**Un librero con libros

Descripción generada automáticamenteFACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA**

**ING. RUBÉN MIGUEL RIOJAS RODRIGUEZ**

**TECNOLOGÍAS DE INFORMACIÓN**

**“SISTEMA DE TIENDA DEL LIBROS”**

**EQUIPO:**

**•DANIELA LÓPEZ RODRIGUEZ - 19189830**

**•DENISE VILLARREAL HINOJOSA 19194929**

**9° A ISC T.V.**

**AGOSTO – DICIEMBRE 2023**

**27/11/2023**

Índice

[DIAGRAMA ENTIDAD – RELACIÓN 4](#_Toc151997341)

[DIAGRAMA RELACIONAL 5](#_Toc151997342)

[DICCIONARIO DE DATOS. 5](#_Toc151997343)

[Descuento 5](#_Toc151997344)

[Tipo\_Cliente 5](#_Toc151997345)

[Venta 6](#_Toc151997346)

[Detalle\_Venta 6](#_Toc151997347)

[Rating 6](#_Toc151997348)

[Puesto 6](#_Toc151997349)

[Usuario 6](#_Toc151997350)

[Log 7](#_Toc151997351)

[Tipo\_Usuario 7](#_Toc151997352)

[Tipo\_Pago 7](#_Toc151997353)

[Categoria 8](#_Toc151997354)

[Nivel 8](#_Toc151997355)

[Libro 8](#_Toc151997356)

[Tipo\_Formato 8](#_Toc151997357)

[Departamento 9](#_Toc151997358)

[Autor 9](#_Toc151997359)

[Editorial 9](#_Toc151997360)

[Pais 9](#_Toc151997361)

[Estado 10](#_Toc151997362)

[Ciudad 10](#_Toc151997363)

[Libro\_Autores 10](#_Toc151997364)

[BASES DE DATOS 11](#_Toc151997365)

[Microsoft SQL Server 11](#_Toc151997366)

[MariaDB 14](#_Toc151997367)

[PostgreSQL 18](#_Toc151997368)

[18](#_Toc151997369)

[18](#_Toc151997370)

[Programas 20](#_Toc151997371)

[C# 20](#_Toc151997372)

[VB.net 86](#_Toc151997373)

[HTML 152](#_Toc151997374)

## Diagrama, Esquemático Descripción generada automáticamenteDIAGRAMA ENTIDAD – RELACIÓN

## Diagrama, Esquemático Descripción generada automáticamenteDIAGRAMA RELACIONAL

## DICCIONARIO DE DATOS.

### Descuento

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo de dato | PK | Null/Not null |
| Id\_Descuento | int | PK | Not Null |
| Porcentaje | Decimal(3,2) |  | Not Null |
| Estatus | bit |  | Not Null |

### Tipo\_Cliente

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo de dato | PK | Null/Not null |
| Id\_Tipo\_Cliente | int | PK | Not Null |
| Nombre | Varchar(30) |  | Not Null |
| Id\_Usuario\_Creacion | Int | FK | Not Null |
| Fecha\_Hora\_creacion | Datetime |  | Not Null |
| Id\_Usuario\_Modificacion | Int | FK | Not Null |
| Fecha\_Hora\_modificacion | Datetime |  | Not Null |
| Estatus | bit |  | Not Null |

### Venta

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo de dato | PK | Null/Not null |
| Id\_Venta | Int | PK | Not Null |
| Id\_tipo\_Cliente | Int | FK | Not Null |
| Id\_Usuario\_Creacion | Int | FK | Not Null |
| Fecha\_Hora\_creacion | Datetime |  | Not Null |
| Id\_Usuario\_Modificacion | Int | FK | Not Null |
| Fecha\_Hora\_modificacion | Datetime |  | Not Null |
| Estatus | Bit |  | Not Null |
| Id\_Tipo\_Pago | Int | FK | Not Null |

### Detalle\_Venta

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo de dato | PK | Null/Not null |
| Id\_Detalle | Int | PK | Not Null |
| Id\_Venta | Int | FK | Not Null |
| Id\_Descuento | Int | FK | Null |
| Id\_Libro\_ ISBN | Varchar(50) | FK | Not Null |
| Id\_Usuario\_Creacion | Int | FK | Not Null |
| Fecha\_Hora\_creacion | Datetime |  | Not Null |
| Id\_Usuario\_Modificacion | Int | FK | Not Null |
| Fecha\_Hora\_modificacion | Datetime |  | Not Null |
| Estatus | Bit |  | Not Null |

### Rating

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo de dato | PK | Null/Not null |
| Id\_Rating | Int | PK | Not Null |
| Valor | Int |  | Not Null |
| Estatus | Bit |  | Not Null |

### Puesto

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo de dato | PK | Null/Not null |
| Id\_Puesto | Int | PK | Not Null |
| Nombre | Varchar(30) |  | Not Null |
| Id\_Usuario\_Creacion | Int | FK | Not Null |
| Fecha\_Hora\_creacion | Datetime |  | Not Null |
| Id\_Usuario\_Modificacion | Int | FK | Not Null |
| Fecha\_Hora\_modificacion | Datetime |  | Not Null |
| Estatus | Bit |  | Not Null |

### Usuario

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo de dato | PK | Null/Not null |
| Id\_Usuario | Int | PK | Not Null |
| Nombre\_Usuario | Varchar(30) |  | Not Null |
| Contraseña | Varchar(max) |  | Not Null |
| Nombre | Varchar(30) |  | Not Null |
| Apellido\_Paterno | Varchar(50) |  | Not Null |
| Apellido\_Materno | Varchar(50) |  | Null |
| Email | Varchar(200) |  | Not Null |
| Teléfono | Varchar(10) |  | Null |
| Id\_Puesto | Int | FK | Not Null |
| Id\_Departamento | Int | FK | Not Null |
| Id\_Usuario\_Creacion | Int | FK | Not Null |
| Fecha\_Hora\_creacion | Datetime |  | Not Null |
| Id\_Usuario\_Modificacion | Int | FK | Not Null |
| Fecha\_Hora\_modificacion | Datetime |  | Not Null |
| Id\_Tipo\_Usuario | Int | FK | Not Null |
| Estatus | bit |  | Not Null |

### Log

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo de dato | PK | Null/Not null |
| Id\_Log | Int | PK | Not Null |
| Id\_Usuario | Int | FK | Not Null |
| Fecha\_Hora | Datetime |  | Not Null |
| Accion | Varchar(50) |  | Not Null |
| Tabla | Varchar(50) |  | Not Null |
| Estatus | bit |  | Not Null |
| Id\_nivel | int | FK | Not Null |

### Tipo\_Usuario

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo de dato | PK | Null/Not null |
| Id\_Tipo\_Usuario | Int | PK | Not Null |
| Nombre | Varchar(30) |  | Not Null |
| Id\_Usuario\_Creacion | Int | FK | Not Null |
| Fecha\_Hora\_creacion | Datetime |  | Not Null |
| Id\_Usuario\_Modificacion | Int | FK | Not Null |
| Fecha\_Hora\_modificacion | Datetime |  | Not Null |
| Estatus | bit |  | Not Null |

### Tipo\_Pago

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo de dato | PK | Null/Not null |
| Id\_Tipo\_Pago | Int | PK | Not Null |
| Nombre | Varchar(30) |  | Not Null |
| Id\_Usuario\_Creacion | Int | FK | Not Null |
| Fecha\_Hora\_creacion | Datetime |  | Not Null |
| Id\_Usuario\_Modificacion | Int | FK | Not Null |
| Fecha\_Hora\_modificacion | Datetime |  | Not Null |
| Estatus | bit |  | Not Null |

### Categoria

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo de dato | PK | Null/Not null |
| Id\_Categoria | Int | PK | Not Null |
| Nombre | Varchar(30) |  | Not Null |
| Id\_Usuario\_Creacion | Int | FK | Not Null |
| Fecha\_Hora\_creacion | Datetime |  | Not Null |
| Id\_Usuario\_Modificacion | Int | FK | Not Null |
| Fecha\_Hora\_modificacion | Datetime |  | Not Null |
| Estatus | bit |  | Not Null |

### Nivel

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo de dato | PK | Null/Not null |
| Id\_Nivel | Int | PK | Not Null |
| Valor | Varchar(50) |  | NotNull |
| Estatus | bit |  | Not Null |

### Libro

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo de dato | PK | Null/Not null |
| Id\_Libro\_ISBN | Varchar(50) | PK | Not Null |
| Titulo | Varchar(250) |  | Not Null |
| Id\_Categoria | Int | FK | Not Null |
| Volumen | Varchar(5) |  | Null |
| Id\_Tipo\_Formato | int | FK | Not Null |
| Id\_Editorial | Varchar(3) | FK | Not Null |
| Fecha\_publicacion | Date |  | Null |
| Paginas | Int |  | Null |
| Precio | Decimal(10,2) |  | Not Null |
| Id\_Usuario\_Creacion | Int | FK | Not Null |
| Fecha\_Hora\_creacion | Datetime |  | Not Null |
| Id\_Usuario\_Modificacion | Int | FK | Not Null |
| Fecha\_Hora\_modificacion | Datetime |  | Not Null |
| Id\_Rating | int | FK | Not Null |
| Estatus | bit |  | Not Null |

### Tipo\_Formato

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo de dato | PK | Null/Not null |
| Id\_Tipo\_Formato | Int | PK | Not Null |
| Nombre | Varchar(30) |  | Not Null |
| Id\_Usuario\_Creacion | Int | FK | Not Null |
| Fecha\_Hora\_creacion | Datetime |  | Not Null |
| Id\_Usuario\_Modificacion | Int | FK | Not Null |
| Fecha\_Hora\_modificacion | Datetime |  | Not Null |
| Estatus | bit |  | Not Null |

### Departamento

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo de dato | PK | Null/Not null |
| Id\_Departamento | Int | PK | Not Null |
| Departamento | Varchar(50) |  | Not Null |
| Id\_Usuario\_Creacion | Int | FK | Not Null |
| Fecha\_Hora\_creacion | Datetime |  | Not Null |
| Id\_Usuario\_Modificacion | Int | FK | Not Null |
| Fecha\_Hora\_modificacion | Datetime |  | Not Null |
| Estatus | bit |  | Not Null |

### Autor

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo de dato | PK | Null/Not null |
| Id\_Autor | Varchar(10) | PK | Not Null |
| Nombre | Varchar(50) |  | Not Null |
| Apellido | Varchar(50) |  | Not Null |
| Fecha\_Nacimiento | Date |  | Null |
| Id\_pais | Int | FK | Null |
| Id\_Usuario\_Creacion | Int | FK | Not Null |
| Fecha\_Hora\_creacion | Datetime |  | Not Null |
| Id\_Usuario\_Modificacion | Int | FK | Not Null |
| Fecha\_Hora\_modificacion | Datetime |  | Not Null |
| Estatus | bit |  | Not Null |

### Editorial

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo de dato | PK | Null/Not null |
| Id\_Editorial | Varchar(3) | PK | Not Null |
| Nombre | Varchar(50) |  | Not Null |
| Id\_Ciudad | Int | FK | Not Null |
| Año\_Inauguración | Varchar(4) |  | Null |
| Id\_Usuario\_Creacion | Int | FK | Not Null |
| Fecha\_Hora\_creacion | Datetime |  | Not Null |
| Id\_Usuario\_Modificacion | Int | FK | Not Null |
| Fecha\_Hora\_modificacion | Datetime |  | Not Null |
| Estatus | bit |  | Not Null |

### Pais

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo de dato | PK | Null/Not null |
| Id\_Pais | Int | PK | Not Null |
| Nombre | Varchar(150) |  | Not Null |
| Nacionalidad | Varchar(150) |  | Not Null |
| Estatus | bit |  | Not Null |

### Estado

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo de dato | PK | Null/Not null |
| Id\_Estado | Int | PK | Not Null |
| Id\_Pais | Int | FK | Not Null |
| Nombre | Varchar(100) |  | Not Null |
| Estatus | bit |  | Not Null |

### Ciudad

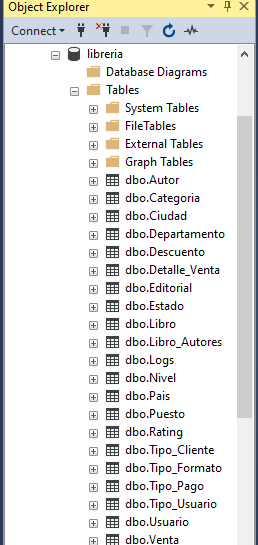
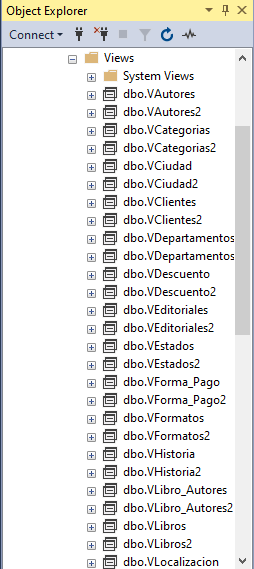
|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo de dato | PK | Null/Not null |
| Id\_Ciudad | Int | PK | Not Null |
| Id\_Estado | Int | FK | Not Null |
| Nombre | Varchar(250) |  | Not Null |
| Estatus | bit |  | Not Null |

