

**Universidad de El salvador**

**Facultad Multidisciplinaria Oriental**

**Departamento de ingeniería y arquitectura.**

**Ingeniería en sistemas informáticos.**

**Manual de programación.**

**Sistema Informático de Administración de Gimnasios. SIAG.**

**Ing. Diego Herrera**

**Por: Elvin Josué Morales Hernández mh16025**

**Fecha: 20/06/2021**

Contenido

**No se encontraron entradas de tabla de contenido.**

# **Introducción.**

El presente documento describe los aspectos técnicos informáticos del sistema de

información. El documento familiariza al personal técnico especializado encargado de las actividades de mantenimiento, revisión, solución de problemas, instalación y configuración del sistema.

# **Objetivo.**

Instruir el uso adecuado del Sistema de Información, para el acceso oportuno y adecuado en la instalación del mismo, mostrando los pasos a seguir en el proceso de instalación, así como la descripción de los archivos relevantes del sistema los cuales nos orienten en la configuración y soporte del mismo.

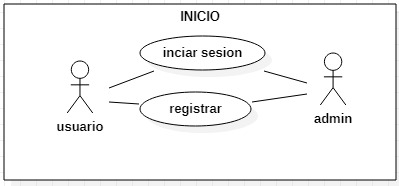
# **Requerimientos técnicos.**

Nuestro sistema de un control en un gimnasio ocupa el sistema operativo Windows 10. El lenguaje de programación que utilice es java NetBeans IDE 8.2

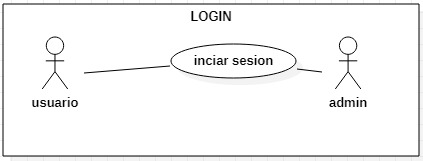
# **Diagramas DDC y DDU**

## **4.1 Diagrama de clases.**

## **4.2 Diagrama de caso de uso al inicio del sistema.**



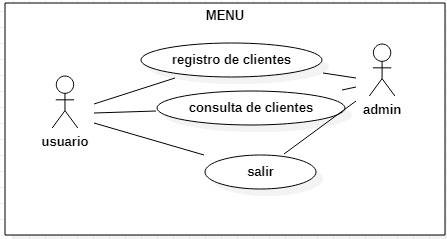
## **4.3 Diagrama de caso de uso al iniciar sesión siendo usuario.**



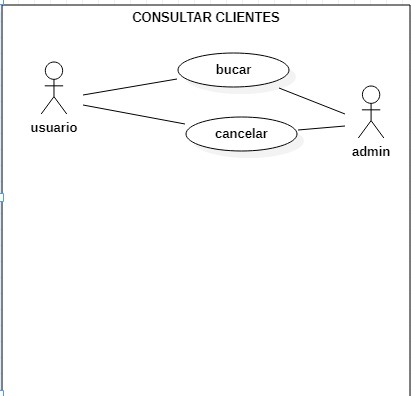
## **4.4 diagrama de caso de uso al registrarse como usuario.**



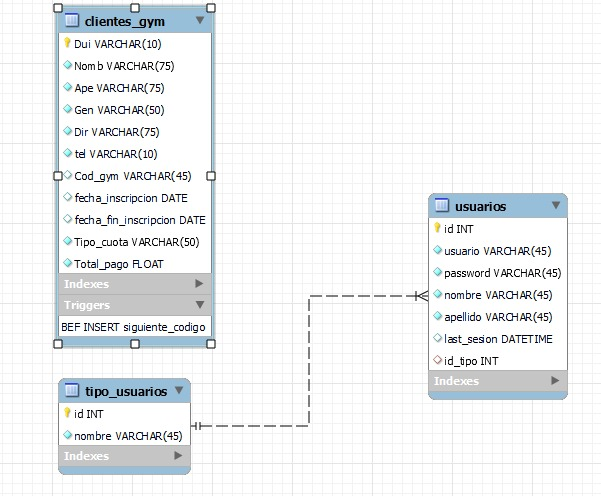
## **4.5 Diagrama de caso de uso de menú del sistema.**



## **4.6 Diagrama de caso de uso al consultar clientes.**



# **Diagrama de entidad relación y normalización de la base de datos.**



# **Descripción de las tablas de la base de datos.**

|  |  |
| --- | --- |
| **Clientes\_gym** | |
| Nombre del campo | Tipo de datos |
| Dui | Varchar (10) |
| nomb | Varchar (75) |
| ape | Varchar (75) |
| Gen | Varchar (50) |
| Dir | Varchar (75) |
| tel | Varchar (10) |
| Cod\_gym | Varchar (45) |
| fecha\_inscripcion | DATE |
| fecha\_fin\_inscripcion | DATE |
| Tipo \_cuota | VARCHAR (50) |
| Total\_pago | FLOAT |

|  |  |
| --- | --- |
| **usuarios** | |
| Nombre del campo | Tipo de datos |
| id | VARCHAR (45) |
| usuario | VARCHAR (45) |
| password | VARCHAR (45) |
| nombre | VARCHAR (45) |
| apellido | VARCHAR (45) |
| last\_sesion | DATETIME |
| id\_tipo | INT |
| **Tipo\_usuarios** | |
| Nombre del campo | Tipo de datos |
| id | INT |
| Nombre | VARCHAR (45) |

