Indice

[1. Jornalizacion 1](#__RefHeading___Toc331_1722455891)

[1.1. Entidades 1](#__RefHeading___Toc333_1722455891)

[1.2. Atributos de las entidades 1](#__RefHeading___Toc335_1722455891)

[1.3. Tipos de datos 2](#__RefHeading___Toc337_1722455891)

[1.4. Llaves primarias y llaves foráneas 3](#__RefHeading___Toc339_1722455891)

[2. Controlador de conexión 4](#__RefHeading___Toc341_1722455891)

[3. Transacciones de insercion, eliminacion y modificacion 6](#__RefHeading___Toc865_3811100194)

[4. Script para la creacion de la base de datos 8](#__RefHeading___Toc343_1722455891)

# Jornalizacion

## Entidades

* **Usuarios:** Se utiliza para guardar los usuarios del sistema, esto permite que el sistema sea seguro ya que solicita una contraseña para ingresar al mismo.
* **Roles:** Se utliza para guardar la jerarquia de privilegios de los usuarios.
* **Clientes:** Se utiliza para guardar la información de los clientes.
* **Productos:** Se utiliza para guardar las información de los prductos
* **Facturas:** Se utiliza para guardar el cliente y la factura.
* **DetalleDeFactura:** Se utiliza para guardar el detalle de una factura.

## Atributos de las entidades

Usuarios

* IdUsuario
* Usuario
* Clave

Roles

* IdRol
* IdUsuario
* Tabla
* TipoDePrivilegio

Clientes:

* IdCliente
* NIT
* Nombre
* Teléfono
* Dirección

Productos

* CodigoProducto
* Nombre
* Descripcion
* Precio
* Cantidad

Facturas

* CodigoFactura
* IdCliente

DetalleDeFactura

* CodigoFactura
* CodigoProducto

## Tipos de datos

Usuarios

* IdUsuario: Numero entero
* Usuario: Cadena
* Clave: Cadena

Roles

* IdRol: Numero entero
* Usuario: Cadena
* Tabla: Cadena
* TipoDePrivilegio: Cadena

Clientes:

* IdCliente: Numero entero
* NIT: Numero entero
* Nombre: Cadena
* Teléfono: Numero entero
* Dirección: Cadena

Productos

* CodigoProducto: Numero entero
* Nombre: Cadena
* Descripcion: Cadena
* Precio: Numero decimal
* Cantidad: Numero entero

Facturas

* CodigoFactura: Numero entero
* IdCliente: Numero entero

DetalleDeFactura

* CodigoFactura: Numero entero
* CodigoProducto: Numero entero

## Llaves primarias y llaves foráneas

Usuarios

* IdUsuario PK
* Usuario
* Clave

Roles

* IdRol PK
* Usuario
* Tabla
* TipoDePrivilegio

Clientes:

* IdCliente PK
* NIT
* Nombre
* Teléfono
* Dirección

Productos

* CodigoProducto PK
* Nombre
* Descripcion
* Precio
* Cantidad

Facturas

* CodigoFactura PK
* IdCliente FK

DetalleDeFactura

* CodigoFactura FK
* CodigoProducto FK

# Controlador de conexión

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace Sistema.Conexion

{

internal class ControladorConexion

{

//Variable para crear la instancia de conexion

SqlConnection instanciaConexion = new SqlConnection();

//Variables para estructurar la cadena de conexion

static string servidor = "localhost";

static string bd = "proyecto";

static string usuario = "admin";

static string password = "";

static string puerto = "1433";

//Varaiable para guardar la cadena de conexion

string cadenaDeConexion = "Data Source=" + servidor + "," + puerto + ";" + "user id=" + usuario + ";" + "password=" + password + ";" + "Initial Catalog=" + bd + ";" + "Persist Security Info=true";

//Metodo para realizar prueba de conexion

public SqlConnection pruebaDeConexion()

{

try

{

instanciaConexion.ConnectionString = cadenaDeConexion;

instanciaConexion.Open();

MessageBox.Show("Se conectó correctamente a la Base de Datos");

}

catch (SqlException e)

{

MessageBox.Show("No se logró conectar a la Base de Datos" + e.ToString());

}

return instanciaConexion;

}

//Metodo para validar usuario

public bool validarUsuario(string usuario, string clave)

{

instanciaConexion.ConnectionString = cadenaDeConexion;

instanciaConexion.Open();

SqlCommand comando = new SqlCommand("SELECT \* FROM Usuarios WHERE(Usuario='" + usuario + "' and Clave='" + clave + "')", instanciaConexion);

SqlDataReader reader = comando.ExecuteReader();

if (reader.HasRows)

{

return true;

}

else

{

return false;