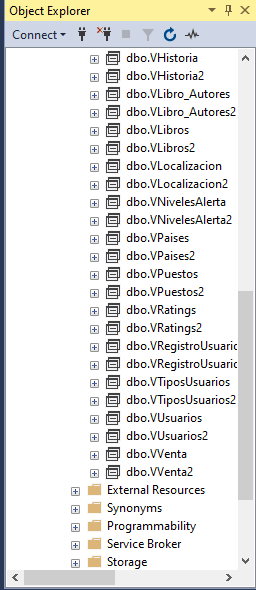
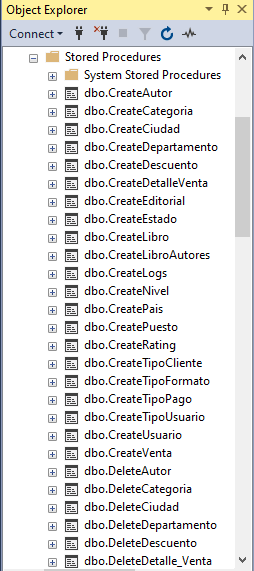
### Libro\_Autores

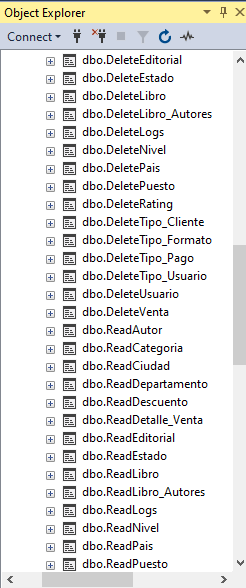
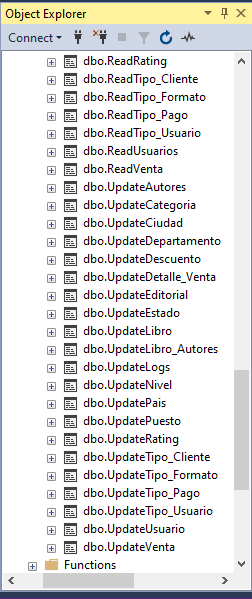
|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo de dato | PK | Null/Not null |
| Id\_Libro\_Autores | Int | PK | Not Null |
| Id\_Libro\_ISBN | Varchar(50) | FK | Not Null |
| Id\_Autor | Varchar(10) | FK | Not Null |
| Id\_Usuario\_Creacion | Int | FK | Not Null |
| Fecha\_Hora\_creacion | Datetime |  | Not Null |
| Id\_Usuario\_Modificacion | Int | FK | Not Null |
| Fecha\_Hora\_modificacion | Datetime |  | Not Null |
| Estatus | bit |  | Not Null |