# **Definición y descripción de variables usadas.**

**CONEXIÓN**

* public class Conexión: conexión a la base de datos

**SQLUSUARIOS**

* public boolean registrar (usuarios usr) throws SQLException: registrar un usuario y lleva los campos de la base de datos de la tabla usuarios
* public boolean login (usuarios usr) throws SQLException: sirve para el login para que inicie sesión
* public int existeUsuario(String usuario) throws SQLException: si existen usuario si un usuario ya está registrado

**HASH**

* public static String getHash (String txt, String hasType) throws NoSuchAlgorithmException: sirve para confirmar la contraseña

**USUARIOS**

* public class usuarios: contiene todos los métodos ste y get de usuarios id contraseña, nombre, apellido, last\_sesion

**INTERFAZ CLIENTES**

* private final String URL: sirve para la direccion a la base de datoa y se agrega el local sino dará
* public Connection getConexion() throws SQLException : sirve para los driver a la conexión a la base de datos.
* private void BguardarActionPerformed: sirve para hacer un registro para un cliente se conecta a la base de datos hace la consulta PreparedStatement ps con cada campo de texto para que lo que se escriba lo vaya guardando al final
* private void Limpiar() : limpia las cajas de dexto después hacer una acción
* private void BcargarActionPerformed: carga los datos a la tabla haciendo la consulta SELECT.
* private void BeliminarActionPerformed: elimina un registro que hay en la tabla y con la base de datos
* private void BModificarActionPerformed: modifica algún dato que queramos solo seleccionamos un campo en la tabla para poder hacer la modificación
* private void BbuscarActionPerformed: busca un registro que se haiga hecho con el código del gym que se le asigne

**CONSULTA CLIENTE**

* public class ConsultaCliente: se conecta a la base de datos con el url,usuario, contraseña
* private void jButton1ActionPerformed: busca un cliente
* private void jButton2ActionPerformed: abrir la interfaz home y cierra esta interfaz consultar cliente.

**RCLIENTES**

* public class RClientes extends: hacemo el permiso para tener acceso a la base de datos
* private void BbuscarActionPerformed: busca un dato por medio del código
* private void BcargarActionPerformed: cargar los datos de la base de datos hacia la tabla
* private void BguardarActionPerformed: hace un registro
* private void Limpiar(): limpia las cajas de texto después de hacer una acción

**REGISTRO**

* private void BRegistrarActionPerformed: hace un registro para registrar un usuario válido para que no haiga campos vacíos
* private void Limpiar(): limpia las cajas de textos después de hacer una acción

**HOME**

* private void BverActionPerformed: al seleccionar un botón abre y cierra otro conectarse con interfaces

**INICIO**

* private void BRegistrarseActionPerformed: abre para que registre un usuario si dado caso no sea registrado
* private void BIniciarSActionPerformed: abre una interfaz donde iniciara sesión el usuario

**LOGIN**

* BIngresarActionPerformed: se conecta a la base de datos para que pueda iniciar sesión si dado caso hay un usuario y la contraseña.

# **Diccionario de datos.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Llave** | **Nombre** | **Campo** | **Tipo** | **Tamaño** | **Descripción** |
| Pk | Dui | Dui | Varchar | 10 | Documento único de identidad. |
|  | Nombre | Nomb | Varchar | 75 | Nombre del cliente. |
|  | Apellido | Ape | Varchar | 75 | Apellido del cliente. |
|  | Genero | Gen | Varchar | 50 | Genero del cliente. |
|  | Dirección | Dir | Varchar | 75 | Dirección del cliente. |
|  | Teléfono | tel | Varchar | 10 | Teléfono del cliente. |
|  | Código de gimnasio | Cod\_gym | Varchar | 45 | Código auto generable del gimnasio. |
|  | Fecha de inscripción | Fecha\_inscripcion | Date |  | Fecha de la inscripción al gimnasio. |
|  | Fecha de fin de la inscripción. | fecha\_fin\_inscripcion | Date |  | Fecha de finalización de la inscripción al gimnasio. |
|  | Tipo de cuota | Tipo\_cuota | Varchar | 50 | Tipo de la cuota. |
|  | Total, a pagar | Total\_pago | Float | 4 | Total, a cancelar. |

**clientes.**

**tipo\_ usuario**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Llave** | **Nombre** | **Campo** | **Tipo** | **Tamaño** | | **Descripción** |
| Pk | Id | id | INT | |  | Clave del usuario donde define si es administrador o usuario. |
|  | Nombre | nombre | Varchar | | 45 | Nombre del tipo de usuario. |

**usuarios.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Llave** | **Nombre** | **Campo** | **Tipo** | **Tamaño** | **Descripción** |
| Pk | Id | id | INT |  | Es el código del usuario o su identificación. |
|  | Usuario | usuario | Varchar | 45 | El usuario. |
|  | Contraseña | password | Varchar | 45 | Contraseña del usuario. |
|  | Nombre | nombre | Varchar | 45 | Nombre del usuario. |
|  | Apellido | apellido | Varchar | 45 | Apellido del usuario. |
|  | Ultima sesión | last\_sesion | DATETIME |  | Ultimo inicio de sesión. |
| fk | Id del tipo de usuario. | id\_tipo | INT |  | Tipo de usuario según el id asignado. |