

Universidad Autónoma Metropolitana  
Unidad Azcapotzalco

División de Ciencias Básicas e Ingeniería

Licenciatura en Ingeniería en Computación

Proyecto Tecnológico

Marzo de 2014

# “Sistema de información para el control de empleados eventuales”

Reporte Final

Christian Daniel Ibarra Hernández  
208367246  
Trimestre 2014 Invierno

Asesora: Rafaela Blanca Silva López, Profesor Titular, Departamento de Sistemas  
Asesor Externo: Heber Torres y Torres

Yo, Rafaela Blanca Silva López, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.

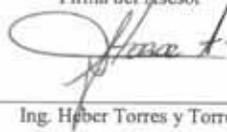
Firma del Asesor



M en C. Rafaela Blanca Silva López

Yo, Heber Torres y Torres, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.

Firma del Asesor



Ing. Heber Torres y Torres

Yo, Christian Daniel Ibarra Hernández, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.

Firma del Asesor



Christian Daniel Ibarra Hernández

## Contenido

<b>A. Objetivos</b> .....	7
1.1. General .....	7
1.2. Particulares .....	7
<b>B. Resumen</b> .....	7
<b>C. Introducción</b> .....	8
<b>D. JUSTIFICACIÓN</b> .....	9
<b>E. Marco Teórico</b> .....	9
<b>F. Desarrollo del proyecto</b> .....	10
<b>G. Casos de Uso</b> .....	11
1. Identificación de los Casos de Uso relevantes para la arquitectura.....	11
1.1. Descripción de los Casos de Uso más relevantes para la arquitectura .....	11
1.1.1. Módulo 1: Acceso al Sistema .....	13
1.1.2. Módulo 2: Características de los empleados eventuales. ....	14
1.1.3. Módulo 3: Registro de asistencia y pagos. ....	16
1.1.4. Módulo 4: Lector de huella digital. ....	17
1.1.5. Módulo 5: Registro de permisos. ....	18
1.1.6. Módulo 6: Reportes. ....	19
<b>H. Estructura de la base de datos</b> .....	19
1. Entidades .....	19
2. Atributos.....	19
3. Diagrama entidad-relación .....	20
4. Diccionario de datos.....	20
5. Esquema Físico de la Base de Datos .....	22
1. <b>TABLA: USUARIOS</b> .....	22
2. <b>TABLA: EVENTO</b> .....	22
3. <b>TABLA: PERMISOS</b> .....	23
4. <b>TABLA: PUESTO</b> .....	23
5. <b>TABLA: EVENTUALES</b> .....	24
6. <b>TABLA: ASISTENCIA</b> .....	25
<b>I. Diagrama de Clases</b> .....	26
<b>J. Navegación</b> .....	27
1. Diagrama de Navegación .....	27

2. Manual de Navegación.....	27
PASO 1: INGRESAR AL SISTEMA .....	27
PASO 2: SELECCIONAR UNA OPCIÓN .....	28
PASO 3: REGISTRAR USUARIO .....	28
PASO 4: ELIMINAR USUARIO .....	29
PASO 5: ACTUALIZAR USUARIO.....	29
PASO 6: MODIFICAR CONTRASEÑA.....	30
PASO 7: REGISTRAR EVENTUAL.....	30
PASO 8: ELIMINAR EVENTUAL.....	31
PASO 9: ACTUALIZAR EVENTUAL.....	31
PASO 10: REGISTRAR ASISTENCIA .....	32
PASO 11: REGISTRAR PERMISOS .....	32
PASO 12: VISUALIZAR E IMPRIMIR REPORTE .....	33
PASO 13: CERRAR SESIÓN .....	33
<b>K. Manual de instalación .....</b>	<b>34</b>
<b>A. SOFTWARE NECESARIO .....</b>	<b>34</b>
<b>B. INSTALACION DE XAMPP 1.8.3.....</b>	<b>34</b>
<b>C. INSTALACION DE ORACLE 11G EXPRESS EDITION.....</b>	<b>43</b>
<b>L. RESULTADOS.....</b>	<b>48</b>
<b>A. FUNCIONES .....</b>	<b>48</b>
<b>B. DESCRIPCION DEL FUNCIONAMIENTO.....</b>	<b>49</b>
<b>C. ESTRUCTURA .....</b>	<b>59</b>
1. PANTALLA PRINCIPAL .....	59
1.1. DESCRIPCIÓN.....	60
1.2. FUNCIONAMIENTO.....	60
2. PANTALLA DE USUARIO.....	60
2.1. DESCRIPCIÓN.....	61
2.2. FUNCIONAMIENTO.....	62
3. PANTALLA ASISTENCIA EVENTUAL Y EVENTO .....	63
3.1. DESCRIPCIÓN.....	63
3.2. FUNCIONAMIENTO.....	63
4. PANTALLA ASISTENCIA EVENTUAL Y EVENTO .....	64
4.1. DESCRIPCIÓN.....	64

4.2.	FUNCIONAMIENTO.....	65
5.	PANTALLA REGISTRAR USUARIO.....	65
5.1.	DESCRIPCIÓN.....	65
5.2.	FUNCIONAMIENTO.....	66
6.	PANTALLA ELIMINAR USUARIO.....	66
6.1.	DESCRIPCIÓN.....	66
6.2.	FUNCIONAMIENTO.....	67
7.	PANTALLA ACTUALIZAR USUARIO.....	67
7.1.	DESCRIPCIÓN.....	67
7.2.	FUNCIONAMIENTO.....	68
8.	PANTALLA REGISTRAR EVENTUAL.....	68
8.1.	DESCRIPCIÓN.....	68
8.2.	FUNCIONAMIENTO.....	69
9.	PANTALLA ELIMINAR EVENTUAL.....	69
9.1.	DESCRIPCIÓN.....	70
9.2.	FUNCIONAMIENTO.....	70
10.	PANTALLA ACTUALIZAR EVENTUAL.....	70
10.1.	DESCRIPCIÓN.....	70
10.2.	FUNCIONAMIENTO.....	71
11.	PANTALLA PERMISOS.....	71
11.1.	DESCRIPCIÓN.....	72
11.2.	FUNCIONAMIENTO.....	72
12.	PANTALLA ASISTENCIA.....	73
12.1.	DESCRIPCIÓN.....	73
12.2.	FUNCIONAMIENTO.....	74
13.	PANTALLA CAMBIAR CONTRASEÑA.....	74
13.1.	DESCRIPCIÓN.....	74
13.2.	FUNCIONAMIENTO.....	74
14.	PANTALLA APPLET.....	75
14.1.	DESCRIPCIÓN.....	75
14.2.	FUNCIONAMIENTO.....	75
<b>M.</b>	<b>Análisis y Discusión de Resultados.....</b>	<b>76</b>
1.	Huella Digital con Lector Digital Persona.....	76

2.	Seguridad del sistema algoritmo de encriptación Blowfish .....	76
<b>N.</b>	<b>Conclusiones</b> .....	77
<b>O.</b>	<b>Bibliografía</b> .....	77
<b>P.</b>	<b>Apéndices</b> .....	78
1.1.	Página Web (HTML5 + PHP + JAVASCRIPT) .....	78
1.1.1.	Conexión a base de datos.....	78
1.1.2.	Funciones SQL .....	79
1.1.3.	Página Index .....	80
1.1.4.	Página Principal .....	82
1.1.5.	Script Validar_usuario .....	83
1.1.6.	Pantalla Usuario .....	84
1.1.7.	Script Seguridad .....	86
1.1.8.	Pantalla Asistencia_eventual .....	87
1.1.9.	Script Asistencia .....	94
1.1.10.	Pantalla Registrar_usuario .....	95
1.1.11.	Pantalla Registrar_eventual .....	98
1.1.12.	Script Registrar .....	101
1.1.13.	Pantalla Borrar_usuario .....	103
1.1.14.	Pantalla Borrar_eventual .....	106
1.1.15.	Script Borrar .....	108
1.1.16.	Pantalla Actualizar_usuario.....	109
1.1.17.	Pantalla Actualizar_eventual.....	118
1.1.18.	Script Actualizar.....	122
1.1.19.	Pantalla Permisos .....	123
1.1.20.	Script Permiso .....	128
1.1.21.	Script Registrar_evento.....	130
1.1.22.	Pantalla Reportes .....	130
1.1.23.	Script ReportePDF .....	140
1.1.24.	Script Barra_Asistencia.....	146
1.1.25.	Script Barra_Pagos .....	146
1.1.26.	Script Barra_Permisos .....	146
1.1.27.	Pantalla Cambiar_pass .....	147
1.1.28.	Script Salir.....	149

1.1.29. Script Blow.....	149
1.2. Applet (JAVASWING).....	150
1.2.1. Conexión a base de datos.....	150
1.2.2. Huella .....	152
<b>Q. Entregables Comprometidos en la Propuesta.....</b>	<b>167</b>

## **A. Objetivos**

### **1.1. General**

- Diseñar e implementar un sistema de información que permita llevar un mejor control y organización de los empleados eventuales.

### **1.2. Particulares**

- Mantener actualizada y centralizada la información de los empleados eventuales.
- Contar con un sistema de información que permita realizar el registro de asistencia y los pagos correspondientes a los empleados.
- El sistema debe realizar la administración de inasistencias y permisos de los empleados.
- Crear un módulo que registre y modifique la información de los empleados eventuales.
- Realizar el diseño e implementación de un módulo que registre la asistencia y los pagos semanales de cada empleado eventual a través de su huella digital.
- Diseñar e implementar un módulo que registre las horas extras y permisos de los empleados eventuales.
- Crear un diseño que implemente un módulo que permita el envío de reportes por correo electrónico.
- Diseñar e implementar una base de datos que permita administrar la información de los empleados.

## **B. Resumen**

Este sistema de información se elaboró, con la finalidad de mantener una mejor organización los empleados eventuales; como son: sus pagos, asistencias y registros logrando así los objetivos propuestos.

Este sistema considera la seguridad como parte de las contraseñas de usuarios e inicios de sesión, también la lectura biométrica de huella digital para autenticación de usuarios.

## C. Introducción

La crisis económica actual ha cambiado el panorama del empleo. Los empleadores han aprendido lecciones sobre la necesidad de transformar sus estrategias para la adquisición de talento de cara a una creciente competencia global.

En la medida en que la recesión disminuye, los empleadores dependen cada vez más del talento externo o de trabajadores eventuales; ya sean empleados temporales, contratistas, trabajadores de terceros o consultores.

Las razones más comunes para la contratación de trabajadores eventuales, es cubrir la demanda de trabajo "por temporada". Sin embargo por lo menos el 14% de las compañías en todo el mundo ahora contemplan contratar empleados eventuales por razones más estratégicas; para probar a los candidatos antes de contratarlos en puestos permanentes, para proporcionar flexibilidad a largo plazo, para encontrar rápidamente a personas talentosas que posean habilidades especializadas y para funciones terceras que no son fundamentales para la operación del negocio.

Las empresas que utilizan empleados eventuales, se ven en la necesidad de llevar un registro de estos como son: el control e ingreso a las instalaciones. El control se lleva a cabo por las personas de recursos humanos, tomando la asistencia y el registro con el programa de más fácil uso para ellos, sin tomar en cuenta los problemas que ocasiona no tener un sistema especializado en ello como son:

Perdida de información por virus que atacan la computadora o accesos no permitidos a los archivos.

Registros inadecuados por un mal almacenamiento o error humano.

Asistencia de empleados eventuales que pudieron no presentarse a laborar o un mal registro de aquellos que sí asistieron, provocando pagos erróneos.

Por tratarse de personal eventual existe mucha rotación, en cada momento se agregan más y más empleados eventuales, haciendo que las hojas de cálculo sean inoperables y en ocasiones no abran correctamente o se vean corruptas.

Cuando se requiere realizar cambios a las hojas de cálculo por más de un usuario, estos deben esperar que les compartan la información para tener los registros actualizados.

Después de un tiempo las hojas de cálculo crecen tanto que ya no es posible abrirlas y se opta por seccionar en archivos más pequeños; con lo que se limita su consulta a únicamente el archivo que se abre.

A su vez las empresas piensan en estos problemas y contratan empresas dedicadas al control de personal, lo que genera un aumento en sus costos y reduce su independencia.

Por lo tanto se propone este sistema para empresas con pocos recursos y que requieren un sistema dedicado al registro y asistencia de los empleados eventuales. El sistema ayudará a las empresas a llevar un mejor control de sus empleados eventuales y permitiéndoles determinar cuándo es conveniente volver a contratarlos.

## D. JUSTIFICACIÓN

El sistema se desarrollará para sustituir el uso de hojas de cálculo en la captura de datos del personal eventual; ya que estas no cubren la necesidad de almacenar la cantidad de registros que requieren las empresas.

El sistema se empleará para tener actualizada la información de todo el personal eventual; ya que al usar un archivo en Excel; los administrativos tienen que esperar que se comparta dicho archivo para conocer las altas, bajas y cambios del personal.

Actualmente cualquiera que revise el archivo en Excel puede consultar la información del personal; en cambio en el sistema se requiere tener una cuenta válida para ingresar a la información y realizar cambios o ajustes necesarios.

## E. Marco Teórico

El sistema de información a desarrollado consiste en una aplicación Web que consta de 7 módulos relacionados de la siguiente forma

a) **Acceso al sistema.**

El acceso al sistema, como su nombre lo indica; realizará la validación del usuario por medio de una cuenta y una contraseña (previamente asignadas). Al enviar los datos y ser validados se tendrá acceso a los demás módulos del sistema, dependiendo del perfil asignado a cada usuario.

b) **Características de los empleados eventuales.**

Módulo que permitirá registrar y actualizar información de los empleados eventuales, siempre que el usuario este autorizado para ello. Este módulo tendrá conexión con el módulo de huella digital. A éste módulo tendrán acceso solo los usuarios con privilegios de administrador, supervisor y asistente.

c) **Registro de asistencia y pagos.**

Este módulo permitirá tomar el registro de las asistencias e inasistencias (por medio de la huella digital) de los empleados eventuales para posteriormente asignar los pagos que serán enviados al módulo de reportes. Para registrar la asistencia por huella digital el módulo tendrá conexión al módulo de huella digital.

d) **Lector de huella digital**

Modulo que se encargara de la comunicación entre el sistema y el dispositivo lector de huella digital. Permitirá registrar la huella digital y almacenarla en la base datos, también se encargará de registrar la asistencia con la lectura de la huella digital. Este módulo tendrá conexión con el módulo de características de los usuarios y registro de asistencia y pagos.

e) **Registro de horas extras y permisos.**

La función principal del módulo será llevar un registro en la base de datos de los permisos que tiene cada empleado eventual. Información que servirá al módulo de reportes.

f) **Reportes.**

Módulo que genera estadísticas (horas extras, permisos, pagos, asistencias e inasistencias) y gráficas de todos los empleados eventuales. Enviando estos reportes a los diferentes empleados permanentes y directivos.

g) **Base de datos.**

La base de datos almacena la información de los empleados eventuales, esto evita registros duplicados, perdida de información, registros inadecuados y asistencias mal tomadas. También tendrá la información de acceso de los empleados que usen el sistema. La base tendrá comunicación con todos los módulos del sistema.

## F. Desarrollo del proyecto

Los módulos con los que cuenta el sistema son los siguientes:

➤ Módulo 1: Acceso al sistema.

Se realiza la validación del usuario por medio de una cuenta y una contraseña (previamente asignadas). Al enviar los datos y ser validados se tendrá acceso a los demás módulos del sistema, dependiendo del perfil asignado a cada usuario.

➤ Módulo 2: Características de los empleados eventuales.

Permite registrar y actualizar información de los empleados eventuales, siempre que el usuario este autorizado para ello. A éste solo tienen accesos los usuarios con privilegios de administrador, supervisor y asistente.

➤ Módulo 3: Registro de asistencia y pagos.

Permite tomar el registro de las asistencias e inasistencias (por medio de la huella digital) de los empleados eventuales, estos datos son almacenados en la base de datos.

➤ Módulo 4: Lector de huella digital

Se encarga de la comunicación entre el sistema y el dispositivo lector de huella digital. Permite a su vez registrar los datos de huella digital y almacenarlos en la

base datos, también se encarga verificar que la huella coincida con la del usuario eventual.

➤ **Módulo 5: Registro de permisos.**

Se lleva a cabo un registro en la base de datos, del tipo de permiso tomado por un empleado eventual que hubiera asistido o no.

➤ **Módulo 6: Reportes.**

Genera estadísticas (permisos, pagos, asistencias) y gráficas de todos los empleados eventuales. Permitiendo su envío a un archivo digital (PDF) para su almacenamiento de forma local.

## **G. Casos de Uso**

En esta sección se describen los casos de uso o escenarios que representan funcionalidades centrales del sistema final, estos casos de uso permitirán descubrir, describir y diseñar la arquitectura del sistema.

### **1. Identificación de los Casos de Uso relevantes para la arquitectura**

Para el diseño del Sistema de información para el control de empleados eventuales se identifican los siguientes casos de uso:

Módulo 1: Acceso al sistema.

Módulo 2: Características de los empleados eventuales.

Módulo 3: Registro de asistencia y pagos.

Módulo 4: Lector de huella digital.

Módulo 5: Registro de permisos.

Módulo 6: Reportes.

#### **1.1. Descripción de los Casos de Uso más relevantes para la arquitectura**

##### **Descripción de actores**

###### **Actor 1**

**Nombre:** Administrador

**Descripción:** Es la persona responsable del manejo del sistema y la base de datos, normalmente será una persona que cuente con conocimientos en sistemas de informáticos y el proceso del negocio.

### Actor 2

**Nombre:** Supervisor

**Descripción:** Es la persona encargada de que los registros, asistencias, permisos y pagos de los empleados eventuales se lleven correctamente basándose en los datos arrojados por el sistema.

### Actor 3

**Nombre:** Ayudante

**Descripción:** Es la persona que apoya al supervisor en los registros, asistencias y permisos de los empleados eventuales, a través del sistema.

### Actor 3

**Nombre:** Usuario General

**Descripción:** Visualiza e imprime el reporte generado por el sistema.

### Lista de Casos de Uso

<i>Actor Primario</i>	<i>Casos de uso</i>	
<b>Administrador</b>	<i>Acceso al sistema</i>	<i>1.1</i>
<b>Administrador</b>	<i>Registrar Cuenta</i>	<i>1.2</i>
<b>Administrador</b>	<i>Registrar usuario</i>	<i>1.3</i>
<b>Administrador</b>	<i>Borrar usuario</i>	<i>1.4</i>
<b>Administrador</b>	<i>Modificar usuario</i>	<i>1.5</i>
<b>Supervisor</b>	<i>Registrar empleado eventual</i>	<i>2.1</i>
<b>Supervisor</b>	<i>Borrar empleado eventual</i>	<i>2.2</i>
<b>Supervisor</b>	<i>Modificación de datos del empleado eventual</i>	<i>2.3</i>
<b>Supervisor</b>	<i>Registrar asistencia</i>	<i>3.1</i>
<b>Supervisor</b>	<i>Generar pago</i>	<i>3.2</i>
<b>Supervisor</b>	<i>Validar Huella</i>	<i>4.1</i>
<b>Supervisor</b>	<i>Guardar Huella</i>	<i>4.2</i>
<b>Supervisor</b>	<i>Registrar Tipo Permiso</i>	<i>5.1</i>
<b>Supervisor</b>	<i>Reporte</i>	<i>6.1</i>

1.1.1. Módulo 1: Acceso al Sistema.

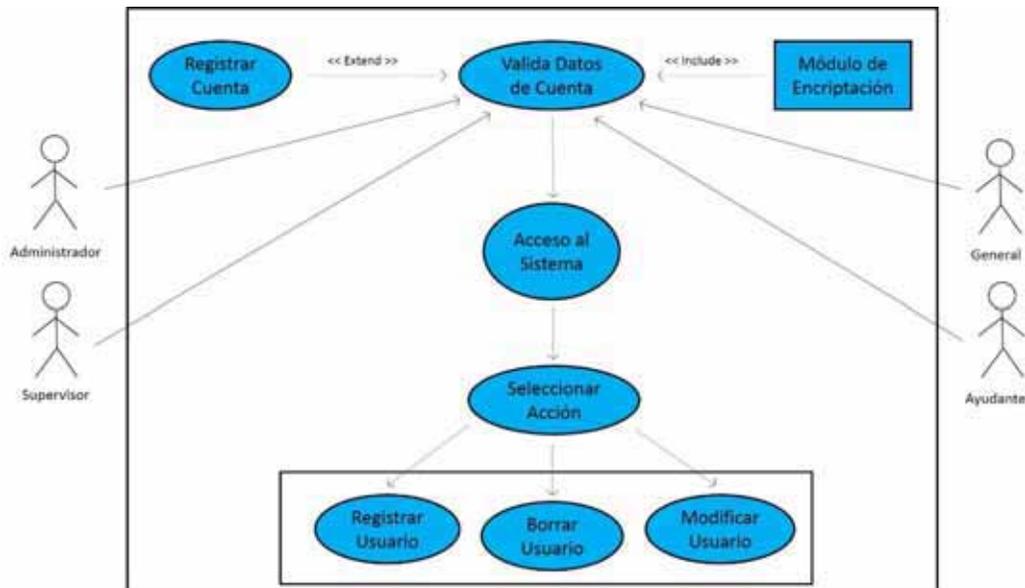


Figura 1. Diagrama de casos de uso: Acceso al sistema.

Caso de Uso	
<b>ID del caso de uso:</b>	CU 1.1
<b>Nombre del caso de uso:</b>	Acceso al sistema
<b>Actores:</b>	Administrador, Supervisor, Ayudante y General.
<b>Propósito:</b>	Tener acceso al sistema
<b>Descripción:</b>	El usuario ingresara su usuario y contraseña.
<b>Pre-condición</b>	<ul style="list-style-type: none"> <li>➤ El usuario existe.</li> <li>➤ Se tiene una contraseña.</li> </ul>
<b>Pos-condición</b>	<ul style="list-style-type: none"> <li>➤ Ingreso al sistema principal.</li> </ul>

Caso de Uso	
<b>ID del caso de uso:</b>	CU 1.1
<b>Nombre del caso de uso:</b>	Registar Cuenta
<b>Actores:</b>	Administrador
<b>Propósito:</b>	Crear cuentas de usuarios predeterminadas
<b>Descripción:</b>	El usuario asignara una nueva cuenta a los nuevos usuarios del sistema.
<b>Pre-condición</b>	<ul style="list-style-type: none"> <li>➤ No existe el usuario</li> <li>➤ Al menos existe un usuario administrador</li> </ul>
<b>Pos-condición</b>	<ul style="list-style-type: none"> <li>➤ Crear nuevos usuarios</li> <li>➤ Asignar una contraseña.</li> </ul>

Caso de Uso	
<b>ID del caso de uso:</b>	CU 1.2

<b>Nombre del caso de uso:</b>	<i>Registrar Usuario</i>
<b>Actores:</b>	Administrador, Supervisor, Ayudante y General.
<b>Propósito:</b>	Registrar nuevos usuarios.
<b>Descripción:</b>	El usuario creara un nuevo usuario de acceso al sistema con iguales o menores privilegios.
<b>Pre-condición</b>	<ul style="list-style-type: none"> <li>➤ No existe el usuario.</li> <li>➤ Al menos existe un usuario administrador.</li> </ul>
<b>Pos-condición</b>	<ul style="list-style-type: none"> <li>➤ Crear nuevos usuarios .</li> <li>➤ Asignar una contraseña.</li> </ul>

Caso de Uso	
<b>ID del caso de uso:</b>	CU 1.3
<b>Nombre del caso de uso:</b>	<i>Borrar Usuario</i>
<b>Actores:</b>	Administrador, Supervisor, Ayudante y General.
<b>Propósito:</b>	Eliminar usuarios.
<b>Descripción:</b>	El usuario eliminara aquellos usuarios con iguales o menores privilegios que por algún motivo ya no deban tener acceso al sistema.
<b>Pre-condición</b>	<ul style="list-style-type: none"> <li>➤ Existe el usuario.</li> <li>➤ Al menos existe un usuario.</li> </ul>
<b>Pos-condición</b>	<ul style="list-style-type: none"> <li>➤ Eliminar usuarios.</li> </ul>

Caso de Uso	
<b>ID del caso de uso:</b>	CU 1.4
<b>Nombre del caso de uso:</b>	<i>Modificar Usuario</i>
<b>Actores:</b>	Administrador, Supervisor, Ayudante y General.
<b>Propósito:</b>	Modificar datos de los usuarios.
<b>Descripción:</b>	El usuario cambiara aquellos registros que sean necesarios o modificara privilegios en el sistema.
<b>Pre-condición</b>	<ul style="list-style-type: none"> <li>➤ Existe el usuario</li> <li>➤ Al menos existe un usuario.</li> </ul>
<b>Pos-condición</b>	<ul style="list-style-type: none"> <li>➤ Modifica datos de los usuarios.</li> </ul>

### 1.1.2. Módulo 2: Características de los empleados eventuales.

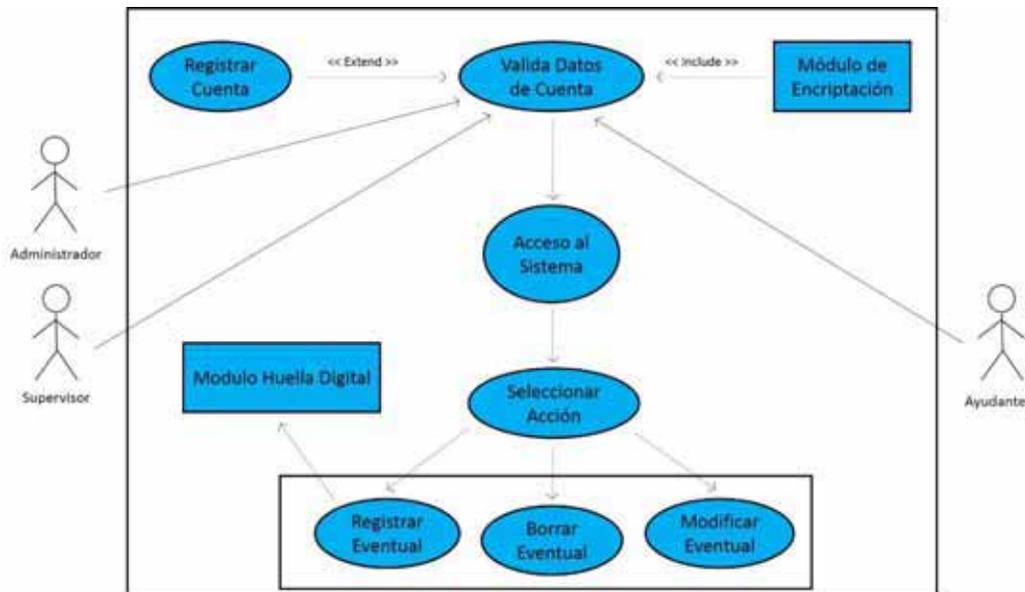


Figura 2. Diagrama de casos de uso: Características de los empleados eventuales.

Caso de Uso	
<b>ID del caso de uso:</b>	CU 2.1
<b>Nombre del caso de uso:</b>	<i>Registrar Eventual</i>
<b>Actores:</b>	Administrador, Supervisor, Ayudante.
<b>Propósito:</b>	Registrar nuevos empleados eventuales.
<b>Descripción:</b>	El usuario creara un nuevo empleado eventual.
<b>Pre-condición</b>	<ul style="list-style-type: none"> <li>➤ No existe el empleado eventual.</li> <li>➤ Al menos existe un usuario administrador, supervisor o ayudante.</li> </ul>
<b>Pos-condición</b>	<ul style="list-style-type: none"> <li>➤ El nuevo empleado eventual es creado.</li> </ul>

Caso de Uso	
<b>ID del caso de uso:</b>	CU 2.2
<b>Nombre del caso de uso:</b>	<i>Borrar Eventual</i>
<b>Actores:</b>	Administrador, Supervisor y Ayudante.
<b>Propósito:</b>	Eliminar empleados eventuales.
<b>Descripción:</b>	El usuario eliminara aquellos eventuales que por algún motivo ya no deban estar en el sistema.
<b>Pre-condición</b>	<ul style="list-style-type: none"> <li>➤ Existe el empleado eventual.</li> <li>➤ Al menos existe un usuario administrador, supervisor o ayudante.</li> </ul>
<b>Pos-condición</b>	<ul style="list-style-type: none"> <li>➤ Eliminar empleado eventual.</li> </ul>

### Caso de Uso

<b>ID del caso de uso:</b>	CU 2.3
<b>Nombre del caso de uso:</b>	<i>Modificar Eventual</i>
<b>Actores:</b>	Administrador, Supervisor y Ayudante.
<b>Propósito:</b>	Modificar información de los empleados eventuales.
<b>Descripción:</b>	El usuario cambiara aquellos registros que sean necesarios.
<b>Pre-condición</b>	<ul style="list-style-type: none"> <li>➤ Existe el usuario.</li> <li>➤ Al menos existe un usuario administrador, supervisor o ayudante.</li> </ul>
<b>Pos-condición</b>	<ul style="list-style-type: none"> <li>➤ Modifica información de los empleados eventuales.</li> </ul>

### 1.1.3. Módulo 3: Registro de asistencia y pagos.

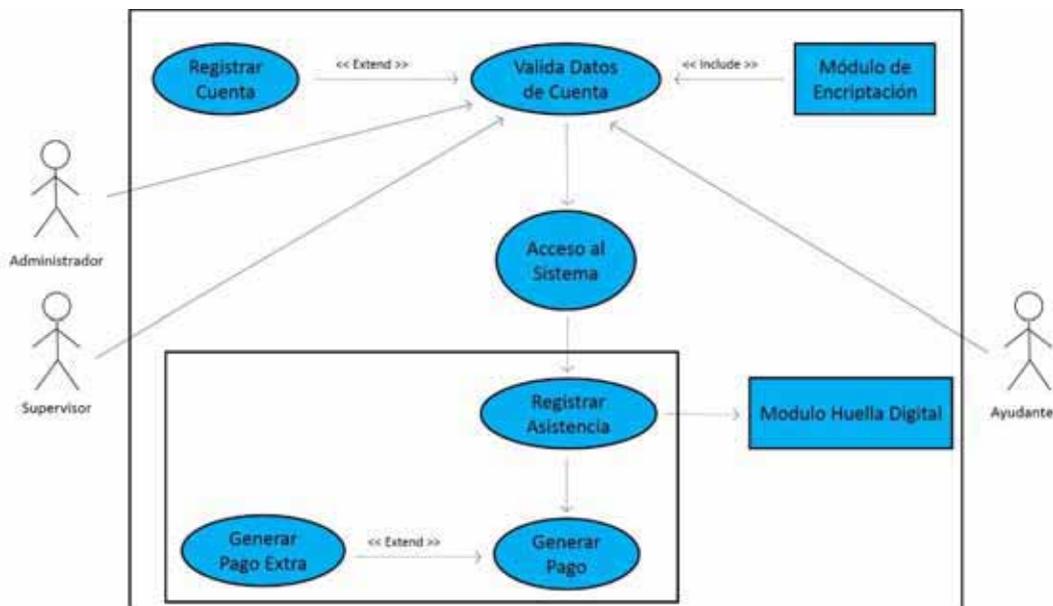


Figura 3. Diagrama de casos de uso: Registro de asistencia y pagos.

Caso de Uso	
<b>ID del caso de uso:</b>	CU 3.1
<b>Nombre del caso de uso:</b>	<i>Registrar Asistencia</i>
<b>Actores:</b>	Administrador, Supervisor y Ayudante.
<b>Propósito:</b>	Tomar la asistencia de los empleados eventuales.
<b>Descripción:</b>	El usuario registra la asistencia de los empleados eventuales.
<b>Pre-condición</b>	<ul style="list-style-type: none"> <li>➤ Existe el usuario.</li> <li>➤ Al menos existe un usuario administrador, supervisor o ayudante.</li> <li>➤ No tiene asistencia para ese evento.</li> </ul>
<b>Pos-condición</b>	<ul style="list-style-type: none"> <li>➤ Se registra la asistencia.</li> <li>➤ Se genera un pago.</li> </ul>

Caso de Uso	
<b>ID del caso de uso:</b>	CU 3.2
<b>Nombre del caso de uso:</b>	<i>Generar Pago</i>
<b>Actores:</b>	Administrador, Supervisor y Ayudante.
<b>Propósito:</b>	Guarda el pago del evento para el empleado eventual.
<b>Descripción:</b>	El usuario registra el pago que se pagara al empleado eventual.
<b>Pre-condición</b>	<ul style="list-style-type: none"> <li>➤ Existe el usuario.</li> <li>➤ Al menos existe un usuario administrador, supervisor o ayudante.</li> <li>➤ Tiene asistencia para ese evento.</li> <li>➤ No se ha registrado pago.</li> </ul>
<b>Pos-condición</b>	<ul style="list-style-type: none"> <li>➤ Se genera un pago.</li> </ul>

1.1.4. Módulo 4: Lector de huella digital.

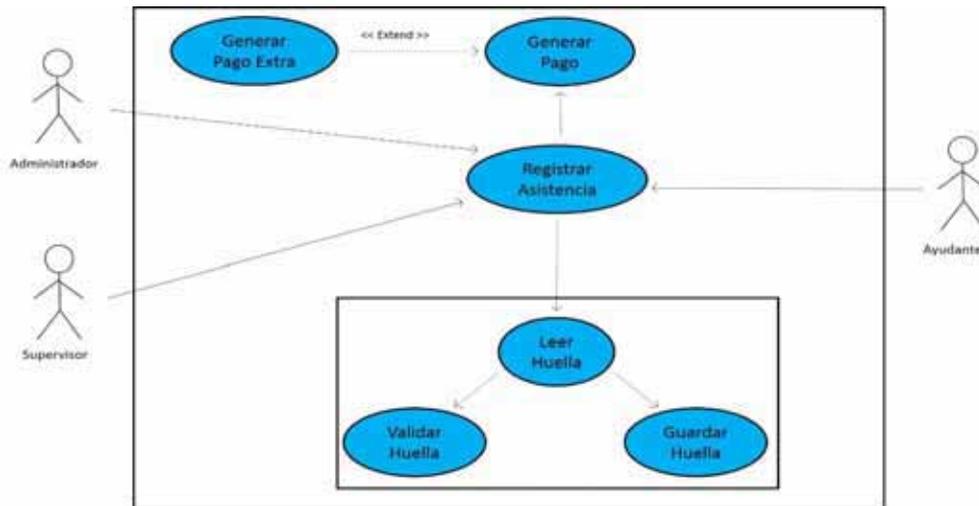


Figura 4. Diagrama de casos de uso: Lector de huella digital.