## BASES DE DATOS

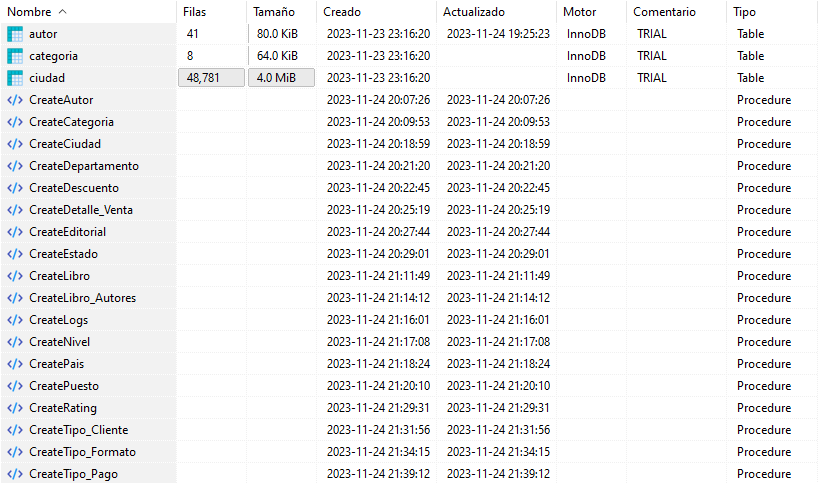
### Microsoft SQL Server

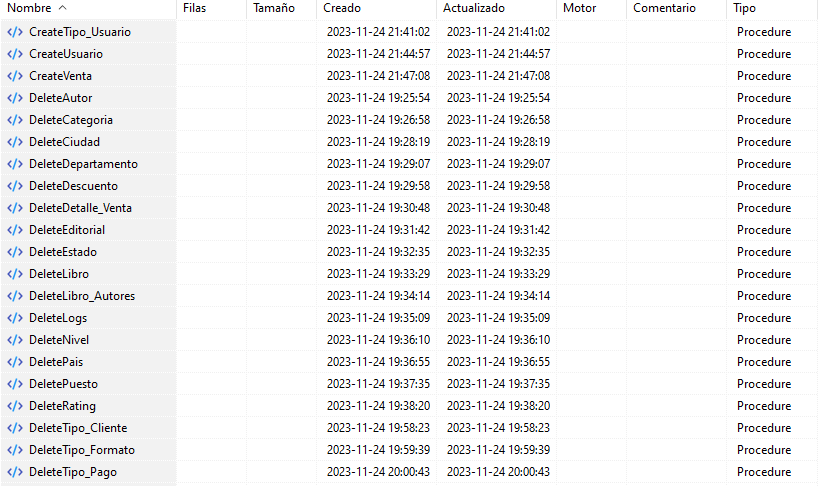
 

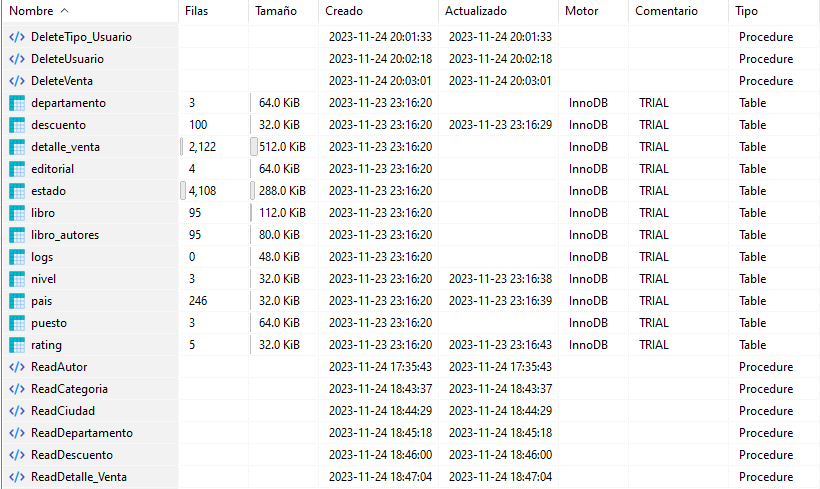
 

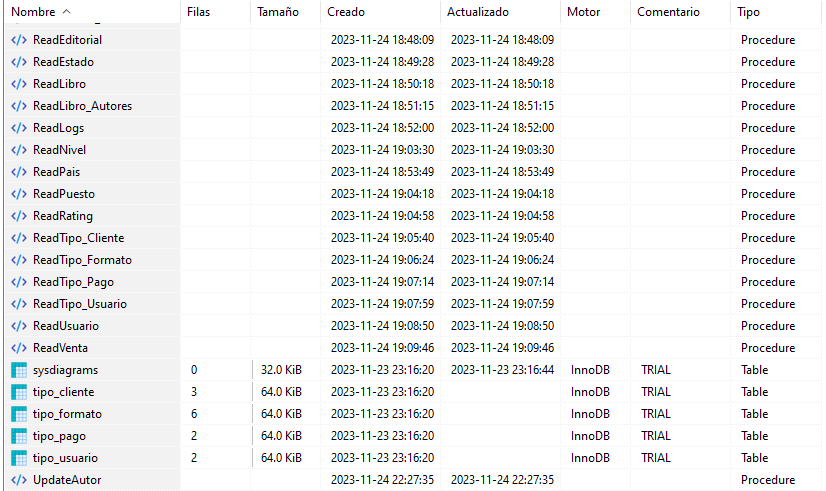
 

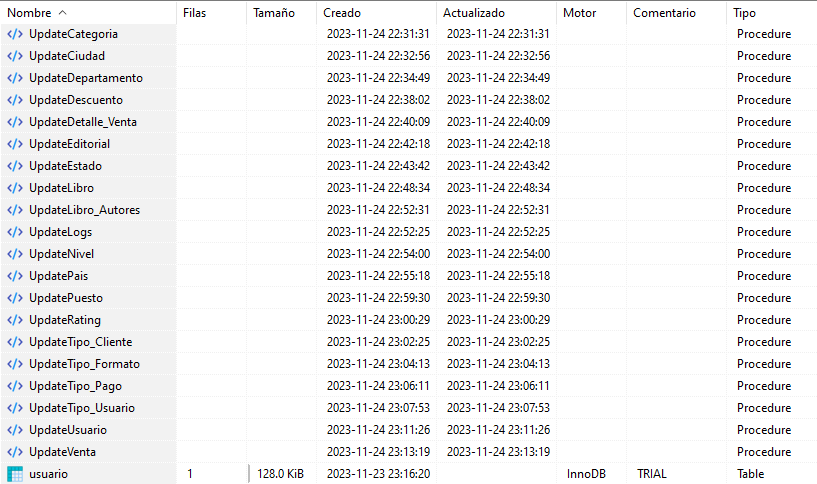
### MariaDB

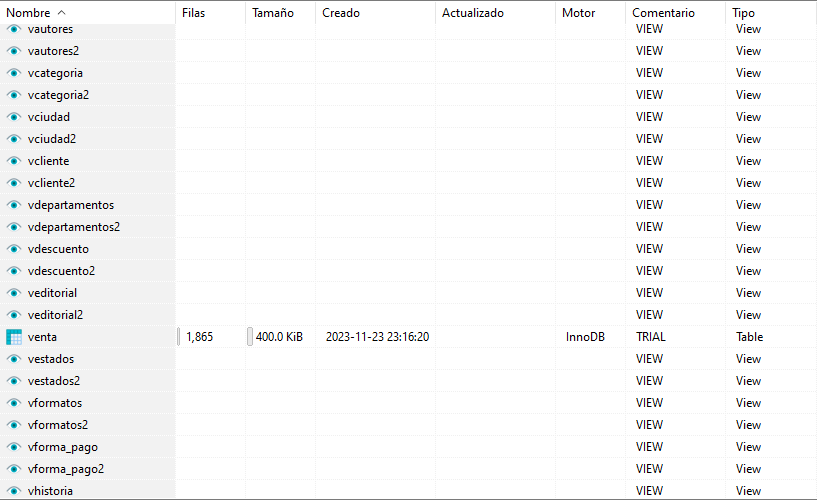


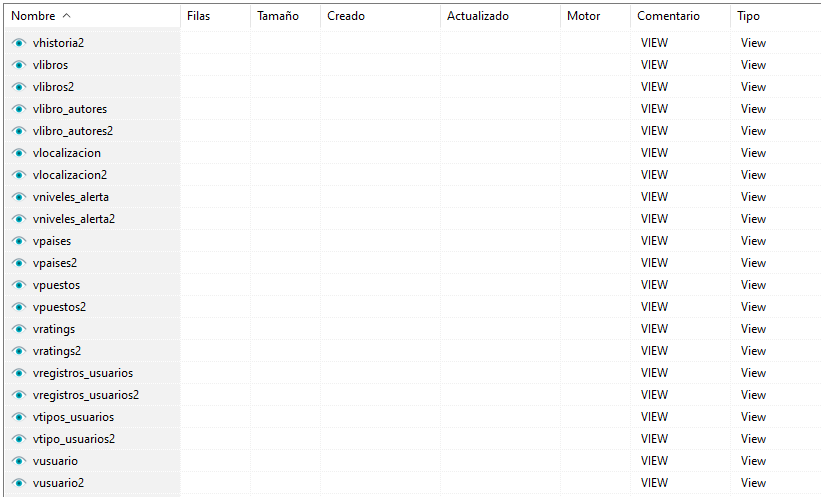








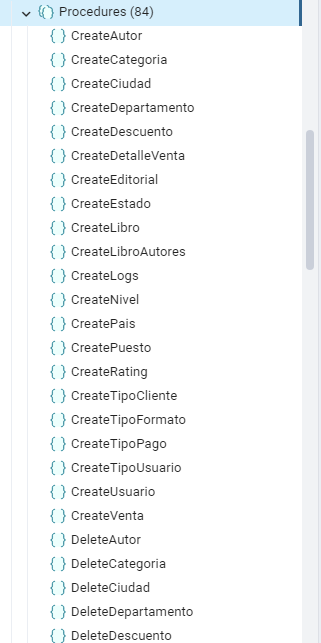
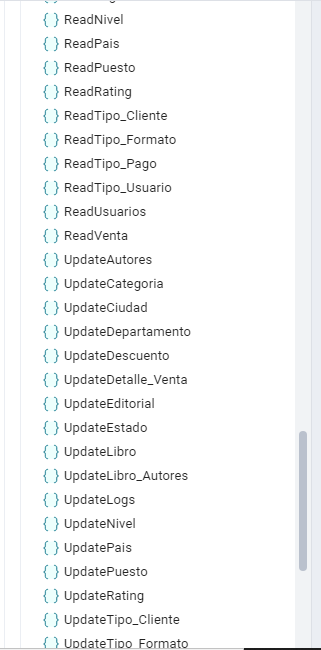
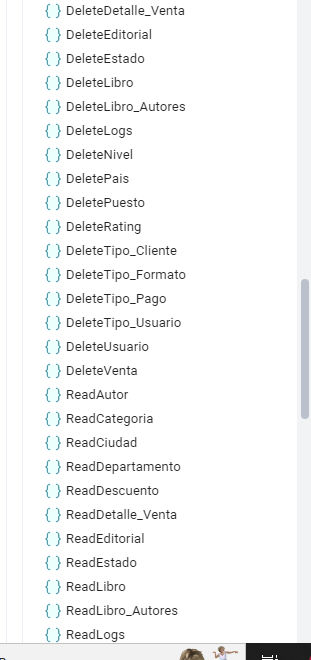
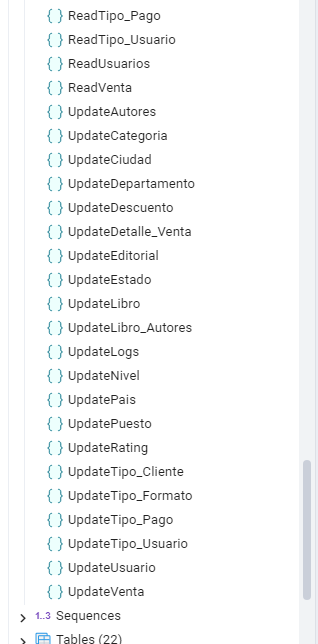




### PostgreSQL

### 

### 

## Programas

### C#

using libreriaC.libreriaDataSetTableAdapters;

using System;

using System.Collections;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Data.SqlTypes;

using System.Drawing;

using System.Linq;

using System.Net.NetworkInformation;

using System.Security.Cryptography;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using static System.Net.Mime.MediaTypeNames;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

using static System.Windows.Forms.VisualStyles.VisualStyleElement.ListView;

namespace libreriaC

{

public partial class Form1 : Form

{

private SqlCommand cmd;

private int IDuser = 0,tipo\_User=0;

private string con="";

private int modCategoria=0;

private string modEditorial = "";

private int modCliente = 0;

private int modPago = 0;

private int modDepartamento = 0;

private int modPuesto = 0;

private int modFormato = 0;

private string modLibro = "";

private string modAutor = "";

private int modUsuario = 0;

private string modcon = "";

private int modVenta = 0;

public Form1()

{

InitializeComponent();

}

private void btnAcceder\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(txtCorreoLogin.Text)) {

txtCorreoLogin.Select();

MessageBox.Show("Favor de llenar todos los campos solicitados");

}

else if (string.IsNullOrEmpty(txtContraLogin.Text)) {

txtContraLogin.Select();

MessageBox.Show("Favor de llenar todos los campos solicitados");

}

else {

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

string consulta = "Select Id\_Usuario,Id\_Tipo\_Usuario from Usuario where Email='" + txtCorreoLogin.Text + "'and Contraseña='" + Encriptado(txtContraLogin.Text) + "'";

cmd = new SqlCommand(consulta, sql);

SqlDataReader lector;

lector = cmd.ExecuteReader();

if (lector.HasRows)

{

while (lector.Read())

{

IDuser = (int)lector[0];

tipo\_User = (int)lector[1];

MessageBox.Show("Se inicio la sesion");

}

if (tipo\_User == 1)

{

Menu.Enabled = true;

Menu.Visible = true;

}

else

{

UsuariosToolStripMenuItem.Visible = false;

UsuariosToolStripMenuItem.Enabled = false;

CatalogosToolStripMenuItem.Visible = false;

CatalogosToolStripMenuItem.Visible = false;

Menu.Enabled = true;

Menu.Visible = true;

}

txtContraLogin.Clear();

PanelLogin.Enabled = false;

PanelLogin.Visible = false;

LogsAceptable("Iniciar Sesion", "Logs");

}

else

{

txtContraLogin.Clear();

txtCorreoLogin.Clear();

MessageBox.Show("El usuario no existe o datos incorrectos");

}

}

}

catch(Exception ex) { MessageBox.Show("Error en conexion"); }

}

}

private string Encriptado(string co)

{

using (var sha256 = new SHA256Managed())

{

return BitConverter.ToString(sha256.ComputeHash(Encoding.UTF8.GetBytes(co))).Replace("-", "");

}

}

private void Form1\_Load(object sender, EventArgs e)

{

cerrarPaneles();

Menu.Visible=false;

Menu.Enabled=false;

PanelLogin.Enabled = true;

PanelLogin.Visible= true;

}

private void LogsAceptable(string accion,string tabla ) {

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("CreateLogs", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Accion", accion));

cmd.Parameters.Add(new SqlParameter("@Tabla", tabla));

cmd.Parameters.Add(new SqlParameter("@Estatus", 1));

cmd.Parameters.Add(new SqlParameter("@Id\_Nivel", 1));

cmd.ExecuteReader();

}

}

catch (Exception ex)

{

}

}

private void Label4\_Click(object sender, EventArgs e)

{

}

private void cerrarPaneles() {

PanelRegistra.Enabled = false;

PanelRegistra.Visible= false;

PanelCateEditor.Visible = false;

PanelCateEditor.Enabled = false;

gbCategoria.Enabled = false;

gbCategoria.Visible = false;

gbEditorial.Visible = false;

gbEditorial.Enabled = false;

pTiClienteTiPago.Enabled = false;

pTiClienteTiPago.Visible = false;

gbClientes.Visible = false;

gbClientes.Enabled = false;

gpPago.Visible = false;

gpPago.Enabled = false;

PanelDepartamentosPuestos.Visible = false;

PanelDepartamentosPuestos.Enabled = false;

gbPuestos.Enabled = false;

gbPuestos.Visible = false;

gbDepartamento.Enabled = false;

gbDepartamento.Visible = false;

panelLibrosFormatos.Enabled = false;

panelLibrosFormatos.Visible = false;

gbFormato.Enabled = false;

gbFormato.Visible = false;

panelLibros.Enabled = false;

panelLibros.Visible = false;

PanelAutor.Enabled = false;

PanelAutor.Visible = false;

panelUsuarioAM.Enabled = false;

panelUsuarioAM.Visible = false;

panelVentaRegistro.Enabled = false;

panelVentaRegistro.Visible = false;

PanelDetalleVenta.Enabled = false;

PanelDetalleVenta.Visible = false;

PanelVentas.Enabled = false;

PanelVentas.Visible = false;

PanelUsuarios.Enabled = false;

PanelUsuarios.Visible = false;

PanelHistorial.Enabled = false;

PanelHistorial.Visible = false;

PanelAutorAM.Enabled = false;

PanelAutorAM.Visible = false;

}

private void cerrarSesiónToolStripMenuItem\_Click(object sender, EventArgs e)

{

cerrarPaneles();

Menu.Visible = false;

Menu.Enabled= false;

MessageBox.Show("Sesión cerrada");

PanelLogin.Enabled = true;

PanelLogin.Visible = true;

txtContraLogin.Clear();

txtCorreoLogin.Clear();

IDuser = 0;

tipo\_User = 0;

}

private void txtContraMiCuenta\_Enter(object sender, EventArgs e)

{

txtContraMiCuenta.Text = "";

}

private void btnGuardarReg\_Click(object sender, EventArgs e)

{

if (!(string.IsNullOrEmpty(txtContraMiCuenta.Text) || string.IsNullOrWhiteSpace(txtContraMiCuenta.Text)) && !(txtContraMiCuenta.Text.Equals(con)))

{

con = Encriptado(txtContraMiCuenta.Text);

}

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

string consulta = "update Usuario set Nombre\_Usuario='" + txtUsuarioMiCuenta.Text + "',Nombre='" + txtNombreMiCuenta.Text + "'," +

"Contraseña='" + con + "',Apellido\_Paterno='" + txtApellidoPMiCuenta.Text + "',Apellido\_Materno='" + txtApellidoMMiCuenta.Text +

"',Email='" + txtCorreoMiCuenta.Text + "',Telefono='" + txtTelefMiCuenta.Text + "',Id\_Usuario\_Modificacion=" + IDuser.ToString() +

",Fecha\_Hora\_Modificacion= '" + DateTime.Now.ToString("yyyy/MM/dd HH:mm:ss.fff") +

"' where Id\_Usuario=" + IDuser.ToString();

cmd = new SqlCommand(consulta, sql);

cmd.ExecuteReader();

PanelRegistra.Visible = true;

PanelRegistra.Enabled = true;

LogsAceptable("Modificar mi perfil", "Usuario");

sql.Close();

MessageBox.Show("Guardado");

}

}

catch (Exception ex) { MessageBox.Show("Error en conexion"); }

}

private void radioButton1\_CheckedChanged(object sender, EventArgs e)

{

}

private void dgvLibros\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void dgvLibros\_CellClick(object sender, DataGridViewCellEventArgs e)

{

}

private void btnAgregCate\_Click(object sender, EventArgs e)

{

gbCategoria.Visible = true;

gbCategoria.Enabled=true;

gbCategoria.BringToFront();

txtUpCreNombreCat.Text = "";

ckbCat.Checked = true;

modCategoria = 0;

}

private void btnUpdCat\_Click(object sender, EventArgs e)

{

if (dgvCategoria.SelectedRows.Count == 1) {

modCategoria =(int) dgvCategoria.SelectedCells[0].Value;

txtUpCreNombreCat.Text = (string)dgvCategoria.SelectedCells[1].Value;

if ((bool)dgvCategoria.SelectedCells[2].Value) {

ckbCat.Checked = true;

}

else {

ckbCat.Checked = false;

}

gbCategoria.Visible = true;

gbCategoria.Enabled = true;

gbCategoria.BringToFront();

}

else { MessageBox.Show("Seleccione una categoria."); }

}

private void recargaCatalogoCategoria() {

DataTable dt = new DataTable();

try {

using (SqlConnection sql= new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd= new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select Id\_Categoria,Nombre,Estatus from Categoria";

using (SqlDataAdapter da =new SqlDataAdapter(cmd))

{

da.Fill(dt);

}

}

sql.Close();

}

}

catch (Exception ex) {

MessageBox.Show(ex.Message);

}

dgvCategoria.DataSource = dt;

//dgvCategoria.Columns["Nombre"].AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;

dgvCategoria.Columns["Id\_Categoria"].Visible = false;

dgvCategoria.Columns["Estatus"].Visible = false;

}

private void recargaCatalogoEditorial()

{

DataTable dt = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "SELECT e.Nombre AS Editorial, e.Año\_Inauguracion, e.Id\_Ciudad, c.Nombre AS Ciudad, es.Nombre AS Estado, p.Nombre AS Pais, c.Id\_Estado, es.Id\_Pais, e.Id\_Editorial, e.Estatus FROM Editorial e INNER JOIN Ciudad c ON e.Id\_Ciudad = c.Id\_Ciudad INNER JOIN Estado es ON c.Id\_Estado = es.Id\_Estado INNER JOIN Pais p ON es.Id\_Pais = p.Id\_Pais";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

dgvEditorial.DataSource = dt;

//dgvEditorial.Columns["Editorial"].AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;

dgvEditorial.Columns["Id\_Ciudad"].Visible = false;

dgvEditorial.Columns["Id\_Estado"].Visible = false;

dgvEditorial.Columns["Id\_Pais"].Visible = false;

dgvEditorial.Columns["Id\_Editorial"].Visible = false;

dgvEditorial.Columns["Estatus"].Visible = false;

}

private void recargaCatalogoTipoCliente()

{

DataTable dt = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = " select Id\_Tipo\_Cliente,Nombre,Estatus from Tipo\_Cliente";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

dgCliente.DataSource = dt;

dgCliente.Columns["Nombre"].AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;

dgCliente.Columns["Id\_Tipo\_Cliente"].Visible = false;

dgCliente.Columns["Estatus"].Visible = false;

