		Christian Eugenio	María de Lourdes	María de Lourdes
10:00 a 11:00	Christian Eugenio	Christian Eugenio	Christian Eugenio María de Lourdes		Christian Eugenio	María de Lourdes	María de Lourdes
11:00 a 12:00	Christian Eugenio	Christian Eugenio	Christian Eugenio María de Lourdes		Christian Eugenio	María de Lourdes	María de Lourdes
12:00 a 13:00	Christian Eugenio	Christian Eugenio	Christian Eugenio María de Lourdes		Christian Eugenio	María de Lourdes	María de Lourdes
13:00 a 14:00						María de Lourdes	María de Lourdes
14:00 a 15:00	Ariana	Ariana		Ariana María de Lourdes	Ariana		
15:00 a 16:00	Ariana	Ariana		Ariana María de Lourdes	Ariana		
16:00 a 17:00	Ariana	Ariana		Ariana María de Lourdes	Ariana		
17:00 a 18:00	Ariana	Ariana		Ariana María de Lourdes	Ariana		

At the bottom of the window, there is a "Terminar" button.

# **Glosario de términos**



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**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

## **Apache**

Servidor HTTP de dominio público el cual está basado en el sistema operativo Linux. Fue desarrollado en 1995 y actualmente es uno de los servidores HTTP más utilizados en la red.

## **Applet**

Pequeña aplicación escrita en Java la cual se difunde a través de la red en orden de ejecutarse en el navegador cliente.

# B

## **Bit (Dígito Binario)**

Unidad mínima de almacenamiento de la información cuyo valor puede ser 0 ó 1; o bien verdadero o falso.

## **Bits por Segundo (bps)**

Velocidad a la que se transmiten los bits en un medio de comunicación.

## **Byte**

Conjunto de 8 bits el cual suele representar un valor asignado a un carácter.

## **Byte codes**

Son el resultado de la compilación de un programa Java. Es un código de máquina virtual que es interpretado por el intérprete Java. No es el código máquina directamente entendible por el hardware, pero ya ha pasado todas las fases del compilador: análisis de instrucciones, orden de operadores, etc., y ya tiene generada la pila de ejecución de órdenes.

# C

## **C/ C++**

Lenguajes de programación (orientado a objetos en el caso de C++) utilizados en el WWW a través de un CGI, principalmente para realizar consultas a bases de datos tipo Oracle, SQL-Server, SyBase, etc.; o a herramientas locales como

WAIS. Generalmente el servidor donde se encuentra el programa funciona en ambiente UNIX.

### **Cabecera (Header)**

Parte inicial de un paquete que precede a los datos propiamente dichos y que contiene las direcciones del remitente y del destinatario, control de errores y otros campos. Porción de un mensaje de correo electrónico que precede al mensaje propiamente dicho y contiene, entre otras cosas, el remitente del mensaje, la fecha y la hora.

### **Cache (Copia de Visitas)**

Copia que mantiene un ordenador de las páginas web visitadas últimamente de forma que si el usuario vuelve a solicitarlas, las mismas son leídas desde el disco duro sin necesidad de tener que conectarse de nuevo a la red; consiguiéndose así una mejora muy apreciable del tiempo de respuesta.

### **CGI (Common Gateway Interface)**

Lo que normalmente se conoce por CGI son pequeños códigos de programa que se adaptan al estándar Common Gateway Interface (CGI) mediante el cual se puede acceder a servidores de Internet que envían información a los usuarios. Mediante este sistema la Web se comunica con recursos externos a la misma, como pueden ser bases de datos, y facilita la interacción del internauta directamente con la máquina.

## **D**

### **Dirección de Protocolo de Internet (IP)**

Dirección numérica de una computadora en Internet de forma que cada dirección electrónica se asigna a una computadora conectada a Internet y por lo tanto es única. La dirección IP esta compuesta de cuatro octetos como 132.248.53.10

### **DTD (Document Type Definition.)**

El DTD es una declaración en un documento SGML ó XML que especifica restricciones en la estructura del mismo. El DTD puede ser incluido dentro del archivo del documento, pero normalmente se almacena en un fichero ASCII

de texto separado. La sintaxis de los DTDs para SGML y XML es similar pero no idéntica.

## E

### **Ethernet**

Tipo de red de área local desarrollada en forma conjunta por Xerox, Intel y Digital Equipment. Se apoya en la topología de bus, tiene ancho de banda de 10 Mbps de forma que presenta una elevada velocidad de transmisión; y se ha convertido en un estándar de red corporativa.

### **EBNF (Backus Naun Form Extendido)**

Es una notación formal para describir la sintaxis. Es un Metalenguaje. Un metalenguaje es un lenguaje que es usado para describir otro lenguaje. Usa símbolos como ::=, <, >, | así como \* y +, a estos símbolos se les conoce como metasímbolos.

## F

### **FTP -- File Transfer Protocol**

Protocolo de transferencia de archivos. Se usan programas para FTP como son CuteFTP o LeapFTP para Windows, por ejemplo, que permiten la conexión entre dos computadoras, usando por lo general el puerto 21 para conectarse (aunque se pueden usar otros puertos). Por medio del Protocolo de transferencia de archivos se pueden subir y bajar archivos entre el cliente y el servidor.