Caso de Uso	
<b>ID del caso de uso:</b>	CU 4.1
<b>Nombre del caso de uso:</b>	<i>Validar Huella</i>
<b>Actores:</b>	Administrador, Supervisor y Ayudante.
<b>Propósito:</b>	Validar la huella digital del empleado eventual.
<b>Descripción:</b>	El sistema válida que la huella leída desde el lector corresponda al del usuario en cuestión.
<b>Pre-condición</b>	<ul style="list-style-type: none"> <li>➤ Existe el usuario.</li> <li>➤ Al menos existe un usuario administrador, supervisor o ayudante.</li> <li>➤ Se leyó la huella.</li> <li>➤ La huella se encuentra almacenada.</li> </ul>
<b>Pos-condición</b>	<ul style="list-style-type: none"> <li>➤ Valida al empleado eventual.</li> <li>➤ Permite el registro de la asistencia.</li> </ul>

Caso de Uso	
<b>ID del caso de uso:</b>	CU 4.2
<b>Nombre del caso de uso:</b>	<i>Guardar Huella</i>
<b>Actores:</b>	Administrador, Supervisor y Ayudante.
<b>Propósito:</b>	Almacena la huella digital del empleado eventual.
<b>Descripción:</b>	El sistema almacena la huella leída desde el lector en un registro.
<b>Pre-condición</b>	<ul style="list-style-type: none"> <li>➤ Al menos existe un usuario administrador, supervisor o ayudante.</li> <li>➤ Se leyó la huella.</li> </ul>
<b>Pos-condición</b>	<ul style="list-style-type: none"> <li>➤ Guarda la huella en el registro.</li> <li>➤ Permite el registro del empleado.</li> </ul>

### 1.1.5. Módulo 5: Registro de permisos.

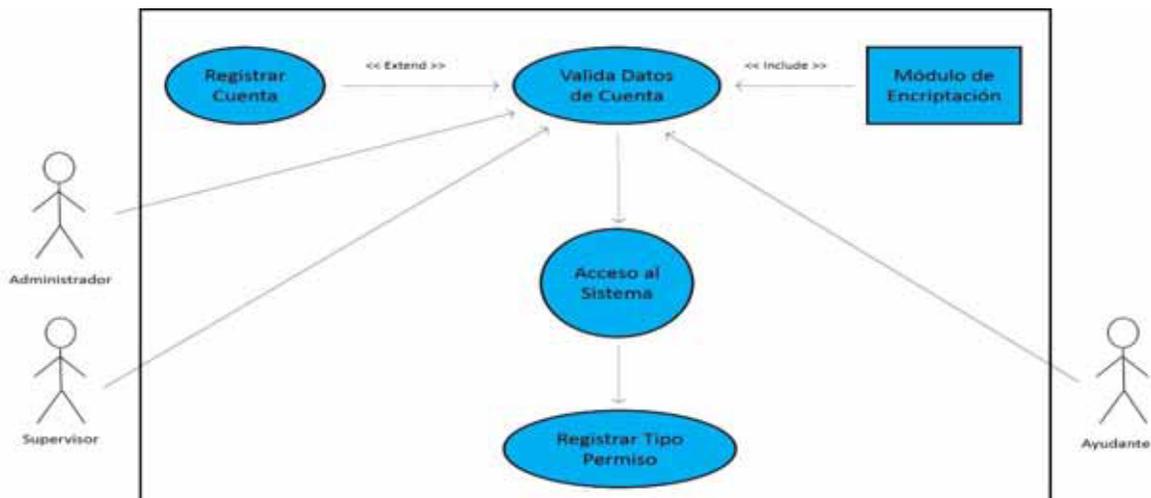


Figura 5. Diagrama de casos de uso: Registro de permisos.

Caso de Uso	
<b>ID del caso de uso:</b>	CU 5.1
<b>Nombre del caso de uso:</b>	<i>Registrar Tipo Permiso</i>
<b>Actores:</b>	Administrador, Supervisor y Ayudante.
<b>Propósito:</b>	Registrar algún tipo de permiso al empleado eventual.
<b>Descripción:</b>	El usuario registra un tipo de permiso según sea el caso a un empleado eventual.
<b>Pre-condición</b>	<ul style="list-style-type: none"> <li>➤ Existe el empleado eventual</li> <li>➤ Al menos existe un usuario administrador, supervisor o ayudante.</li> </ul>
<b>Pos-condición</b>	<ul style="list-style-type: none"> <li>➤ Se registra el permiso.</li> </ul>

1.1.6. Módulo 6: Reportes.

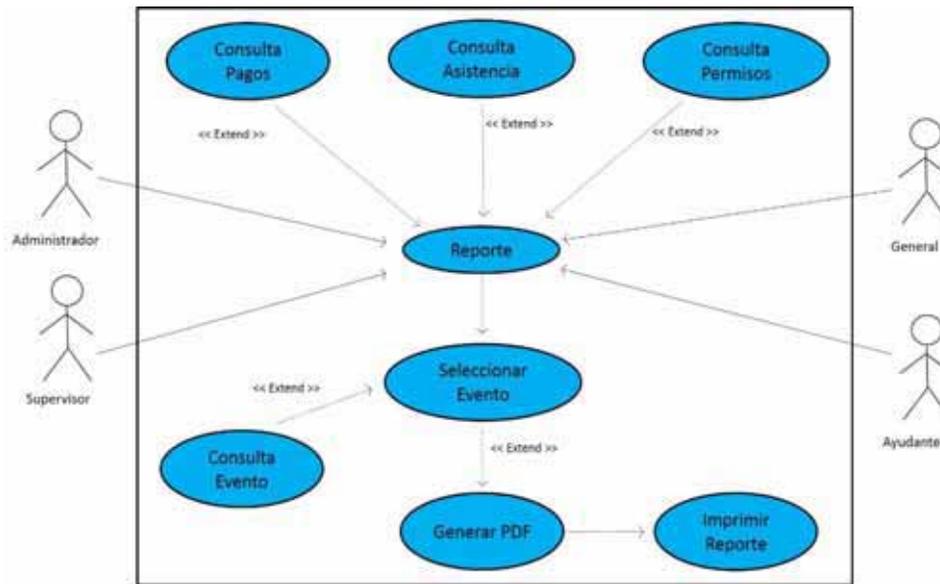
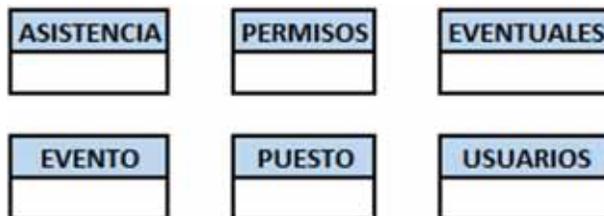


Figura 6. Diagrama de casos de uso: Reportes.

Caso de Uso	
<b>ID del caso de uso:</b>	CU 6.1
<b>Nombre del caso de uso:</b>	Reporte
<b>Actores:</b>	Administrador, Supervisor, Ayudante y General.
<b>Propósito:</b>	Obtener un reporte de algún evento.
<b>Descripción:</b>	El sistema generara un reporte del evento consultado.
<b>Pre-condición</b>	<ul style="list-style-type: none"> <li>➤ Al menos existe un usuario administrador, supervisor, ayudante o general.</li> <li>➤ Existe un evento.</li> </ul>
<b>Pos-condición</b>	<ul style="list-style-type: none"> <li>➤ Generar reporte.</li> <li>➤ Imprimir reporte.</li> </ul>

## H. Estructura de la base de datos

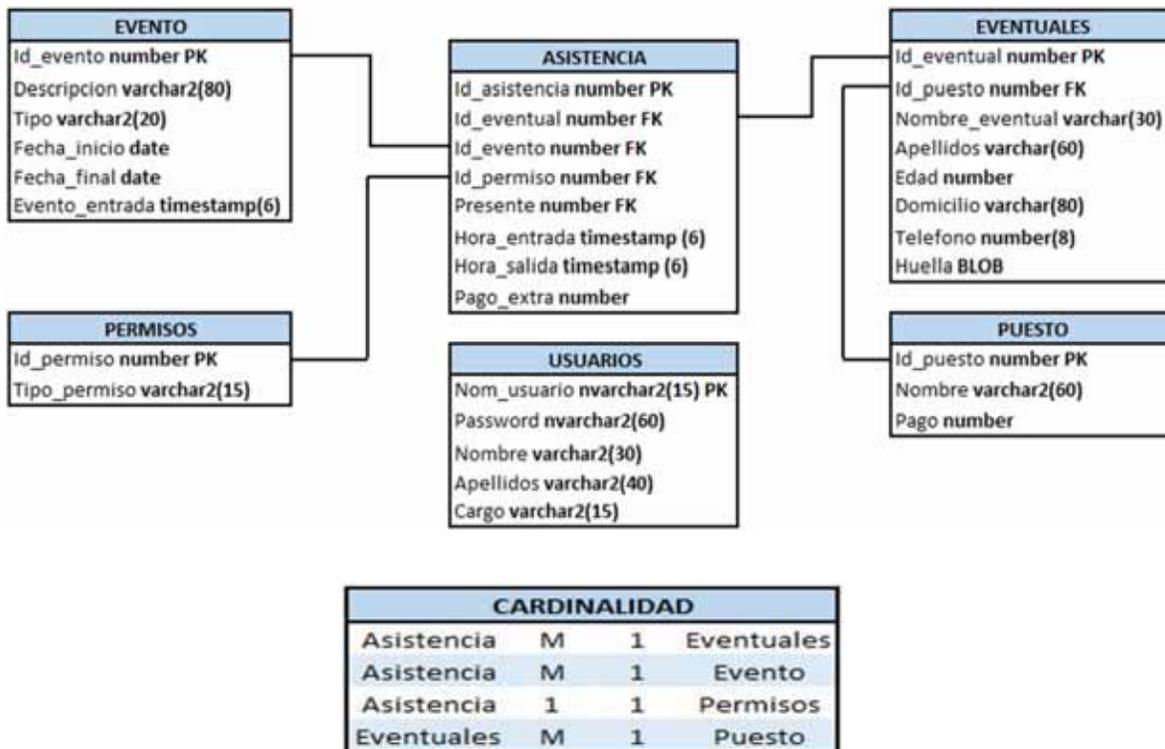
### 1. Entidades



### 2. Atributos



### 3. Diagrama entidad-relación



### 4. Diccionario de datos

Diccionario de Datos						
Columna	Tipo	Unique	Not null	Check	Llave	Descripción

TABLA ASISTENCIA						
Id_asistencia	number	Unique	Not null	Check	Primary key	Identificador de la asistencia
Id_eventual	number	Unique	Not null	Check	Foreign key	Identificador del empleado eventual
Id_evento	number		Not null	Check	Foreign key	Identificador del evento
Id_permiso	number				Foreign key	Identificador del tipo de permiso
Presente	number		Not null	Check		Asistencia
Hora_entrada	timestamp		Not null	Check		Hora en que ingresa el empleado eventual al evento
Hora_salida	timestamp					Hora en que sale el empleado eventual al evento
Pago_extra	timestamp					Pago extra
TABLA EVENTO						
Id_evento	number	Unique	Not null	Check	Primary key	Identificador del evento
Descripcion	varchar2		Not null	Check		Descripción o nombre del evento
Tipo	varchar2		Not null	Check		Tipo de evento
Fecha_inicio	date	Unique	Not null	Check		Fecha de inicio del evento
Fecha_final	date	Unique	Not null	Check		Fecha de termino del evento
Evento_entrada	timestamp		Not null	Check		Hora de entrada al evento
TABLA EVENTUALES						
Id_eventual	number	Unique	Not null	Check	Primary key	Identificador del empleado eventual
Id_puesto	number		Not null	Check	Foreign key	Identificador del puesto
Nombre_eventual	varchar2		Not null	Check		Nombre del empleado eventual
Apellidos	varchar2					Apellidos del empleado eventual
Edad	number					Edad del empleado eventual
Domicilio	varchar2					Domicilio empleado eventual
Telefono	number					Telefono empleado eventual
Huella	blob	Unique		Check		Patron de la huella digital del empleado eventual
TABLA PERMISOS						
Id_permiso	number	Unique	Not null	Check	Primary key	Identificador del permiso
Tipo_permiso	varchar2	Unique	Not null	Check		Descripción del tipo de permiso

TABLA PUESTO						
Id_puesto	number	Unique	Not null	Check	Primary key	Identificador del puesto
Nombre	varchar2	Unique	Not null	Check		Descripción o nombre del puesto
Pago	number		Not null	Check		Cantidad a pagar por puesto
TABLA USUARIOS						
Nom_usuario	nvarchar2	Unique	Not null	Check	Primary key	Nombre de usuario para acceso al sistema
Password	nvarchar2		Not null	Check		Contraseña del usuario
Nombre	varchar2		Not null	Check		Nombre del usuario que tendrá acceso al sistema
Apellidos	varchar2					Apellidos del usuario
Cargo	varchar2		Not null	Check		Cargo que tendrá el usuario

## 5. Esquema Físico de la Base de Datos

### 1. TABLA: USUARIOS

CREATE TABLE USUARIOS

```
(
  NOM_USUARIO NVARCHAR2(15) NOT NULL
, PASSWORD NVARCHAR2(60) NOT NULL
, NOMBRE VARCHAR2(30) NOT NULL
, APELLIDOS VARCHAR2(40)
, CARGO VARCHAR2(15) NOT NULL
, CONSTRAINT USUARIOS_PK PRIMARY KEY
  (
    NOM_USUARIO
  )
)
ENABLE
);
INSERT INTO "SYSTEM"."USUARIOS" (NOM_USUARIO, PASSWORD, NOMBRE, APELLIDOS, CARGO)
VALUES ('user', '12345', 'USUARIO', 'GENERICO', 'ADMINISTRADOR');
```

### 2. TABLA: EVENTO

CREATE TABLE EVENTO

```
(
  ID_EVENTO NUMBER NOT NULL
, DESCRIPCION VARCHAR2(80) NOT NULL
, TIPO VARCHAR2(20) NOT NULL
, FECHA_INICIO DATE NOT NULL
, FECHA_FINAL DATE NOT NULL
, EVENTO_ENTRADA TIMESTAMP(6) NOT NULL
, CONSTRAINT EVENTO_PK PRIMARY KEY

```

```
(
  ID_EVENTO
)
ENABLE
);

ALTER TABLE EVENTO
ADD CONSTRAINT EVENTO_UK1 UNIQUE
(
  FECHA_INICIO
, FECHA_FINAL
, ID_EVENTO
)
ENABLE;
```

### 3. TABLA: PERMISOS

```
CREATE TABLE PERMISOS
(
  ID_PERMISO NUMBER NOT NULL
, TIPO_PERMISO VARCHAR2(15) NOT NULL
, CONSTRAINT PERMISOS_PK PRIMARY KEY
(
  ID_PERMISO
)
ENABLE
);

ALTER TABLE PERMISOS
ADD CONSTRAINT PERMISOS_UK1 UNIQUE
(
  ID_PERMISO
, TIPO_PERMISO
)
ENABLE;

INSERT INTO "SYSTEM"."PERMISOS" (ID_PERMISO, TIPO_PERMISO) VALUES ('0', 'NO PERMISOS')
INSERT INTO "SYSTEM"."PERMISOS" (ID_PERMISO, TIPO_PERMISO) VALUES ('1', 'ENFERMEDAD')
INSERT INTO "SYSTEM"."PERMISOS" (ID_PERMISO, TIPO_PERMISO) VALUES ('2', 'ESPECIAL')
INSERT INTO "SYSTEM"."PERMISOS" (ID_PERMISO, TIPO_PERMISO) VALUES ('3', 'OTRO')
```

### 4. TABLA: PUESTO

```
CREATE TABLE PUESTO
(
  ID_PUESTO NUMBER NOT NULL
, NOMBRE VARCHAR2(60) NOT NULL
, PAGO NUMBER NOT NULL
```

```
, CONSTRAINT PUESTO_PK PRIMARY KEY
(
ID_PUESTO
)
ENABLE
);

ALTER TABLE PUESTO
ADD CONSTRAINT PUESTO_UK1 UNIQUE
(
ID_PUESTO
, NOMBRE
)
ENABLE;
INSERT INTO "SYSTEM"."PUESTO" (ID_PUESTO, NOMBRE, PAGO) VALUES ('1', 'SOPORTE TECNICO',
'400')
INSERT INTO "SYSTEM"."PUESTO" (ID_PUESTO, NOMBRE, PAGO) VALUES ('2', 'CAJERO', '400')
INSERT INTO "SYSTEM"."PUESTO" (ID_PUESTO, NOMBRE, PAGO) VALUES ('3', 'AYUDANTE
GENERAL', '300')
INSERT INTO "SYSTEM"."PUESTO" (ID_PUESTO, NOMBRE, PAGO) VALUES ('4', 'BAR TENDER', '350')
```

## 5. TABLA: EVENTUALES

```
CREATE TABLE EVENTUALES
(
ID_EVENTUAL NUMBER NOT NULL
, ID_PUESTO NUMBER NOT NULL
, NOMBRE_EVENTUAL VARCHAR2(30) NOT NULL
, APELLIDOS VARCHAR2(60)
, EDAD NUMBER
, DOMICILIO VARCHAR2(80)
, TELEFONO NUMBER(8)
, HUELLA BLOB
, CONSTRAINT EVENTUALES_PK PRIMARY KEY
(
ID_EVENTUAL
)
ENABLE
);

ALTER TABLE EVENTUALES
ADD CONSTRAINT EVENTUALES_PUESTO_FK1 FOREIGN KEY
(
ID_PUESTO
)
REFERENCES PUESTO
(
ID_PUESTO
```

```
)  
ENABLE;  
INSERT INTO "SYSTEM"."EVENTUALES" (ID_EVENTUAL, ID_PUESTO, NOMBRE_EVENTUAL,  
APELLIDOS) VALUES ('00000000000', '1', 'Eventual', 'Prueba')
```

## 6. TABLA: ASISTENCIA

```
CREATE TABLE ASISTENCIA  
(  
  ID_ASISTENCIA NUMBER NOT NULL  
, ID_EVENTUAL NUMBER NOT NULL  
, ID_EVENTO NUMBER NOT NULL  
, ID_PERMISO NUMBER  
, PRESENTE NUMBER DEFAULT 1 NOT NULL  
, HORA_ENTRADA TIMESTAMP(6) NOT NULL  
, HORA_SALIDA TIMESTAMP(6)  
, PAGO_EXTRA NUMBER DEFAULT 0  
, CONSTRAINT ASISTENCIA_PK PRIMARY KEY  
(  
  ID_ASISTENCIA  
)  
ENABLE  
);  
  
ALTER TABLE ASISTENCIA  
ADD CONSTRAINT ASISTENCIA_UK1 UNIQUE  
(  
  ID_ASISTENCIA  
, ID_EVENTUAL  
)  
ENABLE;  
  
ALTER TABLE ASISTENCIA  
ADD CONSTRAINT ASISTENCIA_EVENTO_FK1 FOREIGN KEY  
(  
  ID_EVENTO  
)  
REFERENCES EVENTO  
(  
  ID_EVENTO  
)  
ENABLE;  
  
ALTER TABLE ASISTENCIA  
ADD CONSTRAINT ASISTENCIA_EVENTUALES_FK1 FOREIGN KEY  
(  
  ID_EVENTUAL  
)
```

REFERENCES EVENTUALES

```
(
  ID_EVENTUAL
)
```

ENABLE;

ALTER TABLE ASISTENCIA

ADD CONSTRAINT ASISTENCIA\_PERMISOS\_FK1 FOREIGN KEY

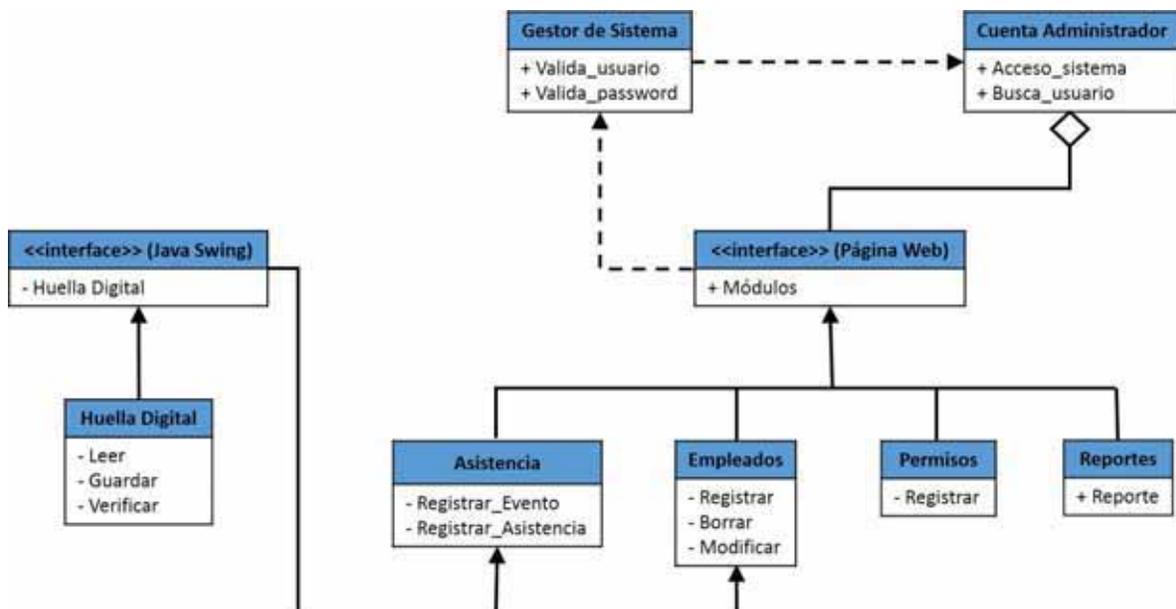
```
(
  ID_PERMISO
)
```

REFERENCES PERMISOS

```
(
  ID_PERMISO
)
```

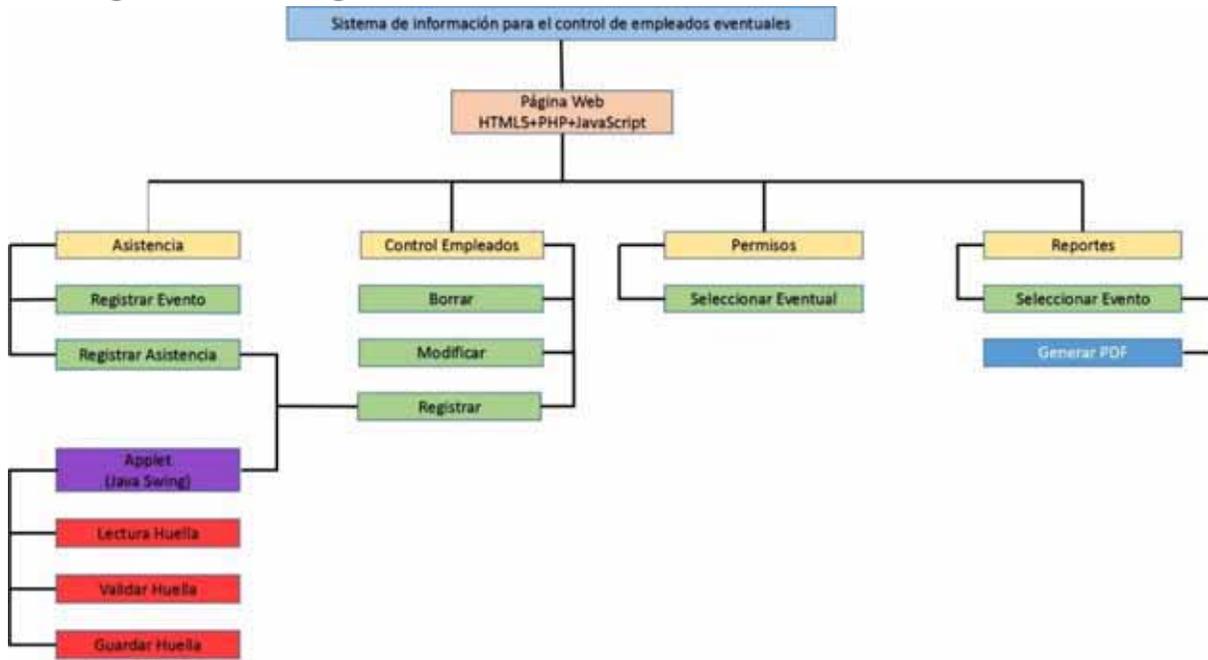
ENABLE;

## I. Diagrama de Clases



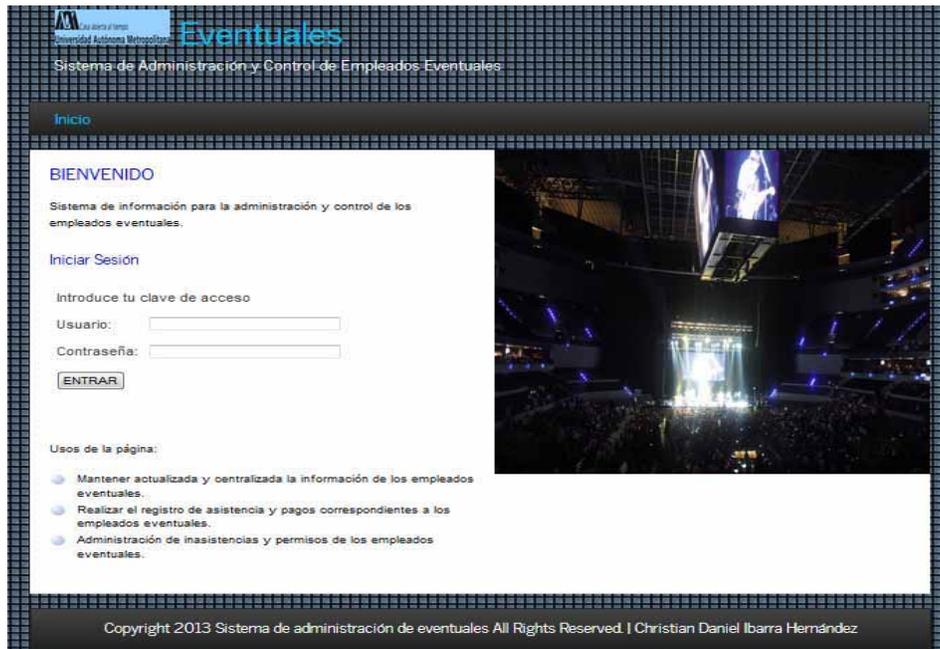
## J. Navegación

### 1. Diagrama de Navegación



### 2. Manual de Navegación

#### PASO 1: INGRESAR AL SISTEMA



## PASO 2: SELECCIONAR UNA OPCIÓN



## PASO 3: REGISTRAR USUARIO



#### PASO 4: ELIMINAR USUARIO

**Eventuales**  
Sistema de Administración y Control de Empleados Eventuales

Inicio Asistencia Empleados Permisos Reportes Chris080190 / ADMINISTRADOR

### ELIMINAR USUARIO

Introduce los datos que se te piden:

Usuario:

Chris080190 | Christian Daniel Ibarra Hernández -  
Heber10 | Heber Torres y Torres  
MarcoPacheco | Marco Pacheco

Copyright 2013 Sistema de administración de eventuales All Rights Reserved | Christian Daniel Ibarra Hernández

#### PASO 5: ACTUALIZAR USUARIO

**Eventuales**  
Sistema de Administración y Control de Empleados Eventuales

Inicio Asistencia Empleados Permisos Reportes Chris080190 / ADMINISTRADOR

### ACTUALIZAR USUARIO

Introduce los datos que se te piden:

Usuario:

Chris080190 | Christian Daniel Ibarra Hernández  
MarcoPacheco | Marco Pacheco

Usuario:   
Nombre(s):   
Apellidos:   
Cargo:

Copyright 2013 Sistema de administración de eventuales All Rights Reserved | Christian Daniel Ibarra Hernández

## PASO 6: MODIFICAR CONTRASEÑA

Logo: Universidad Autónoma Metropolitana

# Eventuales

Sistema de Administración y Control de Empleados Eventuales

Inicio Asistencia Empleados Permisos Reportes Chris080190 / ADMINISTRADOR

### CAMBIAR CONTRASEÑA

Introduce los datos que se te piden:

Usuario:

Contraseña anterior:

Nueva contraseña:

Repite contraseña:

Copyright 2013 Sistema de administración de eventuales All Rights Reserved | Christian Daniel Ibarra Hernández

## PASO 7: REGISTRAR EVENTUAL

Logo: Universidad Autónoma Metropolitana

# Eventuales

Sistema de Administración y Control de Empleados Eventuales

Inicio Asistencia Empleados Permisos Reportes Chris080190 / ADMINISTRADOR

### REGISTRAR EVENTUAL

Introduce los datos que se te piden:

Foto:

Puesto:

Nombre(s):

Apellidos:

Edad:

Domicilio:

Teléfono:

Copyright 2013 Sistema de administración de eventuales All Rights Reserved | Christian Daniel Ibarra Hernández

## PASO 8: ELIMINAR EVENTUAL

The screenshot shows the 'ELIMINAR EVENTUAL' page of the 'Sistema de Administración y Control de Empleados Eventuales'. The page has a dark blue header with the system name and a navigation menu with 'Inicio', 'Asistencia', 'Empleados', 'Permisos', and 'Reportes'. The user is logged in as 'Chris080190 / ADMINISTRADOR'. The main content area is titled 'ELIMINAR EVENTUAL' and contains the instruction 'Introduce los datos que se te piden:'. Below this is a search form with a text input labeled 'Folio: Eventual a Buscar' and a 'Buscar' button. A dropdown list shows three results: '5281 | Alejandro Gudiño Perez', '5237 | Christian Daniel Ibarra Hernández', and '5199 | Omar Rodriguez Mora'. An 'Enviar' button is located at the bottom left of the search area. The footer contains the copyright notice: 'Copyright 2013 Sistema de administración de eventuales All Rights Reserved. | Christian Daniel Ibarra Hernández'.

## PASO 9: ACTUALIZAR EVENTUAL

The screenshot shows the 'ACTUALIZAR EVENTUAL' page of the 'Sistema de Administración y Control de Empleados Eventuales'. The page has a dark blue header with the system name and a navigation menu with 'Inicio', 'Asistencia', 'Empleados', 'Permisos', and 'Reportes'. The user is logged in as 'Chris080190 / ADMINISTRADOR'. The main content area is titled 'ACTUALIZAR EVENTUAL' and contains the instruction 'Introduce los datos que se te piden:'. Below this is a search form with a text input labeled 'Folio: Usuario a Buscar' and a 'Buscar' button. A dropdown list shows two results: '5281 | Alejandro Gudiño Perez' and '5237 | Christian Daniel Ibarra Hernández'. An 'Enviar' button is located at the bottom left of the search area. Below the search area is a form with the following fields: 'Folio:' with value '5281', 'Puesto:' with a dropdown menu showing 'SOPORTE TECNICO', 'Nombre(s):' with value 'Alejandro', 'Apellidos:' with value 'Gudiño Perez', 'Edad:' with value '22', 'Domicilio:' with value 'El Trébol', and 'Teléfono:' with value '23873276'. An 'Actualizar' button is located at the bottom left of the form. The footer contains the copyright notice: 'Copyright 2013 Sistema de administración de eventuales All Rights Reserved. | Christian Daniel Ibarra Hernández'.

PASO 10: REGISTRAR ASISTENCIA

The screenshot shows the 'ASISTENCIA EVENTUAL' registration form. At the top left is the logo of Universidad Autónoma Metropolitana with the tagline 'Cada día es un tiempo'. The main title 'Eventuales' is in large blue letters, followed by the subtitle 'Sistema de Administración y Control de Empleados Eventuales'. A navigation bar contains 'Inicio', 'Asistencia', 'Empleados', 'Permisos', 'Reportes', and a user profile 'Chris080190 / ADMINISTRADOR'. The form title is 'ASISTENCIA EVENTUAL'. Below it, the instruction 'Registrar Evento:' is followed by four input fields: 'Ingresa nombre de evento:' with the value 'Prueba', 'Selecciona Tipo de Evento:' with a dropdown menu showing 'GRUPERO/SALSA', 'Fecha de Termino del evento:' with the value '23/03/14', and 'Hora de Entrada:' with the value '09:25:14'. A 'Registrar Evento' button is at the bottom. A footer contains the copyright notice: 'Copyright 2013 Sistema de administración de eventuales All Rights Reserved. | Christian Daniel Ibarra Hernández'.

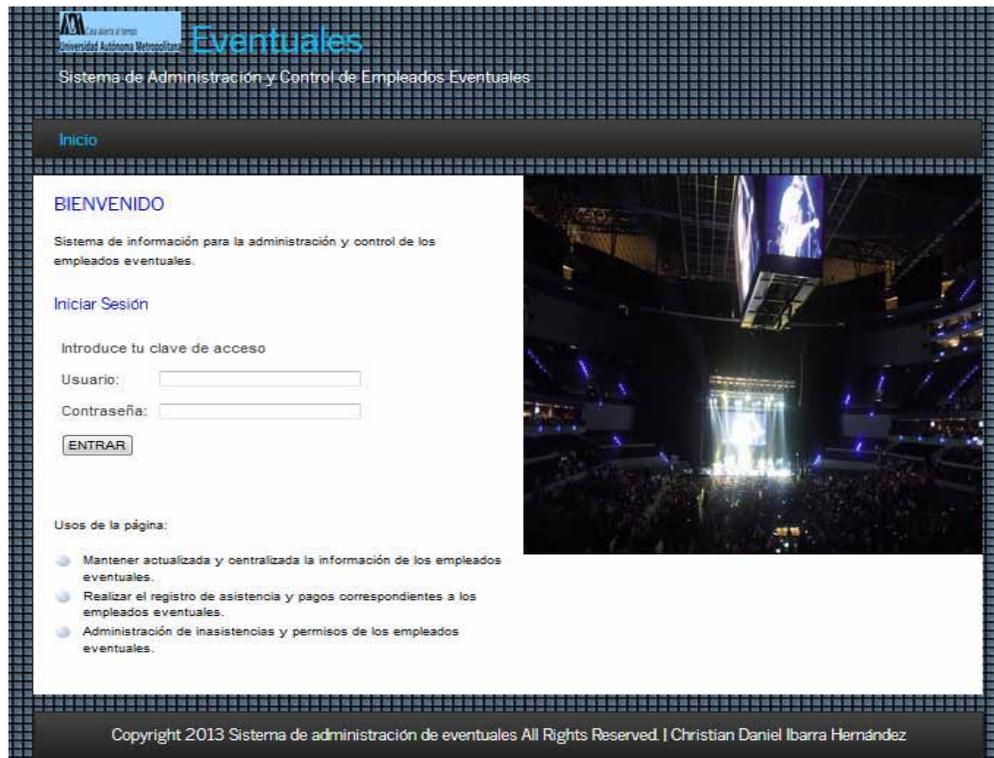
PASO 11: REGISTRAR PERMISOS

The screenshot shows the 'PERMISOS' registration form. It features the same header and navigation bar as the previous form. The title is 'PERMISOS'. The instruction 'Introduce los datos que se te piden:' is followed by a search section with 'Eventual:' and a text input 'Eventual a Buscar', and a 'Buscar' button. Below this is a list box containing two entries: '5281 | Alejandro Gudiño Perez' and '5237 | Christian Daniel Ibarra Hernández'. An 'Enviar' button is below the list. The 'Selección del Permiso:' section has three fields: 'Folio:' with the value '5237', 'Nombre:' with the value 'Christian Daniel Ibarra Hernández', and 'Permiso:' with a dropdown menu showing 'ENFERMEDAD'. A 'Registrar Permiso' button is at the bottom. The footer contains the copyright notice: 'Copyright 2013 Sistema de administración de eventuales All Rights Reserved. | Christian Daniel Ibarra Hernández'.

PASO 12: VISUALIZAR E IMPRIMIR REPORTE



PASO 13: CERRAR SESIÓN



## K. Manual de instalación

### A. SOFTWARE NECESARIO

✓ **XAMPP 1.8.3**

Es una aplicación para cualquier sistema operativo, que pone a disposición los servicios de Apache, PHP y MySQL de manera fácil y asistida.

✓ **PHP 5.5.9**

Versión incluida en la instalación de XAMP

✓ **Oracle 11g Express Edition**

**Oracle Database Express Edition 11g XE Release 2** es la versión del mundialmente conocido gestor de base de datos de Oracle en su versión 11g, para Windows es gratuita y muy sencilla de administrar, gracias a su carácter free es ideal para aquellos que quieran iniciarse el mundo de las bases de datos Oracle.

✓ **Copiar carpeta "eventuales" en:**

C:\xampp\htdocs

### B. INSTALACION DE XAMPP 1.8.3

Cuando instalas XAMPP, al igual que cuando instalas cualquier Apache, tendrás que desinstalar o deshabilitar los programas que utilizan los puertos 80, 443 y 3306.