}

private void recargaHistorial() {

DataTable dt = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select l.Fecha\_Hora,u.Nombre\_Usuario,concat(u.Nombre, ' ', u.Apellido\_Paterno, ' ', u.Apellido\_Materno) Nombre\_Completo,l.Accion, l.Tabla,n.Valor from Logs l left join Usuario u on l.Id\_Usuario = u.Id\_Usuario left join Nivel n on l.Id\_Nivel = n.Id\_Nivel order by l.Fecha\_Hora";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

dgvHistorial.DataSource = dt;

}

private void recargaCatalogoTipoPago()

{

DataTable dt = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select Id\_Tipo\_Pago,Nombre,Estatus from Tipo\_Pago";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

dgvPago.DataSource = dt;

dgvPago.Columns["Nombre"].AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;

dgvPago.Columns["Id\_Tipo\_Pago"].Visible = false;

dgvPago.Columns["Estatus"].Visible = false;

}

private void recargaCatalogoTipoFormato()

{

DataTable dt = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select Id\_Tipo\_Formato,Nombre,Estatus from Tipo\_Formato";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

dgvFormatos.DataSource = dt;

dgvFormatos.Columns["Nombre"].AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;

dgvFormatos.Columns["Id\_Tipo\_Formato"].Visible = false;

dgvFormatos.Columns["Estatus"].Visible = false;

}

private void recargaCatalogoAutor()

{

DataTable dt = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select a.Id\_Autor,a.Nombre,a.Apellido,CONCAT(a.Nombre,' ',a.Apellido) Nombre\_Completo,a.Fecha\_Nacimiento,a.Id\_Pais,p.Nacionalidad,a.Estatus from Autor a left join Pais p on a.Id\_Pais=p.Id\_Pais";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

dgvAutor.DataSource = dt;

//dgvAutor.Columns["Nombre"].AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;

dgvAutor.Columns["Id\_Autor"].Visible = false;

dgvAutor.Columns["Nombre"].Visible = false;

dgvAutor.Columns["Apellido"].Visible = false;

dgvAutor.Columns["Id\_Pais"].Visible = false;

dgvAutor.Columns["Estatus"].Visible = false;

}

private void recargaCatalogoLibro()

{

DataTable dt = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select l.Id\_Libro\_ISBN,l.Titulo,l.Id\_Categoria,c.Nombre Categoria,isnull(l.Volumen,'') Volumen,l.Id\_Tipo\_Formato,f.Nombre Formato,l.Id\_Editorial, e.Nombre Editorial,isnull(l.Fecha\_Publicacion,'') [Fecha de publicacion],isnull(l.Paginas,'') Paginas,l.Precio,l.Id\_Rating,r.Valor Rating,l.Estatus from Libro l left join Categoria c on l.Id\_Categoria=c.Id\_Categoria left join Tipo\_Formato f on l.Id\_Tipo\_Formato=f.Id\_Tipo\_Formato left join Editorial e on l.Id\_Editorial= e.Id\_Editorial left join Rating r on l.Id\_Rating=r.Id\_Rating\r\n";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

dgvLibros.DataSource = dt;

dgvLibros.Columns["Id\_Libro\_ISBN"].Visible = false;

dgvLibros.Columns["Id\_Categoria"].Visible = false;

dgvLibros.Columns["Id\_Tipo\_Formato"].Visible = false;

dgvLibros.Columns["Id\_Editorial"].Visible = false;

dgvLibros.Columns["Id\_Rating"].Visible = false;

dgvLibros.Columns["Estatus"].Visible = false;

}

private void recargaCatalogoDepartamento()

{

DataTable dt = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select Id\_Departamento,Departamento,Estatus from Departamento";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

dgvDepartamentos.DataSource = dt;

dgvDepartamentos.Columns["Departamento"].AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;

dgvDepartamentos.Columns["Id\_Departamento"].Visible = false;

dgvDepartamentos.Columns["Estatus"].Visible = false;

}

private void recargaCatalogoUsuario()

{

DataTable dt = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select u.Id\_Usuario,u.Nombre\_Usuario,u.Nombre,u.Contraseña,u.Apellido\_Paterno,u.Apellido\_Materno,u.Email,u.Telefono,u.Id\_Puesto,p.Nombre Puesto,u.Id\_Departamento,d.Departamento,u.Id\_Tipo\_Usuario,t.Nombre [Tipo de Usuario],u.Estatus from Usuario u left join Puesto p on u.Id\_Puesto=p.Id\_Puesto left join Departamento d on u.Id\_Departamento=d.Id\_Departamento left join Tipo\_Usuario t on u.Id\_Tipo\_Usuario=t.Id\_Tipo\_Usuario\r\n";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

dgvUsuarios.DataSource = dt;

dgvUsuarios.Columns["Id\_Usuario"].Visible = false;

dgvUsuarios.Columns["Contraseña"].Visible = false;

dgvUsuarios.Columns["Id\_Puesto"].Visible = false;

dgvUsuarios.Columns["Id\_Departamento"].Visible = false;

dgvUsuarios.Columns["Id\_Tipo\_Usuario"].Visible = false;

dgvUsuarios.Columns["Estatus"].Visible = false;

}

private void recargaCatalogoPuestos()

{

DataTable dt = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select Id\_Puesto,Nombre,Estatus from Puesto";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

dgvPuestos.DataSource = dt;

dgvPuestos.Columns["Nombre"].AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;

dgvPuestos.Columns["Id\_Puesto"].Visible = false;

dgvPuestos.Columns["Estatus"].Visible = false;

}

private void cargacombosUsuarios()

{

DataTable t = new DataTable();

DataTable de = new DataTable();

DataTable p = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select Id\_Tipo\_Usuario, Nombre from Tipo\_Usuario";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(t);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

cbTipoUsuarioUAM.DataSource = t;

cbTipoUsuarioUAM.DisplayMember = "Nombre";

cbTipoUsuarioUAM.ValueMember = "Id\_Tipo\_Usuario";

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select Id\_Departamento,Departamento from Departamento where Estatus=1";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(de);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

cbDepartamentoUAM.DataSource = de;

cbDepartamentoUAM.DisplayMember = "Departamento";

cbDepartamentoUAM.ValueMember = "Id\_Departamento";

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select Id\_Puesto,Nombre from Puesto where Estatus=1";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(p);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

cbPuestoUAM.DataSource = p;

cbPuestoUAM.DisplayMember = "Nombre";

cbPuestoUAM.ValueMember = "Id\_Puesto";

}

private void cargaCombosVentaLibros()

{

DataTable l = new DataTable();

DataTable c = new DataTable();

DataTable p = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select l.Id\_Libro\_ISBN,CONCAT(l.Titulo,', ',f.Nombre) Nombre from Libro l left join Tipo\_Formato f on l.Id\_Tipo\_Formato=f.Id\_Tipo\_Formato where l.Estatus=1";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(l);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

cbLibrosCaja.DataSource = l;

cbLibrosCaja.DisplayMember = "Nombre";

cbLibrosCaja.ValueMember = "Id\_Libro\_ISBN";

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select Id\_Tipo\_Cliente,Nombre from Tipo\_Cliente where Estatus=1\r\n";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(c);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

cbClienteCaja.DataSource = c;

cbClienteCaja.DisplayMember = "Nombre";

cbClienteCaja.ValueMember = "Id\_Tipo\_Cliente";

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select Id\_Tipo\_Pago,Nombre from Tipo\_Pago where Estatus=1";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(p);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

cbPagoCaja.DataSource = p;

cbPagoCaja.DisplayMember = "Nombre";

cbPagoCaja.ValueMember = "Id\_Tipo\_Pago";

}

private void cargacombosLibros() {

DataTable cat = new DataTable();

DataTable Ed = new DataTable();

DataTable ra = new DataTable();

DataTable fo = new DataTable();

DataTable au = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select Id\_Categoria,Nombre from Categoria where Estatus=1";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(cat);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

cbCategoriaLibro.DataSource = cat;

cbCategoriaLibro.DisplayMember = "Nombre";

cbCategoriaLibro.ValueMember = "Id\_Categoria";

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select Id\_Editorial,Nombre from Editorial where Estatus=1";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(Ed);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

cbEditorialLibro.DataSource = Ed;

cbEditorialLibro.DisplayMember = "Nombre";

cbEditorialLibro.ValueMember = "Id\_Editorial";

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select Id\_Rating,Valor from Rating where Estatus=1";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(ra);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

cbRatingLibro.DataSource = ra;

cbRatingLibro.DisplayMember = "Valor";

cbRatingLibro.ValueMember = "Id\_Rating";

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select Id\_Tipo\_Formato,Nombre from Tipo\_Formato where Estatus=1";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(fo);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

cbFormatoLibro.DataSource = fo;

cbFormatoLibro.DisplayMember = "Nombre";

cbFormatoLibro.ValueMember = "Id\_Tipo\_Formato";

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select a.Id\_Autor,CONCAT(a.Nombre,' ',a.Apellido) Nombre\_Completo from Autor a where a.Estatus=1\r\n";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(au);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

cbAutoresLibro.DataSource = au;

cbAutoresLibro.DisplayMember = "Nombre\_Completo";

cbAutoresLibro.ValueMember = "Id\_Autor";

}

private void CategoriasYEditorialesToolStripMenuItem\_Click(object sender, EventArgs e)

{

cerrarPaneles();

PanelCateEditor.Visible = true;

PanelCateEditor.Enabled= true;

recargaCatalogoCategoria();

recargaCatalogoEditorial();

}

private void btnElimCate\_Click(object sender, EventArgs e)

{

if (dgvCategoria.SelectedRows.Count == 1)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("DeleteCategoria", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Id", dgvCategoria.SelectedCells[0].Value));

cmd.ExecuteReader();

LogsAceptable("Eliminar Categoria", "Categoria");

sql.Close();

MessageBox.Show("Eliminada");

recargaCatalogoCategoria();

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else { MessageBox.Show("Seleccione una categoria."); }

}

private void btnUpdEdit\_Click(object sender, EventArgs e)

{

if (dgvEditorial.SelectedRows.Count == 1)

{

DataTable dt = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "SELECT c.Id\_Ciudad, concat(c.Nombre,', ',es.Nombre,', ',p.Nombre) as localizacion FROM Ciudad c INNER JOIN Estado es ON c.Id\_Estado = es.Id\_Estado INNER JOIN Pais p ON es.Id\_Pais = p.Id\_Pais";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

string año = "2023";

if (!string.IsNullOrEmpty((string)dgvEditorial.SelectedCells[1].Value)) {

año = (string)dgvEditorial.SelectedCells[1].Value;

}

cbCiudadEditorial.DataSource = dt;

cbCiudadEditorial.DisplayMember = "localizacion";

cbCiudadEditorial.ValueMember = "Id\_Ciudad";

cbCiudadEditorial.SelectedValue= (int)dgvEditorial.SelectedCells[2].Value;

modEditorial = (string)dgvEditorial.SelectedCells[8].Value;

txtNombreEditorial.Text = (string)dgvEditorial.SelectedCells[0].Value;

dtañoinaguracionEditorial.Value=new DateTime(Convert.ToInt32(año),1,1);

if ((bool)dgvEditorial.SelectedCells[9].Value)

{

chkEditorial.Checked = true;

}

else

{

chkEditorial.Checked = false;

}

gbEditorial.Visible = true;

gbEditorial.Enabled = true;

gbEditorial.BringToFront();

}

else { MessageBox.Show("Seleccione una Editorial."); }

}

private string SiglaEditExis(string nombre) {

string siglas = "";

for (int i=0; i<nombre.Length-1;i++) {

if (i == 0) {

siglas = siglas + nombre.Substring(i, 1);

}

if (nombre.Substring(i,1).Equals(" ")){

siglas = siglas + nombre.Substring(i+1, 1);

}

}

if (siglas.Length > 3){

siglas = siglas.Substring(0, 3);

}

return siglas.ToUpper();

}

private void btnElimEdit\_Click(object sender, EventArgs e)

{

if (dgvEditorial.SelectedRows.Count == 1)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("DeleteEditorial", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Id", dgvEditorial.SelectedCells[8].Value));

cmd.ExecuteReader();

LogsAceptable("Eliminar Editorial", "Editorial");

sql.Close();

MessageBox.Show("Eliminada");

recargaCatalogoEditorial();

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else { MessageBox.Show("Seleccione una Editorial."); }

}

private void btnAgregaEdit\_Click(object sender, EventArgs e)

{

gbEditorial.Visible = true;

gbEditorial.Enabled = true;

gbEditorial.BringToFront();

txtNombreEditorial.Text = "";

dtañoinaguracionEditorial.Value = DateTime.Now;