}

}

//Metodo para validad el privilegio que tiene el usuario para realizar transacciones

public bool validarPrivilegio(string usuario, string tabla, string transaccion)

{

instanciaConexion.ConnectionString = cadenaDeConexion;

instanciaConexion.Open();

SqlCommand comando = new SqlCommand("SELECT \* FROM Roles WHERE(Usuario='" + usuario + "' and Tabla='" + tabla + "' and TipoDePrivilegio='" + transaccion + "')", instanciaConexion);

SqlDataReader reader = comando.ExecuteReader();

if (reader.HasRows)

{

return true;

}

else

{

return false;

}

}

//Metodo para la ejecucion de transacciones

public void ejecutarTransaccion(string parametroTabla, string parametroValores, string parametroUsuario, string parametroTransaccion)

{

bool resultadoDePrivilegio = false;

resultadoDePrivilegio = validarPrivilegio(parametroUsuario, parametroTabla, parametroTransaccion);

if (resultadoDePrivilegio == true)

{

instanciaConexion.ConnectionString = cadenaDeConexion;

instanciaConexion.Open();

SqlCommand comando = new SqlCommand(parametroValores, instanciaConexion);

comando.ExecuteNonQuery();

instanciaConexion.Close();

}

if (resultadoDePrivilegio == true)

{

MessageBox.Show("No tienes privilegios para la transaccion.");

}

}

//metodo para prueba de instancia

public void pruebaDeInstancia()

{

MessageBox.Show("hola");

}

}//Fin class controlador conexion

}

# Transacciones de insercion, eliminacion y modificacion

**Transacciones para usuarios**

instanciaConexion.ejecutarTransaccion("Usuarios","INSERT INTO Usuarios VALUES('"+textBox1.Text+"','"+textBox2.Text+"');",Form1.usuarioInstanciado,"Insertar");

instanciaConexion.ejecutarTransaccion("Usuarios","DELETE FROM Usuarios WHERE(Usuario='"+textBox1.Text+"');",Form1.usuarioInstanciado,"Eliminar");

instanciaConexion.ejecutarTransaccion("Usuarios","UPDATE Usuarios SET Usuario='"+textBox1.Text+"',Clave='"+textBox2.Text+"' WHERE Usuario='"+textBox1.Text+"';",Form1.usuarioInstanciado,"Modificar");

**Transacciones para Roles**

instanciaConexion.ejecutarTransaccion("Roles","INSERT INTO Roles VALUES("+textBox1.Text+",'"+textBox2.Text+"','"+textBox3.Text+"');",Form1.usuarioInstanciado,"Insertar");

instanciaConexion.ejecutarTransaccion("Roles","DELETE FROM Roles WHERE(IdUsuario="+textBox1.Text+");",Form1.usuarioInstanciado,"Eliminar");

instanciaConexion.ejecutarTransaccion("Roles","UPDATE Roles SET IdUsuario="+textBox1.Text+",Tabla='"+textBox2.Text+"',TipoDePrivilegio='"+textBox3.Text+"' WHERE IdUsuario="+textBox1.Text+";",Form1.usuarioInstanciado,"Modificar");

**Transacciones para Clientes**

instanciaConexion.ejecutarTransaccion("Clientes","INSERT INTO Clientes VALUES("+textBox1.Text+",'"+textBox2.Text+"',"+textBox3.Text+",'"+textBox4.Text+"');",Form1.usuarioInstanciado,"Insertar");

instanciaConexion.ejecutarTransaccion("Clientes","DELETE FROM Clientes WHERE(NIT="+textBox1.Text+");",Form1.usuarioInstanciado,"Eliminar");

instanciaConexion.ejecutarTransaccion("Clientes","UPDATE Clientes SET NIT="+textBox1.Text+",Nombre='"+textBox2.Text+"',Telefono="+textBox3.Text+",Direccion='"+textBox4.Text+"' WHERE NIT='"+textBox1.Text+"';",Form1.usuarioInstanciado,"Modificar");

**Transacciones para Productos**

instanciaConexion.ejecutarTransaccion("Productos","INSERT INTO Productos VALUES("+textBox1.Text+",'"+textBox2.Text+"','"+textBox3.Text+"',"+textBox4.Text+","+textBox5.Text+");",Form1.usuarioInstanciado,"Insertar");

instanciaConexion.ejecutarTransaccion("Productos","DELETE FROM Productos WHERE(CodigoProducto='"+textBox1.Text+"');",Form1.usuarioInstanciado,"Eliminar");

instanciaConexion.ejecutarTransaccion("Productos","UPDATE Productos SET CodigoProducto="+textBox1.Text+",Nombre='"+textBox2.Text+"',Descripcion='"+textBox3.Text+"',Precio="+textBox4.Text+",Cantidad="+textBox4.Text+" WHERE CodigoProducto="+textBox1.Text+";",Form1.usuarioInstanciado,"Modificar");