Por ejemplo, el TeamViewer, el IIS, el Skype dan problemas, ya que todos utilizan el puerto 80

Descargamos XAMPP de la página de

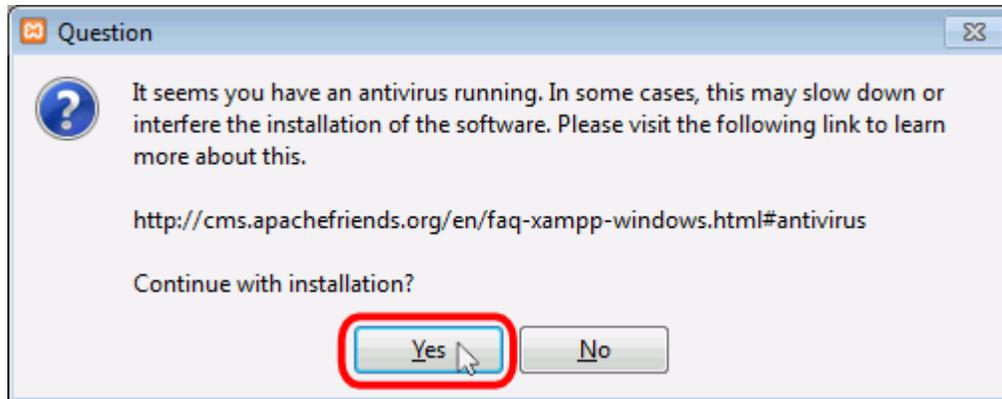
Apachefriends:<http://www.apachefriends.org/es/xampp.html>

Una vez obtenido el archivo de instalación de XAMPP, hay que hacer doble clic sobre él para ponerlo en marcha. Las imágenes que se muestran a continuación corresponden a la instalación de XAMPP 1.8.3.1 en Windows 7 (a partir de XAMPP 1.8.3, XAMPP no se puede instalar en Windows XP ya que PHP 5.5 no se puede instalar en Windows XP).

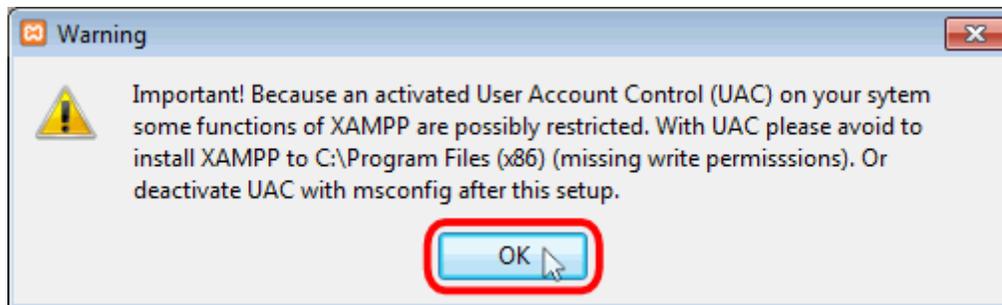
---

Al poner en marcha el instalador XAMPP nos muestra dos avisos:

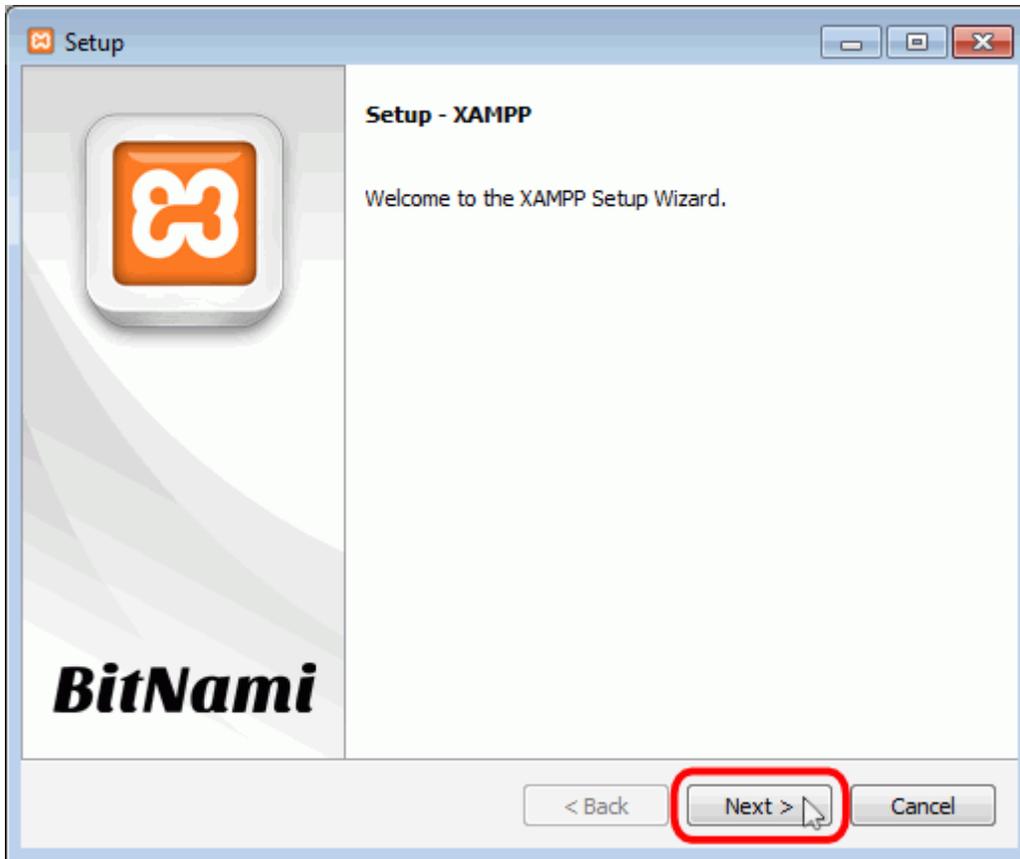
- El primero aparece si en el ordenador hay instalado un antivirus:



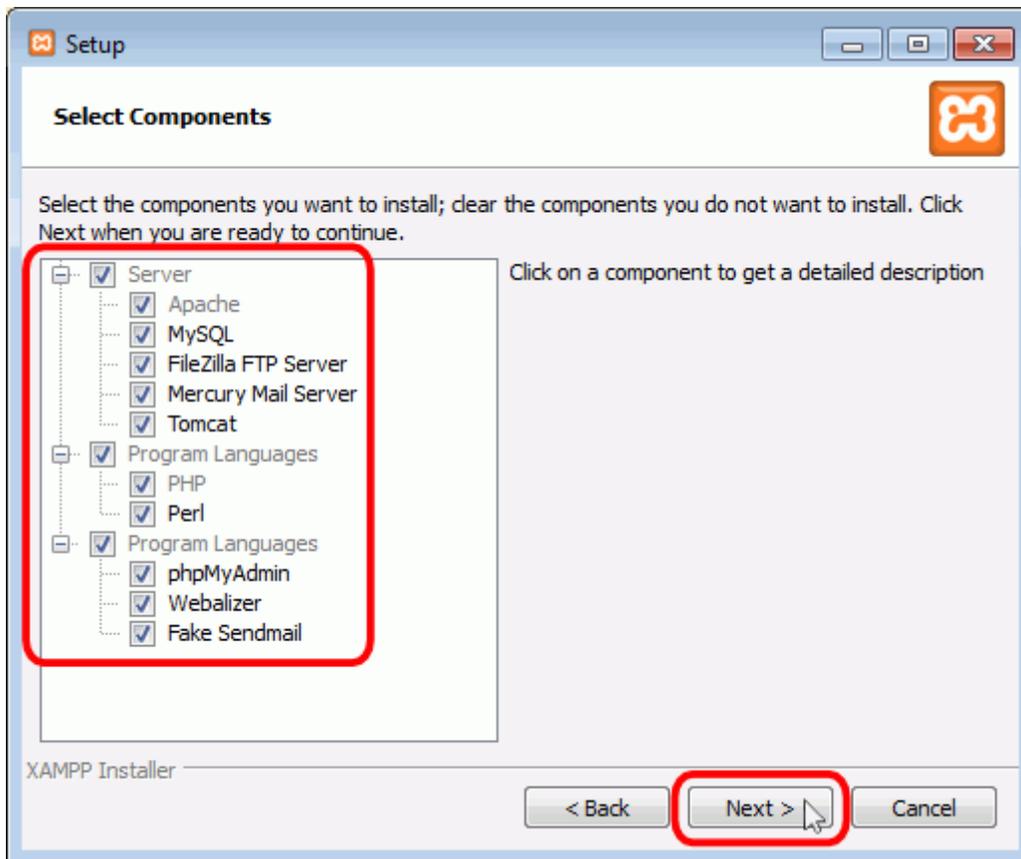
- El segundo aparece si está activado el Control de Cuentas de Usuario y recuerda que algunos directorios tienen permisos restringidos:



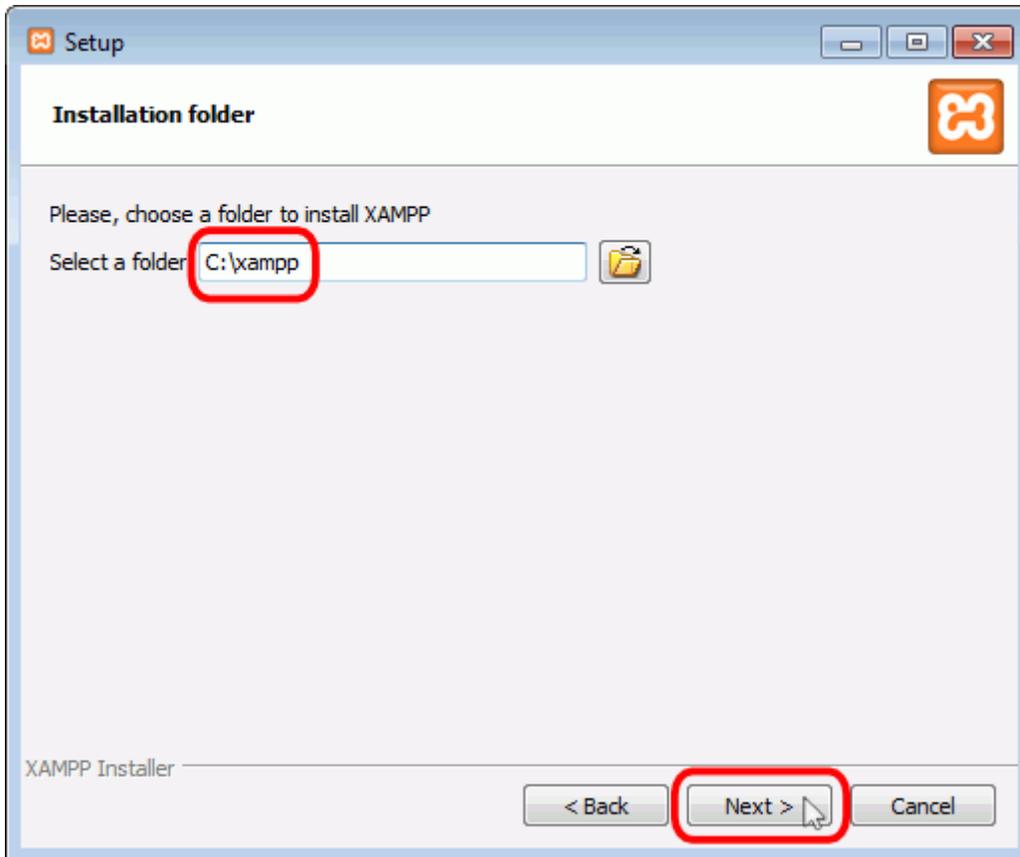
A continuación se inicia el asistente de instalación. Para continuar, hay que hacer clic en el botón "Next".



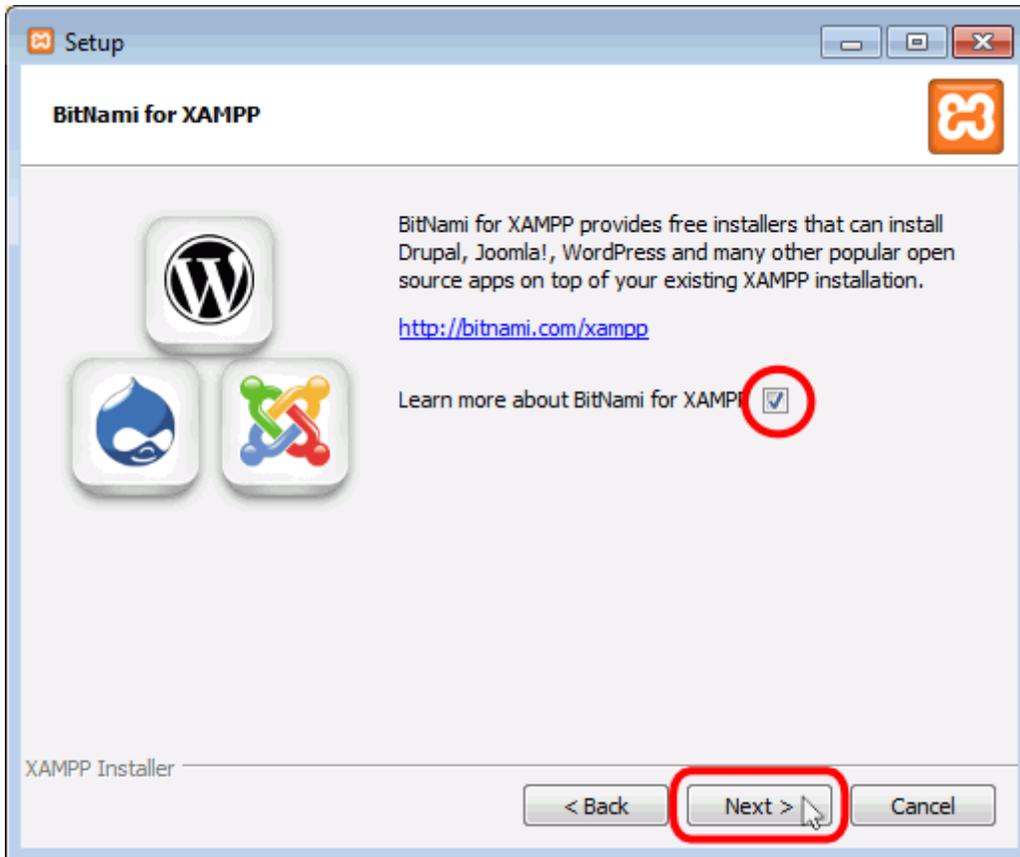
Los componentes mínimos que instala XAMPP son el servidor Apache y el lenguaje PHP, pero XAMPP también instala otros elementos (Para propósitos del sistema no instalaremos lo elementos adicionales). En la pantalla de selección de componentes puede elegirse la instalación o no de estos componentes para propósitos del sistema no instalaremos lo elementos adicionales.



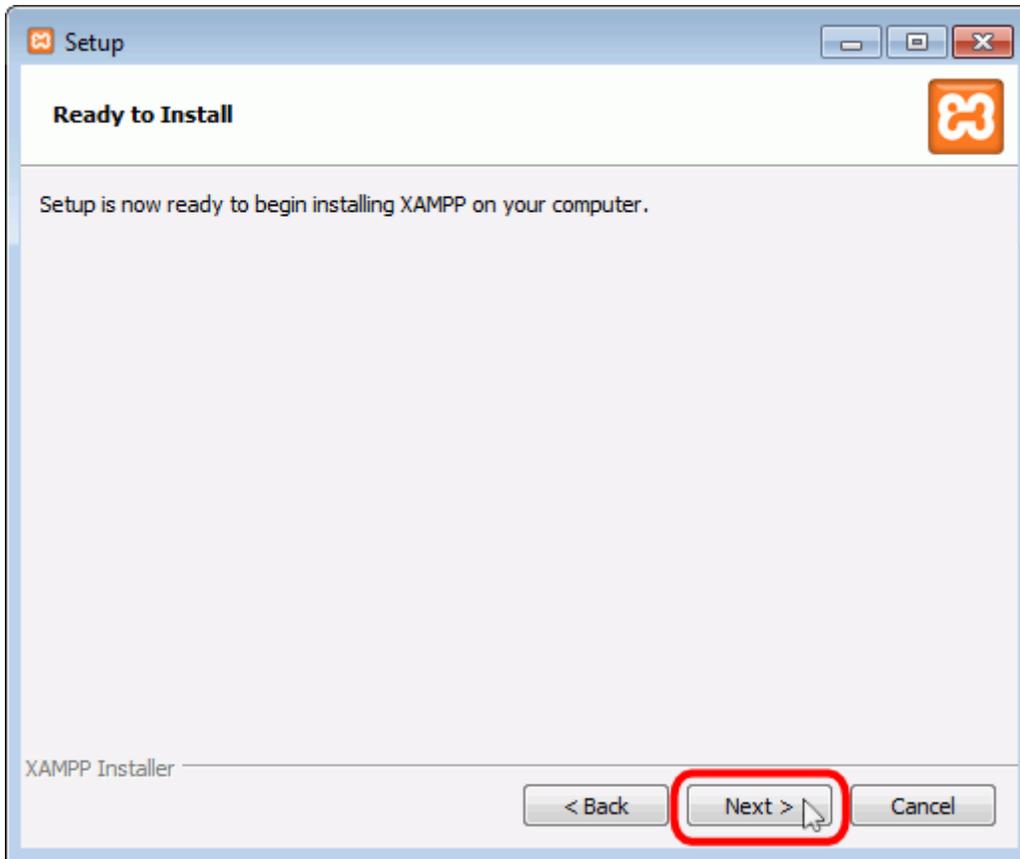
En la siguiente pantalla se puede elegir la carpeta de instalación de XAMPP. La carpeta de instalación predeterminada es **C:\xampp**. Si se quiere cambiar, hay que hacer clic en el icono de carpeta y seleccionar la carpeta donde se quiere instalar XAMPP. Para continuar la configuración de la instalación, hay que hacer clic en el botón "Next".



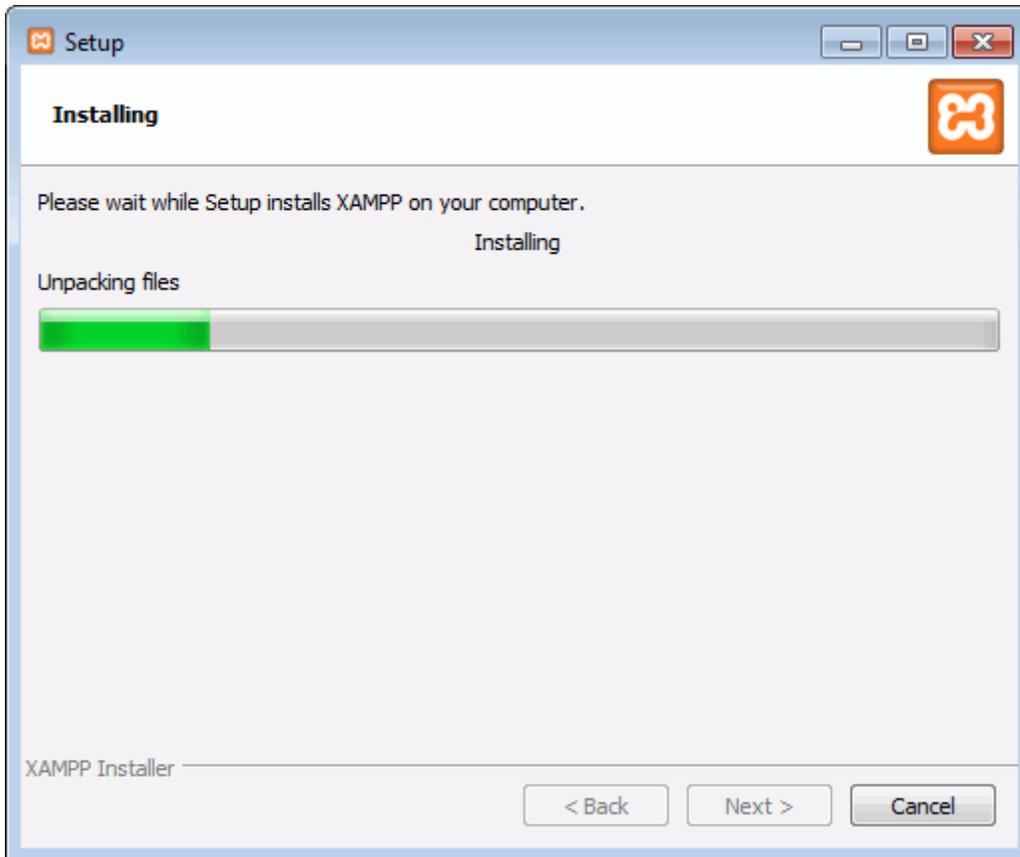
La siguiente pantalla nos ofrece información sobre los instaladores de aplicaciones para XAMPP creados por Bitnami. Para que no se abra la página web de Bitnami, habría que desmarcar la casilla correspondiente.



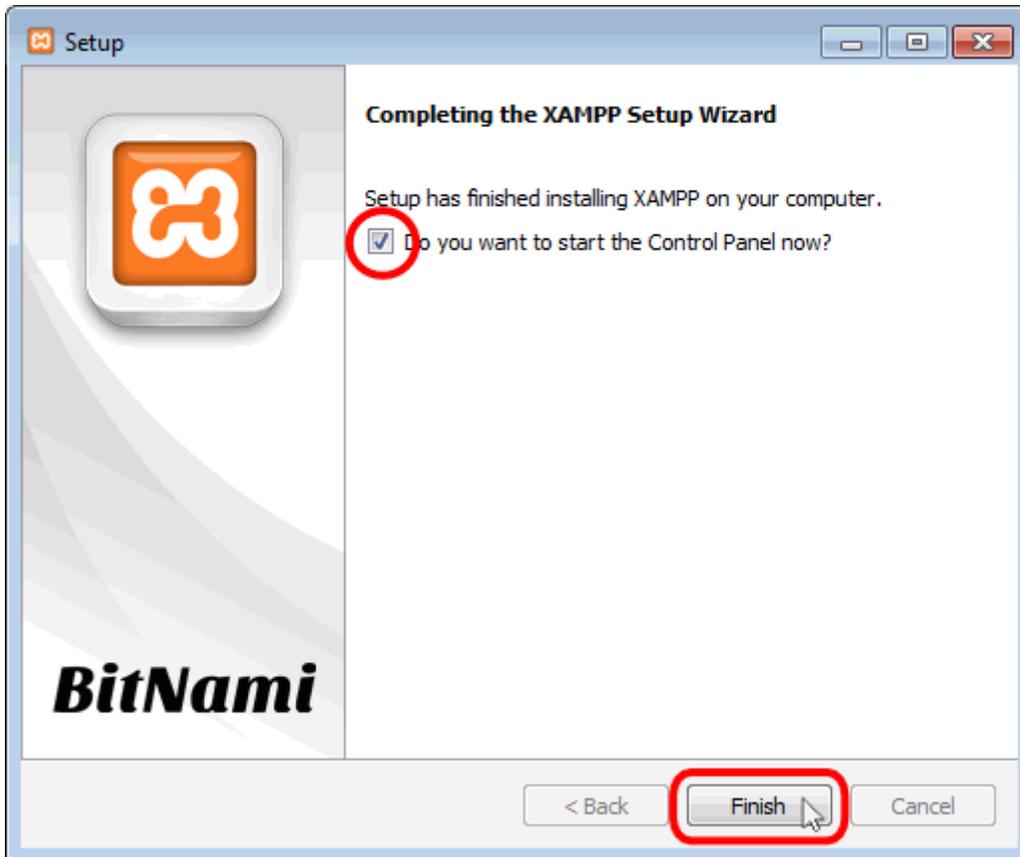
Para empezar la instalación de XAMPP, hay que hacer clic en el botón "Next" en la pantalla siguiente.



A continuación, se inicia el proceso de copia de archivos, que puede durar unos minutos.



Una vez terminada la copia de archivos, se muestra la pantalla que confirma que XAMPP ha sido instalado. Hay que hacer clic en el botón "Finish". Para no abrir a continuación el panel de control de XAMPP hay que desmarcar la casilla correspondiente.



Cuando el servicio Apache este instalado, ábrete una ventana de Firefox (u otro navegador si es que utilizas otro para navegar) y comprueba que el servidor funciona con **http://localhost**



[English](#) / [Deutsch](#) / [Français](#) / [Nederlands](#) / [Polski](#) / [Italiano](#) / [Norsk](#) / [Español](#) / [中文](#) / [Português \(Brasil\)](#) / [日本語](#)

La primera vez saldrá esta página. Elegimos español. Y nos aparecerá la siguiente:

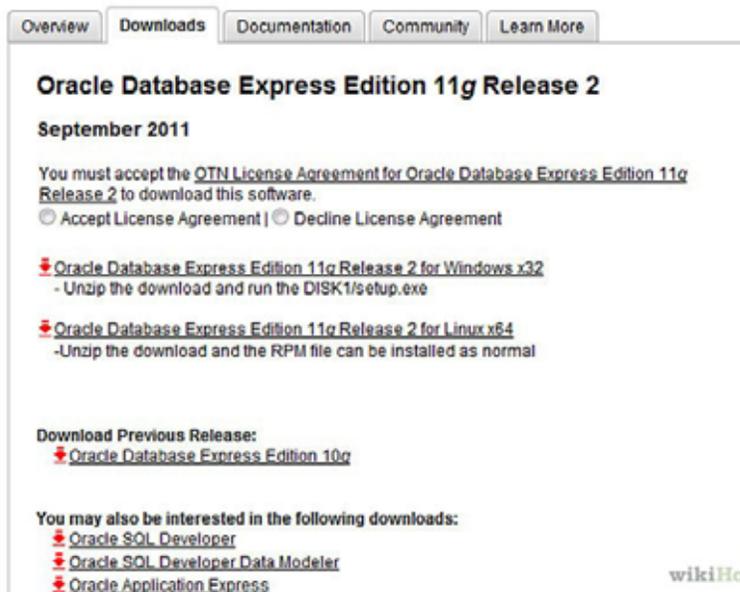


Pulsando Estado podremos ver qué servicios se están ejecutando.

## C. INSTALACION DE ORACLE 11G EXPRESS EDITION

Descargamos XAMPP de la página de Apachefriends:

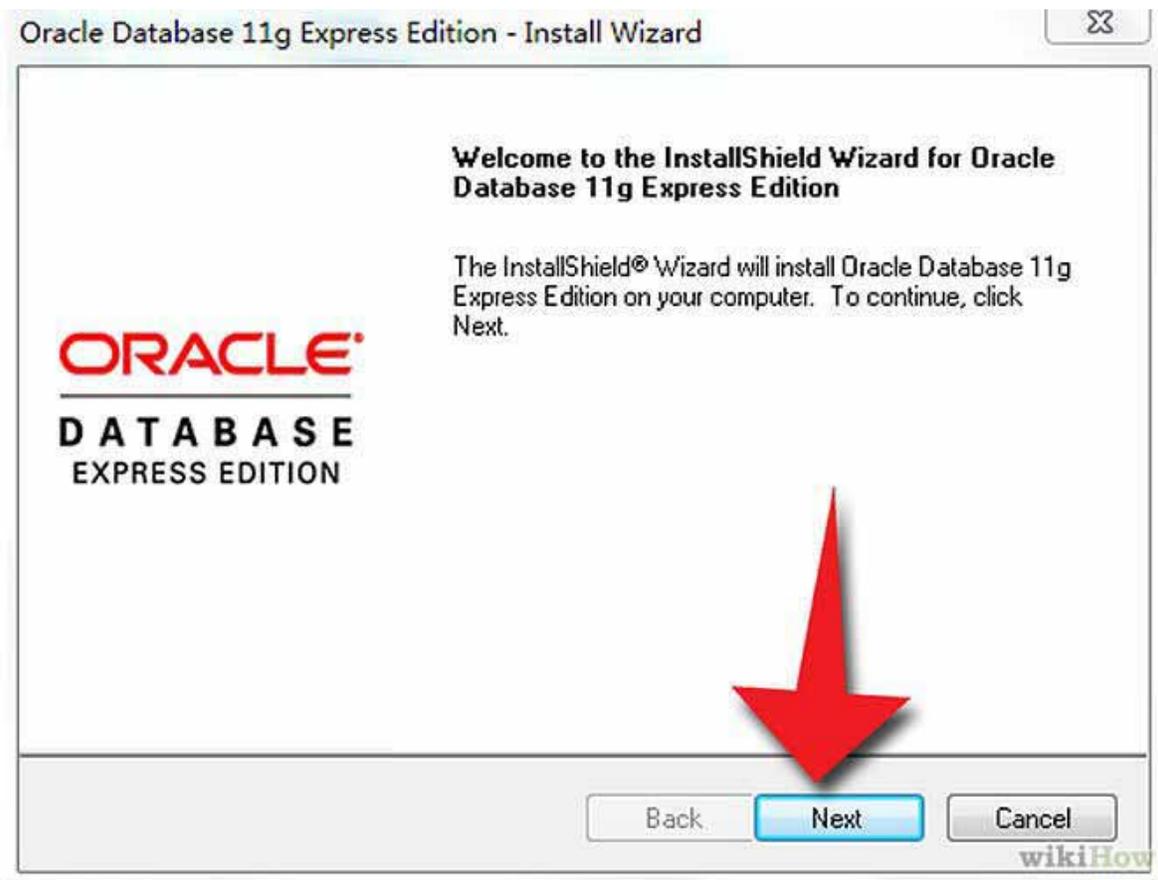
<http://www.oracle.com/technetwork/database/database-technologies/express-edition/downloads/index.html>



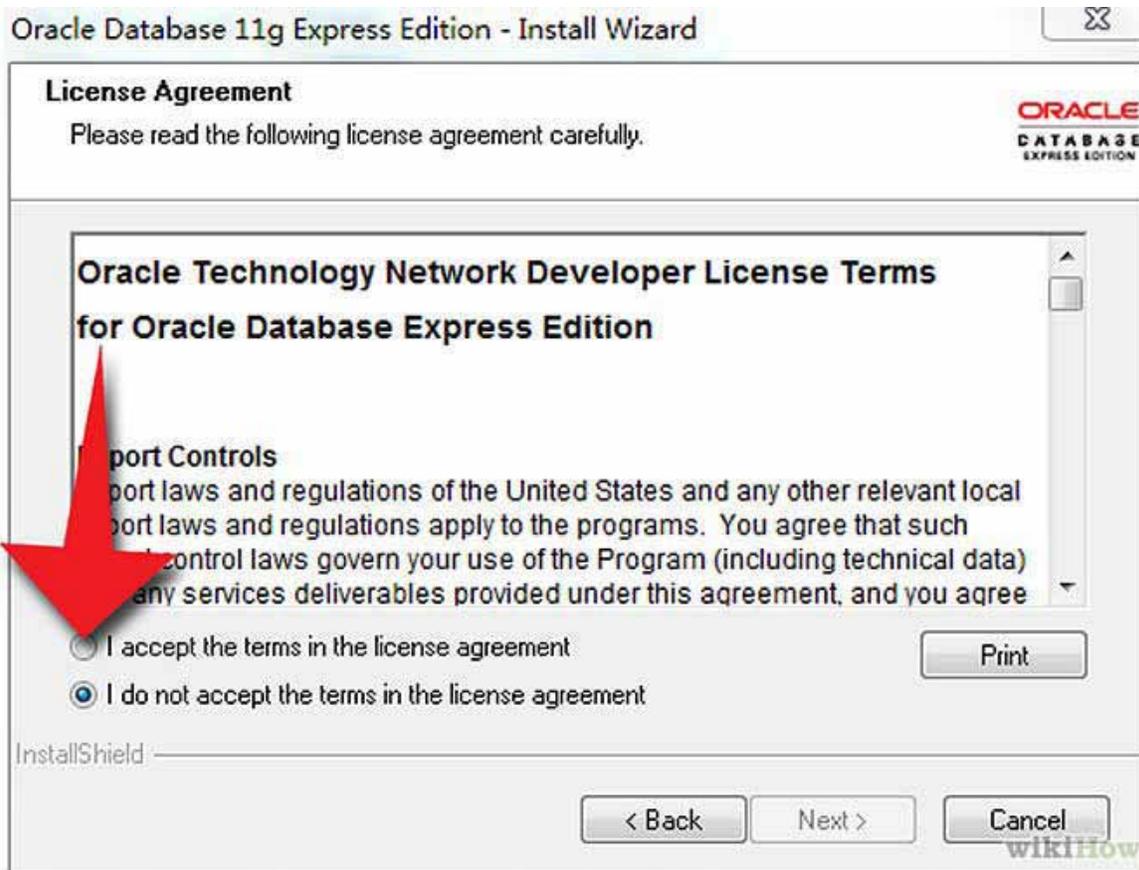
Escoge "Aceptar acuerdos de licencia" primero. Descarga el archivo correcto para tu computadora, ya sea Windows o Linux. Guarda el archivo.

•**Escoge "Aceptar acuerdos de licencia" primero.** Descarga el archivo correcto para tu computadora, ya sea Windows o Linux. Guarda el archivo.