DataTable dt = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "SELECT c.Id\_Ciudad, concat(c.Nombre,', ',es.Nombre,', ',p.Nombre) as localizacion FROM Ciudad c INNER JOIN Estado es ON c.Id\_Estado = es.Id\_Estado INNER JOIN Pais p ON es.Id\_Pais = p.Id\_Pais";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

cbCiudadEditorial.DataSource = dt;

cbCiudadEditorial.DisplayMember = "localizacion";

cbCiudadEditorial.ValueMember = "Id\_Ciudad";

chkEditorial.Checked = true;

modEditorial = "0";

}

private void categ0ToolStripButton\_Click(object sender, EventArgs e)

{

}

private void btnGuarModCat\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(txtUpCreNombreCat.Text) || string.IsNullOrWhiteSpace(txtUpCreNombreCat.Text)) {

MessageBox.Show("Ingrese los datos solicitados");

}

else

{

string a = "0";

if (ckbCat.Checked)

{

a = "1";

}

if (modCategoria == 0)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("CreateCategoria", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Nombre", txtUpCreNombreCat.Text));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Creacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Creacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Modificacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Modificacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Estatus", a));

cmd.ExecuteReader();

LogsAceptable("Insertar Categoria", "Categoria");

sql.Close();

MessageBox.Show("Guardado");

gbCategoria.Visible = false;

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

string consulta = "Update Categoria set Nombre='" + txtUpCreNombreCat.Text + "',Estatus=" + a + ",Id\_Usuario\_Modificacion=" + IDuser.ToString() +

",Fecha\_Hora\_Modificacion= '" + DateTime.Now.ToString("yyyy/MM/dd HH:mm:ss.fff") + "' where id\_Categoria=" + modCategoria.ToString();

cmd = new SqlCommand(consulta, sql);

cmd.ExecuteReader();

LogsAceptable("Modificar categorias", "Categoria");

sql.Close();

MessageBox.Show("Guardado");

gbCategoria.Visible = false;

}

}

catch (Exception ex) { MessageBox.Show("Error en conexion"); }

}

recargaCatalogoCategoria();

}

}

private void btnModInsEditorial\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(txtNombreEditorial.Text) || string.IsNullOrWhiteSpace(txtNombreEditorial.Text)||string.IsNullOrEmpty(cbCiudadEditorial.Text)||string.IsNullOrWhiteSpace(cbCiudadEditorial.Text) || string.IsNullOrEmpty(dtañoinaguracionEditorial.Text) || string.IsNullOrWhiteSpace(dtañoinaguracionEditorial.Text))

{

MessageBox.Show("Ingrese los datos solicitados");

}

else

{

string a = "0";

if (chkEditorial.Checked)

{

a = "1";

}

if (modEditorial.Equals("0"))

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

//

sql.Open();

cmd = new SqlCommand("CreateEditorial", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Id\_Editorial", SiglaEditExis(txtNombreEditorial.Text)));

cmd.Parameters.Add(new SqlParameter("@Nombre", txtNombreEditorial.Text));

cmd.Parameters.Add(new SqlParameter("@Id\_Ciudad", cbCiudadEditorial.SelectedValue));

cmd.Parameters.Add(new SqlParameter("@Año\_Inaguracion", dtañoinaguracionEditorial.Text));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Creacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Creacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Modificacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Modificacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Estatus", a));

cmd.ExecuteReader();

LogsAceptable("Insertar Editorial", "Editorial");

sql.Close();

MessageBox.Show("Guardado");

gbEditorial.Visible = false;

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

//ciudad año inaugura

string consulta = "Update Editorial set Nombre='" + txtNombreEditorial.Text + "',Estatus=" + a + ",Id\_Usuario\_Modificacion=" + IDuser.ToString() + ",Año\_Inauguracion='" +dtañoinaguracionEditorial.Text+ "', Id\_Ciudad=" + cbCiudadEditorial.SelectedValue.ToString() + ",Fecha\_Hora\_Modificacion= '" + DateTime.Now.ToString("yyyy/MM/dd HH:mm:ss.fff") + "' where Id\_Editorial='" + modEditorial+"'";

cmd = new SqlCommand(consulta, sql);

cmd.ExecuteReader();

LogsAceptable("Modificar Editorial", "Editorial");

sql.Close();

MessageBox.Show("Guardado");

gbEditorial.Visible = false;

}

}

catch (Exception ex) { MessageBox.Show("Error en conexion"); }

}

recargaCatalogoEditorial();

}

}

private void btnAgregaCliente\_Click(object sender, EventArgs e)

{

gbClientes.Visible = true;

gbClientes.Enabled = true;

gbClientes.BringToFront();

txtNombreCliente.Text = "";

chkTiCliente.Checked = true;

modCliente = 0;

}

private void btnModCliente\_Click(object sender, EventArgs e)

{

if (dgCliente.SelectedRows.Count == 1)

{

txtNombreCliente.Text = (string)dgCliente.SelectedCells[1].Value;

if ((bool)dgCliente.SelectedCells[2].Value)

{

chkTiCliente.Checked = true;

}

else

{

chkTiCliente.Checked = false;

}

modCliente = (int)dgCliente.SelectedCells[0].Value;

gbClientes.Visible = true;

gbClientes.Enabled = true;

gbClientes.BringToFront();

}

else { MessageBox.Show("Seleccione un Tipo de Cliente."); }

}

private void btnEliminaCliente\_Click(object sender, EventArgs e)

{

if (dgCliente.SelectedRows.Count == 1)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("DeleteTipo\_Cliente", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Id", dgCliente.SelectedCells[0].Value));

cmd.ExecuteReader();

LogsAceptable("Eliminar Tipo Cliente", "Tipo\_Cliente");

sql.Close();

MessageBox.Show("Eliminada");

recargaCatalogoTipoCliente();

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else { MessageBox.Show("Seleccione un Tipo de Cliente."); }

}

private void btnAgregaPago\_Click(object sender, EventArgs e)

{

gpPago.Visible = true;

gpPago.Enabled = true;

gpPago.BringToFront();

txtNombrePago.Text = "";

chkPago.Checked = true;

modPago = 0;

}

private void btnModificaPago\_Click(object sender, EventArgs e)

{

if (dgvPago.SelectedRows.Count == 1)

{

txtNombrePago.Text = (string)dgvPago.SelectedCells[1].Value;

if ((bool)dgvPago.SelectedCells[2].Value)

{

chkPago.Checked = true;

}

else

{

chkPago.Checked = false;

}

modPago = (int)dgvPago.SelectedCells[0].Value;

gpPago.Visible = true;

gpPago.Enabled = true;

gpPago.BringToFront();

}

else { MessageBox.Show("Seleccione una Forma de Pago."); }

}

private void btnEliminaPago\_Click(object sender, EventArgs e)

{

if (dgvPago.SelectedRows.Count == 1)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("DeleteTipo\_Pago", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Id", dgvPago.SelectedCells[0].Value));

cmd.ExecuteReader();

LogsAceptable("Eliminar Tipo de Pago", "Tipo\_Pago");

sql.Close();

MessageBox.Show("Eliminada");

recargaCatalogoTipoPago();

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else { MessageBox.Show("Seleccione un Tipo de Pago."); }

}

private void TiposDeClientesYTiposDePagosToolStripMenuItem\_Click(object sender, EventArgs e)

{

cerrarPaneles();

pTiClienteTiPago.Visible = true;

pTiClienteTiPago.Enabled = true;

recargaCatalogoTipoPago();

recargaCatalogoTipoCliente();

}

private void btnAgregaModificaPago\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(txtNombrePago.Text) || string.IsNullOrWhiteSpace(txtNombrePago.Text))

{

MessageBox.Show("Ingrese los datos solicitados");

}

else

{

string a = "0";

if (chkPago.Checked)

{

a = "1";

}

if (modPago == 0)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("CreateTipoPago", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Nombre", txtNombrePago.Text));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Creacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Creacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Modificacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Modificacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Estatus", a));

cmd.ExecuteReader();

LogsAceptable("Insertar Forma de Pago", "Tipo\_Pago");

sql.Close();

MessageBox.Show("Guardado");

gbCategoria.Visible = false;

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

string consulta = "Update Tipo\_Pago set Nombre='" + txtNombrePago.Text + "',Estatus=" + a + ",Id\_Usuario\_Modificacion=" + IDuser.ToString() +

",Fecha\_Hora\_Modificacion= '" + DateTime.Now.ToString("yyyy/MM/dd HH:mm:ss.fff") + "' where Id\_Tipo\_Pago=" + modPago.ToString();

cmd = new SqlCommand(consulta, sql);

cmd.ExecuteReader();

LogsAceptable("Modificar Forma de Pago", "Tipo\_Pago");

sql.Close();

MessageBox.Show("Guardado");

gbCategoria.Visible = false;

}

}

catch (Exception ex) { MessageBox.Show("Error en conexion"); }

}

recargaCatalogoTipoPago();

}

}

private void btnAgregaModificaTiCliente\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(txtNombreCliente.Text) || string.IsNullOrWhiteSpace(txtNombreCliente.Text))

{

MessageBox.Show("Ingrese los datos solicitados");

}

else

{

string a = "0";

if (chkTiCliente.Checked)

{

a = "1";

}

if (modCliente == 0)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("CreateTipoCliente", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Nombre", txtNombreCliente.Text));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Creacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Creacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Modificacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Modificacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Estatus", a));

cmd.ExecuteReader();

LogsAceptable("Insertar Tipo de Cliente", "Tipo\_Cliente");

sql.Close();

MessageBox.Show("Guardado");

gbCategoria.Visible = false;

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

string consulta = "Update Tipo\_Cliente set Nombre='" + txtNombreCliente.Text + "',Estatus=" + a + ",Id\_Usuario\_Modificacion=" + IDuser.ToString() +

",Fecha\_Hora\_Modificacion= '" + DateTime.Now.ToString("yyyy/MM/dd HH:mm:ss.fff") + "' where id\_Tipo\_Cliente=" + modCliente.ToString();

cmd = new SqlCommand(consulta, sql);

cmd.ExecuteReader();

LogsAceptable("Modificar Tipo de Cliente", "Tipo\_Cliente");

sql.Close();

MessageBox.Show("Guardado");

gbCategoria.Visible = false;

}

}

catch (Exception ex) { MessageBox.Show("Error en conexion"); }

}

recargaCatalogoTipoCliente();

}

}

private void LibrosToolStripMenuItem\_Click(object sender, EventArgs e)

{

cerrarPaneles();

panelLibrosFormatos.Visible = true;

panelLibrosFormatos.Enabled = true;

recargaCatalogoTipoFormato();

recargaCatalogoLibro();

}

private void DepartamentosYPuestosToolStripMenuItem\_Click(object sender, EventArgs e)

{

cerrarPaneles();

PanelDepartamentosPuestos.Visible = true;

PanelDepartamentosPuestos.Enabled = true;

recargaCatalogoPuestos();

recargaCatalogoDepartamento();

}

private void btnAgregaDepartamento\_Click(object sender, EventArgs e)

{

gbDepartamento.Visible = true;

gbDepartamento.Enabled = true;

gbDepartamento.BringToFront();

txtNombreDepartamento.Text = "";

chkDepartamento.Checked = true;

modDepartamento = 0;

}

private void btnAgregaPuestos\_Click(object sender, EventArgs e)

{

gbPuestos.Visible = true;

gbPuestos.Enabled = true;

gbPuestos.BringToFront();

txtNombrePuestos.Text = "";

chkPuesto.Checked = true;

modPuesto = 0;

}

private void btnAgregaModificaDepartamento\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(txtNombreDepartamento.Text) || string.IsNullOrWhiteSpace(txtNombreDepartamento.Text))

{

MessageBox.Show("Ingrese los datos solicitados");

}

else

{

string a = "0";

if (chkDepartamento.Checked)

{

a = "1";

}

if (modDepartamento == 0)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("CreateDepartamento", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Departamento", txtNombreDepartamento.Text));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Creacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Creacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Modificacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Modificacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Estatus", a));

cmd.ExecuteReader();

LogsAceptable("Insertar Departamento", "Departamento");

sql.Close();

MessageBox.Show("Guardado");

gbDepartamento.Visible = false;

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

string consulta = "Update Departamento set Departamento='" + txtNombreDepartamento.Text + "',Estatus=" + a + ",Id\_Usuario\_Modificacion=" + IDuser.ToString() +

",Fecha\_Hora\_Modificacion= '" + DateTime.Now.ToString("yyyy/MM/dd HH:mm:ss.fff") + "' where Id\_Departamento=" + modDepartamento.ToString();

cmd = new SqlCommand(consulta, sql);

cmd.ExecuteReader();

LogsAceptable("Modificar Departamento", "Departamento");

sql.Close();

MessageBox.Show("Guardado");

gbDepartamento.Visible = false;

}

}

catch (Exception ex) { MessageBox.Show("Error en conexion"); }

}

recargaCatalogoDepartamento();

}

}

private void btnAgregaModificaPuesto\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(txtNombrePuestos.Text) || string.IsNullOrWhiteSpace(txtNombrePuestos.Text))