## G

### **GNU**

Proyecto creado en 1984 con el fin de desarrollar un sistema operativo tipo Unix según la filosofía del "software libre".

### **GUI**

Graphic user interface (Interfaz grafica para usuario)

# H

## **Htm o Html**

Acrónimo de HyperText Markup Language: Lenguaje de hipertexto que emplea órdenes simples para generar pantallas multimediales (se utiliza .htm o .html como extensión de los archivos escritos en este lenguaje).

## **HTTP**

Herramienta de acceso a sitios Web (Web sites).

# I

## **Intercambio Dinámico de Datos (DDE)**

Conjunto de especificaciones de Microsoft para el intercambio de datos y control de flujo entre aplicaciones.

## **Intercambio Electrónico de Datos (EDI)**

Sistemas y protocolos estandarizados de intercambio de datos a través de la red utilizado principalmente por empresas.

## **IEEE**

Instituto de Ingenieros Electricistas y Electrónicos

## **Interconexión de Sistemas Abiertos (OSI)**

Protocolo en el que se apoya Internet debido a que establece la manera como se realiza la comunicación entre dos computadoras a través de siete capas Física, Datos, Red, Transporte, Sesión, Presentación y Aplicación.

# J

## **Java**

Lenguaje de programación que permite ejecutar programas escritos en un lenguaje muy parecido al C++, llamados applets, a través del WWW. Se diferencia de un CGI en el hecho de que la ejecución es completamente realizada en la computadora cliente, en lugar del servidor. Java fue originalmente desarrollado por Sun Microsystems y su principal objetivo fue crear un lenguaje que fuera capaz de ser ejecutado de una forma segura a

través de Internet. Esta característica requiere la eliminación de muchas construcciones y usos de C y C++; entre los que se destaca la eliminación de punteros. Java no puede acceder arbitrariamente a direcciones de memoria y es un lenguaje compilado en un código llamado "byte-code". Este código es interpretado "en vuelo" por el intérprete Java.

### **JavaScript**

Lenguaje desarrollado por Netscape y aunque es parecido a Java se diferencia de él en que los programas están incorporados en el archivo HTML.

## **L**

### **Linux**

Versión de libre distribución del sistema operativo UNIX el cual tiene todas las características que se pueden esperar de un moderno y flexible UNIX. Incluye multitarea real, memoria virtual, librerías compartidas, dirección y manejo propio de memoria y TCP/IP.

## **M**

### **Macintosh**

Serie de computadoras de Apple Computer cuyo sistema operativo fue el primero totalmente gráfico y basado en ventanas. El entorno es intuitivo, eliminando el teclado de los comandos del sistema de forma que a todos los objetos se le asigna una representación gráfica (iconos).

### **MacOS**

Sistema operativo desarrollado por la empresa Apple Computer para la PC Macintosh, aparecido en 1984.

### **Microsoft**

Compañía creadora de los sistemas operativos Windows 95, Windows NT; de los controles Active X, y desarrolladora del navegador IE del WWW entre otros recursos.

## O

OSX versión última del sistema operativo para Mac OS  
Véase Mac OS

## P

POO (Programación Orientada a Objetos)

## Q

### Quicktime

Formato popular de video el cual puede desplegar películas, sonido y panoramas envolventes de forma que los objetos pueden girar en la pantalla.

## S

### Sistema Operativo (OS)

Programa especial el cual se carga en una computadora tras ser encendido y cuya función es gestionar los demás programas, o aplicaciones, que se ejecutarán en dicho ordenador, como, por ejemplo, un procesador de texto o una hoja de cálculo, o la impresión de un texto en una impresora o una conexión a Internet.

### Sistema Operativo en Disco (MS-DOS)

Fue el primer sistema operativo para computadoras personales el cual se basa en mandatos que se escriben línea por línea y fue desarrollado por Bill Gates para IBM. Se destaca que antes de la aparición de las computadoras personales IBM desarrolló otro DOS para computadoras anteriores.

### SMTP (Protocolo Simple de Transferencia de Correo)

Dicho protocolo es definido en STD 10, RFC 821, y se usa para la transferencia de correo electrónico entre computadoras. Es un protocolo de servidor a servidor, de forma que para poder leer los mensajes se deben utilizar otros protocolos.

## **SNMP**

Acrónimo de Simple Network Management Protocol. Protocolo estándar para la administración de red en Internet. Prácticamente todos los sistemas operativos, routers, switches, módems cable o ADSL módem, firewalls, etc. se ofrecen con este servicio.

## **Solaris**

Es el nombre con el que se conoce el sistema operativo de Sun Microsystems. Originalmente se llamó SunOS, pero posteriormente, debido a la presentación de UNIX Sistema V se desarrolló una nueva versión a la que se le llamó Solaris.

## **SQL**

Lenguaje de Consulta Estructurado (SQL siglas en ingles): Un lenguaje desarrollado por IBM y estandarizado por ANSI para direccionar, crear, actualizar o consultar bases de datos relacionales.

# **W**

## **Windows NT**

Sistema operativo gráfico de 32 bits desarrollado por Microsoft muy similar al Windows, pero con más prestaciones.