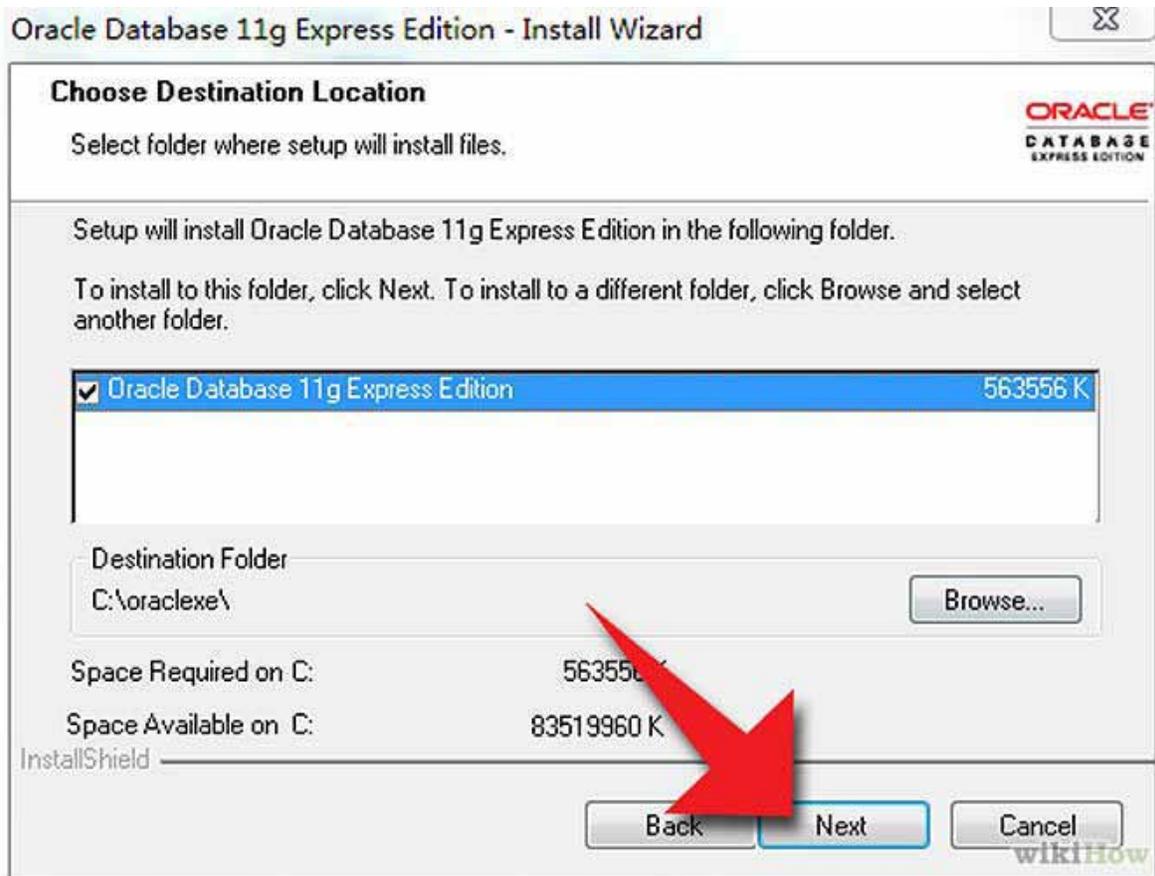
**Encuentra el archivo que descargaste, descomprímelo, haz doble clic en el botón "Instalar" para instalar Oracle.**



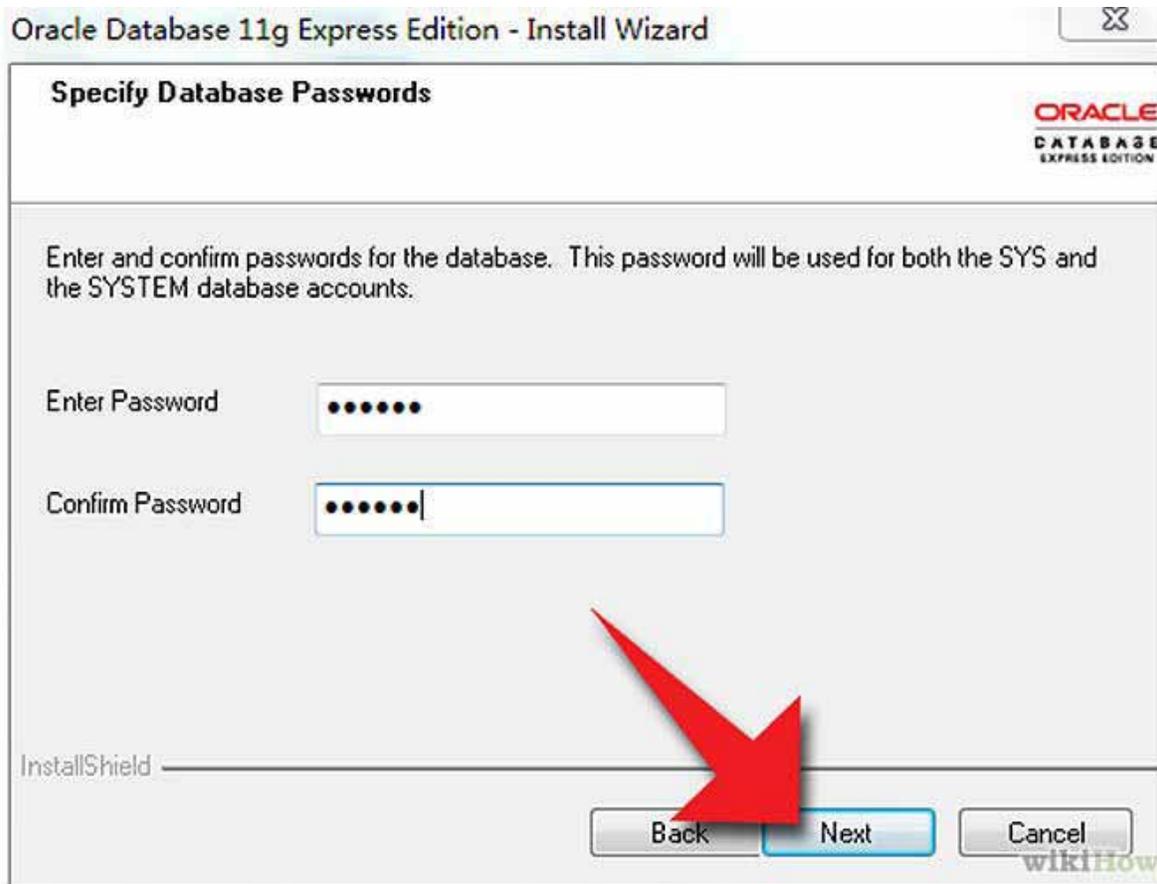
**Haz clic en "siguiente".**



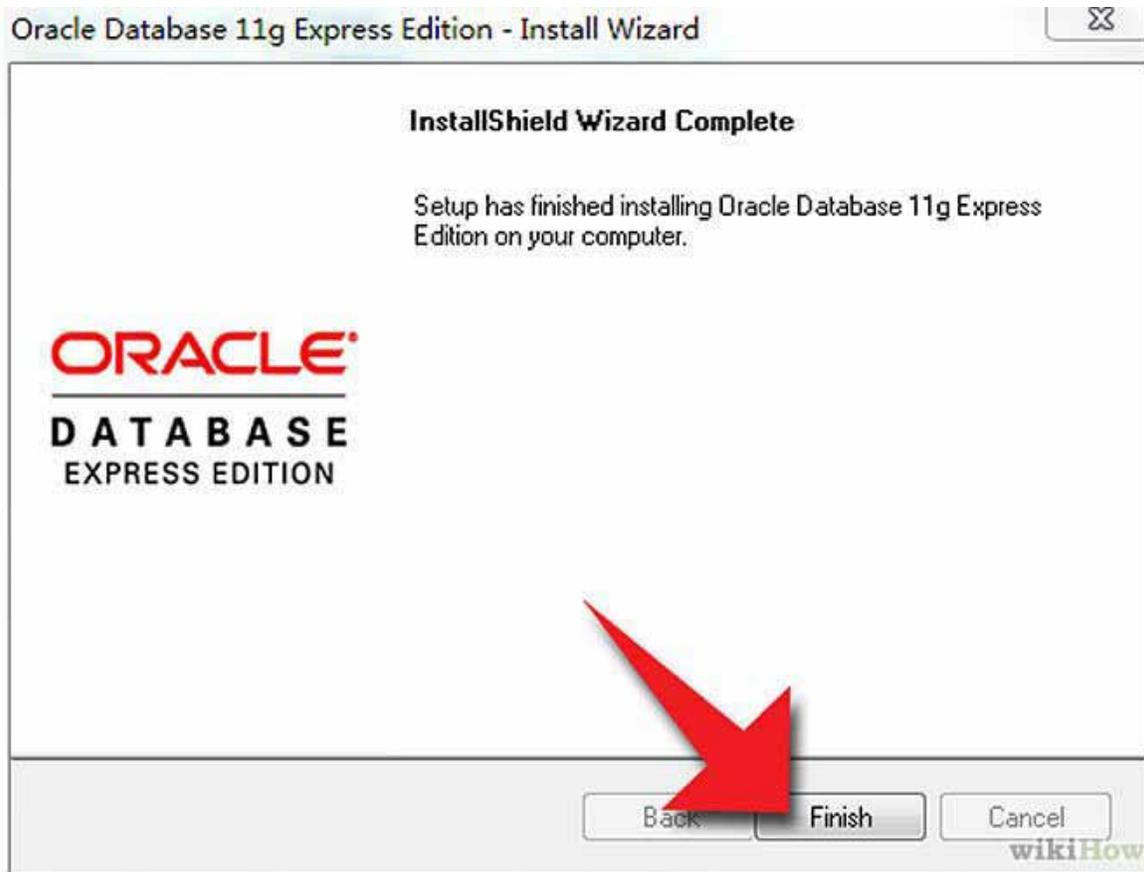
Elige "Yo acepto los términos en el acuerdo de licencia", luego haz clic en siguiente.



**Escoge una carpeta destino y haz clic en "siguiente".**



**Entra y confirma la contraseña de la base de datos (para que el proyecto funcione la contraseña será "oracle"), haz clic en "siguiente".**



Haz clic en "terminar".

## L. RESULTADOS

### A. FUNCIONES

El sistema de información web "Sistema de información para el control de empleados eventuales" tiene como objetivo principal llevar un mejor control de los empleados eventuales, manteniendo de forma centralizada y organizada toda la información.

Sus objetivos específicos son:

- ✓ Mantener actualizada y centralizada la información de los empleados eventuales.
- ✓ Contar con un sistema de información que permita realizar el registro de asistencia y los pagos correspondientes a los empleados.
- ✓ El sistema debe realizar la administración de inasistencias y permisos de los empleados.
- ✓ Crear un módulo que registre y modifique la información de los empleados eventuales.

- ✓ Realizar el diseño e implementación de un módulo que registre la asistencia y los pagos semanales de cada empleado eventual a través de su huella digital.
- ✓ Diseñar e implementar un módulo que registre las horas extras y permisos de los empleados eventuales.
- ✓ Crear un diseño que implemente un módulo que permita el envío de reportes por correo electrónico.
- ✓ Diseñar e implementar una base de datos que permita administrar la información de los empleados.

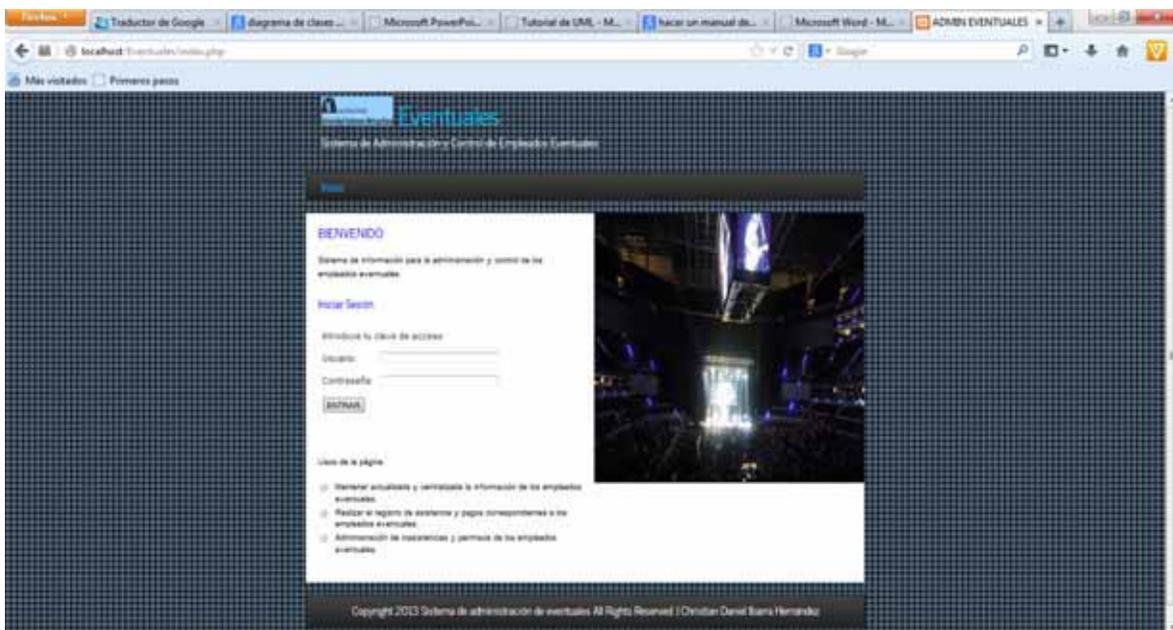
El usuario y contraseña por defecto son:

Usuario: user

Contraseña: 12345

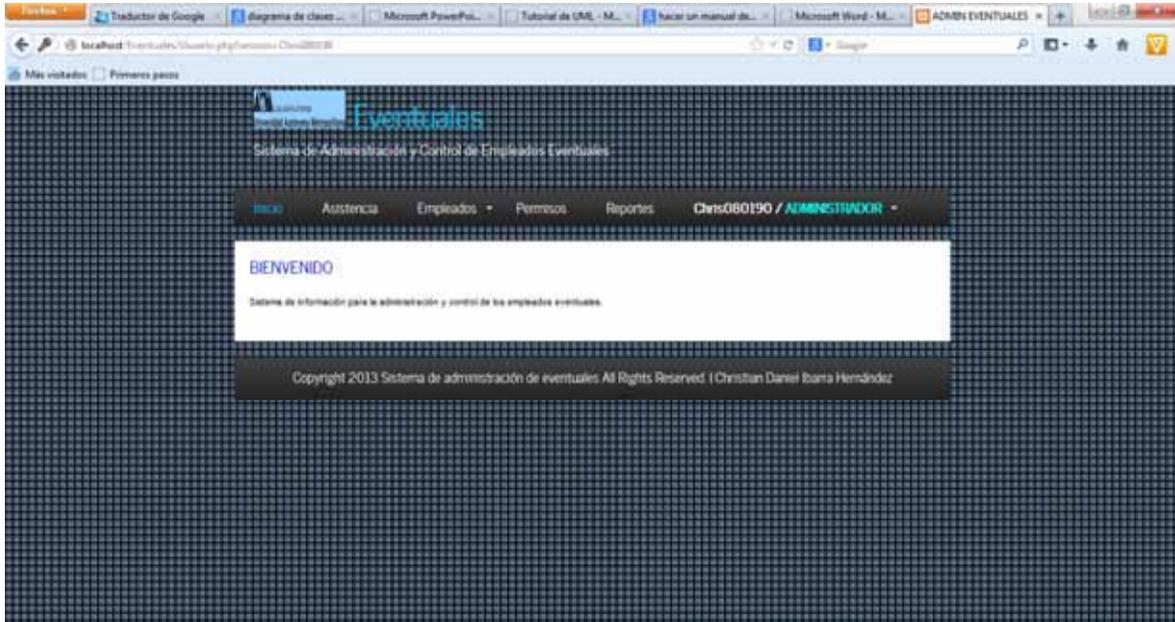
## B. DESCRIPCION DEL FUNCIONAMIENTO

La aplicación web cuenta con una PANTALLA PRINCIPAL en la que se deberá introducir un usuario y contraseña.



Pantalla principal.

De esta forma se va a tener acceso a la PANTALLA DE USUARIO.



Pantalla de Usuarios

### Si desea registrar o tomar asistencia de los empleados eventuales

Haga clic en la palabra asistencia, en la parte superior de la pantalla. Aparecerá la PANTALLA EVENTO cuando no existe un evento la fecha o ya se ha vencido otro.

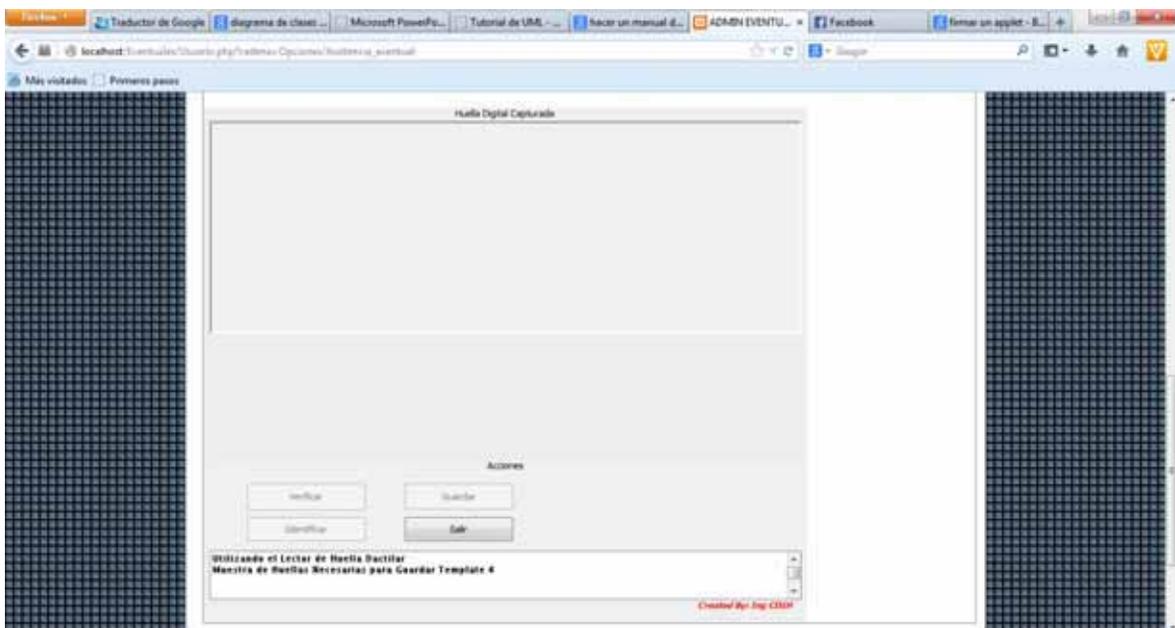


Después de registrar el evento aparecerá la PANTALLA ASISTENCIA EVENTUAL



- ✧ Busque por folio, nombre o apellidos del eventual.
- ✧ Haga clic en buscar.
- ✧ Seleccione y haga clic en el nombre del eventual que desea registrar la asistencia.
- ✧ Haga clic en el botón enviar.
- ✧ Confirme eventual con los datos que aparecen.
- ✧ Marque la casilla de pago extra en caso de que aplique.
- ✧ Haga clic en el botón registrar asistencia.

### Confirmar la identidad del usuario con su huella digital

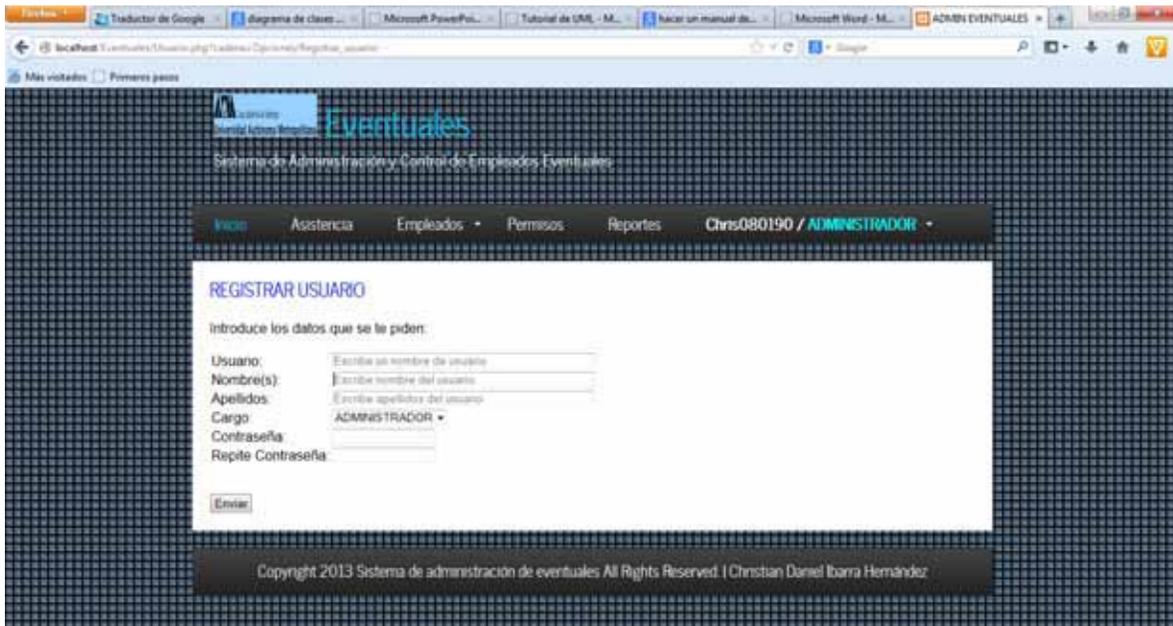


- ✧ Poner en el lector el dedo para escanear huella.

- ✧ Con una lectura se activan los botones verificar e identificar que podrá utilizar para reconocer la huella si ya existe una guardada.
- ✧ Poner 4 veces en el lector el dedo para escanear huella.
- ✧ Se activara el botón guardar que nos permitirá almacenar la huella.
- ✧ Ingrese folio de eventual.

### Si desea registrar usuarios

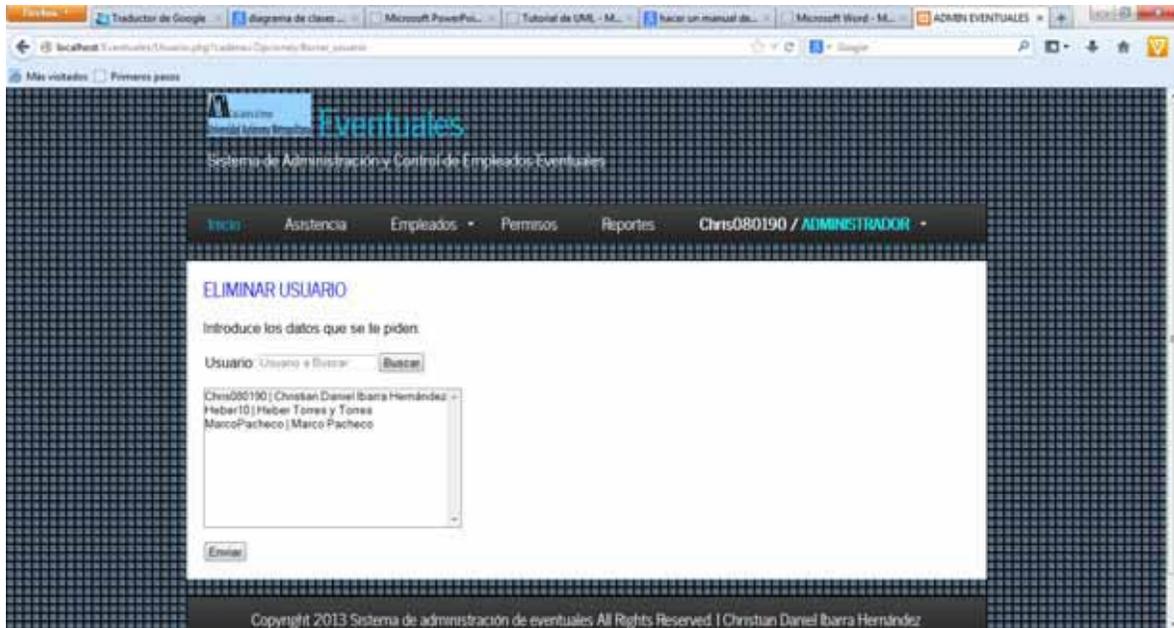
Para registrar un usuario posicione el puntero en la palabra empleados, después posicione el puntero en la palabra Usuarios haga clic sobre la opción registrar en la parte superior de la pantalla. Aparecerá la PANTALLA REGISTRAR USUARIO.



- ✧ Escribir el nombre de usuario que asignara.
- ✧ Escribir nombre del usuario.
- ✧ Escribir Apellidos del usuario.
- ✧ Seleccionar y hacer clic sobre el cargo.
- ✧ Escribir una contraseña temporal que permitirá el acceso al usuario.
- ✧ Escribir la contraseña anterior.
- ✧ Haga clic en enviar para registrar el nuevo usuario.

### Si desea eliminar usuarios

Para eliminar un usuario posicione el puntero en la palabra empleados, después posicione el puntero en la palabra Usuarios haga clic sobre la opción Borrar en la parte superior de la pantalla. Aparecerá la PANTALLA ELIMINAR USUARIO.



- ✧ Busque por nombre de usuario, nombre o apellidos del usuario.
- ✧ Haga clic en buscar.
- ✧ Seleccione y haga clic en el nombre del usuario que desea eliminar.
- ✧ Haga clic en el botón enviar.

### Si desea modificar datos de los usuarios

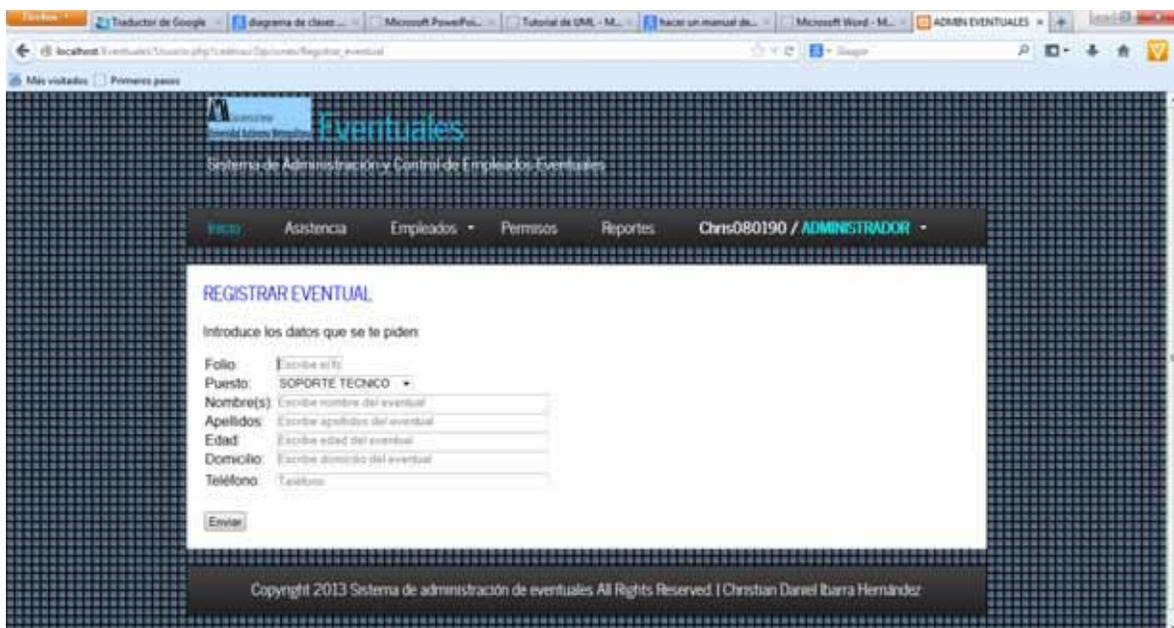
Para modificar un usuario posicione el puntero en la palabra empleados, después posicione el puntero en la palabra Usuarios haga clic sobre la opción Modificar en la parte superior de la pantalla. Aparecerá la PANTALLA ACTUALIZAR USUARIO.



- ✧ Busque por nombre de usuario, nombre o apellidos del usuario.
- ✧ Haga clic en buscar.
- ✧ Seleccione y haga clic en el nombre del usuario que desea modificar.
- ✧ Haga clic en el botón enviar.
- ✧ Modifique Nombre, Apellidos o Cargo del usuario.
- ✧ Haga clic en el botón Actualizar.

### Si desea registrar eventuales

Para registrar un empleado debe posicione el puntero en la palabra empleados, después posicione el puntero en la palabra Eventuales haga clic sobre la opción registrar en la parte superior de la pantalla. Aparecerá la PANTALLA REGISTRAR EVENTUAL.



- ✧ Escribir el folio del eventual.
- ✧ Seleccionar y hacer clic sobre el puesto.
- ✧ Escribir Nombre del eventual.
- ✧ Escribir Apellidos del eventual.
- ✧ Escribir Edad del eventual.
- ✧ Escribir Edad del eventual.
- ✧ Escribir Domicilio del eventual.
- ✧ Escribir Teléfono del eventual.
- ✧ Haga clic en enviar para registrar el nuevo eventual.

### Si desea eliminar eventuales

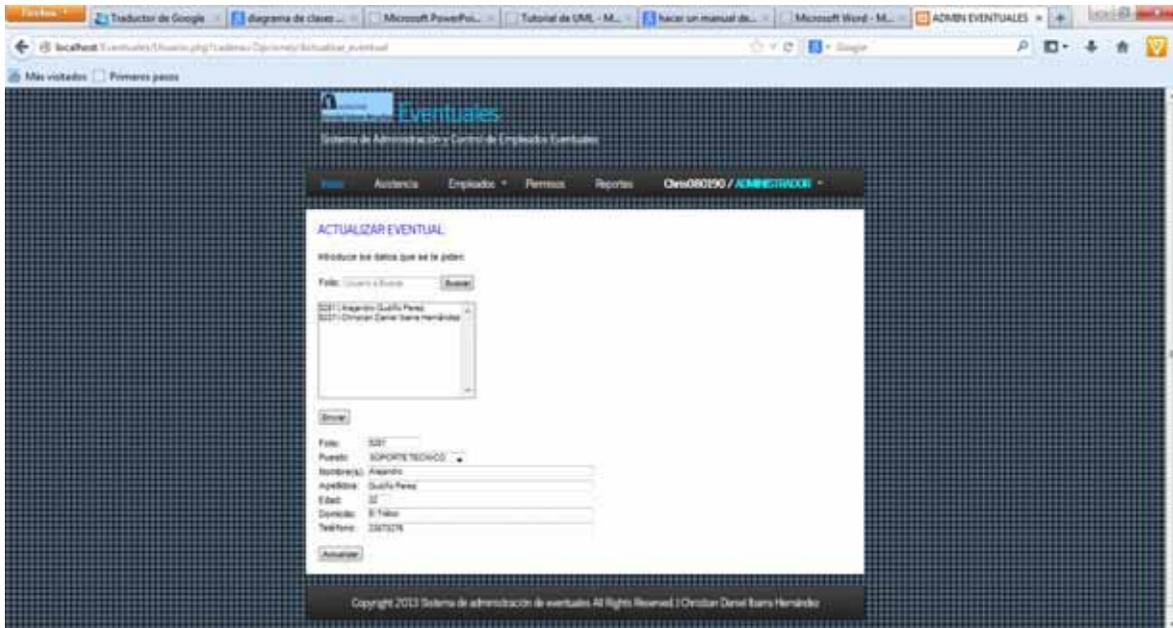
Para eliminar un empleado eventual posicione el puntero en la palabra empleados, después posicione el puntero en la palabra Eventuales haga clic sobre la opción Borrar en la parte superior de la pantalla. Aparecerá la PANTALLA ELIMINAR EVENTUAL.



- ✧ Busque por folio, nombre o apellidos del eventual.
- ✧ Haga clic en buscar.
- ✧ Seleccione y haga clic en el nombre del eventual que desea eliminar.
- ✧ Haga clic en el botón enviar.

#### Si desea modificar datos de los eventuales

Para modificar un usuario posicione el puntero en la palabra empleados, después posicione el puntero en la palabra Eventuales haga clic sobre la opción Modificar en la parte superior de la pantalla. Aparecerá la PANTALLA ACTUALIZAR EVENTUAL.



- ✧ Busque por folio, nombre o apellidos del eventual.
- ✧ Haga clic en buscar.
- ✧ Seleccione y haga clic en el nombre del eventual que desea modificar.
- ✧ Haga clic en el botón enviar.
- ✧ Modifique Puesto, Nombre, Apellidos, Edad, Domicilio o Teléfono del eventual.
- ✧ Haga clic en el botón Actualizar.

### Si desea registrar permisos

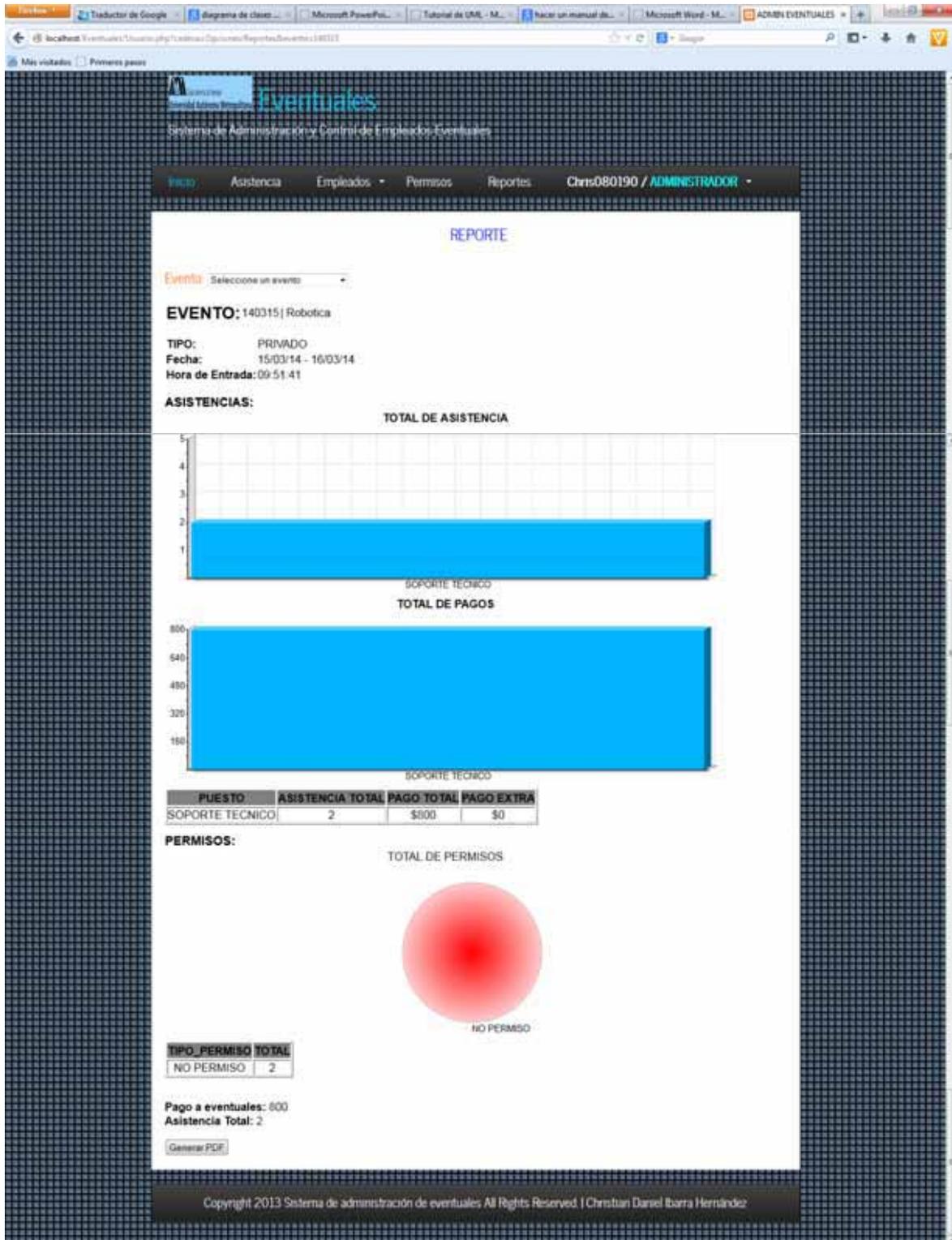
Para registrar un permiso posicione el puntero y haga clic en la palabra Permisos en la parte superior de la pantalla. Aparecerá la PANTALLA PERMISOS.



- ✧ Busque por folio, nombre o apellidos del eventual.
- ✧ Haga clic en buscar.
- ✧ Seleccione y haga clic en el nombre del eventual que desea registrar un permiso.
- ✧ Haga clic en el botón enviar.
- ✧ Confirme datos.
- ✧ Seleccione y de clic en el tipo permiso.
- ✧ Haga clic en Registrar Permiso.

#### Si desea ver estadísticas y generar un reporte

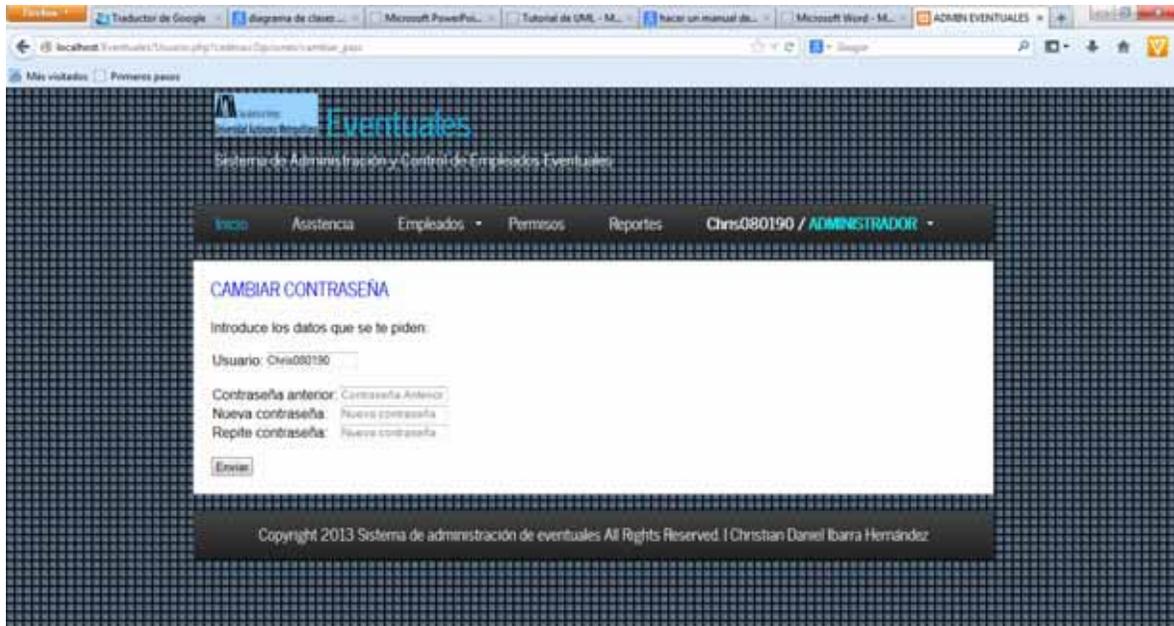
Para visualizar un reporte de las asistencias, pagos y permisos, posicione el puntero y haga clic en la palabra Reportes en la parte superior de la pantalla. Aparecerá la PANTALLA REPORTES.