{

MessageBox.Show("Ingrese los datos solicitados");

}

else

{

string a = "0";

if (chkPuesto.Checked)

{

a = "1";

}

if (modPuesto == 0)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("CreatePuesto", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Nombre", txtNombrePuestos.Text));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Creacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Creacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Modificacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Modificacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Estatus", a));

cmd.ExecuteReader();

LogsAceptable("Insertar Puesto", "Puesto");

sql.Close();

MessageBox.Show("Guardado");

gbPuestos.Visible = false;

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

string consulta = "Update Puesto set Nombre='" + txtNombrePuestos.Text + "',Estatus=" + a + ",Id\_Usuario\_Modificacion=" + IDuser.ToString() +

",Fecha\_Hora\_Modificacion= '" + DateTime.Now.ToString("yyyy/MM/dd HH:mm:ss.fff") + "' where Id\_Puesto=" + modPuesto.ToString();

cmd = new SqlCommand(consulta, sql);

cmd.ExecuteReader();

LogsAceptable("Modificar Puesto", "Puesto");

sql.Close();

MessageBox.Show("Guardado");

gbPuestos.Visible = false;

}

}

catch (Exception ex) { MessageBox.Show("Error en conexion"); }

}

recargaCatalogoPuestos();

}

}

private void btnModificaDepartamento\_Click(object sender, EventArgs e)

{

if (dgvDepartamentos.SelectedRows.Count == 1)

{

txtNombreDepartamento.Text = (string)dgvDepartamentos.SelectedCells[1].Value;

if ((bool)dgvDepartamentos.SelectedCells[2].Value)

{

chkDepartamento.Checked = true;

}

else

{

chkDepartamento.Checked = false;

}

modDepartamento = (int)dgvDepartamentos.SelectedCells[0].Value;

gbDepartamento.Visible = true;

gbDepartamento.Enabled = true;

gbDepartamento.BringToFront();

}

else { MessageBox.Show("Seleccione un Departamento."); }

}

private void btnModificaPuestos\_Click(object sender, EventArgs e)

{

if (dgvPuestos.SelectedRows.Count == 1)

{

txtNombrePuestos.Text = (string)dgvPuestos.SelectedCells[1].Value;

if ((bool)dgvPuestos.SelectedCells[2].Value)

{

chkPuesto.Checked = true;

}

else

{

chkPuesto.Checked = false;

}

modPuesto = (int)dgvPuestos.SelectedCells[0].Value;

gbPuestos.Visible = true;

gbPuestos.Enabled = true;

gbPuestos.BringToFront();

}

else { MessageBox.Show("Seleccione un Puesto."); }

}

private void btnEliminaDepartamento\_Click(object sender, EventArgs e)

{

if (dgvDepartamentos.SelectedRows.Count == 1)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("DeleteDepartamento", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Id", dgvDepartamentos.SelectedCells[0].Value));

cmd.ExecuteReader();

LogsAceptable("Eliminar Departamento", "Departamento");

sql.Close();

MessageBox.Show("Eliminada");

recargaCatalogoDepartamento();

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else { MessageBox.Show("Seleccione un Departamento."); }

}

private void btnEliminaPuestos\_Click(object sender, EventArgs e)

{

if (dgvPuestos.SelectedRows.Count == 1)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("DeletePuesto", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Id", dgvPuestos.SelectedCells[0].Value));

cmd.ExecuteReader();

LogsAceptable("Eliminar Puesto", "Puesto");

sql.Close();

MessageBox.Show("Eliminada");

recargaCatalogoPuestos();

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else { MessageBox.Show("Seleccione un Puesto."); }

}

private void btnAgregaFormatos\_Click(object sender, EventArgs e)

{

gbFormato.Visible = true;

gbFormato.Enabled = true;

gbFormato.BringToFront();

txtNombreFormato.Text = "";

chkFormato.Checked = true;

modFormato = 0;

}

private void btnAgregarLibro\_Click(object sender, EventArgs e)

{

dgvAutoresLibro.Rows.Clear();

panelLibros.Visible = true;

panelLibros.Enabled = true;

panelLibros.BringToFront();

txtISBN.Text = "";

txtISBN.Enabled =true;

txtTitulo.Text = "";

cargacombosLibros();

chkLibro.Checked = true;

modLibro = "0";

txtVolumenLibro.Text = "";

dtFechaLibro.Value = DateTime.Now;

txtPaginasLibro.Text = "";

txtPrecioLibro.Text = "";

dgvAutoresLibro.DataSource = null;

}

private void btnModificaFormato\_Click(object sender, EventArgs e)

{

if (dgvFormatos.SelectedRows.Count == 1)

{

txtNombreFormato.Text = (string)dgvFormatos.SelectedCells[1].Value;

if ((bool)dgvFormatos.SelectedCells[2].Value)

{

chkFormato.Checked = true;

}

else

{

chkFormato.Checked = false;

}

modFormato = (int)dgvFormatos.SelectedCells[0].Value;

gbFormato.Visible = true;

gbFormato.Enabled = true;

gbFormato.BringToFront();

}

else { MessageBox.Show("Seleccione un Formato."); }

}

private void cargaLibrosAutoresTabla(string isbn) {

dgvAutoresLibro.Rows.Clear();

DataTable dt = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select l.Id\_Autor,CONCAT(a.Nombre,' ',a.Apellido) Nombre\_Completo from Libro\_Autores l left join Autor a on l.Id\_Autor=a.Id\_Autor where l.Id\_Libro\_ISBN='"+isbn+"'";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

for (int i = 0; i < dt.Rows.Count; i++)

{

dgvAutoresLibro.Rows.Add(dt.Rows[i]["Id\_Autor"].ToString(), dt.Rows[i]["Nombre\_Completo"].ToString());

}

}

private void btnModificaLibro\_Click(object sender, EventArgs e)

{

if (dgvLibros.SelectedRows.Count == 1)

{

cargacombosLibros();

txtISBN.Text = (string)dgvLibros.SelectedCells[0].Value;

txtISBN.Enabled =false;

cargaLibrosAutoresTabla(txtISBN.Text);

txtTitulo.Text= (string)dgvLibros.SelectedCells[1].Value;

cbCategoriaLibro.SelectedValue= (int)dgvLibros.SelectedCells[2].Value;

txtVolumenLibro.Text= (string)dgvLibros.SelectedCells[4].Value;

cbFormatoLibro.SelectedValue = (int)dgvLibros.SelectedCells[5].Value;

cbEditorialLibro.SelectedValue = (string)dgvLibros.SelectedCells[7].Value;

dtFechaLibro.Value=(DateTime)dgvLibros.SelectedCells[9].Value;

txtPaginasLibro.Text= ((int)dgvLibros.SelectedCells[10].Value).ToString();

txtPrecioLibro.Text = ((decimal)dgvLibros.SelectedCells[11].Value).ToString();

cbRatingLibro.SelectedValue = (int)dgvLibros.SelectedCells[12].Value;

if ((bool)dgvLibros.SelectedCells[14].Value)

{

chkLibro.Checked = true;

}

else

{

chkLibro.Checked = false;

}

modLibro = (string)dgvLibros.SelectedCells[0].Value;

panelLibros.Visible = true;

panelLibros.Enabled = true;

panelLibros.BringToFront();

}

else { MessageBox.Show("Seleccione un Libro."); }

}

private void btnEliminaFormato\_Click(object sender, EventArgs e)

{

if (dgvFormatos.SelectedRows.Count == 1)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("DeleteTipo\_Formato", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Id", dgvFormatos.SelectedCells[0].Value));

cmd.ExecuteReader();

LogsAceptable("Eliminar Formato", "Tipo\_Formato");

sql.Close();

MessageBox.Show("Eliminada");

recargaCatalogoTipoFormato();

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else { MessageBox.Show("Seleccione un Formato."); }

}

private void btnEliminaLibro\_Click(object sender, EventArgs e)

{

if (dgvLibros.SelectedRows.Count == 1)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("DeleteLibro", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Id", dgvLibros.SelectedCells[0].Value));

cmd.ExecuteReader();

LogsAceptable("Eliminar Libro", "Libro");

sql.Close();

MessageBox.Show("Eliminada");

recargaCatalogoLibro();

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else { MessageBox.Show("Seleccione un Libro."); }

}

private void btnAgregarModificarFormato\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(txtNombreFormato.Text) || string.IsNullOrWhiteSpace(txtNombreFormato.Text))

{

MessageBox.Show("Ingrese los datos solicitados");

}

else

{

string a = "0";

if (chkFormato.Checked)

{

a = "1";

}

if (modFormato == 0)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("CreateTipoFormato", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Nombre", txtNombreFormato.Text));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Creacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Creacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Modificacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Modificacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Estatus", a));

cmd.ExecuteReader();

LogsAceptable("Insertar Formato", "Tipo\_Formato");

sql.Close();

MessageBox.Show("Guardado");

gbFormato.Visible = false;

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

string consulta = "Update Tipo\_Formato set Nombre='" + txtNombreFormato.Text + "',Estatus=" + a + ",Id\_Usuario\_Modificacion=" + IDuser.ToString() +

",Fecha\_Hora\_Modificacion= '" + DateTime.Now.ToString("yyyy/MM/dd HH:mm:ss.fff") + "' where Id\_Tipo\_Formato=" + modFormato.ToString();

cmd = new SqlCommand(consulta, sql);

cmd.ExecuteReader();

LogsAceptable("Modificar Formato", "Tipo\_Formato");

sql.Close();

MessageBox.Show("Guardado");

gbFormato.Visible = false;

}

}

catch (Exception ex) { MessageBox.Show("Error en conexion"); }

}

recargaCatalogoTipoFormato();

}

}

private void btnAgregarModificarLibro\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(txtISBN.Text) || string.IsNullOrWhiteSpace(txtISBN.Text)

|| string.IsNullOrEmpty(txtTitulo.Text) || string.IsNullOrWhiteSpace(txtTitulo.Text) ||

string.IsNullOrEmpty(txtPrecioLibro.Text) || string.IsNullOrWhiteSpace(txtPrecioLibro.Text) ||

string.IsNullOrEmpty(cbCategoriaLibro.Text) || string.IsNullOrWhiteSpace(cbCategoriaLibro.Text) ||

string.IsNullOrEmpty(cbFormatoLibro.Text) || string.IsNullOrWhiteSpace(cbFormatoLibro.Text) ||

string.IsNullOrEmpty(cbEditorialLibro.Text) || string.IsNullOrWhiteSpace(cbEditorialLibro.Text) ||

string.IsNullOrEmpty(cbRatingLibro.Text) || string.IsNullOrWhiteSpace(cbRatingLibro.Text))

{

MessageBox.Show("Ingrese los datos solicitados");

}

else

{

string vol = null;

string f = null;

string pag = null;

if (!(string.IsNullOrEmpty(txtVolumenLibro.Text) || string.IsNullOrWhiteSpace(txtVolumenLibro.Text))) {

vol = txtVolumenLibro.Text;

}

if (!(dtFechaLibro.Text.Equals(""))) {

f = dtFechaLibro.Value.ToShortDateString();

}

if (!(txtPaginasLibro.Text.Equals(""))) {

pag = txtPaginasLibro.Text;

}

string a = "0";

if (chkLibro.Checked)

{

a = "1";

}

if (modLibro.Equals("0"))

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("CreateLibro", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Id\_Libro\_ISBN", txtISBN.Text));

cmd.Parameters.Add(new SqlParameter("@Titulo", txtTitulo.Text));

cmd.Parameters.Add(new SqlParameter("@Id\_Categoria", cbCategoriaLibro.SelectedValue));

cmd.Parameters.Add(new SqlParameter("@Volumen",vol ));

cmd.Parameters.Add(new SqlParameter("@Id\_Tipo\_Formato", cbFormatoLibro.SelectedValue));

cmd.Parameters.Add(new SqlParameter("@Id\_Editorial", cbEditorialLibro.SelectedValue));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Publicacion",f));

cmd.Parameters.Add(new SqlParameter("@Paginas",pag));

cmd.Parameters.Add(new SqlParameter("@Precio",txtPrecioLibro.Text));

cmd.Parameters.Add(new SqlParameter("@Id\_Rating", cbRatingLibro.SelectedValue));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Creacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Creacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Modificacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Modificacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Estatus", a));

cmd.ExecuteReader();

EliminaTodosAutoresLibro(txtISBN.Text);

for (int i = 0; i < dgvAutoresLibro.Rows.Count; i++)

{

if (!(dgvAutoresLibro.Rows[i].Cells["Id\_Autor"].Value.ToString()).Equals(""))

{

agregarAutoresLibro(txtISBN.Text, dgvAutoresLibro.Rows[i].Cells["Id\_Autor"].Value.ToString());

}

}

LogsAceptable("Insertar Libro", "Libro");

sql.Close();

MessageBox.Show("Guardado");

panelLibros.Visible = false;