# Script para la creacion de la base de datos

-- SQLINES LICENSE FOR EVALUATION USE ONLY

CREATE TABLE Usuarios (

IdUsuario BIGINT NOT NULL IDENTITY(1,1) PRIMARY KEY,

Usuario varchar(200) NOT NULL,

Clave TEXT NOT NULL

);

-- SQLINES LICENSE FOR EVALUATION USE ONLY

CREATE TABLE Roles (

IdRol BIGINT NOT NULL IDENTITY(1,1) PRIMARY KEY,

IdUsuario BIGINT NOT NULL,

Tabla TEXT NOT NULL,

TipoDePrivilegio TEXT NOT NULL,

FOREIGN KEY (IdUsuario) REFERENCES Usuarios (IdUsuario)

);

-- SQLINES LICENSE FOR EVALUATION USE ONLY

CREATE TABLE Clientes (

IdCliente BIGINT NOT NULL IDENTITY(1,1) PRIMARY KEY,

NIT BIGINT NOT NULL,

Nombre TEXT NOT NULL,

Telefono BIGINT NOT NULL,

Direccion TEXT NOT NULL

);

-- SQLINES LICENSE FOR EVALUATION USE ONLY

CREATE TABLE Productos (

CodigoProducto BIGINT NOT NULL IDENTITY(1,1) PRIMARY KEY,

Nombre TEXT NOT NULL,

Descripcion TEXT NOT NULL,

Precio DECIMAL(10,2) NOT NULL,

Cantidad BIGINT NOT NULL

);

-- SQLINES LICENSE FOR EVALUATION USE ONLY

CREATE TABLE Facturas (

CodigoFactura BIGINT NOT NULL IDENTITY(1,1) PRIMARY KEY,

IdCliente BIGINT NOT NULL,

FOREIGN KEY (IdCliente) REFERENCES Clientes (IdCliente)

);

-- SQLINES LICENSE FOR EVALUATION USE ONLY

CREATE TABLE DetalleDeFactura (

CodigoFactura BIGINT NOT NULL,

CodigoProducto BIGINT NOT NULL IDENTITY(1,1) PRIMARY KEY,

FOREIGN KEY (CodigoFactura) REFERENCES Facturas (CodigoFactura),

FOREIGN KEY (CodigoProducto) REFERENCES Productos (CodigoProducto)

);

insert into Usuarios values('admin','admin');

/\*Creamos el cliente consumidor final\*/

insert into Clientes values(0,'CF',0,'CF');

/\*CONFIGURACION INICIAR DE PRIVILEGIOS PARA EL USUARIO SUPER ADMINISTRADOR\*/

/\*--------------------------------------------------------------------------\*/

/\*--------------------------------------------------------------------------\*/

/\*Privilegios del usuario super administrador\*/

/\*Privilegios de insercion\*/

insert into Roles values(1,'Usuarios','Insertar');

insert into Roles values(1,'Roles','Insertar');

insert into Roles values(1,'Clientes','Insertar');

insert into Roles values(1,'Productos','Insertar');

insert into Roles values(1,'Facturas','Insertar');

insert into Roles values(1,'DetalleDeFactura','Insertar');

/\*Privilegios de modificacion\*/

insert into Roles values(1,'Usuarios','Modificar');

insert into Roles values(1,'Roles','Modificar');

insert into Roles values(1,'Clientes','Modificar');

insert into Roles values(1,'Productos','Modificar');

insert into Roles values(1,'Facturas','Modificar');

insert into Roles values(1,'DetalleDeFactura','Modificar');

/\*Privilegios de eliminacion\*/

insert into Roles values(1,'Usuarios','Eliminar');

insert into Roles values(1,'Roles','Eliminar');

insert into Roles values(1,'Clientes','Eliminar');

insert into Roles values(1,'Productos','Eliminar');

insert into Roles values(1,'Facturas','Eliminar');

insert into Roles values(1,'DetalleDeFactura','Eliminar');

/\*Privilegios de visualizacion\*/

insert into Roles values(1,'Usuarios','Visualizar');

insert into Roles values(1,'Roles','Visualizar');

insert into Roles values(1,'Clientes','Visualizar');

insert into Roles values(1,'Productos','Visualizar');

insert into Roles values(1,'Facturas','Visualizar');

insert into Roles values(1,'DetalleDeFactura','Visualizar');

/\*Fin privilegios del usuario super administrador\*/

/\*--------------------------------------------------------------------------\*/