- ❖ Visualice el reporte preliminar.
- ❖ Haga clic en Generar PDF para generar un reporte en formato digital.

Si desea cambiar su contraseña

Para cambiar contraseña posicione el puntero en la donde aparece su nombre y tipo de usuario, haga clic en la palabra Cambiar Contraseña en la parte superior de la pantalla. Aparecerá la PANTALLA CAMBIAR CONTRASEÑA.



- ✧ Confirme usuario.
- ✧ Escribir contraseña anterior.
- ✧ Escribir una nueva contraseña.
- ✧ Escribir repite la nueva contraseña.
- ✧ Haga clic en enviar para cambiar la contraseña.

Si desea cerrar su sesión en el sistema

Para cerrar sesión posicione el puntero en la donde aparece su nombre y tipo de usuario, haga clic en la palabra Cerrar Sesión en la parte superior de la pantalla. Aparecerá la PANTALLA PRINCIPAL.

## C. ESTRUCTURA

Esta aplicación consta de una misma pantalla por la que el usuario se podrá mover y ver diferentes contenidos para obtener orealizar las operaciones que desee.

### 1. PANTALLA PRINCIPAL



### 1.1. DESCRIPCIÓN

Esta pantalla tiene varias partes informativas que le permitirán darse una idea general de los usos de la página, además contiene los campos necesarios para el iniciar una sesión en el sistema.

- ✓ Campo usuario.  
En este campo deberás introducir el usuario que se te ha proporcionado.
- ✓ Campo contraseña.  
En este campo deberás introducir la contraseña de usuario.
- ✓ Botón Entrar.  
Este botón validar los datos ingresados y te dará acceso a la ventana de usuario.
- ✓ Imágenes de la empresa.  
Da vista de las imágenes de la empresa (instalaciones, logo, personal, etc.).
- ✓ Usos de la pagina.  
Muestra una breve descripción de los usos o utilidades que le puedes dar al sistema.

### 1.2. FUNCIONAMIENTO

- ✧ Ingresar usuario en el campo de usuario
- ✧ Ingresar contraseña en el campo contraseña
- ✧ Presionar la tecla enter o bien dar clic en el botón entrar.
- ✧ Si la validación es correcta aparecerá la PANTALLA DE USUARIO si no aparecerá un mensaje de datos incorrectos.

## 2. PANTALLA DE USUARIO



## 2.1. DESCRIPCIÓN

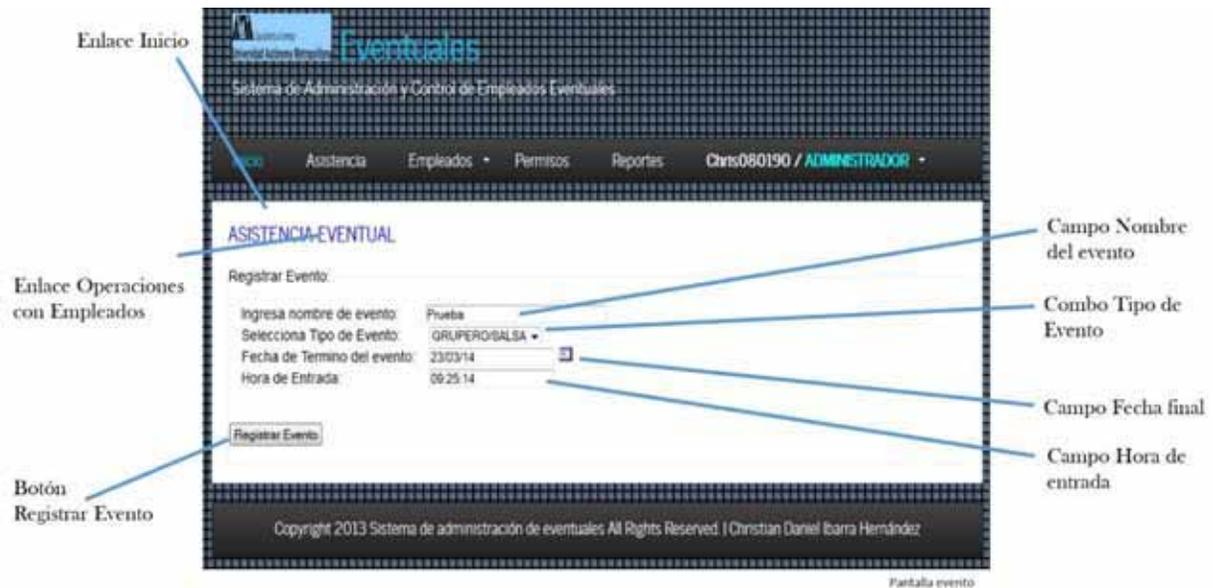
Esta pantalla contiene todos los enlaces necesarios para alta, baja y cambios de empleados, también contiene los enlaces para registrar una asistencia, un permiso o generar un reporte. Tiene un enlace de usuario que nos permitirá cerrar nuestra sesión o cambiar nuestra contraseña y consta de un enlace inicio que nos permitirá volver a nuestra ventana de usuario.

- ✓ Enlace inicio.  
Este enlace permite siempre volver a la ventana de usuario.
- ✓ Enlace Asistencia.  
Este enlace nos muestra la PANTALLA ASISTENCIA.
- ✓ Enlace Operaciones con Empleados.  
Este enlace nos puede mostrar múltiples pantalla dependiendo de la operación que deseamos realizar, las pantallas mostradas por este enlace son:
  - ✧ PANTALLA REGISTRAR USUARIO
  - ✧ PANTALLA ELIMINAR USUARIO
  - ✧ PANTALLA ACTUALIZAR USUARIO
  - ✧ PANTALLA REGISTRAR EVENTUAL
  - ✧ PANTALLA ELIMINAR EVENTUAL
  - ✧ PANTALLA ACTUALIZAR EVENTUAL
- ✓ Enlace Permisos.  
Este enlace nos muestra la PANTALLA PERMISOS.
- ✓ Enlace Reporte.  
Este enlace nos muestra la PANTALLA REPORTES.
- ✓ Enlace Usuario.  
Este enlace nos proporciona dos opciones una de ella nos muestra la PANTALLA CAMBIAR CONTRASEÑA y la otra opción nos permite nuestra sesión activa.

## 2.2. FUNCIONAMIENTO

- ✧ Si hace clic en "inicio".  
Le lleva a la PANTALLA DE USUARIO.
- ✧ Si hace clic en "asistencia".  
Le lleva a la PANTALLA ASISTENCIA EVENTUAL.
- ✧ Si posiciona el puntero en "empleados".  
Se abrirá un menú donde seleccionará el tipo de empleado (usuario/eventual).
  - Si posiciona el puntero en "usuario".  
Se abrirá un submenú donde seleccionará la operación a realizar (registrar/borrar/modificar).
    - Si hace clic en "registrar".  
Le lleva a la PANTALLA REGISTRAR USUARIO.
    - Si hace clic en "borrar".  
Le lleva a la PANTALLA ELIMINAR USUARIO.
    - Si hace clic en "modificar".  
Le lleva a la PANTALLA ACTUALIZAR USUARIO.
  - Si posiciona el puntero en "eventual".  
Se abrirá un submenú donde seleccionará la operación a realizar (registrar/borrar/modificar).
    - Si hace clic en "registrar".  
Le lleva a la PANTALLA REGISTRAR EVENTUAL.
    - Si hace clic en "borrar".  
Le lleva a la PANTALLA ELIMINAR EVENTUAL.
    - Si hace clic en "modificar".  
Le lleva a la PANTALLA ACTUALIZAR EVENTUAL.
- ✧ Si hace clic en "permisos".  
Le lleva a la PANTALLA PERMISOS.
- ✧ Si hace clic en "reportes".  
Le lleva a la PANTALLA REPORTES.
- ✧ Si posiciona el puntero en "usuario/tipo permiso".  
Se abrirá un menú donde seleccionará si desea cambiar contraseña o cerrar sesión.
  - Si hace clic en "cambiar contraseña".  
Le lleva a la PANTALLA CAMBIAR CONTRASEÑA.
  - Si hace clic en "cerrar sesión".  
Cierra su sesión y le lleva a la PANTALLA PRINCIPAL.

### 3. PANTALLA ASISTENCIA EVENTUAL Y EVENTO



#### 3.1. DESCRIPCIÓN

Esta pantalla permite registrar un nuevo evento, aunque solo será mostrada si no existiera un evento o el ultimo evento registrado ya está vencido.

- ✓ Campo evento.  
Este campo de texto permite dar un nombre al evento o descripción.
- ✓ Combo tipo de evento.  
Este es un combo especial que nos permitirá seleccionar un tipo de evento.
- ✓ Campo fecha final.  
Este campo de texto es de un tipo especial porque permite ingresar una fecha de término para el evento de forma escrita o con la imagen de calendario que se encuentra a un lado.
- ✓ Campo Hora de entrada.  
Este campo de texto permite sólo visualizar la hora porque no se podrá escribir en este campo, por defecto toma como hora de entrada; la hora en que se registra el evento.
- ✓ Botón Registrar Evento.  
Este permite validar los campos anteriores y registrar el evento.

#### 3.2. FUNCIONAMIENTO

- ✧ Llenar campo "nombre del evento".
- ✧ Seleccionar tipo de evento y dar clic.
- ✧ Dar clic el calendario y elegir una fecha.
- ✧ Presionar la tecla "enter" o dar clic en el botón "registrar evento".

#### 4. PANTALLA ASISTENCIA EVENTUAL Y EVENTO



##### 4.1. DESCRIPCIÓN

Esta pantalla permite registrar la asistencia de un empleado eventual, aunque solo será mostrada si existe un evento nuevo o el ultimo evento no se ha vencido.

- ✓ Campo buscar.  
Este campo de texto permite buscar un eventual ya sea por nombre, apellido o folio sin tener que saberlo por completo o se puede dejar sin nada.
- ✓ Botón buscar.  
Valida lo escrito en el campo buscar y muestra todas las coincidencias, en caso de no haberse escrito nada muestra la lista completa de los eventuales en el cuadro lista de eventuales.
- ✓ Cuadro lista eventuales.  
Muestra la lista de los eventuales.
- ✓ Botón enviar.  
Valida al empleado eventual seleccionado y muestra la segunda parte de la pantalla.
- ✓ Campo folio.  
Muestra el folio del eventual seleccionado.
- ✓ Campo nombre.  
Muestra el nombre del eventual seleccionado.
- ✓ Campo edad.  
Muestra el nombre del eventual seleccionado.
- ✓ Campo puesto.  
Muestra el puesto del eventual seleccionado.
- ✓ Campo domicilio.  
Muestra el domicilio del eventual seleccionado.
- ✓ Campo teléfono.  
Muestra el teléfono del eventual seleccionado.

- ✓ Check pago extra  
Si es activado este campo permite agregar un pago extra al eventual seleccionado.
- ✓ Botón registrar asistencia  
Este permite validar los campos anteriores y registrar una asistencia al eventual seleccionado.

#### 4.2. FUNCIONAMIENTO

- ✧ Llenar campo "buscar" opcional.
- ✧ Dar clic en el botón "buscar" opcional.
- ✧ Seleccionar y dar clic en el eventual elegido.
- ✧ Verificar datos.
- ✧ Activar la casilla de pago extra opcional.
- ✧ Presionar la tecla "enter" o dar clic en el botón "registrar asistencia".

### 5. PANTALLA REGISTRAR USUARIO

The screenshot shows the 'REGISTRAR USUARIO' form within the 'Eventuales' system. The form is titled 'REGISTRAR USUARIO' and asks the user to 'Introduce los datos que se te piden:'. The fields are: 'Usuario' (with placeholder 'Escribe un nombre de usuario'), 'Nombre(s)' (with placeholder 'Escribe nombre del usuario'), 'Apellidos' (with placeholder 'Escribe apellidos del usuario'), 'Cargo' (a dropdown menu currently showing 'ADMINISTRADOR'), 'Contraseña', and 'Repite Contraseña'. There is an 'Enviar' button at the bottom left. On the right side, blue lines connect labels to the form elements: 'Campo usuario' points to the 'Usuario' field, 'Campo nombre' points to the 'Nombre(s)' field, 'Campo apellidos' points to the 'Apellidos' field, 'Combo cargo' points to the 'Cargo' dropdown, 'Campo contraseña' points to the 'Contraseña' field, and 'Campo repite contraseña' points to the 'Repite Contraseña' field. A label 'Botón enviar' points to the 'Enviar' button. The footer of the page reads 'Copyright 2013 Sistema de administración de eventuales All Rights Reserved. | Christian Daniel Ibarra Hernández' and 'Pantalla registrar usuario'.

#### 5.1. DESCRIPCIÓN

Esta pantalla permite registrar en el sistema un usuario que pueda tener acceso al sistema.

- ✓ Campo usuario.  
Este campo de texto es para escribir el nombre de usuario que tendrá el nuevo usuario del sistema.
- ✓ Campo nombre.  
Este campo de texto es para escribir el nombre del nuevo usuario del sistema.
- ✓ Campo apellidos.  
Este campo de texto es para escribir los apellidos del nuevo usuario del sistema.
- ✓ Combo cargo.  
Este combo permite elegir los privilegios del nuevo usuario del sistema.

- ✓ Campo contraseña.  
Este campo de texto es para escribir una contraseña provisional del nuevo usuario del sistema.
- ✓ Campo repite contraseña.  
Este campo de texto es para escribir una contraseña provisional del nuevo usuario del sistema debe ser la misma que la del campo anterior.
- ✓ Botón enviar.  
Este permite validar los campos anteriores y registrar el nuevo usuario.

## 5.2. FUNCIONAMIENTO

- ✧ Llenar campo "usuario".
- ✧ Llenar campo "nombre".
- ✧ Llenar campo "apellidos".
- ✧ Seleccionar privilegios y dar clic en la selección del combo "cargo".
- ✧ Llenar campo "contraseña".
- ✧ Llenar campo "repite contraseña".
- ✧ Presionar la tecla "enter" o dar clic en el botón "enviar".

## 6. PANTALLA ELIMINAR USUARIO



### 6.1. DESCRIPCIÓN

Esta pantalla permite eliminar del sistema un usuario que por alguna razón ya no deba tener acceso al sistema.

- ✓ Campo buscar.  
Este campo de texto permite buscar un usuario ya sea por nombre, apellido o usuario sin tener que saberlo por completo o se puede dejar sin nada.
- ✓ Botón buscar.

Valida lo escrito en el campo buscar y muestra todas las coincidencias, en caso de no haberse escrito nada muestra la lista completa de los usuarios en el cuadro lista de usuarios.

- ✓ Cuadro lista usuarios.  
Muestra la lista de los usuarios.
- ✓ Botón enviar.  
Valida al usuario seleccionado y lo elimina del sistema.

## 6.2. FUNCIONAMIENTO

- ✧ Llenar campo "buscar" opcional.
- ✧ Dar clic en el botón "buscar" opcional.
- ✧ Seleccionar y dar clic en el usuario elegido.
- ✧ Presionar la tecla "enter" o dar clic en el botón "enviar".

## 7. PANTALLA ACTUALIZAR USUARIO



### 7.1. DESCRIPCIÓN

Esta pantalla permite modificar o actualizar datos de un usuario en el sistema.

- ✓ Campo buscar.  
Este campo de texto permite buscar un usuario ya sea por nombre, apellido o usuario sin tener que saberlo por completo o se puede dejar sin nada.
- ✓ Botón buscar.  
Valida lo escrito en el campo buscar y muestra todas las coincidencias, en caso de no haberse escrito nada muestra la lista completa de los usuarios en el cuadro lista de usuarios.
- ✓ Cuadro lista usuarios.  
Muestra la lista de los usuarios.
- ✓ Botón enviar.  
Valida al usuario seleccionado y lo elimina del sistema.

- Valida al usuario seleccionado y muestra la segunda parte de la pantalla.
- ✓ Campo usuario.  
Muestra el nombre de usuario que será modificado o actualizado.
- ✓ Campo nombre.  
Muestra el nombre y permite escribir otro en caso de que lo requiera.
- ✓ Campo apellidos.  
Muestra los apellidos y permite escribir otros en caso de que lo requiera.
- ✓ Combo cargo.  
Permite modificar los privilegios de usuario para acceso al sistema.
- ✓ Botón enviar.  
Valida los campos modificados y los actualiza en el sistema.

## 7.2. FUNCIONAMIENTO

- ✧ Llenar campo "buscar" opcional.
- ✧ Dar clic en el botón "buscar" opcional.
- ✧ Seleccionar y dar clic en el usuario elegido.
- ✧ Presionar la tecla "enter" o dar clic en el botón "enviar" esto mostrara la siguiente parte de la ventana.
- ✧ Cambiar campo "nombre" opcional.
- ✧ Cambiar campo "apellidos" opcional.
- ✧ Seleccionar y dar clic en el nuevo cargo opcional.
- ✧ Presionar la tecla "enter" o dar clic en el botón "actualizar".

## 8. PANTALLA REGISTRAR EVENTUAL

Botón enviar

Campo folio

Combo puesto

Campo nombre

Campo apellidos

Campo edad

Campo domicilio

Campo teléfono

Copyright 2013 Sistema de administración de eventuales. All Rights Reserved | Christian Daniel Ibarra Hernández

Pantalla registrar eventual

### 8.1. DESCRIPCIÓN

Esta pantalla permite registrar un eventual en el sistema.

- ✓ Campo folio.  
Este campo de texto es para escribir el folio asignado al eventual.
- ✓ Combo puesto.  
Este combo permite elegir el puesto que tiene el eventual.
- ✓ Campo nombre.  
Este campo de texto es para escribir el nombre del eventual.
- ✓ Campo apellidos.  
Este campo de texto es para escribir los apellidos del eventual.
- ✓ Campo edad.  
Este campo de texto es para escribir la edad del eventual.
- ✓ Campo domicilio.  
Este campo de texto es para escribir el domicilio del eventual.
- ✓ Campo edad.  
Este campo de texto es para escribir el teléfono del eventual.
- ✓ Botón enviar.  
Este permite validar los campos anteriores y registrar el nuevo eventual.

## 8.2. FUNCIONAMIENTO

- ✧ Llenar campo "folio".
- ✧ Seleccionar puesto y dar clic en la selección del combo "puesto".
- ✧ Llenar campo "nombre".
- ✧ Llenar campo "apellidos"
- ✧ Llenar campo "edad"
- ✧ Llenar campo "domicilio"
- ✧ Llenar campo "teléfono"
- ✧ Presionar la tecla "enter" o dar clic en el botón "enviar".

## 9. PANTALLA ELIMINAR EVENTUAL



## 9.1. DESCRIPCIÓN

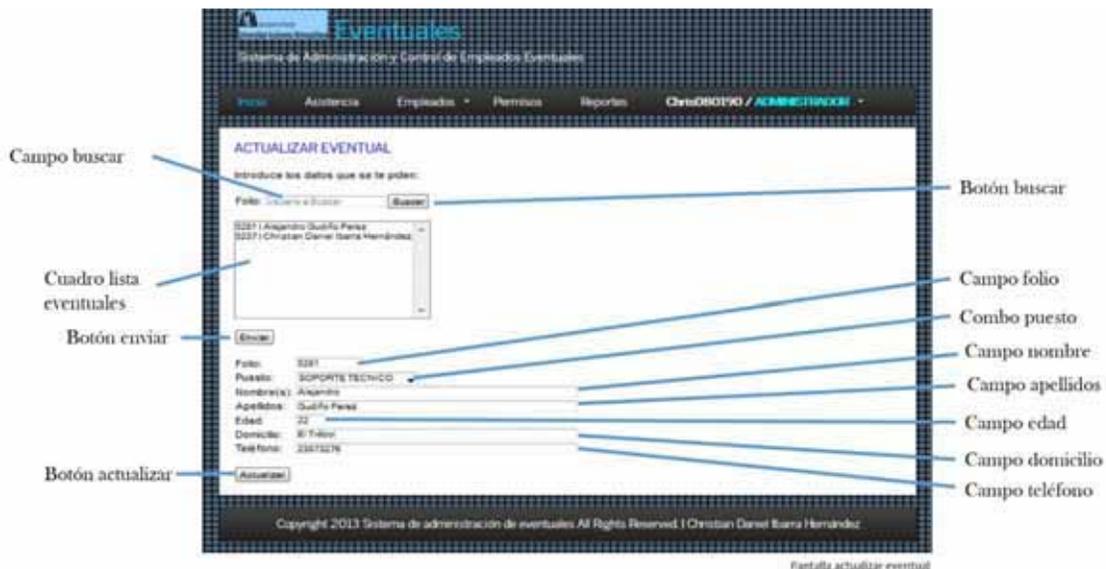
Esta pantalla permite eliminar del sistema un eventual que por alguna razón ya no deba estar en el sistema.

- ✓ Campo buscar.  
Este campo de texto permite buscar un eventual ya sea por nombre, apellido o folio sin tener que saberlo por completo o se puede dejar sin nada.
- ✓ Botón buscar.  
Valida lo escrito en el campo buscar y muestra todas las coincidencias, en caso de no haberse escrito nada muestra la lista completa de los eventuales en el cuadro lista de eventuales.
- ✓ Cuadro lista eventuales.  
Muestra la lista de los eventuales.
- ✓ Botón enviar.  
Valida al eventual seleccionado y lo elimina del sistema.

## 9.2. FUNCIONAMIENTO

- ✧ Llenar campo "buscar" opcional.
- ✧ Dar clic en el botón "buscar" opcional.
- ✧ Seleccionar y dar clic en el eventual elegido.
- ✧ Presionar la tecla "enter" o dar clic en el botón "enviar".

## 10. PANTALLA ACTUALIZAR EVENTUAL



### 10.1. DESCRIPCIÓN

Esta pantalla permite modificar o actualizar datos de un eventual en el sistema.

- ✓ Campo buscar.

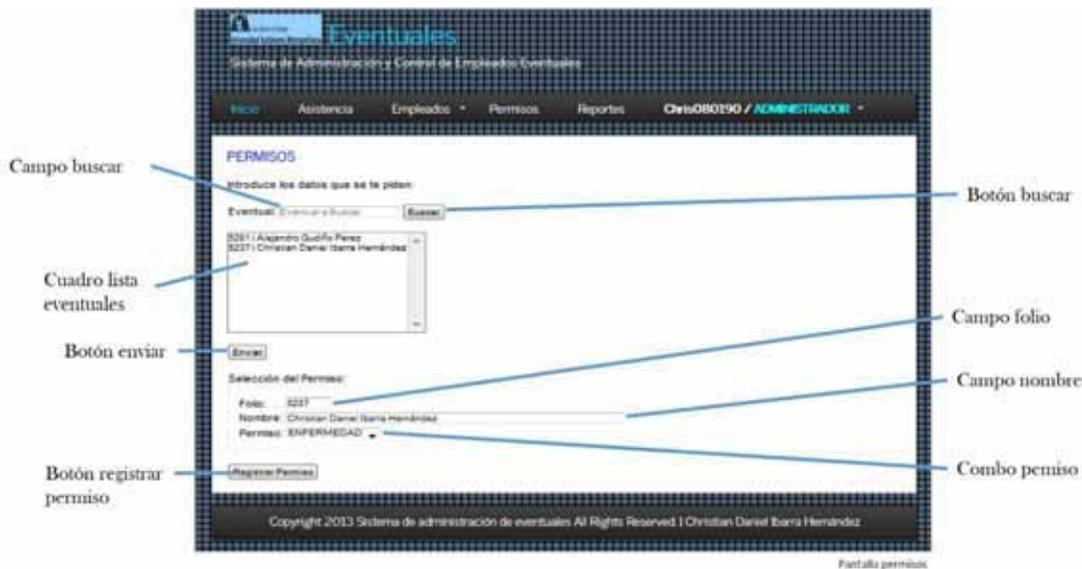
Este campo de texto permite buscar un usuario ya sea por nombre, apellido o usuario sin tener que saberlo por completo o se puede dejar sin nada.

- ✓ Botón buscar.  
Valida lo escrito en el campo buscar y muestra todas las coincidencias, en caso de no haberse escrito nada muestra la lista completa de los usuarios en el cuadro lista de eventuales.
- ✓ Cuadro lista eventuales.  
Muestra la lista de los eventuales.
- ✓ Botón enviar.  
Valida el eventual seleccionado y muestra la segunda parte de la pantalla.
- ✓ Campo folio.  
Muestra el folio del eventual que será modificado o actualizado.
- ✓ Combo puesto.  
Permite modificar el puesto del eventual en el sistema.
- ✓ Campo nombre.  
Muestra el nombre y permite escribir otro en caso de que lo requiera.
- ✓ Campo apellidos.  
Muestra los apellidos y permite escribir otros en caso de que lo requiera.
- ✓ Campo edad.  
Muestra la edad y permite escribir otra en caso de que lo requiera.
- ✓ Campo domicilio.  
Muestra el domicilio y permite escribir otro en caso de que lo requiera.
- ✓ Campo teléfono.  
Muestra el teléfono y permite escribir otro en caso de que lo requiera.
- ✓ Botón enviar.  
Valida los campos modificados y los actualiza en el sistema.

## 10.2.FUNCIONAMIENTO

- ✧ Llenar campo "buscar" opcional.
- ✧ Dar clic en el botón "buscar" opcional.
- ✧ Seleccionar y dar clic en el eventual elegido.
- ✧ Presionar la tecla "enter" o dar clic en el botón "enviar" esto mostrara la siguiente parte de la ventana.
- ✧ Cambiar campo "nombre" opcional.
- ✧ Cambiar campo "apellidos" opcional.
- ✧ Cambiar campo "edad" opcional.
- ✧ Cambiar campo "domicilio" opcional.
- ✧ Cambiar campo "teléfono" opcional.
- ✧ Presionar la tecla "enter" o dar clic en el botón "actualizar".

## 11. PANTALLA PERMISOS



### 11.1. DESCRIPCIÓN

Esta pantalla permite registrar un permiso a un empleado eventual.

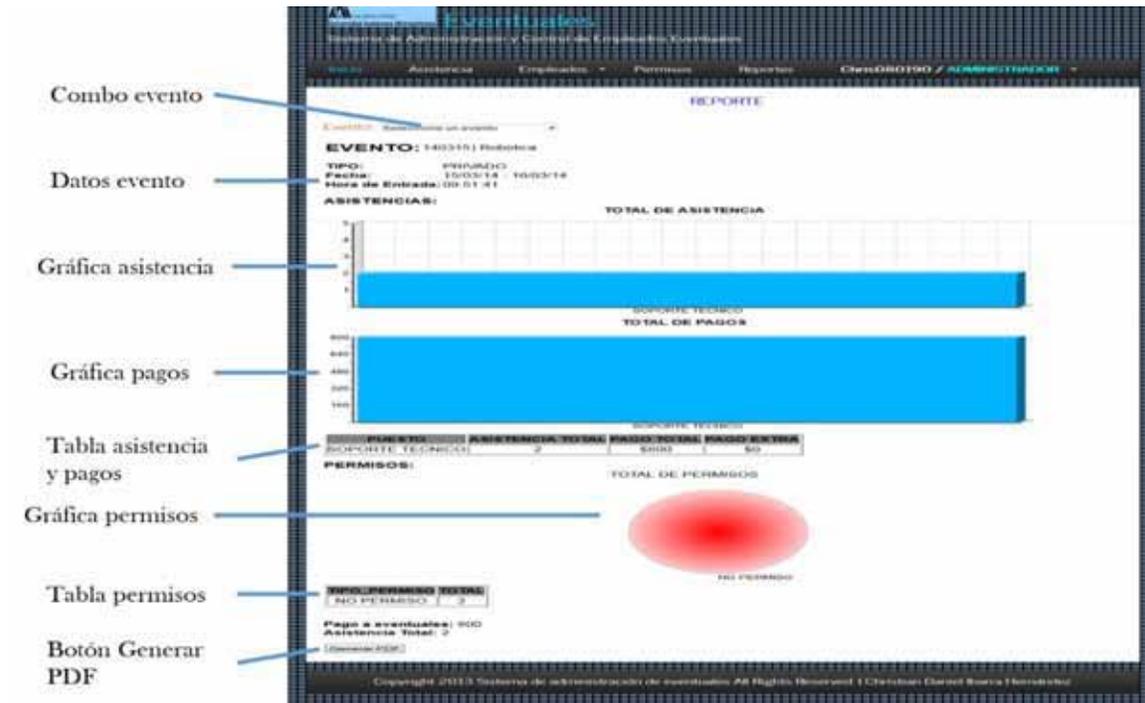
- ✓ Campo buscar.  
Este campo de texto permite buscar un usuario ya sea por nombre, apellido o usuario sin tener que saberlo por completo o se puede dejar sin nada.
- ✓ Botón buscar.  
Valida lo escrito en el campo buscar y muestra todas las coincidencias, en caso de no haberse escrito nada muestra la lista completa de los usuarios en el cuadro lista de eventuales.
- ✓ Cuadro lista eventuales.  
Muestra la lista de los eventuales.
- ✓ Botón enviar.  
Valida el eventual seleccionado y muestra la segunda parte de la pantalla.
- ✓ Campo folio.  
Muestra el folio del eventual al que se le registrara el permiso.
- ✓ Campo nombre.  
Muestra el nombre del eventual al que se le registrara el permiso.
- ✓ Combo permiso.  
Permite asignar el tipo de permiso que se le asignara la eventual.
- ✓ Botón registrar permiso.  
Valida los campos y los registra el permiso en el sistema.

### 11.2. FUNCIONAMIENTO

- ✧ Llenar campo "buscar" opcional.
- ✧ Dar clic en el botón "buscar" opcional.
- ✧ Seleccionar y dar clic en el eventual elegido.
- ✧ Presionar la tecla "enter" o dar clic en el botón "enviar" esto mostrara la siguiente parte de la ventana.
- ✧ Validar "folio".

- ✧ Validar "nombre".
- ✧ Seleccionar permiso y dar clic en la selección del combo "permiso".
- ✧ Presionar la tecla "enter" o dar clic en el botón "registrar permiso".

## 12. PANTALLA ASISTENCIA



### 12.1. DESCRIPCIÓN

Esta pantalla visualizar las estadísticas de un evento seleccionado.

- ✓ Combo evento.  
Permite seleccionar el evento que nos interesa para mostrar su reporte.
- ✓ Datos evento.  
Muestra los datos generales del evento seleccionado.
- ✓ Grafica asistencia  
Muestra la gráfica general de la asistencia del evento seleccionado.
- ✓ Grafica pagos.  
Muestra la gráfica general de los pagos realizados para el evento seleccionado.
- ✓ Tabla asistencia y pagos.  
Muestra la tabla de los datos generales mostrados en las gráficas anteriores.
- ✓ Grafica permisos.  
Muestra la gráfica general de los permisos solicitados para el evento seleccionado.
- ✓ Tabla permisos.  
Muestra la tabla de los datos generales mostrados en la gráfica anterior.
- ✓ Botón generar reporte.

Envía a una ventana donde se puede visualizar el reporte anterior en un formato digital (PDF).

## 12.2. FUNCIONAMIENTO

- ✧ Seleccionar y dar clic en el evento que deseas consultar.
- ✧ Dar clic en el botón "generar reporte".

## 13. PANTALLA CAMBIAR CONTRASEÑA

La imagen muestra una captura de pantalla de la interfaz de usuario de un sistema de administración de empleados eventuales. El título de la página es "Eventuales" y el subtítulo es "Sistema de Administración y Control de Empleados Eventuales". En la parte superior, hay un menú de navegación con opciones como Inicio, Asistencia, Empleados, Permisos, Reportes, y un usuario logueado como "Chris080190 / ADMINISTRADOR".

El formulario principal se titula "CAMBIAR CONTRASEÑA" y contiene el texto "Introduce los datos que se te piden:". Los campos de entrada son:

- Usuario: Chris080190
- Contraseña anterior: Contraseña anterior
- Nueva contraseña: Nueva contraseña
- Repite contraseña: Nueva contraseña

Debajo de los campos hay un botón "Enviar".

Las etiquetas de los campos a la derecha del formulario son:

- Campo usuario
- Campo contraseña anterior
- Campo nueva contraseña
- Campo repite contraseña

El botón "Enviar" está etiquetado como "Botón Enviar".

En la parte inferior de la pantalla, se muestra el copyright: "Copyright 2013 Sistema de administración de eventuales All Rights Reserved | Christian Daniel Ibarra Hernández".

### 13.1. DESCRIPCIÓN

Esta pantalla cambiar la contraseña de usuario.

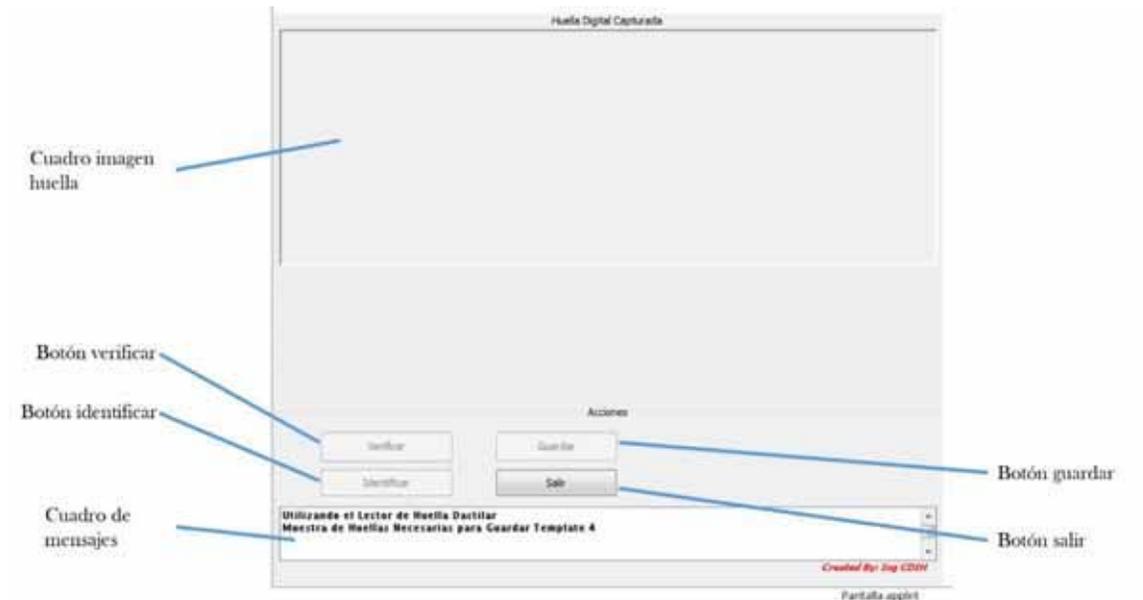
- ✓ Campo usuario  
Muestra el nombre de usuario que se está modificando su contraseña.
- ✓ Campo contraseña  
En este campo se debe ingresar la contraseña actual.
- ✓ Campo nueva contraseña  
En este campo se debe ingresar la nueva contraseña que se desea.
- ✓ Campo nueva contraseña  
En este campo se debe ingresar y repetir la nueva contraseña que se desea.
- ✓ Botón enviar  
Valida los campos y modifica la contraseña de usuario.

### 13.2. FUNCIONAMIENTO

- ✧ Validar nombre de usuario.
- ✧ Llenar campo "contraseña anterior".

- ✧ Llenar campo "nueva contraseña".
- ✧ Llenar campo "repetir contraseña" con la que se ingresó en el campo anterior.
- ✧ Dar clic en el botón "enviar".

## 14. PANTALLA APPLLET



### 14.1. DESCRIPCIÓN

Esta pantalla nos permitirá leer las características de la huella, así como identificarla, verificarla y almacenarla.