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

if (!(string.IsNullOrEmpty(txtVolumenLibro.Text) || string.IsNullOrWhiteSpace(txtVolumenLibro.Text)))

{

vol ="'"+txtVolumenLibro.Text+"'";

}

else {

vol = "NULL";

}

if (!(dtFechaLibro.Text.Equals("")))

{

f = "'" + dtFechaLibro.Value.ToShortDateString() + "'";

}

else {

f = "NULL";

}

if (!(txtPaginasLibro.Text.Equals("")))

{

pag = txtPaginasLibro.Text;

}

else {

pag = "NULL";

}

string consulta = "Update Libro set Titulo='" + txtTitulo.Text + "',Estatus=" + a + ",Id\_Usuario\_Modificacion=" + IDuser.ToString() +

",Fecha\_Hora\_Modificacion= '" + DateTime.Now.ToString("yyyy/MM/dd HH:mm:ss.fff") + "',"+

"Id\_Categoria="+ cbCategoriaLibro.SelectedValue.ToString()+",Volumen="+vol+","+

"Paginas="+pag+ ",Id\_Tipo\_Formato="+cbEditorialLibro.SelectedValue.ToString() + ",Id\_Editorial="+cbEditorialLibro.SelectedValue.ToString() +

",Id\_Rating="+cbRatingLibro.SelectedValue.ToString()+ ",Precio="+txtPrecioLibro.Text+ ",Fecha\_Publicacion="+f +

" where Id\_Libro\_ISBN='" + modLibro.ToString()+"'";

cmd = new SqlCommand(consulta, sql);

cmd.ExecuteReader();

LogsAceptable("Modificar Libro", "Libro");

EliminaTodosAutoresLibro(txtISBN.Text);

for (int i = 0; i < dgvAutoresLibro.Rows.Count; i++)

{

if (!(dgvAutoresLibro.Rows[i].Cells["Id\_Autor"].Value.ToString()).Equals(""))

{

agregarAutoresLibro(txtISBN.Text, dgvAutoresLibro.Rows[i].Cells["Id\_Autor"].Value.ToString());

}

}

sql.Close();

MessageBox.Show("Guardado");

panelLibros.Visible = false;

}

}

catch (Exception ex) { MessageBox.Show("Error en conexion"); }

}

recargaCatalogoLibro();

}

}

private void AutoresToolStripMenuItem\_Click(object sender, EventArgs e)

{

cerrarPaneles();

PanelAutor.Visible = true;

PanelAutor.Enabled = true;

recargaCatalogoAutor();

}

private void btnagregaAutorLibro\_Click(object sender, EventArgs e)

{ bool existe = false;

string id=cbAutoresLibro.SelectedValue.ToString();

for (int i=0;i< dgvAutoresLibro.Rows.Count;i++) {

if ((dgvAutoresLibro.Rows[i].Cells["Id\_Autor"].Value.ToString()).Equals(id)) {

existe = true;

}

}

if (!existe){

dgvAutoresLibro.Rows.Add(id, cbAutoresLibro.Text);

}

}

private void agregarAutoresLibro(string isbn, string autor) {

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("CreateLibroAutores", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Id\_Libro\_ISBN", isbn));

cmd.Parameters.Add(new SqlParameter("@Id\_Autor", autor));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Creacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Creacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Modificacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Modificacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Estatus", "1"));

cmd.ExecuteReader();

sql.Close();

}

}

catch (Exception ex)

{

throw;

}

}

private void btnQuitarLibroAutores\_Click(object sender, EventArgs e)

{

if (dgvAutoresLibro.SelectedRows.Count == 1)

{

dgvAutoresLibro.Rows.Remove(dgvAutoresLibro.SelectedRows[0]);

}

else { MessageBox.Show("Seleccione un Autor."); }

}

private void EliminaTodosAutoresLibro(string isbn)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

string consulta = "delete from Libro\_Autores where Id\_Libro\_ISBN='" + isbn + "'";

cmd = new SqlCommand(consulta, sql);

cmd.ExecuteReader();

sql.Close();

}

}

catch (Exception ex) { throw; }

}

private void HistorialToolStripMenuItem\_Click(object sender, EventArgs e)

{

cerrarPaneles();

PanelHistorial.Visible = true;

PanelHistorial.Enabled = true;

recargaHistorial();

}

private void btnAgregaAutor\_Click(object sender, EventArgs e)

{

PanelAutorAM.Visible = true;

PanelAutorAM.Enabled = true;

PanelAutorAM.BringToFront();

txtNombreAutor.Text = "";

txtApellidoAutor.Text = "";

dtNacioAutor.Value = DateTime.Now;

DataTable dt = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select Id\_Pais,Nacionalidad from Pais";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

cbNacionalidadAutor.DataSource = dt;

cbNacionalidadAutor.DisplayMember = "Nacionalidad";

cbNacionalidadAutor.ValueMember = "Id\_Pais";

chkAutor.Checked = true;

modAutor = "0";

}

private void btnModificaAutor\_Click(object sender, EventArgs e)

{

if (dgvAutor.SelectedRows.Count == 1)

{

DataTable dt = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select Id\_Pais,Nacionalidad from Pais";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

cbNacionalidadAutor.DataSource = dt;

cbNacionalidadAutor.DisplayMember = "Nacionalidad";

cbNacionalidadAutor.ValueMember = "Id\_Pais";

cbNacionalidadAutor.SelectedValue=(int)dgvAutor.SelectedCells[5].Value;

if ((bool)dgvAutor.SelectedCells[7].Value)

{

chkAutor.Checked = true;

}

else

{

chkAutor.Checked = false;

}

txtNombreAutor.Text = (string)dgvAutor.SelectedCells[1].Value; ;

txtApellidoAutor.Text = (string)dgvAutor.SelectedCells[2].Value;

dtNacioAutor.Value = (DateTime)dgvAutor.SelectedCells[4].Value;

modAutor = (string)dgvAutor.SelectedCells[0].Value;

PanelAutorAM.Visible = true;

PanelAutorAM.Enabled = true;

PanelAutorAM.BringToFront();

}

else { MessageBox.Show("Seleccione un Autor."); }

}

private void btnEliminaAutor\_Click(object sender, EventArgs e)

{

if (dgvAutor.SelectedRows.Count == 1)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("DeleteAutor", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Id", dgvAutor.SelectedCells[0].Value));

cmd.ExecuteReader();

LogsAceptable("Eliminar Autor", "Autor");

sql.Close();

MessageBox.Show("Eliminada");

recargaCatalogoAutor();

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else { MessageBox.Show("Seleccione un Autor."); }

}

private void btnAgregarModificarAutor\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(txtNombreAutor.Text) || string.IsNullOrWhiteSpace(txtNombreAutor.Text) || string.IsNullOrEmpty(txtApellidoAutor.Text) || string.IsNullOrWhiteSpace(txtApellidoAutor.Text) || string.IsNullOrEmpty(dtNacioAutor.Text) || string.IsNullOrWhiteSpace(dtNacioAutor.Text))

{

MessageBox.Show("Ingrese los datos solicitados");

}

else

{

string a = "0";

if (chkAutor.Checked)

{

a = "1";

}

if (modAutor.Equals("0"))

{

string f = null;

string p = null;

if (!(cbNacionalidadAutor.Text.Equals(""))) {

p = cbNacionalidadAutor.SelectedValue.ToString();

}

if (!(dtNacioAutor.Text.Equals(""))) {

f = dtNacioAutor.Value.ToShortDateString();

}

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("CreateAutor", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Id\_Autor", SiglaAutor(txtNombreAutor.Text,txtApellidoAutor.Text, dtNacioAutor.Value.ToString("ddMMyyyy"))));

cmd.Parameters.Add(new SqlParameter("@Nombre", txtNombreAutor.Text));

cmd.Parameters.Add(new SqlParameter("@Apellido", txtApellidoAutor.Text));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Nacimiento", f));

cmd.Parameters.Add(new SqlParameter("@Id\_Pais",p ));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Creacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Creacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Modificacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Modificacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Estatus", a));

cmd.ExecuteReader();

LogsAceptable("Insertar Autor", "Autor");

sql.Close();

MessageBox.Show("Guardado");

PanelAutorAM.Visible = false;

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else

{

try

{

string f = "NULL";

string p = "NULL";

if (!(cbNacionalidadAutor.Text.Equals("")))

{

p = cbNacionalidadAutor.SelectedValue.ToString();

}

if (!(dtNacioAutor.Text.Equals("")))

{

f ="'"+ dtNacioAutor.Value.ToShortDateString()+ "'";

}

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

//ciudad año inaugura

string consulta = "Update Autor set Nombre='" + txtNombreAutor.Text + "',Estatus=" + a + ",Id\_Usuario\_Modificacion=" + IDuser.ToString() + ",Apellido='" + txtApellidoAutor.Text + "',Fecha\_Nacimiento="+f+", Id\_Pais=" +p + ",Fecha\_Hora\_Modificacion= '" + DateTime.Now.ToString("yyyy/MM/dd HH:mm:ss.fff") + "' where Id\_Autor='" + modAutor + "'";

cmd = new SqlCommand(consulta, sql);

cmd.ExecuteReader();

LogsAceptable("Modificar Autor", "Autor");

sql.Close();

MessageBox.Show("Guardado");

PanelAutorAM.Visible = false;

}

}

catch (Exception ex) { MessageBox.Show("Error en conexion"); }

}

recargaCatalogoAutor();

}

}

private string SiglaAutor(string nombre,string apellido, string fecha\_nacimiento) {

string resul="";

resul = nombre.Substring(0, 1) + apellido.Substring(0,1) + fecha\_nacimiento;

return resul.ToUpper();

}

private void UsuariosToolStripMenuItem1\_Click(object sender, EventArgs e)

{

cerrarPaneles();

PanelUsuarios.Visible = true;

PanelUsuarios.Enabled = true;

recargaCatalogoUsuario();

}

private void ventasToolStripMenuItem\_Click(object sender, EventArgs e)

{

cerrarPaneles();

PanelVentas.Visible = true;

PanelVentas.Enabled = true;

recargaCatalogoVentas();

}

private void cajaToolStripMenuItem\_Click(object sender, EventArgs e)

{

cerrarPaneles();

panelVentaRegistro.Visible = true;

panelVentaRegistro.Enabled = true;

cargaCombosVentaLibros();

}

private void btnAgregaUsuario\_Click(object sender, EventArgs e)

{

cargacombosUsuarios();

panelUsuarioAM.Visible = true;

panelUsuarioAM.Enabled = true;

panelUsuarioAM.BringToFront();

txtNombreUsuarioUAM.Text = "";

txtNombreUAM.Text = "";

txtContraseñaUAM.Text = "";

txtApellidoPatUAM.Text = "";

txtApellidoMatUAM.Text = "";

txtCorreoUAM.Text = "";

txtTelefonoUAM.Text = "";

chkUsuarioUAM.Checked = true;

modUsuario = 0;

modcon = "";

}

private void btnModificaUsuario\_Click(object sender, EventArgs e)

{

if (dgvUsuarios.SelectedRows.Count == 1)

{

cargacombosUsuarios();

txtNombreUsuarioUAM.Text = (string)dgvUsuarios.SelectedCells[1].Value;

txtNombreUAM.Text = (string)dgvUsuarios.SelectedCells[2].Value;

txtContraseñaUAM.Text = (string)dgvUsuarios.SelectedCells[3].Value;

modcon = txtContraseñaUAM.Text;

txtApellidoPatUAM.Text = (string)dgvUsuarios.SelectedCells[4].Value;

txtApellidoMatUAM.Text = (string)dgvUsuarios.SelectedCells[5].Value;

txtCorreoUAM.Text = (string)dgvUsuarios.SelectedCells[6].Value;

txtTelefonoUAM.Text = (string)dgvUsuarios.SelectedCells[7].Value;

cbPuestoUAM.SelectedValue = (int)dgvUsuarios.SelectedCells[8].Value;

cbDepartamentoUAM.SelectedValue = (int)dgvUsuarios.SelectedCells[10].Value;

cbTipoUsuarioUAM.SelectedValue = (int)dgvUsuarios.SelectedCells[12].Value;

if ((bool)dgvUsuarios.SelectedCells[14].Value)

{

chkUsuarioUAM.Checked = true;

}

else

{

chkUsuarioUAM.Checked = false;

}

modUsuario = (int)dgvUsuarios.SelectedCells[0].Value;

panelUsuarioAM.Visible = true;

panelUsuarioAM.Enabled = true;

panelUsuarioAM.BringToFront();

}

else { MessageBox.Show("Seleccione un Usuario."); }

}

private void btnEliminaUsuario\_Click(object sender, EventArgs e)

{

if (dgvUsuarios.SelectedRows.Count == 1)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("DeleteUsuario", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Id", dgvUsuarios.SelectedCells[0].Value));

cmd.ExecuteReader();

LogsAceptable("Eliminar Usuario", "Usuario");

sql.Close();