- ✓ Cuadro imagen.  
Nos mostrara una imagen de la huella.
- ✓ Botón verificar.  
Nos permitirá validar la huella digital en base al folio del eventual.
- ✓ Botón verificar.  
Nos permitirá validar la huella digital en base al folio del eventual.
- ✓ Botón identificar.  
Nos permitirá validar la huella digital directamente.
- ✓ Botón identificar.  
Nos permitirá validar la huella digital directamente.
- ✓ Botón guardar.  
Permite almacenar la huella y pedirá el folio del eventual para ese propósito.

### 14.2. FUNCIONAMIENTO

- ✧ Para validar huella.
  - Colocar una vez el dedo sobre el lector el dedo sobre el lector.
  - Dar clic en el botón verificar.

- Ingresar folio del eventual
  - Dar clic en botón identificar
- ✧ Para guardar huella.
  - Colocar una cuatro veces el dedo sobre el lector el dedo sobre el lector.
  - Dar clic en el botón guardar.
  - Ingresar folio del eventual.
  - Dar clic en botón identificar.

## M. Análisis y Discusión de Resultados

### 1. Huella Digital con Lector Digital Persona

La tecnología de digital Persona está reconocida actualmente como una de las mejores del mundo, e incluye todos componentes necesarios de Hardware y Software para poder ofrecer una solución completa para cualquier tipo de negocio, incluyendo soluciones de identificación vía Web. Nuestra estrecha relación con digital Persona nos permite llevar soluciones probadas inmediatamente a nuestros clientes, con varios casos de éxitos probados en México y más de 70,000 sensores biométricos instalados a la fecha.

Utilizando un lector biométrico de huella digital se reconoce al empleado eventual, utilizando algoritmo librerías desarrolladas por la empresa digital persona que nos permiten leer los patrones de huella, mostrar imagen de la lectura y guardar plantillas. Con la ayuda de una base de datos esas características se guardan para después ser comparadas para próximas lecturas.

En este sistema en particular la parte más complicada fue la creación de la interfaz applet porque si bien lee, verifica, valida y guarda los patrones de la huella de cada empleado eventual también se tiene que como applet no funciona bien como debería se queda trabado cuando quiere mostrar la huella, sin embargo en forma de aplicación local funciona excelente. Derivado de esto se hace notar que esto pasa con el applet puesto que la capacidad de procesamiento es mayor y no se logra de forma correcta su funcionamiento.

### 2. Seguridad del sistema algoritmo de encriptación Blowfish

Blowfish es un codificador de bloques simétricos, diseñado por Bruce Schneier en 1993 e incluido en un gran número de conjuntos de codificadores y productos de cifrado. Las claves de expansión convierten una clave de más de 448 bits en varias subclaves que totalizan 4168 bytes. Blowfish utiliza un gran número de subclaves que deben ser preprocesadas antes de cualquier proceso de cifrado o descifrado.

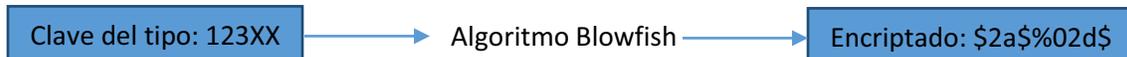
Utilizando el algoritmo de encriptación blowfish implementado en php se permite al usuario encriptar su contraseña así ningún usuario que tuviera acceso a la base de datos podría visualizar dicha contraseña así en algoritmo se utiliza de la forma siguiente con el siguiente código php.

```
<?php
function crypt_blowfish($password, $digito = 8)
{
```

```

$set_salt = './1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz';
$salt = sprintf('$2a$%02d$', $digito);
for($i = 0; $i < 22; $i++)
{
    $salt .= $set_salt[mt_rand(0, 62)];
}
return crypt($password, $salt);
}
?>

```



## N. Conclusiones

El desarrollo de este proyecto servirá para el control de los eventuales, permitirá saber cuantos asisten, cuando se paga por su asistencia, cuantos permisos solicitan y controlar de mejor forma su asistencia.

Utilizar lectores biométricos son actualmente de los métodos más seguros para el reconocimiento de identidad y si a eso le agregamos métodos de encriptación esto permitirá mantener datos más seguros para las empresas y evitar robos de identidad

La implementación de Java Swing, Javascript, HTML5 y PHP en el sector industrial pueden contribuir en gran medida a la solución de problemas en los que se requiera un amplio análisis de cientos o millones de datos, ya que nos proveen herramientas las cuales podemos explotar obteniendo ganancias en costos y tiempos.

Finalmente, la correcta aplicación de las metodologías de programación, la interoperabilidad que actualmente se encuentra en desarrollo entre los diferentes lenguajes de programación, servirá para incrementar la productividad y la máxima reducción de costos dentro de un sector por ejemplo industrial.

## O. Bibliografía

- A. Álvarez Hernández. "Programación de cursos y gestión de inscripciones". Proyecto terminal. División de Ciencias Básicas e Ingeniería. Universidad Autónoma Metropolitana Azcapotzalco. México. 2012.
- C. G. Rubio Alemán. "Sistema de preregistro para aspirantes de la Maestría en Ciencias de la Computación". Proyecto terminal. División de Ciencias Básicas e Ingeniería. Universidad Autónoma Metropolitana Azcapotzalco. México. 2013.

- H. Iturbe García. "Sistema de gestión de nómina para PyMEs". Proyecto terminal. División de Ciencias Básicas e Ingeniería. Universidad Autónoma Metropolitana Azcapotzalco. México. 2010.
- C. A Carrizales Mayorga. "Sistema de huella digital para el registro de asistencia de la D.C.B.". Tesis. Facultad de Ingeniería. Universidad Nacional Autónoma de México. México. 2011.
- O. Ibáñez Orozco. "Diseño de un sistema reacción-difusión para huellas digitales". Tesis doctoral. Facultad de Ingeniería. Universidad Nacional Autónoma de México. México. 2008.
- E. Martínez Gonzáles. "Desarrollo de un sistema de control automatizado para registro de asistencia". Tesis. Facultad de Estudios Superiores Cuautitlán. Universidad Nacional Autónoma de México. 2010.
- Convertir ficheros php a pdf [En línea]. Disponible en: <http://krack.artinfinity.net/programacion/convertir-ficheros-php-a-pdf/>
- Como encriptar contraseñas y hacerlas más seguras con PHP [En línea]. Disponible en: <http://www.dinvaders.com/como-encriptar-contrasenas-y-hacerlas-mas-seguras-con-php/>
- Tutorial de sesiones en PHP [En línea]. Disponible en: <http://www.desarrolloweb.com/articulos/235.php>
- Programación de U.are.U 4500 Fingerprint Reader Digital Persona en Java con W7 [En línea]. Disponible en: <http://blogjarc.blogspot.mx/2011/12/programacion-de-4500-fingerprint-reader.html>
- RGraph: HTML5 charts library [En línea]. Disponible en: <http://www.rgraph.net/>
- Formularios [En línea]. Disponible en: [http://www.mclibre.org/consultar/amaya/xhtml/xhtml\\_formularios.html](http://www.mclibre.org/consultar/amaya/xhtml/xhtml_formularios.html)
- SQL Consultas más complejas [En línea]. Disponible en: <http://www.slideshare.net/alexmerono/sql-consultas-ms-complejas>

## P. Apéndices

### 1.1. Página Web (HTML5 + PHP + JAVASCRIPT)

#### 1.1.1. Conexión a base de datos

```
<?php
$dbuser = "system";
$dbpass = "oracle";
$db = "127.0.0.1:1521/xe";

//creación conexión a oracle
$conn = oci_connect($dbuser, $dbpass, $db);

if(!$conn)
{
    $m = OCI_ERROR();
    echo $m['message']."\n";
    die("Conexion fallida");
}
```

```

        exit;
    }
?>

```

### 1.1.2. Funciones SQL

```
<?php
```

```

function selectsql($campos, $tabla, $filtro)
{
    include "conexion.php";
    $sql = "SELECT ".$campos."
           FROM ".$tabla."
           WHERE ".$filtro;
    $query = oci_parse($conn, $sql);
    oci_execute($query);

    $row = oci_fetch_all($query, $res);

    oci_free_statement($query);
    oci_close($conn);
    return ($row);
}

function selectsql_datos($campos, $tabla, $filtro)
{
    include "conexion.php";
    $sql = "SELECT ".$campos."
           FROM ".$tabla."
           WHERE ".$filtro;
    $query = oci_parse($conn, $sql);
    oci_execute($query);

    $row = oci_fetch_array($query, OCI_BOTH);

    oci_free_statement($query);
    oci_close($conn);
    return ($row);
}

function insertarsql($tabla, $parametros, $valores)
{
    include "conexion.php";
    $sql = "INSERT INTO ".$tabla."
           (".$parametros.")
           VALUES
           (".$valores.")";
    $query = oci_parse($conn, $sql);
    oci_execute($query);
}

```

```

        oci_free_statement($query);
        oci_close($conn);
    }

function borrarsql($tabla, $filtro)
{
    include "conexion.php";
    $sql = "DELETE FROM ".$tabla." WHERE ".$filtro;
    $query = oci_parse($conn, $sql);

    oci_execute($query);

    oci_free_statement($query);
    oci_close($conn);
}

function actualizarsql($tabla, $parametros, $filtro)
{
    include "conexion.php";

    $sql = "UPDATE ".$tabla." set ".$parametros." WHERE ".$filtro;
    $query = oci_parse($conn, $sql);
    oci_execute($query);

    oci_free_statement($query);
    oci_close($conn);
}
?>

```

### 1.1.3. Página Index

```

<?php
    session_start();
?>
<HTML lang="es">
    <HEAD>
        <TITLE>ADMIN EVENTUALES</TITLE>
        <meta name="description" content="website description" />
        <meta name="keywords" content="website keywords, website keywords" />
        <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
        <link rel="stylesheet" type="text/css" href="css/style.css" />
        <script type="text/javascript" src="js/modernizr-1.5.min.js"></script>
    </HEAD>

    <BODY>
        <div id="main">
            <header>
                <div id="logo">

```

```

        <div id="logo_text">
            <h1><a href="index.php"><img src =
"images/Logo.jpg" width=120 height=50/><span class="logo_colour">
Eventuales</span></a></h1>
            <h2>Sistema de Administración y Control de
Empleados Eventuales</h2>
        </div>
    </div>
    <nav>
        <ul class="sf-menu" id="nav">
            <li class="selected"><a
href="index.php">Inicio</a></li>
        </ul>
    </nav>
</header>
<div id="site_content">
    <div id="sidebar_container">
        <div class="gallery">
            <ul class="images">
                <li class="show"></li>
                <li></li>
                <li></li>
                <li></li>
                <li></li>
                <li></li>
                <li></li>
            </ul>
        </div>
    </div>
    <div id="content">
        <?php
            include ("Principal.php");
        ?>
    </div>
</div>
<footer>
    <p>Copyright 2013 Sistema de administración de eventuales All
Rights Reserved. | <a href="http://localhost">Christian Daniel Ibarra Hernández</a></p>
</footer>
</div>
<script type="text/javascript" src="js/jquery.js"></script>

```

```

<script type="text/javascript" src="js/jquery.easing-sooper.js"></script>
<script type="text/javascript" src="js/jquery.sooperfish.js"></script>
<script type="text/javascript" src="js/image_fade.js"></script>
</BODY>
</HTML>

```

#### 1.1.4. Página Principal

```

<h1>BIENVENIDO</h1>
<p>Sistema de información para la administración y
control de los empleados eventuales.</p>
<h3>Iniciar Sesión</h3>
<table align="left" border="0" bgcolor="white">
  <form action="Validar_usuario.php"
method="POST">
    <tr>
      <td colspan="2" align="center"
bgcolor="white">
        Introduce tu clave de
        acceso
        <?php
        if(isset($_GET["errorusuario"]) && $_GET["errorusuario"]=="si")
        {
          <br><span
style="color: ff0000"><b>Datos incorrectos</b></span>
        <?php
        }
        <?>
      </td>
    </tr>
    <tr>
      <td align="right">Usuario:</td>
      <td><input type="Text"
name="usuario" size="25" maxlength="20"></td>
    </tr>
    <tr>
      <td align="right">Contraseña:</td>
      <td><input type="password"
name="contrasena" size="25" maxlength="16"></td>
    </tr>
    <tr>
      <td colspan="2"
align="center"><input type="Submit" value="ENTRAR"></td>
    </tr>
  </form>
</table>

```

```

<br><br><br><br><br><br><br><br><br><br>
        <p>Usos de la página:</p>
        <ul>
            <li>Mantener actualizada y centralizada la
información de los empleados eventuales.</li>
            <li>Realizar el registro de asistencia y pagos
correspondientes a los empleados eventuales.</li>
            <li>Administración de inasistencias y permisos de
los empleados eventuales.</li>
        </ul>

```

### 1.1.5. Script Validar\_usuario

```

<?php
session_start();// inicio la sesión
include("conexion.php");
include("SQL.php");

//si es necesario cambiar la config. del php.ini desde tu script
ini_set("session.use_only_cookies","1");
ini_set("session.use_trans_sid","0");

$user = $_POST["usuario"];
$password = $_POST["contrasena"];

//realizamos una consulta a la bd
$passwordBD = selectsql_datos("password", "usuarios", "nom_usuario = ".$user."");
$row = selectsql("nom_usuario", "usuarios", "nom_usuario = ".$user."");

//vemos si el usuario y contraseña es válido
if($row && crypt($password, $passwordBD[0]) == $passwordBD[0])
{
    //usuario y contraseña válidos
    $row = selectsql_datos("cargo", "usuarios", "nom_usuario = ".$user."");

    $_SESSION['nombre'] = $user;//asigno un nombre a la sesión para poder guardar
diferentes datos
    $_SESSION['usuario'] = $row[0];

    //cambiamos la duración a la cookie de la sesión
    session_set_cookie_params(0, "/", $_SERVER["HTTP_HOST"], 0);

    $_SESSION["autenticado"] = "SI";//defino la sesión que demuestra que el usuario
está autorizado

    $_SESSION["ultimoAcceso"] = date("Y-n-j H:i:s");//defino la fecha y hora de inicio
de sesión en formato aaaa-mm-dd hh:mm:ss

```

```

        header ("Location: Usuario.php?session=". $user);
    }
    else
    {
        //si no existe le mando otra vez a la portada
        header("Location: index.php?errorusuario=si");
    }
?>

```

### 1.1.6. Pantalla Usuario

```

<?php
    session_start();
    include ("seguridad.php");
?>
<HTML lang="es">
    <HEAD>
        <TITLE>ADMIN EVENTUALES</TITLE>
        <meta name="description" content="website description" />
        <meta name="keywords" content="website keywords, website keywords" />
        <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
        <link rel="stylesheet" type="text/css" href="css/style2.css" />
        <script type="text/javascript" src="js/modernizr-1.5.min.js"></script>
        <link href="css/calendario.css" type="text/css" rel="stylesheet">
        <script src="js/calendar.js" type="text/javascript"></script>
        <script src="js/calendar-es.js" type="text/javascript"></script>
        <script src="js/calendar-setup.js" type="text/javascript"></script>
    </HEAD>

    <BODY>
        <div id="main">
            <header>
                <div id="logo">
                    <div id="logo_text">
                        <h1><a href="Usuario.php"><img src =
"images/Logo.jpg" width=120 height=50/><span class="logo_colour">
Eventuales</span></a></h1>
                        <h2>Sistema de Administración y Control de
Empleados Eventuales</h2>
                    </div>
                </div>
            </div>
            <nav>
                <ul class="sf-menu" id="nav">
                    <li class="selected"><a
href="Usuario.php">Inicio</a></li>
                    <li><a
href="Usuario.php?cadena=Opciones/Asistencia_eventual">Asistencia</a></li>
                    <li><a>Empleados</a>
                </ul>

```

```

        <li><a>Usuarios</a>
            <ul>
                <li><a
href="Usuario.php?cadena=Opciones/Registrar_usuario">Registrar</a></li>
                <li><a
href="Usuario.php?cadena=Opciones/Borrar_usuario">Borrar</a></li>
                <li><a
href="Usuario.php?cadena=Opciones/Actualizar_usuario">Modificar</a></li>
            </ul>
        </li>
        <li><a>Eventuales</a>
            <ul>
                <li><a
href="Usuario.php?cadena=Opciones/Registrar_eventual">Registrar</a></li>
                <li><a
href="Usuario.php?cadena=Opciones/Borrar_eventual">Borrar</a></li>
                <li><a
href="Usuario.php?cadena=Opciones/Actualizar_eventual">Modificar</a></li>
            </ul>
        </li>
    </ul>
    </li>
    <li><a
href="Usuario.php?cadena=Opciones/Permisos">Permisos</a></li>
    <li><a
href="Usuario.php?cadena=Opciones/Reportes">Reportes</a></li>
    <li>
        <a>
            <b>
                <?= $_SESSION['nombre'].>
            </b>
            <span style="color: #00ffff">
                <b>
                    <?=
$_SESSION['usuario']; ?>
                </b>
            </span>
        </a>
    </li>
    <li><a
href="Usuario.php?cadena=Opciones/cambiar_pass">Cambiar Contraseña</a></li>
    <li><a
href="salir.php?cadena=Opciones/Borrar_eventual">Cerrar Sesión</a></li>
</ul>
</li>
</ul>

```

```

        </nav>
    </header>
    <div id="site_content">
        <div id="content">

            <?php
                if(isset($_GET["cadena"]))

include(utf8_encode($_GET["cadena"].".php"));
                else
                {
                    ?>
                        <h1>BIENVENIDO</h1>
                        <p>Sistema de información para la
administración y control de los empleados eventuales.</p>
                    <?php
                        }
                    ?>
                </div>
            </div>
        <footer>
            <p>Copyright 2013 Sistema de administración de eventuales All
Rights Reserved. | <a href="http://localhost">Christian Daniel Ibarra Hernández</a></p>
        </footer>
    </div>
    <script type="text/javascript" src="js/jquery.js"></script>
    <script type="text/javascript" src="js/jquery.easing-sooper.js"></script>
    <script type="text/javascript" src="js/jquery.sooperfish.js"></script>
    <script type="text/javascript" src="js/image_fade.js"></script>
    <script type="text/javascript">
        $(document).ready(function()
        {
            $('ul.sf-menu').sooperfish();
        });
    </script>
</BODY>
</HTML>

```

### 1.1.7. Script Seguridad

```

<?php
if(!$_SESSION['nombre'])
{
    //cambiamos la duración a la cookie de la sesión
    session_set_cookie_params(0, "/", $_SERVER["HTTP_HOST"], 0);

    //COMPRUEBA QUE EL USUARIO ESTA AUTENTICADO
    if($_SESSION["autenticado"] != "SI")
    {

```

```

        //si no existe, envio a la página de autentificación
        header("Location: index.php");
        //ademas salgo de este script
        exit();
    }
    else
    {
        //sino, calculamos el tiempo transcurrido
        $fechaGuardada = $_SESSION["ultimoAcceso"];
        $ahora = date("Y-n-j H:i:s");
        $tiempo_transcurrido = (strtotime($ahora)-strtotime($fechaGuardada));
        //comparamos el tiempo transcurrido
        if($tiempo_transcurrido >= 600)
        {
            //si pasaron 10 minutos o más
            session_destroy(); // destruyo la sesión
            header("Location: index.php"); //envío al usuario a la pag. de
autentificación
        }
        else //sino, actualizo la fecha de la sesión
        {
            $_SESSION["ultimoAcceso"] = $ahora;
        }
    }
}
?>

```

### 1.1.8. Pantalla Asistencia\_eventual

```

        <h1>ASISTENCIA EVENTUAL</h1>
        <?php
            include ("conexion.php");
            include ("SQL.php");

            date_default_timezone_set('America/Mexico_City');

            $valida_evento = selectsql_datos("id_evento,
fecha_final", "evento", "id_evento = (SELECT max(id_evento) FROM evento)");
            $entrada = date("H:i:s");
            $fecha_termino = date("d/m/y");

            if($valida_evento[0] != date("ymd") &&
$fecha_termino != $valida_evento[1])
        {
            ?>
                <fieldset>
                <legend><font size = 3>Registrar Evento:
</font></legend>

```







```

if(isset($nom[0]))

$sql = "SELECT id_eventual, nombre_eventual, apellidos

FROM eventuales

WHERE id_eventual NOT IN (SELECT id_eventual

FROM asistencia

WHERE id_evento = ".date("ymd").") AND
(nombre_eventual LIKE '%".$nom[0]."' OR apellidos LIKE '%".$nom[0]."'";

if(isset($nom[1]))

$sql = $sql." OR nombre_eventual LIKE '%".$nom[1]."' OR apellidos LIKE
'".$nom[1]."'";

$sql.) ORDER BY nombre_eventual ASC";

}

}

$query = oci_parse($conn,
oci_execute($query);

while ($row =
{
?>
<option>
<?php echo
$row[0]." | ".$row[1]." ".$row[2]; ?>
</option>
<?php
}
?>
</select>
<br><br>
<input type="submit"
value="Enviar">

</form>
<br>
<?php
}
//realizamos una consulta a la bd

```





```

width =
700,
height =
600>
</applet>
<br>
</fieldset>
<br>
<input
type="submit" value="Registrar Asistencia">
<?php
    }
}
?>
</form>

```

### 1.1.9. Script Asistencia

```

<?php
include ("../conexion.php");
include ("../SQL.php");

$folio = $_REQUEST["folio"];
$pago_extra = 0;

if(isset($_REQUEST["pago_extra"]) && $_REQUEST["pago_extra"] == "on")
    echo $pago_extra = selectsql_datos("puesto.pago/2", "eventuales, evento,
puesto", "puesto.id_puesto = eventuales.id_puesto AND eventuales.id_eventual = ".$folio;
else
    echo $pago_extra = 0;

date_default_timezone_set('America/Mexico_City');
$evento = date("ymd");
$entrada = date("d/m/y H:i:s");
$asist = selectsql_datos("max(id_asistencia)", "asistencia", "id_asistencia = id_asistencia");
$asistencia = $asist[0]+1;

//realizamos una consulta a la bd
$row = selectsql("id_evento", "evento", "id_evento = '".$evento."'");

if($row)
{
    //realizamos una consulta a la bd
    insertarsql("asistencia", "id_asistencia, id_eventual, id_evento, id_permiso,
presente, hora_entrada, pago_extra", $asistencia.", ".$folio.", ".$evento.", 0, 1, ".$entrada.",
".$pago_extra);
?>
<script type="text/javascript">

```

```

        window.onunload = Confirmar();

        function Confirmar()
        {
            alert("Asistencia Registrada con ¡Exito!");
            return
location.href="../Usuario.php?cadena=Opciones/Asistencia_eventual";
        }
    </script>
    <?php
}
else
{
    echo "No existe un evento";
    ?>
    <script type="text/javascript">
        window.onunload = Confirmar();

        function Confirmar()
        {
            alert("NO existe un Evento");
            return
location.href="../Usuario.php?cadena=Opciones/Asistencia_eventual";
        }
    </script>
    <?php
}
if(isset($_REQUEST['src']))
{
    $src = $_REQUEST['src'];
    $src = substr($src, strpos($src, ",") + 1);
    $decoded = base64_decode($src);

    $fp = fopen($filename,'wb');
    fwrite($fp, $decoded);
    fclose($fp);
}
?>

1.1.10. Pantalla Registrar_usuario

<?php
if(isset($_GET["errorusuario"]) && $_GET["errorusuario"]=="si")
{
?>
    <script type="text/javascript" content="text/html; charset=UTF-8">
        window.onunload = Error();

        function Error()

```

```

        {
            alert("¡Error! Verifica el usuario o Contraseñas no coinciden!");
            return
location.href=" ../Usuario.php?cadena=Opciones/Registrar_usuario";
        }
    </script>
<?php
    }
    else
    {
        if(isset($_GET["errorusuario"]) && $_GET["errorusuario"]=="no")
        {
?>
            <script type="text/javascript" content="text/html; charset=UTF-8">
                window.onunload = Confirmar();

                function Confirmar()
                {
                    alert("Usuario REGISTRADO con ¡Exito!");
                    return
location.href=" ../Usuario.php?cadena=Opciones/Registrar_usuario";
                }
            </script>
<?php
        }
    }
?>
        <h1>REGISTRAR USUARIO</h1>
        <form method = "POST" action =
"Opciones/Registrar.php">
            <font size = 3>Introduce los datos que se
te piden: </font>
            <br><br>
            <table border="0" cellpadding = "5">
                <tr>
                    <td>Usuario:</td>
                    <td>
                        <input
name="nom_usuario" size="48" placeholder="Escribe un nombre de usuario" required>
                    </td>
                </tr>
                <tr>
                    <td>Nombre(s):</td>
                    <td>
                        <input
name="nombre" size ="48" placeholder="Escribe nombre del usuario" autofocus required>
                    </td>
                </tr>
            </table>

```

```

        </tr>
        <tr>
            <td>Apellidos:</td>
            <td>
                <input
name="apellidos" size="48" placeholder="Escribe apellidos del usuario" autofocus>
            </td>
        </tr>
        <tr>
            <td>Cargo:</td>
            <td>
                <select name =
"cargo" onChange="combo(this, 'theinput')">
                    <?php
                        if($_SESSION['usuario'] == "ADMINISTRADOR")
                            {
                                ?>
                                <option selected>ADMINISTRADOR</option>
                                <option>SUPERVISOR</option>
                                <option>AYUDANTE</option>
                                <?php
                                    }
                                else
                                    {
                                        if($_SESSION['usuario'] == "SUPERVISOR")
                                            {
                                                ?>
                                                <option selected>SUPERVISOR</option>
                                                <option>AYUDANTE</option>
                                                <?php
                                                    }
                                            else

```

```

    {
        if($_SESSION['usuario'] == "AYUDANTE")
        {
            ?>
            <option selected>AYUDANTE</option>
            <?php
        }
    }
}
?>
<option>GENERAL</option>
</select>
</td>
</tr>
<tr>
<td>Contrase&ntilde;a:</td>
<td>
<input
name="password" type = "password" size ="17" maxlength="16" autofocus required>
</td>
</tr>
<tr>
<td>Repite
Contrase&ntilde;a: </td>
<td>
<input
name="password2" type = "password" size ="17" maxlength="16" autofocus required>
</td>
</tr>
</table>
<br><br>
<input type="submit" value="Enviar">
</form>

```

### 1.1.11. Pantalla Registrar\_eventual

```

<?php
if(isset($_GET["errorusuario"]) && $_GET["errorusuario"]=="si")
{

```

```

?>
        <script type="text/javascript">
            window.onunload = Error();

            function Error()
            {
                alert("Error ya existe ese usuario");
                return
location.href=" ../Usuario.php?cadena=Opciones/Registrar_eventual";
            }
        </script>
<?php
    }
    else
    {
        if(isset($_GET["errorusuario"]) && $_GET["errorusuario"]=="no")
        {
?>
            <script type="text/javascript">
                window.onunload = Confirmar();

                function Confirmar()
                {
                    alert("Usuario REGISTRADO con ¡Exito!");
                    return
location.href=" ../Usuario.php?cadena=Opciones/Registrar_eventual";
                }
            </script>
<?php
        }
    }
?>
<?php
function Puesto()
{
    include "conexion.php";
    //realizamos una consulta a la bd
    $sql = "SELECT nombre FROM puesto";
    $query = oci_parse($conn, $sql);
    oci_execute($query);

    $i=0;
    while($row[$i] = oci_fetch_array($query, OCI_BOTH))
    {
        $i++;
    }
}

```



```

        </td>
    </tr>
    <tr>
        <td>Apellidos:</td>
        <td>
            <input
name="apellidos" size="50" placeholder="Escribe apellidos del eventual" autofocus>
        </td>
    </tr>
    <tr>
        <td>Edad:</td>
        <td>
            <input
name="edad" size="50" placeholder="Escribe edad del eventual" autofocus>
        </td>
    </tr>
    <tr>
        <td>Domicilio:</td>
        <td>
            <input
name="domicilio" size="50" placeholder="Escribe domicilio del eventual" autofocus>
        </td>
    </tr>
    <tr>
        <td><p
class="uso">Tel&eacute;fono:</td>
        <td>
            <input
name="telefono" type="tel" pattern="\S{8,10}" size="50" placeholder="Tel&eacute;fono"
maxlength="10" autofocus></p>
        </td>
    </tr>
</table>
<applet
    code = "HuellaDigital.Huella",
    archive =
"AppletHuellaDigitalPersona.jar, lib/dpfpenrollment.jar, lib/dpfverification.jar, lib/dpotapi.jar,
lib/dpotjni.jar, lib/ojdbc6.jar",
    width = 700,
    height = 600>
</applet>
<br>
<input type="submit" value="Enviar">
</form>

```

### 1.1.12. Script Registrar

```

<?php
include("../Conexion.php");

```

```

include("../SQL.php");
include("Blow.php");

if(isset($_GET["pag"]) && $_GET["pag"] == 2)
{
    $folio = $_REQUEST["folio"];
    $puesto = $_REQUEST["puesto"];
    $nombre = $_REQUEST["nombre"];
    $apellidos = $_REQUEST["apellidos"];
    $edad = $_REQUEST["edad"];
    $domicilio = $_REQUEST["domicilio"];
    $telefono = $_REQUEST["telefono"];

    $row = selectsql("id_eventual", "eventuales", "id_eventual = '". $folio. "'");

    if(!$row)
    {
        $id_puesto = selectsql_datos("id_puesto", "puesto", "nombre =
        '". $puesto. "'");
        insertarsql("eventuales", "id_eventual, id_puesto, nombre_eventual,
        apellidos, edad, domicilio, telefono", "'". $folio. "', ". $id_puesto[0]. ", '". $nombre. "', '". $apellidos. "',
        '". $edad. "', '". $domicilio. "', '". $telefono. "'");

        header("Location: Registrar_eventual.php?errorusuario=no");
    }
    else
    //si existe le mando otra vez a la portada
    header("Location: Registrar_eventual.php?errorusuario=si");
}
else
{
    $user = $_REQUEST["nom_usuario"];
    $pwd = $_REQUEST["password"];
    $pwd2 = $_REQUEST["password2"];
    $nombre = $_REQUEST["nombre"];
    $apellidos = $_REQUEST["apellidos"];
    $cargo = $_REQUEST["cargo"];

    if ($pwd == $pwd2 && $user != "")
    {
        $password = crypt_blowfish($pwd);

        //realizamos una consulta a la bd
        $row = selectsql("nom_usuario", "usuarios", "NOM_USUARIO =
        '". $user. "'");

        if(!$row)
        {

```

```

        insertarsql("usuarios", "nom_usuario, password, nombre,
apellidos, cargo", "".$user.", "".$password.", "".$nombre.", "".$apellidos.", "".$cargo."");
        header("Location: Registrar_usuario.php?errorusuario=no");
    }
    else
    {
        header("Location: Registrar_usuario.php?errorusuario=si");
    }
}
else
{
    //si usuario esta vacío o no conciden las contraseñas
    header("Location: Registrar_usuario.php?errorusuario=si");
}
}
?>

```

### 1.1.13. Pantalla Borrar\_usuario

```

<?php
if(isset($_GET["errorusuario"]) && $_GET["errorusuario"]=="si")
{
?>
    <script type="text/javascript" content="text/html; charset=UTF-8">
        window.onunload = Error();

        function Error()
        {
            alert("Error usuario NO se encuentra en la Base de Datos");
            return
location.href=" ../Usuario.php?cadena=Opciones/Borrar_usuario";
        }
    </script>
<?php
}
else
{
    if(isset($_GET["errorusuario"]) && $_GET["errorusuario"]=="no")
    {
?>
        <script type="text/javascript" content="text/html; charset=UTF-8">
            window.onunload = Confirmar();

            function Confirmar()
            {
                alert("Usuario ELIMINADO con ¡Exito!");
                return
location.href=" ../Usuario.php?cadena=Opciones/Borrar_usuario";
            }

```

```

        </script>
<?php
    }
}
?>

<h1>ELIMINAR USUARIO</h1>
<font size = 3>Introduce los datos que se te piden: </font>
<br><br>
<table border="0" width="48" cellpadding
= "0">
    <tr>
        <td>Usuario:</td>
        <td>
            <form
method="POST" action="Usuario.php?cadena=Opciones/Borrar_usuario">
                <input
name="buscar_nom" placeholder="Usuario a Buscar">
                <td><input
type="submit" value="Buscar"></td>
            </form>
        </td>
    </tr>
</table>
<br>
<form method = "POST" action =
"Opciones/Borrar.php">
    <select name="nom_usuario" size="10"
onChange="list(this, 'theinput')">
        <?php
            include ("conexion.php");

            if(!isset($_REQUEST["buscar_nom"]) || $_REQUEST["buscar_nom"] == "")
            {
                //realizamos una consulta
                a la bd
                if($_SESSION['usuario'] ==
"ADMINISTRADOR")
                {
                    $sql = "SELECT *
FROM usuarios ORDER BY nombre ASC";
                }
                else
                {
                    if($_SESSION['usuario'] == "SUPERVISOR")
                    {

```

```

                                                                    $sql =
"SELECT * FROM usuarios WHERE cargo != 'ADMINISTRADOR' ORDER BY nombre ASC";
                                                                    }
                                                                    else
                                                                    {

                                                                    if($_SESSION['usuario'] == "AYUDANTE")
                                                                    {

                                                                    $sql = "SELECT * FROM usuarios WHERE cargo != 'ADMINISTRADOR' AND cargo !=
'SUPERVISOR' ORDER BY nombre ASC";
                                                                    }
                                                                    else
                                                                    {

                                                                    $sql = "SELECT * FROM usuarios WHERE cargo != 'ADMINISTRADOR' AND cargo !=
'SUPERVISOR' AND cargo != 'AYUDANTE' ORDER BY nombre ASC";
                                                                    }
                                                                    }
                                                                    }
                                                                    }
                                                                    else
                                                                    {
                                                                    //realizamos una consulta
                                                                    a la bd
                                                                    if($_SESSION['usuario'] ==
                                                                    "ADMINISTRADOR")
                                                                    {
                                                                    $sql = "SELECT *
                                                                    FROM usuarios WHERE (nom_usuario LIKE '%" . $_REQUEST["buscar_nom"] . "%' OR nombre LIKE
                                                                    '%" . $_REQUEST["buscar_nom"] . "%' OR apellidos LIKE '%" . $_REQUEST["buscar_nom"] . "%') ORDER
                                                                    BY nombre ASC";
                                                                    }
                                                                    else
                                                                    {

                                                                    if($_SESSION['usuario'] == "SUPERVISOR")
                                                                    {
                                                                    $sql =
                                                                    "SELECT * FROM usuarios WHERE cargo != 'ADMINISTRADOR' AND (nom_usuario LIKE
                                                                    '%" . $_REQUEST["buscar_nom"] . "%' OR nombre LIKE '%" . $_REQUEST["buscar_nom"] . "%' OR
                                                                    apellidos LIKE '%" . $_REQUEST["buscar_nom"] . "%') ORDER BY nombre ASC";
                                                                    }
                                                                    else
                                                                    {

                                                                    if($_SESSION['usuario'] == "AYUDANTE")
                                                                    {

```

```

    $sql = "SELECT * FROM usuarios WHERE cargo != 'ADMINISTRADOR' AND cargo !=
'SUPERVISOR' AND (nom_usuario LIKE '%" .$_REQUEST["buscar_nom"]."% OR nombre LIKE
 '%" .$_REQUEST["buscar_nom"]."% OR apellidos LIKE '%" .$_REQUEST["buscar_nom"]."% ) ORDER
BY nombre ASC";

```

```

    }
    else
    {

```

```

    $sql = "SELECT * FROM usuarios WHERE cargo != 'ADMINISTRADOR' AND cargo !=
'SUPERVISOR' AND cargo != 'AYUDANTE' AND (nom_usuario LIKE
 '%" .$_REQUEST["buscar_nom"]."% OR nombre LIKE '%" .$_REQUEST["buscar_nom"]."% OR
apellidos LIKE '%" .$_REQUEST["buscar_nom"]."% ) ORDER BY nombre ASC";

```

```

    }

```

```

    }

```

```

    }

```

```

    $query = oci_parse($conn, $sql);
    oci_execute($query);

```

```

oci_fetch_array($query, OCI_BOTH)

```

```

while ($row =

```

```

{

```

```

?>

```

```

    <option>

```

```

        <?php echo

```

```

$row[0]." | ".$row[2]." ".$row[3]; ?>

```

```

    </option>

```

```

<?php

```

```

}

```

```

    oci_free_statement($query);
    oci_close($conn);

```

```

?>

```

```

</select>

```

```

<br><br>

```

```

<input type="submit" value="Enviar">

```

```

</form>

```

#### 1.1.14. Pantalla Borrar\_eventual

```

<?php

```

```

if(isset($_GET["errorusuario"]) && $_GET["errorusuario"]=="si")

```

```

{

```

```

?>

```

```

    <script type="text/javascript" content="text/html; charset=UTF-8">
        window.onunload = Error();

```

```

        function Error()

```

```

        {
            alert("Error eventual NO se encuentra en la Base de Datos");
            return
location.href=" ../Usuario.php?cadena=OpcionesBorrar_eventual";
        }
    </script>
<?php
    }
    else
    {
        if(isset($_GET["errorusuario"]) && $_GET["errorusuario"]=="no")
        {
?>
            <script type="text/javascript" content="text/html; charset=UTF-8">
                window.onunload = Confirmar();

                function Confirmar()
                {
                    alert("Eventual ELIMINADO con ¡Exito!");
                    return
location.href=" ../Usuario.php?cadena=Opciones/Borrar_eventual";
                }
            </script>
<?php
        }
    }
?>
        <h1>ELIMINAR EVENTUAL</h1>
        <font size = 3>Introduce los datos que se te piden: </font>
        <br><br>
        <table border="0" width="48" cellpadding
= "0">
            <tr>
                <td>Folio:</td>
                <td>
                    <form
method="POST" action="Usuario.php?cadena=Opciones/Borrar_eventual">
                        <input
name="buscar_nom" placeholder="Eventual a Buscar">
                        <td><input
type="submit" value="Buscar"></td>
                    </form>
                </td>
            </tr>
        </table>
        <br>
        <form method = "POST" action =
"Opciones/Borrar.php?pag=2">

```

```

onChange="list(this, 'theinput')">
                                <select name="nom_usuario" size="10"
                                <?php
                                    include ("conexion.php");

                                if(!isset($_REQUEST["buscar_nom"]) || $_REQUEST["buscar_nom"] == "")
                                    {
                                        //realizamos una consulta
                                a la bd
                                        $sql = "SELECT * FROM
                                eventuales ORDER BY nombre_eventual ASC";
                                    }
                                    else
                                    {
                                        //realizamos una consulta
                                a la bd
                                        $sql = "SELECT * FROM
                                eventuales WHERE (nombre_eventual LIKE '%" . $_REQUEST["buscar_nom"] . "%' OR apellidos LIKE
                                '%" . $_REQUEST["buscar_nom"] . "%') ORDER BY nombre_eventual ASC";
                                    }

                                $query = oci_parse($conn, $sql);
                                oci_execute($query);

                                while ($row =
                                oci_fetch_array($query, OCI_BOTH))
                                {
                                ?>
                                    <option>
                                <?php echo
                                $row[0]." | ".$row[2]." ".$row[3]; ?>
                                    </option>
                                <?php
                                    }
                                ?>
                                </select>
                                <br><br>
                                <input type="submit" value="Enviar">
                                </form>

```

### 1.1.15. Script Borrar

```

<?php
include ("../conexion.php");
include ("../SQL.php");

$usuario = preg_split("/[\s|]+/", $_REQUEST["nom_usuario"]);

```

```

if(count($usuario) > 1)
    $user = $usuario[0];

//realizamos una consulta a la bd
if(isset($_GET["pag"]) && $_GET["pag"] == 2)
    $row = selectsql("id_eventual", "eventuales", "id_eventual = '".$_user.'"");
else
    $row = selectsql("nom_usuario", "usuarios", "nom_usuario = '".$_user.'"");

if($row)
{
    if(isset($_GET["pag"]) && $_GET["pag"] == 2)
    {
        borrarsql("eventuales", "id_eventual = '".$_user.'"");
        header("Location: Borrar_eventual.php?errorusuario=no");
    }
    else
    {
        borrarsql("usuarios", "nom_usuario = '".$_user.'"");
        header("Location: Borrar_usuario.php?errorusuario=no");
    }
}
else
{
    header("Location: Borrar_usuario.php?errorusuario=si");
}
?>

```

### 1.1.16. Pantalla Actualizar\_usuario

```

<h1>ACTUALIZAR USUARIO</h1>
<font size = 3>Introduce los datos que se te piden:
</font>
<br><br>
<table border="0" width="48" cellpadding
= "5">
    <tr>
        <td>Usuario:</td>
        <td>
            <form
method="POST" action="Usuario.php?cadena=Opciones/Actualizar_usuario">
                <input
name="buscar_nom" placeholder="Usuario a Buscar">
                <td><input
type="submit" value="Buscar"></td>
            </form>
        </td>
    </tr>
</table>

```

```

        <br>
        <form method = "POST" action =
"Usuario.php?cadena=Opciones/Actualizar_usuario">
        <select name="usuario" size="10"
onChange="list(this, 'theinput')">
        <?php
            include ("conexion.php");

            if(!isset($_REQUEST["buscar_nom"]) || $_REQUEST["buscar_nom"] == "")
                {
                    //realizamos una consulta
                    a la bd
                    if($_SESSION['usuario'] ==
"ADMINISTRADOR")
                        {
                            $sql = "SELECT *
FROM usuarios ORDER BY nombre ASC";
                        }
                    else
                        {

                            if($_SESSION['usuario'] == "SUPERVISOR")
                                {
                                    $sql =
"SELECT * FROM usuarios WHERE cargo != 'ADMINISTRADOR' ORDER BY nombre ASC";
                                }
                            else
                                {

                                    if($_SESSION['usuario'] == "AYUDANTE")
                                        {

                                            $sql = "SELECT * FROM usuarios WHERE cargo != 'ADMINISTRADOR' AND cargo !=
'SUPERVISOR' ORDER BY nombre ASC";
                                        }
                                    else
                                        {

                                            $sql = "SELECT * FROM usuarios WHERE cargo != 'ADMINISTRADOR' AND cargo !=
'SUPERVISOR' AND cargo != 'AYUDANTE' ORDER BY nombre ASC";
                                        }
                                    }
                                }
                            }
                    }
                }
            }
        }
    }
}
else
{

```

```

a la bd //realizamos una consulta

"ADMINISTRADOR") if($_SESSION['usuario'] ==

{
    $sql = "SELECT *
FROM usuarios WHERE (nom_usuario LIKE '%"$_REQUEST["buscar_nom"]."%' OR nombre LIKE
'"$_REQUEST["buscar_nom"]."%' OR apellidos LIKE '%"$_REQUEST["buscar_nom"]."%' ) ORDER
BY apellidos ASC";
}
else
{

if($_SESSION['usuario'] == "SUPERVISOR")
{
    $sql =
"SELECT * FROM usuarios WHERE cargo != 'ADMINISTRADOR' AND (nom_usuario LIKE
'"$_REQUEST["buscar_nom"]."%' OR nombre LIKE '%"$_REQUEST["buscar_nom"]."%' OR
apellidos LIKE '%"$_REQUEST["buscar_nom"]."%' ) ORDER BY apellidos ASC";
}
else
{

if($_SESSION['usuario'] == "AYUDANTE")
{

    $sql = "SELECT * FROM usuarios WHERE cargo != 'ADMINISTRADOR' AND cargo !=
'SUPERVISOR' AND (nom_usuario LIKE '%"$_REQUEST["buscar_nom"]."%' OR nombre LIKE
'"$_REQUEST["buscar_nom"]."%' OR apellidos LIKE '%"$_REQUEST["buscar_nom"]."%' ) ORDER
BY apellidos ASC";
}
else
{

    $sql = "SELECT * FROM usuarios WHERE cargo != 'ADMINISTRADOR' AND cargo !=
'SUPERVISOR' AND cargo != 'AYUDANTE' AND (nom_usuario LIKE
'"$_REQUEST["buscar_nom"]."%' OR nombre LIKE '%"$_REQUEST["buscar_nom"]."%' OR
apellidos LIKE '%"$_REQUEST["buscar_nom"]."%' ) ORDER BY apellidos ASC";
}
}
}
}

$query = oci_parse($conn, $sql);
oci_execute($query);

while ($row =
oci_fetch_array($query, OCI_BOTH))

```

```

        {
        ?>
                <option>
                    <?php echo
$row[0]." | ".$row[2]." ".$row[3]; ?>
                </option>
        <?php
            }
        ?>
        </select>
        <br>
        <br>
        <input type="submit" value="Enviar">
        <br>
</form>
<?php
    //realizamos una consulta a la bd
    if(isset($_REQUEST["usuario"]))
    {
        $usuario = preg_split("/[\s|]+/",
$_REQUEST["usuario"]);

        if(count($usuario) > 1)
        {
            $user = $usuario[0];
        }
        include ("SQL.php");
        $row = selectsql("nom_usuario",
"usuarios", "nom_usuario = '".$user.'");

        if ($row)
        {
            //usuario y contraseña

            $row = selectsql_datos("*",
"usuarios", "nom_usuario = '".$user.'");

        ?>

            <br><br>
            <form method = "POST"

                <table border="0"

                    <tr>

                        <td>Usuario:</td>

```

```

                <td>
<input name="nom_usuario" size="48" value="<?php echo $row[0]; ?>" readonly>
                </td>
            </tr>
            <tr>
                <td>Nombre(s):</td>
                <td>
<input name="nombre" size ="48" value="<?php echo $row[2]; ?>" autofocus>
                </td>
            </tr>
            <tr>
                <td>Apellidos:</td>
                <td>
<input name="apellidos" size ="48" value="<?php echo $row[3]; ?>">
                </td>
            </tr>
            <tr>
                <td>Cargo:</td>
                <td>
<select name = "cargo" onChange="combo(this, 'theinput')">
<?php
if($_SESSION['usuario'] == "ADMINISTRADOR")
{
    if($row[4] == "ADMINISTRADOR")
    {
                <option selected>ADMINISTRADOR</option>
                <option>SUPERVISOR</option>
                <option>AYUDANTE</option>
                <?>
    }
}
    
```

```
        <option>GENERAL</option>

        <?php
    }
else
    if($row[4] == "SUPERVISOR")
    {
        ?>

        <option>ADMINISTRADOR</option>
        <option selected>SUPERVISOR</option>
        <option>AYUDANTE</option>
        <option>GENERAL</option>

        <?php
    }
else
    if($row[4] == "AYUDANTE")
    {
        ?>

        <option>ADMINISTRADOR</option>
        <option>SUPERVISOR</option>
        <option selected>AYUDANTE</option>
        <option>GENERAL</option>

        <?php
    }
else
```

```
        if($row[4] == "GENERAL")
        {
            ?>
                <option>ADMINISTRADOR</option>
                <option>SUPERVISOR</option>
                <option>AYUDANTE</option>
                <option selected>GENERAL</option>
            <?php
        }
    }
    else
        if($_SESSION['usuario'] == "ADMINISTRADOR" || $_SESSION['usuario'] ==
        "SUPERVISOR")
        {
            if($row[4] == "SUPERVISOR")
            {
                ?>
                    <option selected>SUPERVISOR</option>
                    <option>AYUDANTE</option>
                    <option>GENERAL</option>
                <?php
            }
            else
                if($row[4] == "AYUDANTE")
```

```

        {
            ?>
            <option>SUPERVISOR</option>
            <option selected>AYUDANTE</option>
            <option>GENERAL</option>
            <?php
        }
    else
        if($row[4] == "GENERAL")
        {
            ?>
                <option>SUPERVISOR</option>
                <option>AYUDANTE</option>
                <option selected>GENERAL</option>
            <?php
        }
    }
    else
        if($_SESSION['usuario'] == "ADMINISTRADOR" || $_SESSION['usuario'] ==
        "SUPERVISOR" || $_SESSION['usuario'] == "AYUDANTE")
        {
            if($row[4] == "AYUDANTE")
            {
                ?>
            }
        }
    }
}

```

```

        <option selected>AYUDANTE</option>
        <option>GENERAL</option>
        <?php
    }
    else
        if($row[4] == "GENERAL")
        {
            ?>
                <option>AYUDANTE</option>
                <option selected>GENERAL</option>
            <?php
        }
    }
    else
        if($_SESSION['usuario'] == "ADMINISTRADOR" ||
$_SESSION['usuario'] == "SUPERVISOR" || $_SESSION['usuario'] == "AYUDANTE" ||
$_SESSION['usuario'] == "GENERAL")
        {
            ?>
                <option selected>GENERAL</option>
            <?php
        }
        ?>
    </select>
        </td>
    </tr>
</table>
<br><br>

```

```

value="Actualizar">
                                <input type="submit"
                                <?php
                                }
                                }
                                oci_free_statement($query);
                                oci_close($conn);
                                ?>

```

### 1.1.17. Pantalla Actualizar\_eventual

```

                                <h1>ACTUALIZAR EVENTUAL</h1>
                                <font size = 3>Introduce los datos que se te piden:
</font>
                                <br><br>
                                <table border="0" width="48" cellpadding
= "5">
                                <tr>
                                <td>Folio:</td>
                                <td>
                                <form
method="POST" action="Usuario.php?cadena=Opciones/Actualizar_eventual">
                                <input
name="buscar_nom" placeholder="Usuario a Buscar">
                                <td><input
type="submit" value="Buscar"></td>
                                </form>
                                </td>
                                </tr>
                                </table>
                                <br>
                                <form method = "POST" action =
"Usuario.php?cadena=Opciones/Actualizar_eventual">
                                <select name="usuario" size="10"
onChange="list(this, 'theinput')">
                                <?php
                                include ("conexion.php");

                                if(!isset($_REQUEST["buscar_nom"]) || $_REQUEST["buscar_nom"] == "")
                                {
                                //realizamos una consulta
                                a la bd
                                $sql = "SELECT * FROM
                                eventuales ORDER BY nombre_eventual ASC";
                                }
                                else
                                {

```

```

//realizamos una consulta
a la bd
eventuales WHERE (nombre_eventual LIKE '%".$_REQUEST["buscar_nom"]."%' OR apellidos LIKE
'%".$_REQUEST["buscar_nom"]."%' ORDER BY nombre_eventual ASC";

}

$query = oci_parse($conn, $sql);
oci_execute($query);

while ($row =
oci_fetch_array($query, OCI_BOTH))
{
?>
<option>
<?php echo
$row[0]." | ".$row[2]." ".$row[3]; ?>
</option>
<?php
}
?>
</select>
<br>
<br>
<input type="submit" value="Enviar">
<br>
</form>
<?php
//realizamos una consulta a la bd
if(isset($_REQUEST["usuario"]))
{
$usuario = preg_split("/[\s]+/",
$_REQUEST["usuario"]);

if(count($usuario) > 1)
{
$id_eventual = $usuario[0];
}

include ("SQL.php");
$row = selectsql("id_eventual",
"eventuales", "id_eventual = ".$id_eventual);

if ($row)
{

```

```

//usuario y contraseña
válidos
"eventuales", "id_eventual = ".$id_eventual);
?>
<?php
function Puesto()
{
    include "conexion.php";
    //realizamos una consulta a la bd
    $sql = "SELECT nombre FROM puesto";
    $query = oci_parse($conn, $sql);
    oci_execute($query);

    $i=0;
    while($row[$i] = oci_fetch_array($query, OCI_BOTH))
    {
        $i++;
    }

    return $row;
}
?>

action = "Opciones/Actualizar.php?pag=2">
width="48" cellpadding = "5">

    <td>Folio:</td>

    <input name="folio" size="10" value="<?php echo $row[0]; ?>" readonly>

selectsql_datos("nombre", "puesto", "id_puesto = ".$row[1]);

    <td>Puesto:</td>

    <select name = "puesto" onChange="combo(this, 'theinput')">
        <option selected><?php echo $pues[0]; ?></option>;

```

```

//usuario y contraseña
$row = selectsql_datos("*",
<br>
<form method = "POST"
    <table border="0"
        <tr>
            <td>
            </td>
        </tr>
        <tr>
            <?php
                $pues =
            ?>
            <td>

```

```

<?php
$i=0;

$puestos = Puesto();

while($i < count($puestos)-1)
{
    if($puestos[$i][0] != $pues[0])
        echo "<option>".$puestos[$i][0]."</option>";
    $i++;
}
?>

</select>
</td>
</tr>
<tr>
<td>Nombre(s):</td>
<td>
<input name="nombre" size ="50" value="<?php echo $row[2]; ?>" autofocus>
</td>
</tr>
<tr>
<td>Apellidos:</td>
<td>
<input name="apellidos" size ="50" value="<?php echo $row[3]; ?>">
</td>
</tr>
<tr>
<td>Edad:</td>
<td>
<input name="edad" size ="3" value="<?php echo $row[4]; ?>">
</td>
</tr>
</tr>

```

```

        <tr>
            <td>Domicilio:</td>
            <td>
                <input name="domicilio" size="50" value="<?php echo $row[5]; ?>">
            </td>
        </tr>
        <tr>
            <td>Tel&eacute;fono:</td>
            <td>
                <input name="telefono" size="50" value="<?php echo $row[6]; ?>">
            </td>
        </tr>
    </table>
    <br>
    <input type="submit"
value="Actualizar">
        <?php
        }
    }
?>

```

### 1.1.18. Script Actualizar

```

<?php
include("../conexion.php");
include("../SQL.php");

if(isset($_GET["pag"]) && $_GET["pag"] == 2)
{
    $folio = $_REQUEST["folio"];
    $pues = $_REQUEST["puesto"];
    $nombre = $_REQUEST["nombre"];
    $apellidos = $_REQUEST["apellidos"];
    $edad = $_REQUEST["edad"];
    $domicilio = $_REQUEST["domicilio"];
    $telefono = $_REQUEST["telefono"];

    $puesto = selectsql_datos("id_puesto", "puesto", "nombre = " . $pues . "");

    actualizarsql("eventuales", "id_puesto = " . $puesto[0] . ", nombre_eventual =
    " . $nombre . ", apellidos = " . $apellidos . ", edad = " . $edad . ", domicilio = " . $domicilio . ", telefono =
    " . $telefono . ", id_eventual = " . $folio);
    ?>
    <script type="text/javascript content="text/html">
        window.onunload = Confirmar();
    </script>

```

```

        function Confirmar()
        {
            alert("Eventual ACTUALIZADO con ¡Exito!");
            return
location.href=" ../Usuario.php?cadena=Opciones/Actualizar_eventual";
        }
    </script>
    <?php
}
else
{
    $user = $_REQUEST["nom_usuario"];
    $nombre = $_REQUEST["nombre"];
    $apellidos = $_REQUEST["apellidos"];
    $cargo = $_REQUEST["cargo"];

    //realizamos una consulta a la bd
    actualizarsql("usuarios", "nombre = ".$nombre.", apellidos = ".$apellidos.",
cargo = ".$cargo.", "nom_usuario = ".$user."");
    ?>
    <script type="text/javascript content="text/html; charset=UTF-8">
        window.onunload = Confirmar();

        function Confirmar()
        {
            alert("Usuario ACTUALIZADO con ¡Exito!");
            return
location.href=" ../Usuario.php?cadena=Opciones/Actualizar_usuario";
        }
    </script>
    <?php
}
?>

```

### 1.1.19. Pantalla Permisos

```

<?php
if(isset($_GET["errorusuario"]) && $_GET["errorusuario"]=="si")
{
    ?>
        <script type="text/javascript">
            window.onunload = Error();

            function Error()
            {
                alert("Error eventual NO se encuentra en la Base de Datos");
                return
location.href=" ../Usuario.php?cadena=Opciones/Asistencia_eventual";
            }
        </script>
    ?>
}

```

```

        }
    </script>
<?php
    }
    else
    {
        if(isset($_GET["errorusuario"]) && $_GET["errorusuario"]=="no")
        {
?>
            <script type="text/javascript">
                window.onunload = Confirmar();

                function Confirmar()
                {
                    alert("Eventual ELIMINADO con ¡Exito!");
                    return
location.href="../Usuario.php?cadena=Opciones/Asistencia_eventual";
                }
            </script>
<?php
        }
    }
?>

    <h1>PERMISOS</h1>
    <font size = 3>Introduce los datos que se te piden: </font>
    <br><br>
    <?php
        include ("conexion.php");
        include ("SQL.php");

        date_default_timezone_set('America/Mexico_City');

        $row = selectsql("id_evento", "evento", "id_evento
= ".date("ymd"));

        if(!$row)
        {
?>
            <span style="color: #00ffff">
                <b>
                    NO EXISTE EVENTO
                    No puede tomar registrar
                </b>
            </span>

```

```

        <?php
            }
            else
            {
        ?>

        <table border="0" width="48" cellpadding
= "0">
            <tr>
                <td>Eventual:</td>
                <td>
                    <form
method="POST" action="Usuario.php?cadena=Opciones/Permisos">
                        <input
name="buscar_nom" placeholder="Eventual a Buscar">
                        <td><input
type="submit" value="Buscar"></td>
                    </form>
                </td>
            </tr>
        </table>
        <br>
        <form method = "POST" action =
"Usuario.php?cadena=Opciones/Permisos">
            <select name="usuario" size="10"
onChange="list(this, 'theinput')">
                <?php
                    if(!isset($_REQUEST["buscar_nom"]) || $_REQUEST["buscar_nom"] == "")
                    {
                        //realizamos una
consulta a la bd
                        $sql = "SELECT
eventuales.id_eventual, eventuales.nombre_eventual, eventuales.apellidos
FROM eventuales, evento
WHERE evento.id_evento = ".date("ymd");
                    }
                    else
                    {
                        if ctype_digit($_REQUEST["buscar_nom"]))
                            $sql =
"SELECT id_eventual, nombre_eventual, apellidos
FROM eventuales, evento

```

```

WHERE evento.id_evento = ".date("ymd")." AND id_eventual LIKE
'%" .$_REQUEST["buscar_nom"]."%";

else
{
    $nom =
explode(" ", $_REQUEST["buscar_nom"]);

    if(isset($nom[0]))

        $sql = "SELECT id_eventual, nombre_eventual, apellidos

                FROM eventuales, evento

                WHERE evento.id_evento = ".date("ymd")." AND (nombre_eventual LIKE
'%" . $nom[0]."% ' OR apellidos LIKE '%" . $nom[0]."%";

        if(isset($nom[1]))

            $sql = $sql." OR nombre_eventual LIKE '%" . $nom[1]."% ' OR apellidos LIKE
'%" . $nom[1]."%";

            $sql =
$sql.") ORDER BY nombre_eventual ASC";

        }
    }

    $query = oci_parse($conn,
oci_execute($query);

    while ($row =
    {
        ?>
        <option>
            <?php echo
$row[0]." | ". $row[1]." ". $row[2]; ?>

        </option>
        <?php
    }
    ?>
    </select>
    <br><br>
    <input type="submit"
value="Enviar">

    </form>
<?php

```

```

    }
    //realizamos una consulta a la bd
    if(isset($_REQUEST["usuario"]))
    {
        $usuario = preg_split("/[\s|]+/",
$_REQUEST["usuario"]);

        if(count($usuario) > 1)
        {
            $id_eventual = $usuario[0];
        }

        $sql = "SELECT id_eventual FROM
eventuales WHERE id_eventual = ".$id_eventual."";

        $query = oci_parse($conn, $sql);
        oci_execute($query);

        $row = oci_fetch_all($query, $res);

        //var_dump($res);
        if ($row)
        {
            //usuario y contraseña

            $sql = "SELECT id_eventual,
puesto.nombre, nombre_eventual, apellidos, edad, domicilio, telefono
FROM
eventuales, puesto
WHERE
eventuales.id_puesto = puesto.id_puesto AND id_eventual = ".$id_eventual.";
        $query = oci_parse($conn,
$sql);
        oci_execute($query);

        $row =
oci_fetch_array($query, OCI_BOTH);
    }?>

<br>
<fieldset>
<legend><font size =
3>Selección del Permiso: </font></legend>

<br>
<form method = "POST"

<table border="0"

<tr>

```



```

date_default_timezone_set('America/Mexico_City');
$evento = date("ymd");
$asist = selectsql_datos("max(id_asistencia)", "asistencia", "id_asistencia = id_asistencia");
$asistencia = $asist[0]+1;

//realizamos una consulta a la bd
$row = selectsql("id_evento", "evento", "id_evento = ".$evento."");

if($row)
{
    //realizamos una consulta a la bd
    $per = selectsql_datos("id_permiso", "permisos", "tipo_permiso = ".$permiso."");
    $eventual = selectsql("id_eventual", "asistencia", "id_eventual = ".$folio." AND
id_evento = ".$evento."");

    if($eventual)
        actualizarsql("asistencia", "id_permiso = ".$per[0], "id_eventual = ".$folio."
AND id_evento = ".$evento);
    else
        insertarsql("asistencia", "id_asistencia, id_eventual, id_evento,
id_permiso, presente", ".$asistencia.", ".$folio.", ".$evento.", ".$per[0].", 0");
    ?>
    <script type="text/javascript" content="charset=UTF-8">
        window.onunload = Confirmar();

        function Confirmar()
        {
            alert("Permiso ACTUALIZADO con ¡Exito!");
            return
location.href=" ../Usuario.php?cadena=Opciones/Permisos";
        }
    </script>
    <?php
}
else
{
    echo "No existe un evento";
    ?>
    <script type="text/javascript">
        window.onunload = Confirmar();

        function Confirmar()
        {
            alert("NO existe un Evento");
            return
location.href=" ../Usuario.php?cadena=Opciones/Asistencia_eventual";
        }
}

```

```

        </script>
        <?php
    }
?>

1.1.21.      Script Registrar_evento
<?php
include ("../conexion.php");
include ("../SQL.php");

$evento = $_REQUEST["evento"];
$tipo = $_REQUEST["tipo_evento"];
$fecha_final = $_REQUEST["f_final"];

date_default_timezone_set('America/Mexico_City');
$id_evento = date("ymd");
$entrada = date("d/m/y H:i:s");

//realizamos una consulta a la bd
$row = selectsql("id_evento", "evento", "id_evento = ".$id_evento);

if(!$row)
{
    insertarsql("evento", "id_evento, descripcion, tipo, fecha_inicio, fecha_final,
evento_entrada", $id_evento.", "".$evento.", "".$tipo.", "".$date("d/m/y").", "".$fecha_final.",
"".$entrada."");
    header("Location: ../Usuario.php?cadena=Opciones/Asistencia_eventual");
}
else
{
    echo "Ya existe un evento";
?>
<script type="text/javascript">
    window.onload = Confirmar();

    function Confirmar()
    {
        alert("Ya existe un Evento");
        return
location.href="../Usuario.php?cadena=Opciones/Asistencia_eventual";
    }
}
</script>
<?php
}
?>

```

1.1.22. Pantalla Reportes

```

<?php

```

```

include ("conexion.php");
include ("SQL.php");
?>

<script src="Graficas/libraries/RGraph.common.core.js" ></script>
<script src="Graficas/libraries/RGraph.bar.js" ></script>
<script src="Graficas/libraries/RGraph.pie.js"></script>
<script src="js/jquery.js"></script>

<?php
    if(isset($_GET["evento"]))
    {
        $event = $_GET["evento"];
        $sql = "SELECT puesto.nombre, count(puesto.id_puesto), sum(puesto.pago)
                FROM asistencia, eventuales, evento, puesto
                WHERE eventuales.id_eventual = asistencia.id_eventual
                AND evento.id_evento = asistencia.id_evento
                AND puesto.id_puesto = eventuales.id_puesto
                AND evento.id_evento = ".$event."
                AND asistencia.presente = 1
                GROUP BY eventuales.id_puesto, puesto.nombre";

        $query = oci_parse($conn, $sql);
        oci_execute($query);

        $asist_pagos = oci_fetch_array($query, OCI_BOTH);
        $puesto = "".$asist_pagos[0];
        $asistencia = $asist_pagos[1];
        $pagos = $asist_pagos[2];

        while($asist_pagos = oci_fetch_array($query, OCI_BOTH))
        {
            $puesto = $puesto."", "".$asist_pagos[0];
            $asistencia = $asistencia.", "".$asist_pagos[1];
            $pagos = $pagos.", "".$asist_pagos[2];
        }
        $puesto = $puesto."";

        $sql = "SELECT tipo_permiso, COUNT(asistencia.id_permiso)
                FROM Permisos, Asistencia
                WHERE id_evento = ".$event."
                AND asistencia.id_permiso = permisos.id_permiso
                GROUP BY tipo_permiso";

        $query = oci_parse($conn, $sql);
        oci_execute($query);
        $permisos = oci_fetch_array($query, OCI_BOTH);
        $tipo_permiso = "".$permisos[0];
    }
}

```

```

$total_permisos = $permisos[1];
while($permisos = oci_fetch_array($query, OCI_BOTH))
{
    $tipo_permiso = $tipo_permiso."", " ".$permisos[0];
    $total_permisos = $total_permisos."", " ".$permisos[1];
}
$tipo_permiso = $tipo_permiso."";
}
?>

<h1 align="center">REPORTE</h1>
<form action="" method="POST" >
    <h2>Evento:
    <select onchange="this.form.action =
this.value; this.disabled = true; this.form.submit()" >
        <option selected>Seleccione un
evento</option>
        <?php
            //realizamos una consulta
            $sql = "SELECT id_evento,
descripcion FROM evento";
            $query = oci_parse($conn,
oci_execute($query);
            while ($evento =
oci_fetch_array($query, OCI_BOTH))
            {
                <option value =
"Usuario.php?cadena=Opciones/Reportes&evento=<?php echo $evento[0]; ?>"
                <?php echo
$evento[0]." | ".$evento[1]; ?>
            }
        </option>
        <?php
        }
        ?>
    </select></h2>
</form>

<?php
if(isset($_GET["evento"]))
{
    $event = $_GET["evento"];

    if(selectsql("evento.descripcion,
eventuales.nombre_eventual, eventuales.apellidos, permisos.tipo_permiso",

```

```

Asistencia, Eventuales, Evento",
"Eventuales.id_eventual = asistencia.id_eventual
permisos.id_permiso = asistencia.id_permiso
evento.id_evento = asistencia.id_evento
asistencia.presente = 1
evento.id_evento = ".$event))
selectsql_datos("","evento", "id_evento = ".$event);
cellpadding = "5">
<font size = 5><b>EVENTO:</b></font>
<font size = 3><?= $datos_evento[0];?></font>
<font size = 3> | <?= $datos_evento[1];?></font>
cellpadding = "5">
<td><b>TIPO:</b></td>
$datos_evento[2];?></td>
<td><b>Fecha:</b></td>
if($datos_evento[3] == $datos_evento[4])
"Permisos,
AND
AND
AND
AND
AND
{
$datos_evento =
?>
<table border="0"
<tr>
<td>
</td>
<td>
</td>
</tr>
</table>
<br>
<table border="0"
<tr>
<td><b>TIPO:</b></td>
<td><?=
</tr>
<tr>
<td><b>Fecha:</b></td>
<td><?php

```



```

        .Set('shadow', true)

        .Set('colors.sequential', true)

        .Draw();

var image_asistencia = b_asistencia.canvas.toDataURL("images/png");

$.post("Barra_Asistencia.php", {src:image_asistencia} );

</script>

</td>
</tr>
<tr>
<td>

<b>TOTAL DE PAGOS</b>

</td>
</tr>
<tr>
<td>

<canvas width="800" height="250" id="barra_pagos" ></canvas>

<script>

var b_pagos = new RGraph.Bar('barra_pagos', [<?php print($pagos); ?>])

        .Set('labels', [<?php print($puesto); ?>])

        .Set('gutter.left', 30)

        .Set('variant', '3d')

        .Set('linewidth', 3)

        .Set('shadow', true)

        .Set('colors.sequential', true)

        .Draw();

var image_pagos = b_pagos.canvas.toDataURL("images/png");

```

```

$.post("Barra_Pagos.php", {src:image_pagos});

</script>

border="1" cellpadding = "5">
align="center">
"gray">
color=black<b>PUESTO</b></font></td>
color=black<b>ASISTENCIA TOTAL</b></font></td>
color=black<b>PAGO TOTAL</b></font></td>
color=black<b>PAGO EXTRA</b></font></td>

FROM asistencia, eventuales, evento, puesto

WHERE eventuales.id_eventual = asistencia.id_eventual

AND evento.id_evento = asistencia.id_evento

AND puesto.id_puesto = eventuales.id_puesto

AND evento.id_evento = ".$event."

AND asistencia.presente = 1

GROUP BY eventuales.id_puesto, puesto.nombre, asistencia.pago_extra";

oci_parse($conn, $sql);

oci_execute($query);

```

```

oci_fetch_array($query, OCI_BOTH);

align="center"><?php echo $asistencia[0];?></td>
align="center"><?php echo $asistencia[1];?></td>
align="center">$<?php echo $asistencia[2];?></td>
align="center">$<?php echo $asistencia[3];?></td>

oci_fetch_array($query, OCI_BOTH);

4><b>PERMISOS:</b></font>

cellpadding = "5">
align="center">

TOTAL DE PERMISOS

<canvas width="800" height="250" id="pie_permisos"></canvas>

<script>

var p_permisos = new RGraph.Pie('pie_permisos', [<?php print($total_permisos); ?>])

    .Set('chart.labels', [<?php print($tipo_permiso); ?>])

    .Set('variant', '3d')

```

```
$asistencia =
```

```

do
{
?>
<tr>
<td>
<td>
<td>
<td>
</tr>
<?php
}
while($asistencia =

```

```

?>
</table>
<br>
<font size =
<br>
<table border="0"
<tbody
<tr>
<td>
</td>
</tr>
<tr>
<td>

```

```

        .Set('gutter.left', 100)

        .Draw();

var image_permisos = p_permisos.canvas.toDataURL("images/png");

$.post("Barra_Permisos.php", {src:image_permisos} );

</script>
</td>
</tr>
</tbody>
</table>
<table color = "black"
<tbody align="center">
<tr bgcolor =
<td><font
color=black><b>TIPO_PERMISO</b></font></td>
<td><font
color=black><b>TOTAL</b></font></td>
</tr>
</tbody>
<?php
$sql = "SELECT
tipo_permiso, COUNT(asistencia.id_permiso) AS TOTAL
FROM Permisos, Asistencia
WHERE id_evento = ".$sevent."
AND asistencia.id_permiso = permisos.id_permiso
GROUP BY tipo_permiso";

$query =
oci_parse($conn, $sql);
oci_execute($query);

$permisos =
do

```

```

        {
        ?>
            <tr>
                <td
align="center"><?php echo $permisos[0];?></td>
                <td
align="center"><?php echo $permisos[1];?></td>
            </tr>
        <?php
        }
        while($permisos =
oci_fetch_array($query, OCI_BOTH));
    }
    ?>
</table>
<br>
<br>
<?php
        $pago_asistencia =
selectsql_datos("SUM(puesto.pago), SUM(asistencia.presente)",
                "Asistencia, Eventuales, Evento, Puesto",
                "eventuales.id_eventual = asistencia.id_eventual
AND evento.id_evento = asistencia.id_evento
AND puesto.id_puesto = eventuales.id_puesto
AND asistencia.presente = 1
                AND evento.id_evento = ".$event);
    ?>
    <font size = 3><b>Pago a
eventuales:</b>
        <?="
        ".$pago_asistencia[0];?>
    </font>
    <br>
    <font size = 3><b>Asistencia
        <?="
        ".$pago_asistencia[1];?>
    </font>
    <br>
    <br>
    <form
action="ReportePDF.php?evento=<?=$event;?>" method="POST" >

```

```

value="Generar PDF">
</form>
<?php
}
oci_free_statement($query);
oci_close($conn);
?>

```

### 1.1.23. Script ReportePDF

```

<?php
include ("conexion.php");
include ("SQL.php");

require_once('PDF/html2pdf.class.php');
ob_start();
?>
<h1 align="center"><b>REPORTE</b></h1>
<?php
    if(isset($_REQUEST["evento"]))
    {
        $sevent = $_REQUEST["evento"];
        if(selectsql("evento.descripcion, eventuales.nombre_eventual,
eventuales.apellidos, permisos.tipo_permiso",
"Permisos, Asistencia, Eventuales, Evento",
"Eventuales.id_eventual = asistencia.id_eventual
AND permisos.id_permiso =
asistencia.id_permiso
AND evento.id_evento =
asistencia.id_evento
AND asistencia.presente = 1
AND evento.id_evento =
".$sevent))
        {
            $datos_evento = selectsql_datos("*", "evento", "id_evento =
".$sevent);
            ?>
            <br><br>
            <table border="0" cellpadding = "5">
            <tr>
                <td>
                    <font size = 5><b>EVENTO:</b></font>
                </td>
                <td>
                    <font size = 3><?= $datos_evento[0];?></font>
                </td>
                <td>
                    <font size = 3> | <?= $datos_evento[1];?></font>
                </td>
            </tr>

```

```

        </td>
    </tr>
</table>
<br>
<table border="0" cellpadding = "5">
    <tr>
        <td><b>TIPO:</b></td>
        <td><?= $datos_evento[2];?></td>
    </tr>
    <tr>
        <td><b>Fecha:</b></td>
        <td><?php
            if($datos_evento[3] ==
                echo $datos_evento[3];
            else
                echo $datos_evento[3]." -
        </td>
    </tr>
    <tr>
        <td><b>Hora de Entrada:</b></td>
        <td>
            <?= preg_split("/[\s,.]+"/,
                $datos_evento[5])[1];?>
        </td>
    </tr>
</table>
<br><br><br>
<font size = 4><b>ASISTENCIAS:</b></font>
<br>
<table border="0" cellpadding = "5">
    <tbody align="center">
        <tr>
            <td>
                <br><br>
                <b>TOTAL DE ASISTENCIA</b>
            </td>
        </tr>
        <tr>
            <td>
                
            </td>
        </tr>
    </tbody>
</table>

```

```

                <br><br>
                <b>TOTAL DE PAGOS</b>
            </td>
        </tr>
    </tr>
    <tr>
        <td>
            
        </td>
    </tr>
</tbody>
</table>
<br><br><br><br>
<table color = "black" border="1" cellpadding = "5">
    <tbody align="center">
        <tr bgcolor = "gray">
            <td><font
color=black><b>PUESTO</b></font></td>
            <td><font color=black><b>ASISTENCIA
TOTAL</b></font></td>
            <td><font color=black><b>PAGO
TOTAL</b></font></td>
            <td><font color=black><b>PAGO
EXTRA</b></font></td>
        </tr>
    </tbody>
</table>
<?php
    $sql = "SELECT puesto.nombre,
count(puesto.id_puesto), sum(puesto.pago), asistencia.pago_extra

    FROM asistencia, eventuales, evento, puesto

    WHERE eventuales.id_eventual = asistencia.id_eventual

AND evento.id_evento = asistencia.id_evento

AND puesto.id_puesto = eventuales.id_puesto

    AND evento.id_evento = ".$event."

AND asistencia.presente = 1

    GROUP BY eventuales.id_puesto, puesto.nombre, asistencia.pago_extra";

    $query = oci_parse($conn, $sql);
    oci_execute($query);

    $asistencia = oci_fetch_array($query, OCI_BOTH);

```

```

do
{
    ?>
    <tr>
        <td align="center"><?php echo
$asistencia[0];?></td>
        <td align="center"><?php echo
$asistencia[1];?></td>
        <td align="center">$<?php echo
$asistencia[2];?></td>
        <td align="center">$<?php echo
$asistencia[3];?></td>
    </tr>
    <?php
}
while($asistencia = oci_fetch_array($query,
OCI_BOTH));
?>
</table>
<br><br><br><br><br><br><br><br><br>
<font size = 4><b>PERMISOS:</b></font>
<br>
<table border="0" cellpadding = "5">
    <tbody align="center">
        <tr>
            <td>
                <br><br>
                <b>TOTAL DE PERMISOS</b>
                <br><br><br>
            </td>
        </tr>
        <tr>
            <td>
                <br>
                
            </td>
        </tr>
    </tbody>
</table>
<br><br><br><br>
<table color = "black" border="1" cellpadding = "5">
    <tbody align="center">
        <tr bgcolor = "gray">
            <td><font
color=black><b>TIPO_PERMISO</b></font></td>
            <td><font color=black><b>TOTAL</b></font></td>

```

```

        </tr>
    </tbody>
    <?php
COUNT(asistencia.id_permiso) AS TOTAL
        $sql = "SELECT tipo_permiso,
        FROM Permisos, Asistencia
        WHERE id_evento =
        AND asistencia.id_permiso
        GROUP BY tipo_permiso";
        $query = oci_parse($conn, $sql);
        oci_execute($query);
        $permisos = oci_fetch_array($query,
OCI_BOTH);
        do
        {
            ?>
            <tr>
                <td align="center"><?php
                <td align="center"><?php
            </tr>
            <?php
        }
        while($permisos = oci_fetch_array($query,
OCI_BOTH));
    }
    ?>
</table>
<br><br><br>
<?php
    $pago_asistencia = selectsql_datos("SUM(puesto.pago),
SUM(asistencia.presente)",
"Asistencia, Eventuales, Evento, Puesto",
"eventuales.id_eventual = asistencia.id_eventual
    AND evento.id_evento = asistencia.id_evento
    AND puesto.id_puesto = eventuales.id_puesto
    AND asistencia.presente = 1

```

```

AND evento.id_evento = ".$sevent);
    ?>
        <font size = 3><b>Pago a eventuales:</b>
            <?=" ".$pago_asistencia[0];?>
        </font>
        <br>
        <font size = 3><b>Asistencia Total:</b>
            <?=" ".$pago_asistencia[1];?>
        </font>
    <?php
        oci_free_statement($query);
        oci_close($conn);
    }

$filename = 'images/pagos.jpg';

if(isset($_REQUEST['src']))
{
    $src = $_REQUEST['src'];
    $src = substr($src, strpos($src, ",") + 1);
    $decoded = base64_decode($src);

    $fp = fopen($filename,'wb');
    fwrite($fp, $decoded);
    fclose($fp);
}

$htmlbuffer=ob_get_contents();
ob_end_clean();

try
{
    $html2pdf = new HTML2PDF('P', 'A4', 'es');
    $html2pdf->pdf->SetDisplayMode('real');
    $html2pdf->writeHTML($htmlbuffer, isset($_GET['vuehtml']));
    //$html2pdf->Output('utf8.pdf');
    $html2pdf->Output('utf8.pdf');
    echo "<script>windows.open('utf8.pdf','_self','');</script>";
    exit;
}
catch(HTML2PDF_exception $e)
{
    echo $e;
    exit;
}
?>

```

#### 1.1.24. Script Barra\_Asistencia

```
<?php
$filename = 'images/asistencia.jpg';

if(isset($_REQUEST['src']))
{
    $src = $_REQUEST['src'];
    $src = substr($src, strpos($src, ",") + 1);
    $decoded = base64_decode($src);

    $fp = fopen($filename,'wb');
    fwrite($fp, $decoded);
    fclose($fp);
}
?>
```

#### 1.1.25. Script Barra\_Pagos

```
<?php
$filename = 'images/pagos.jpg';

if(isset($_REQUEST['src']))
{
    $src = $_REQUEST['src'];
    $src = substr($src, strpos($src, ",") + 1);
    $decoded = base64_decode($src);

    $fp = fopen($filename,'wb');
    fwrite($fp, $decoded);
    fclose($fp);
}
?>
```

#### 1.1.26. Script Barra\_Permisos

```
<?php
$filename = 'images/permisos.jpg';

if(isset($_REQUEST['src']))
{
    $src = $_REQUEST['src'];
    $src = substr($src, strpos($src, ",") + 1);
    $decoded = base64_decode($src);

    $fp = fopen($filename,'wb');
    fwrite($fp, $decoded);
    fclose($fp);
}
?>
```

### 1.1.27. Pantalla Cambiar\_pass

```

<h1>CAMBIAR CONTRASEÑA</h1>
<font size = 3>Introduce los datos que se te piden: </font>
<?php
    include ("conexion.php");
    include ("SQL.php");
    include ("Blow.php");

    if(isset($_REQUEST["pass"]) &&
$_REQUEST["pass"] != "")
    {
        $pass = $_REQUEST['pass'];
        $passwordBD =
selectsql_datos("password", "usuarios", "nom_usuario = '" . $_REQUEST['nom_usuario'] . "'");
        if(crypt($pass, $passwordBD[0]) !=
$passwordBD[0])
        {
            ?>
                <br><span style="color:
ff0000"><b>Error contraseña anterior</b></span>
            <?php
            }
            else
            {
                //echo isset($_REQUEST["pass2"]);
                if(isset($_REQUEST["pass2"]) &&
isset($_REQUEST["pass3"]) && $_REQUEST["pass2"] != "" && $_REQUEST["pass3"] != "")
                {
                    $pass2 =
$_REQUEST["pass2"];
                    $pass3 =
$_REQUEST["pass3"];

                    if($pass2 == $pass3)
                    {
                        $password =
crypt_blowfish($pass2);

                        actualizarsql("usuarios", "password = '" . $password . "'", "nom_usuario =
'" . $_REQUEST['nom_usuario'] . "'");
                    }
                }
            }
            ?>
                <br><span
style="color: ff8880"><b>Contraseña modificada</b></span>
            <?php
            }
            else
            {
                ?>

```

```

                                <br><span
style="color: ff0000"><b>Contraseña NO coincide</b></span>
                                <?php
                                }
                                }
                                else
                                {
                                ?>
                                <br><span style="color:
ff0000"><b>Error contraseña nueva</b></span>
                                <?php
                                }
                                }
                                }
                                ?>

                                <br><br>
                                <form method = "POST" action =
"Usuario.php?cadena=Opciones/Cambiar_pass">
                                <?php

                                //realizamos una consulta a la bd
                                $sql = "SELECT nom_usuario FROM
usuarios WHERE nom_usuario = '".$_SESSION['nombre']."'";

                                $query = oci_parse($conn, $sql);
                                oci_execute($query);

                                $row = oci_fetch_array($query,
OCI_BOTH);
                                ?>
                                <table border="0" cellpadding = "5">

                                <tr>
                                <td>Usuario:</td>
                                <td><input
name="nom_usuario" size="15" value="<?php echo $row[0]; ?>" readonly></td>
                                </tr>
                                </table>
                                <br>
                                <table border="0" cellpadding = "5">
                                <tr>
                                <td>Contraseña
anterior:</td>
                                <td><input
type="password" name="pass" maxlength="16" size ="18" placeholder="Contraseña Anterior"
autofocus></td>

```

```

        </tr>
        <tr>
            <td>Nueva
contraseña:</td>
            <td><input
type="password" name="pass2" maxlength="16" size ="18" placeholder="Nueva contraseña"
autofocus></td>
        </tr>
        <tr>
            <td>Repite
contraseña:</td>
            <td><input
type="password" name="pass3" maxlength="16" size ="18" placeholder="Nueva contraseña"
autofocus></td>
        </tr>
    </table>
    <br>
    <input type="submit" value="Enviar">
<?php
        oci_free_statement($query);
        oci_close($conn);
?>

```

#### 1.1.28. Script Salir

```

<?php
    session_start();
    session_destroy();
    header("Location: index.php");
?>
<html>
    <head>
        <title>Has salido!!</title>
    </head>
    <body>
        Gracias por tu acceso
        <br>
        <br>
        <a href="index.php">Formulario de autenticación</a>
    </body>
</html>

```

#### 1.1.29. Script Blow

```

<?php
function crypt_blowfish($password, $digito = 8)
{
    $set_salt =
'./1234567890ABCDEFGHIJKLMNQRSTUvwxyz';
    $salt = sprintf('$2a$%02d$', $digito);
}

```

```

        for($i = 0; $i < 22; $i++)
        {
            $salt .= $set_salt[mt_rand(0, 62)];
        }
        return crypt($password, $salt);
    }
?>

```

## 1.2. Applet (JAVASWING)

### 1.2.1. Conexión a base de datos

```

package HuellaDigital;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JOptionPane;

/**
 *
 * @author Marsopas
 */
public class ConexionBD
{
    private Connection con=null;

    public Connection getConexion()
    {
        return con;
    }

    public void setConexion(Connection conexion)
    {
        this.con = conexion;
    }

    public Connection conectar()
    {
        try
        {
            // Cargamos el driver JDBC de Oracle
            Class.forName("oracle.jdbc.OracleDriver");
            String BaseDeDatos = "jdbc:oracle:thin:@localhost:1521:XE";
            con= DriverManager.getConnection(BaseDeDatos,"system","oracle");

            if (con != null)
            {

```

```
        System.out.println("Conexion exitosa!");
    }
    else
        if (con==null)
        {
            throw new SQLException();
        }
    }
    catch(SQLException e)
    {
        JOptionPane.showMessageDialog(null, e.getMessage());
    }
    catch (ClassNotFoundException e)
    {
        JOptionPane.showMessageDialog(null, "Se produjo el siguiente error: "+e.getMessage());
    }
    catch (NullPointerException e)
    {
        JOptionPane.showMessageDialog(null, "Se produjo el siguiente error 2: "+e.getMessage());
    }
    finally
    {
        return con;
    }
}

public boolean ejecutar(String sql)
{
    try
    {
        Statement sentencia;
        sentencia = getConexion().prepareStatement(sql);
        sentencia.executeUpdate(sql);
        getConexion().commit();
        sentencia.close();
    }
    catch (SQLException e)
    {
        e.printStackTrace();
        return false;
    }
    return true;
}

public void desconectar()
{
    con= null;
    System.out.println("Desconexion a base de datos listo...");
}
```

```
}
```

### 1.2.2. Huella

```
package HuellaDigital;

import com.digitalpersona.onetouch.DPFPPDataPurpose;
import com.digitalpersona.onetouch.DPFPPFeatureSet;
import com.digitalpersona.onetouch.DPFPPGlobal;
import com.digitalpersona.onetouch.DPFPSample;
import com.digitalpersona.onetouch.DPFPTemplate;
import com.digitalpersona.onetouch.capture.DPFPCapture;
import com.digitalpersona.onetouch.capture.event.DPFPPDataAdapter;
import com.digitalpersona.onetouch.capture.event.DPFPPDataEvent;
import com.digitalpersona.onetouch.capture.event.DPFPEErrorAdapter;
import com.digitalpersona.onetouch.capture.event.DPFPEErrorEvent;
import com.digitalpersona.onetouch.capture.event.DPFPPReaderStatusAdapter;
import com.digitalpersona.onetouch.capture.event.DPFPPReaderStatusEvent;
import com.digitalpersona.onetouch.capture.event.DPFPSensorAdapter;
import com.digitalpersona.onetouch.capture.event.DPFPSensorEvent;
import com.digitalpersona.onetouch.processing.DPFPEEnrollment;
import com.digitalpersona.onetouch.processing.DPFPPFeatureExtraction;
import com.digitalpersona.onetouch.processing.DPFPPImageQualityException;
import com.digitalpersona.onetouch.verification.DPFPPVerification;
import com.digitalpersona.onetouch.verification.DPFPPVerificationResult;
import java.awt.Image;
import javax.swing.ImageIcon;
import javax.swing.JOptionPane;
import javax.swing.SwingUtilities;
import javax.swing.UIManager;
import java.io.ByteArrayInputStream;
import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 *
 * @author Marsopas
 */
public class Huella extends javax.swing.JApplet
{
    /**
     * Initializes the applet Huella
     */
    @Override
```

```
public void init()
{
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
    * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Huella.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Huella.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Huella.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Huella.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    }
    //</editor-fold>

    /* Create and display the applet */
    new Huella().setVisible(true);
}

public Huella()
{
    try
    {
        UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
    }
    catch (Exception e)
    {
        JOptionPane.showMessageDialog(null, "Imposible modificar el tema visual", "Lookandfeel
inválido.",
```

```

        JOptionPane.ERROR_MESSAGE);
    }
    initComponents();
    txtArea.setEditable(false);
}

/**
 * This method is called from within the init() method to initialize the
 * form. WARNING: Do NOT modify this code. The content of this method is
 * always regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    panHuellas = new javax.swing.JPanel();
    jPanel1 = new javax.swing.JPanel();
    lblImagenHuella = new javax.swing.JLabel();
    panBtns = new javax.swing.JPanel();
    jPanel2 = new javax.swing.JPanel();
    btnSalir = new javax.swing.JButton();
    btnVerificar = new javax.swing.JButton();
    btnIdentificar = new javax.swing.JButton();
    btnGuardar = new javax.swing.JButton();
    jPanel4 = new javax.swing.JPanel();
    jScrollPane1 = new javax.swing.JScrollPane();
    txtArea = new javax.swing.JTextArea();
    jPanel3 = new javax.swing.JPanel();
    jLabel1 = new javax.swing.JLabel();

    addComponentListener(new java.awt.event.ComponentAdapter() {
        public void componentShown(java.awt.event.ComponentEvent evt) {
            formComponentShown(evt);
        }
    });

    panHuellas.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "Huella Digital
Capturada", javax.swing.border.TitledBorder.CENTER,
javax.swing.border.TitledBorder.DEFAULT_POSITION));
    panHuellas.setPreferredSize(new java.awt.Dimension(400, 270));
    panHuellas.setLayout(new java.awt.BorderLayout());

    jPanel1.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.L
OWERED));
    jPanel1.setLayout(new java.awt.BorderLayout());

    lblImagenHuella.addComponentListener(new java.awt.event.ComponentAdapter() {

```

```
        public void componentShown(java.awt.event.ComponentEvent evt) {
            lblImagenHuellaComponentShown(evt);
        }
    });
    jPanel1.add(lblImagenHuella, java.awt.BorderLayout.CENTER);

    panHuellas.add(jPanel1, java.awt.BorderLayout.CENTER);

    getContentPane().add(panHuellas, java.awt.BorderLayout.NORTH);

    panBtns.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "Acciones",
    javax.swing.border.TitledBorder.CENTER, javax.swing.border.TitledBorder.DEFAULT_POSITION));
    panBtns.setPreferredSize(new java.awt.Dimension(400, 190));
    panBtns.setLayout(new java.awt.BorderLayout());

    jPanel2.setPreferredSize(new java.awt.Dimension(366, 90));

    btnSalir.setText("Salir");
    btnSalir.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnSalirActionPerformed(evt);
        }
    });

    btnVerificar.setText("Verificar");
    btnVerificar.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnVerificarActionPerformed(evt);
        }
    });

    btnIdentificar.setText("Identificar");
    btnIdentificar.setPreferredSize(new java.awt.Dimension(71, 23));
    btnIdentificar.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnIdentificarActionPerformed(evt);
        }
    });

    btnGuardar.setText("Guardar");
    btnGuardar.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnGuardarActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
    jPanel2.setLayout(jPanel2Layout);
```

```

jPanel2Layout.setHorizontalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addGap(43, 43, 43)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
    .addComponent(btnIdentificar, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(btnVerificar, javax.swing.GroupLayout.PREFERRED_SIZE, 139,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(44, 44, 44)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
    .addComponent(btnSalir, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(btnGuardar, javax.swing.GroupLayout.PREFERRED_SIZE, 128,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addContainerGap(34, Short.MAX_VALUE))
);
jPanel2Layout.setVerticalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()
            .addGap(11, 11, 11)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(btnVerificar, javax.swing.GroupLayout.PREFERRED_SIZE, 33,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(btnGuardar, javax.swing.GroupLayout.PREFERRED_SIZE, 32,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(btnIdentificar, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(btnSalir, javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addContainerGap())
);

panBtns.add(jPanel2, java.awt.BorderLayout.NORTH);

jPanel4.setLayout(new java.awt.BorderLayout());

txtArea.setColumns(20);
txtArea.setFont(new java.awt.Font("Lucida Sans", 1, 10)); // NOI18N

```

```

txtArea.setRows(5);
jScrollPane1.setViewportView(txtArea);

jPanel4.add(jScrollPane1, java.awt.BorderLayout.CENTER);

panBtns.add(jPanel4, java.awt.BorderLayout.CENTER);

jPanel3.setPreferredSize(new java.awt.Dimension(366, 20));

jLabel1.setFont(new java.awt.Font("Tahoma", 3, 10)); // NOI18N
jLabel1.setForeground(new java.awt.Color(255, 0, 0));
jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel1.setText("Created By: Ing CDIH");

javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);
jPanel3.setLayout(jPanel3Layout);
jPanel3Layout.setHorizontalGroup(
    jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel3Layout.createSequentialGroup()
            .addGap(256, Short.MAX_VALUE)
            .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 132,
javax.swing.GroupLayout.PREFERRED_SIZE))
        );
jPanel3Layout.setVerticalGroup(
    jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel3Layout.createSequentialGroup()
            .addComponent(jLabel1)
            .addGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );

panBtns.add(jPanel3, java.awt.BorderLayout.SOUTH);

getContentPane().add(panBtns, java.awt.BorderLayout.SOUTH);
} // </editor-fold>

private void btnSalirActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    System.exit(0);
}

private void btnVerificarActionPerformed(java.awt.event.ActionEvent evt) {
String nombre = JOptionPane.showInputDialog("Nombre a verificar:");
    verificarHuella(nombre);
Reclutador.clear();
}

private void btnIdentificarActionPerformed(java.awt.event.ActionEvent evt) {

```

```

try
    {
        identificarHuella();
        Reclutador.clear();
    }
    catch (IOException ex)
    {
        Logger.getLogger(Huella.class.getName()).log(Level.SEVERE, null, ex);
    }
}

private void btnGuardarActionPerformed(java.awt.event.ActionEvent evt) {
guardarHuella();
    Reclutador.clear();
    lblImagenHuella.setIcon(null);
start();
}

private void formComponentShown(java.awt.event.ComponentEvent evt) {
// TODO add your handling code here:
Iniciar();
    start();
    EstadoHuellas();
    btnGuardar.setEnabled(false);
    btnIdentificar.setEnabled(false);
    btnVerificar.setEnabled(false);
    btnSalir.grabFocus();
}

private void lblImagenHuellaComponentShown(java.awt.event.ComponentEvent evt) {
// TODO add your handling code here:
}

//Variable que permite iniciar el dispositivo de lector de huella conectado
// con sus distintos metodos.
private DPFPCapture Lector = DPFPGlobal.getCaptureFactory().createCapture();

//Variable que permite establecer las capturas de la huellas, para determina sus características
// y poder estimar la creacion de un template de la huella para luego poder guardarla
private DPFPEnrollment Reclutador = DPFPGlobal.getEnrollmentFactory().createEnrollment();

//Esta variable tambien captura una huella del lector y crea sus características para autenticarla
// o verificarla con alguna guarda en la BD
private DPFPVerification Verificador = DPFPGlobal.getVerificationFactory().createVerification();

//Variable que para crear el template de la huella luego de que se hallan creado las características
// necesarias de la huella si no ha ocurrido ningun problema
private DPFPTemplate template;

```

```
public static String TEMPLATE_PROPERTY = "template";

protected void Iniciar()
{
    Lector.addDataListener(new DPFPDataAdapter()
    {
        public void dataAcquired(final DPFPDataEvent e)
        {
            SwingUtilities.invokeLater(new Runnable()
            {
                public void run()
            {
                EnviarTexto("La Huella Digital ha sido Capturada");
                ProcesarCaptura(e.getSample());
            }
        });
    });
}

Lector.addReaderStatusListener(new DPFPReaderStatusAdapter()
{
    public void readerConnected(final DPFPReaderStatusEvent e)
    {
        SwingUtilities.invokeLater(new Runnable()
        {
            public void run()
        {
            EnviarTexto("El Sensor de Huella Digital esta Activado o Conectado");
        }
        });
    }

    public void readerDisconnected(final DPFPReaderStatusEvent e)
    {
        SwingUtilities.invokeLater(new Runnable()
        {
            public void run()
        {
            EnviarTexto("El Sensor de Huella Digital esta Desactivado o no Conectado");
        }
        });
    }
});

Lector.addSensorListener(new DPFPSensorAdapter()
{
    public void fingerTouched(final DPFPSensorEvent e)
    {
```

```
        SwingUtilities.invokeLater(new Runnable()
        {
            public void run()
            {
                EnviarTexto("El dedo ha sido colocado sobre el Lector de Huella");
            }
        });
    }

    public void fingerGone(final DPFPSensorEvent e)
    {
        SwingUtilities.invokeLater(new Runnable()
        {
            public void run()
            {
                EnviarTexto("El dedo ha sido quitado del Lector de Huella");
            }
        });
    }

    Lector.addErrorListener(new DPFPErroAdapter()
    {
        public void errorReader(final DPFPErroEvent e)
        {
            SwingUtilities.invokeLater(new Runnable()
            {
                public void run()
                {
                    EnviarTexto("Error: "+e.getError());
                }
            });
        }
    });
}

public DPFPPeatureSet featuresinscripcion;
public DPFPPeatureSet featuresverificacion;

public void ProcesarCaptura(DPFPSample sample)
{
    // Procesar la muestra de la huella y crear un conjunto de características con el propósito de
    inscripción.
    featuresinscripcion = extraerCaracteristicas(sample,
    DPFPPDataPurpose.DATA_PURPOSE_ENROLLMENT);

    // Procesar la muestra de la huella y crear un conjunto de características con el propósito de
    verificacion.
```

```

featuresverificacion = extraerCaracteristicas(sample,
DPFPDataPurpose.DATA_PURPOSE_VERIFICATION);

// Comprobar la calidad de la muestra de la huella y lo añade a su reclutador si es bueno
if (featuresinscripcion != null)
try
{
    System.out.println("Las Caracteristicas de la Huella han sido creada");
    Reclutador.addFeatures(featuresinscripcion);// Agregar las caracteristicas de la huella a la
plantilla a crear

    // Dibuja la huella dactilar capturada.
    Image image=CrearImagenHuella(sample);
    DibujarHuella(image);

btnVerificar.setEnabled(true);
    btnIdentificar.setEnabled(true);
}
catch (DPFPImageQualityException ex)
{
    System.err.println("Error: "+ex.getMessage());
}
finally
{
    EstadoHuellas();
    // Comprueba si la plantilla se ha creado.
    switch(Reclutador.getTemplateStatus())
    {
        case TEMPLATE_STATUS_READY: // informe de éxito y detiene la captura de huellas
            stop();
            setTemplate(Reclutador.getTemplate());
            EnviarTexto("La Plantilla de la Huella ha Sido Creada, ya puede Verificarla o Identificarla");
            btnIdentificar.setEnabled(false);
            btnVerificar.setEnabled(false);
            btnGuardar.setEnabled(true);
            btnGuardar.grabFocus();
            break;

        case TEMPLATE_STATUS_FAILED: // informe de fallas y reiniciar la captura de huellas
            Reclutador.clear();
            stop();
            EstadoHuellas();
            setTemplate(null);
            JOptionPane.showMessageDialog(Huella.this, "La Plantilla de la Huella no pudo ser
creada, Repita el Proceso", "Inscripcion de Huellas Dactilares", JOptionPane.ERROR_MESSAGE);
            start();
            break;
    }
}

```

```

    }
}

public DPFPPFeatureSet extraerCaracteristicas(DPFPSample sample, DPFPPDataPurpose purpose)
{
    DPFPPFeatureExtraction extractor =
DPFPPGlobal.getFeatureExtractionFactory().createFeatureExtraction();
    try
    {
        return extractor.createFeatureSet(sample, purpose);
    }
    catch (DPFPImageQualityException e)
    {
        return null;
    }
}

public Image CrearImagenHuella(DPFPSample sample)
{
    return DPFPPGlobal.getSampleConversionFactory().createImage(sample);
}

public void DibujarHuella(Image image)
{
    lblImagenHuella.setIcon(new
ImageIcon(image.getScaledInstance(lblImagenHuella.getWidth(), lblImagenHuella.getHeight(),
Image.SCALE_REPLICATE)); //Image.SCALE_DEFAULT));
    repaint();
}

public void EstadoHuellas()
{
    EnviarTexto("Muestra de Huellas Necesarias para Guardar Template "+
Reclutador.getFeaturesNeeded());
}

public void EnviarTexto(String string)
{
    txtArea.append(string + "\n");
}

public void start()
{
    Lector.startCapture();
    EnviarTexto("Utilizando el Lector de Huella Dactilar ");
}

public void stop()

```

```

    {
        Lector.stopCapture();
        EnviarTexto("No se está usando el Lector de Huella Dactilar ");
    }

    public DPFPTemplate getTemplate()
    {
        return template;
    }

    public void setTemplate(DPFPTemplate template)
    {
        DPFPTemplate old = this.template;
        this.template = template;
        firePropertyChange(TEMPLATE_PROPERTY, old, template);
    }

    ConexionBD con = new ConexionBD();

    /*
     * Guarda los datos de la huella digital actual en la base de datos
     */
    public void guardarHuella()
    {
        //Obtiene los datos del template de la huella actual
        ByteArrayInputStream datosHuella = new ByteArrayInputStream(template.serialize());
        Integer tamañoHuella=template.serialize().length;

        //Pregunta el folio del eventual a la cual corresponde dicha huella
        String folio = JOptionPane.showInputDialog("Folio Eventual:");

    try
        {
            //Establece los valores para la sentencia SQL
            Connection c = con.conectar(); //establece la conexion con la BD
            //String SQL = "UPDATE eventuales SET huella = "+BinaryStream(datosHuella,tamañoHuella)+"
            WHERE id_eventual = "+folio;
            String SQL = "UPDATE eventuales SET huella = ? WHERE id_eventual = ?";
            PreparedStatement guardarStmt = c.prepareStatement(SQL);

            guardarStmt.setBinaryStream(1, datosHuella, tamañoHuella);
            guardarStmt.setString(2,folio);

            //Ejecuta la sentencia
            guardarStmt.executeUpdate();
            guardarStmt.close();

            JOptionPane.showMessageDialog(null,"Huella Guardada Correctamente");
        }
    }

```

```

        btnGuardar.setEnabled(false);
        btnVerificar.grabFocus();
    }
    catch (SQLException ex)
    {
        //Si ocurre un error lo indica en la consola
        System.err.println("Error al guardar los datos de la huella.");
    }
    finally
    {
        con.desconectar();
        System.exit(0);
    }
}

/**
 * Verifica la huella digital actual contra otra en la base de datos
 */
public void verificarHuella(String folio)
{
    try
    {
        //Establece los valores para la sentencia SQL
        Connection c=con.conectar();
        //Obtiene la plantilla correspondiente a la persona indicada
        PreparedStatement verificarStmt = c.prepareStatement("SELECT huella FROM eventuales WHERE
id_eventual=?");
        verificarStmt.setString(1, folio);
        ResultSet rs = verificarStmt.executeQuery();

        //Si se encuentra el nombre en la base de datos
        if (rs.next())
        {
            //Lee la plantilla de la base de datos
            byte templateBuffer[] = rs.getBytes("huella");
            //Crea una nueva plantilla a partir de la guardada en la base de datos
            DPFPTemplate referenceTemplate =
            DPFPGlobal.getTemplateFactory().createTemplate(templateBuffer);
            //Envia la plantilla creada al objeto contenedor de Template del componente de huella digital
            setTemplate(referenceTemplate);

            // Compara las características de la huella recientemente capturada con la
            // plantilla guardada al usuario específico en la base de datos
            DFPFVerificationResult result = Verificador.verify(featuresverificacion, getTemplate());

            //compara las plantillas (actual vs bd)
            if (result.isVerified())

```

```

        JOptionPane.showMessageDialog(null, "Las huella capturada coinciden con la de
"+folio,"Verificacion de Huella", JOptionPane.INFORMATION_MESSAGE);
    else
        JOptionPane.showMessageDialog(null, "No corresponde la huella con "+folio,
"Verificacion de Huella", JOptionPane.ERROR_MESSAGE);

        //Si no encuentra alguna huella correspondiente al nombre lo indica con un mensaje
    }
    else
    {
        JOptionPane.showMessageDialog(null, "No existe un registro de huella para "+folio,
"Verificacion de Huella", JOptionPane.ERROR_MESSAGE);
    }
}
catch (SQLException e)
{
    //Si ocurre un error lo indica en la consola
    System.err.println("Error al verificar los datos de la huella.");
}
finally
{
    con.desconectar();
    System.exit(0);
}
}

/**
 * Identifica a una persona registrada por medio de su huella digital
 */
public void identificarHuella() throws IOException
{
    try
    {
        //Establece los valores para la sentencia SQL
        Connection c=con.conectar();

        //Obtiene todas las huellas de la bd
        PreparedStatement identificarStmt = c.prepareStatement("SELECT huella,
nombre_eventual, apellidos FROM eventuales");
        ResultSet rs = identificarStmt.executeQuery();

        //Si se encuentra el nombre en la base de datos
        while(rs.next())
        {
            //Lee la plantilla de la base de datos
            byte templateBuffer[] = rs.getBytes("huella");
            String nombre=rs.getString("nombre_eventual");
            String apellidos=rs.getString("apellidos");

```

```

        //Crea una nueva plantilla a partir de la guardada en la base de datos
        DPFPTemplate referenceTemplate =
        DPFPGlobal.getTemplateFactory().createTemplate(templateBuffer);
        //Envia la plantilla creada al objeto contenedor de Template del componente de huella digital
        setTemplate(referenceTemplate);

        // Compara las caracteriticas de la huella recientemente capturda con la
        // alguna plantilla guardada en la base de datos que coincide con ese tipo
        DPFPVerificationResult result = Verificador.verify(featuresverificacion, getTemplate());

        //compara las plantilas (actual vs bd)
        //Si encuentra correspondencia dibuja el mapa
        //e indica el nombre de la persona que coincidió.
        if (result.isVerified())
        {
            //crea la imagen de los datos guardado de las huellas guardadas en la base de datos
            JOptionPane.showMessageDialog(null, "Las huella capturada es de "+nombre+"
"+apellidos,"Verificacion de Huella", JOptionPane.INFORMATION_MESSAGE);
            return;
        }
        //Si no encuentra alguna huella correspondiente al nombre lo indica con un mensaje
        JOptionPane.showMessageDialog(null, "No existe ningún registro que coincida con la
huella", "Verificacion de Huella", JOptionPane.ERROR_MESSAGE);
        setTemplate(null);
    }
    catch (SQLException e)
    {
        //Si ocurre un error lo indica en la consola
        System.err.println("Error al identificar huella dactilar."+e.getMessage());
    }
    finally
    {
        con.desconectar();
        System.exit(0);
    }
}

// Variables declaration - do not modify
private javax.swing.JButton btnGuardar;
private javax.swing.JButton btnIdentificar;
private javax.swing.JButton btnSalir;
private javax.swing.JButton btnVerificar;
private javax.swing.JLabel jLabel1;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JPanel jPanel4;

```

```
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JLabel lblImagenHuella;  
private javax.swing.JPanel panBtns;  
private javax.swing.JPanel panHuellas;  
private javax.swing.JTextArea txtArea;  
// End of variables declaration  
}
```

## **Q. Entregables Comprometidos en la Propuesta**

Al concluir el proyecto terminal, se entregarán tres discos compactos al Coordinador de Estudios de Ingeniería en Computación, que incluirán el reporte final del proyecto terminal en un archivo PDF (sin restricciones) y el código fuente de la aplicación en un archivo comprimido (sin restricciones). El reporte final contendrá al menos: portada, resumen, tabla de contenido, objetivos, introducción, desarrollo del proyecto, conclusiones, bibliografía y apéndices. Los apéndices contendrán al menos un listado del código fuente desarrollado.