MessageBox.Show("Eliminada");

recargaCatalogoUsuario();

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else { MessageBox.Show("Seleccione un Usuario."); }

}

private void btnAgregarModificarUsuario\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(txtNombreUAM.Text) || string.IsNullOrWhiteSpace(txtNombreUAM.Text)

|| string.IsNullOrEmpty(txtNombreUsuarioUAM.Text) || string.IsNullOrWhiteSpace(txtNombreUsuarioUAM.Text)

|| string.IsNullOrEmpty(txtApellidoPatUAM.Text) || string.IsNullOrWhiteSpace(txtApellidoPatUAM.Text)

|| string.IsNullOrEmpty(txtCorreoUAM.Text) || string.IsNullOrWhiteSpace(txtCorreoUAM.Text)

|| string.IsNullOrEmpty(cbPuestoUAM.Text) || string.IsNullOrWhiteSpace(cbPuestoUAM.Text)

|| string.IsNullOrEmpty(cbDepartamentoUAM.Text) || string.IsNullOrWhiteSpace(cbDepartamentoUAM.Text)

|| string.IsNullOrEmpty(cbTipoUsuarioUAM.Text) || string.IsNullOrWhiteSpace(cbTipoUsuarioUAM.Text)

|| ((string.IsNullOrEmpty(txtContraseñaUAM.Text) || string.IsNullOrWhiteSpace(txtContraseñaUAM.Text))

&& (string.IsNullOrEmpty(modcon) || string.IsNullOrWhiteSpace(modcon))))

{

MessageBox.Show("Ingrese los datos solicitados");

}

else

{

if (!(string.IsNullOrEmpty(txtContraseñaUAM.Text) || string.IsNullOrWhiteSpace(txtContraseñaUAM.Text)) && !(txtContraseñaUAM.Text.Equals(modcon)))

{

modcon = Encriptado(txtContraseñaUAM.Text);

}

string mater = "NULL";

string telef = "NULL";

string a = "0";

if (chkUsuarioUAM.Checked)

{

a = "1";

}

if (modUsuario == 0)

{

try

{

if (!(string.IsNullOrEmpty(txtApellidoMatUAM.Text) || string.IsNullOrWhiteSpace(txtApellidoMatUAM.Text)))

{

mater =txtApellidoMatUAM.Text;

}

if (!(string.IsNullOrEmpty(txtTelefonoUAM.Text) || string.IsNullOrWhiteSpace(txtTelefonoUAM.Text)))

{

telef = txtTelefonoUAM.Text;

}

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("CreateTipoCliente", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Nombre", txtNombreUAM.Text));

cmd.Parameters.Add(new SqlParameter("@Contraseña", modcon));

cmd.Parameters.Add(new SqlParameter("@Nombre\_Usuario", txtNombreUsuarioUAM.Text));

cmd.Parameters.Add(new SqlParameter("@Apellido\_Paterno", txtApellidoPatUAM.Text));

cmd.Parameters.Add(new SqlParameter("@Apellido\_Materno", mater));

cmd.Parameters.Add(new SqlParameter("@Email", txtCorreoUAM.Text));

cmd.Parameters.Add(new SqlParameter("@Telefono",telef ));

cmd.Parameters.Add(new SqlParameter("@Id\_Puesto", cbPuestoUAM.SelectedValue));

cmd.Parameters.Add(new SqlParameter("@Id\_Departamento", cbDepartamentoUAM.SelectedValue));

cmd.Parameters.Add(new SqlParameter("@Id\_Tipo\_usuario", cbTipoUsuarioUAM.SelectedValue));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Creacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Creacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Modificacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Modificacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Estatus", a));

cmd.ExecuteReader();

LogsAceptable("Insertar Usuario", "Usuario");

sql.Close();

MessageBox.Show("Guardado");

panelUsuarioAM.Visible = false;

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else

{

try

{

if (!(string.IsNullOrEmpty(txtApellidoMatUAM.Text) || string.IsNullOrWhiteSpace(txtApellidoMatUAM.Text)))

{

mater = "'" + txtApellidoMatUAM.Text + "'";

}

if (!(string.IsNullOrEmpty(txtTelefonoUAM.Text) || string.IsNullOrWhiteSpace(txtTelefonoUAM.Text)))

{

telef = "'" + txtTelefonoUAM.Text + "'";

}

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

string consulta = "update Usuario set Nombre\_Usuario='" + txtNombreUsuarioUAM.Text + "',Nombre='" + txtNombreUAM.Text + "'," +

"Contraseña='" + modcon + "',Apellido\_Paterno='" + txtApellidoPatUAM.Text + "',Apellido\_Materno=" + mater +

",Email='" + txtCorreoUAM.Text + "',Telefono=" + telef + ",Id\_Usuario\_Modificacion=" + IDuser.ToString() +

",Fecha\_Hora\_Modificacion= '" + DateTime.Now.ToString("yyyy/MM/dd HH:mm:ss.fff") +

"',Estatus="+a+ ",Id\_Puesto="+ cbPuestoUAM.SelectedValue.ToString() + ",Id\_Departamento="+ cbDepartamentoUAM.SelectedValue.ToString()+

",Id\_Tipo\_Usuario="+ cbTipoUsuarioUAM.SelectedValue.ToString()+ " where Id\_Usuario=" + modUsuario.ToString();

cmd = new SqlCommand(consulta, sql);

cmd.ExecuteReader();

LogsAceptable("Modificar Usuario", "Usuario");

sql.Close();

MessageBox.Show("Guardado");

panelUsuarioAM.Visible = false;

}

}

catch (Exception ex) { MessageBox.Show("Error en conexion"); }

}

recargaCatalogoUsuario();

}

}

private void btnVerVenta\_Click(object sender, EventArgs e)

{

if (dgvVerVentas.SelectedRows.Count == 1)

{

cargaDetalleVenta((int)dgvVerVentas.SelectedCells[0].Value);

PanelDetalleVenta.Visible = true;

PanelDetalleVenta.Enabled = true;

PanelDetalleVenta.BringToFront();

}

else { MessageBox.Show("Seleccione una Venta."); }

}

private void btnDetalleVentaRegresar\_Click(object sender, EventArgs e)

{

PanelDetalleVenta.Visible = false;

}

private void cargaDetalleVenta(int venta)

{

DataTable dt = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select l.Titulo,(de.Porcentaje\*100) Descuento,l.Precio from Detalle\_Venta d left join Libro l on d.Id\_Libro\_ISBN=l.Id\_Libro\_ISBN left join Descuento de on d.Id\_Descuento= de.id\_Descuento where d.Id\_Venta="+venta.ToString();

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

dgvDetalleVenta.DataSource = dt;

}

private void recargaCatalogoVentas()

{

DataTable dt = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select v.Id\_Venta Venta,v.Fecha\_Hora\_Creacion Fecha, u.Nombre\_Usuario,c.Nombre Cliente,p.Nombre [Forma de Pago] from venta v left join Tipo\_Cliente c on v.Id\_Tipo\_Cliente=c.Id\_Tipo\_Cliente left join Usuario u on v.Id\_Usuario\_Creacion=u.Id\_Usuario left join Tipo\_Pago p on v.Id\_Tipo\_Pago=p.Id\_Tipo\_Pago where v.Estatus=1";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

dgvVerVentas.DataSource = dt;

}

private void btnEliminaVenta\_Click(object sender, EventArgs e)

{

if (dgvVerVentas.SelectedRows.Count == 1)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("DeleteVenta", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Id", dgvVerVentas.SelectedCells[0].Value));

cmd.ExecuteReader();

LogsAceptable("Eliminar Venta", "Venta");

sql.Close();

MessageBox.Show("Eliminada");

recargaCatalogoVentas();

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else { MessageBox.Show("Seleccione una Venta."); }

}

private void btnBuscaCaja\_Click(object sender, EventArgs e)

{

bool existe = false;

string id = cbLibrosCaja.SelectedValue.ToString();

for (int i = 0; i < dgvCaja.Rows.Count; i++)

{

if ((dgvCaja.Rows[i].Cells["ISBN"].Value.ToString()).Equals(id))

{

existe = true;

}

}

if (!existe)

{

string ti = "";

decimal pre = 0;

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

string consulta = "select l.Id\_Libro\_ISBN ISBN,CONCAT(l.Titulo,', ',f.Nombre) Titulo,l.Precio from Libro l left join Tipo\_Formato f on l.Id\_Tipo\_Formato=f.Id\_Tipo\_Formato where l.Id\_Libro\_ISBN='" + id+"'";

cmd = new SqlCommand(consulta, sql);

SqlDataReader lector;

lector = cmd.ExecuteReader();

if (lector.HasRows)

{

while (lector.Read())

{

ti = (string)lector[1];

pre = (decimal)lector[2];

}

}

}

}

catch (Exception ex) { MessageBox.Show("Error en conexion"); }

dgvCaja.Rows.Add(id, ti,pre,"0");

}

}

private void button1\_Click(object sender, EventArgs e)

{

if (dgvCaja.SelectedRows.Count == 1)

{

dgvCaja.Rows.Remove(dgvCaja.SelectedRows[0]);

}

else { MessageBox.Show("Seleccione un Libro."); }

}

private void btnLimpiaCaja\_Click(object sender, EventArgs e)

{

dgvCaja.Rows.Clear();

}

private void btnRegistraVentaCaja\_Click(object sender, EventArgs e)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("CreateVenta", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Id\_Tipo\_Cliente", cbClienteCaja.SelectedValue));

cmd.Parameters.Add(new SqlParameter("@Id\_Tipo\_Pago", cbPagoCaja.SelectedValue));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Creacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Creacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Modificacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Modificacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Estatus", "1"));

cmd.ExecuteReader();

sql.Close();

}

LogsAceptable("Insertar Venta", "Venta");

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

string consulta = "select top(1) id\_Venta from Venta order by Id\_Venta desc";

cmd = new SqlCommand(consulta, sql);

SqlDataReader lector;

lector = cmd.ExecuteReader();

if (lector.HasRows)

{

while (lector.Read())

{

modVenta = (int)lector[0];

}

}

}

}

catch (Exception ex) { MessageBox.Show("Error en conexion"); }

for (int i = 0; i < dgvCaja.Rows.Count; i++)

{

if (!(dgvCaja.Rows[i].Cells["ISBN"].Value.ToString()).Equals(""))

{

string d = "0";

if (string.IsNullOrEmpty(dgvCaja.Rows[i].Cells["Descuento"].Value.ToString()) || string.IsNullOrWhiteSpace(dgvCaja.Rows[i].Cells["Descuento"].Value.ToString()))

{

d = "0";

}

else {

d = dgvCaja.Rows[i].Cells["Descuento"].Value.ToString();

}

agregarDetalleVentaLibros(dgvCaja.Rows[i].Cells["ISBN"].Value.ToString(),d);

LogsAceptable("Insertar Detalle", "Detalle\_Venta");

}

}

MessageBox.Show("Guardado");

}

catch (Exception ex)

{

MessageBox.Show("Error en conexion");

} }

private void agregarDetalleVentaLibros(string isbn, string descuento)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("CreateDetalleVenta", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Id\_Venta", modVenta));

cmd.Parameters.Add(new SqlParameter("@Id\_Descuento", descuento));

cmd.Parameters.Add(new SqlParameter("@Id\_Libro\_ISBN", isbn));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Creacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Creacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Modificacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Modificacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Estatus", "1"));

cmd.ExecuteReader();

sql.Close();

} }

catch (Exception ex)

{

throw;

}}

private void MiCuentaToolStripMenuItem\_Click(object sender, EventArgs e)

{

int id = 0;

cerrarPaneles();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

string consulta = "select Nombre\_Usuario,Nombre,Contraseña,Apellido\_Paterno,Apellido\_Materno,Email,Telefono,Puesto,Departamento from VRegistroUsuarios where Id\_Usuario=" + IDuser.ToString();

cmd = new SqlCommand(consulta, sql);

SqlDataReader lector;

lector = cmd.ExecuteReader();

if (lector.HasRows)

{

while (lector.Read())

{

txtUsuarioMiCuenta.Text = (string)lector[0];

txtNombreMiCuenta.Text = (string)lector[1];

txtContraMiCuenta.Text = (string)lector[2];

txtApellidoPMiCuenta.Text = (string)lector[3];

txtApellidoMMiCuenta.Text = (string)lector[4];

txtCorreoMiCuenta.Text = (string)lector[5];

txtTelefMiCuenta.Text = (string)lector[6];

txtPuestoMiCuenta.Text = (string)lector[7];

txtDepartMiCuenta.Text = (string)lector[8];

con = (string)lector[2];

}

PanelRegistra.Visible = true;

PanelRegistra.Enabled = true;

}

else

{

PanelRegistra.Visible = false;

PanelRegistra.Enabled = false;

MessageBox.Show("El usuario no existe o datos incorrectos");

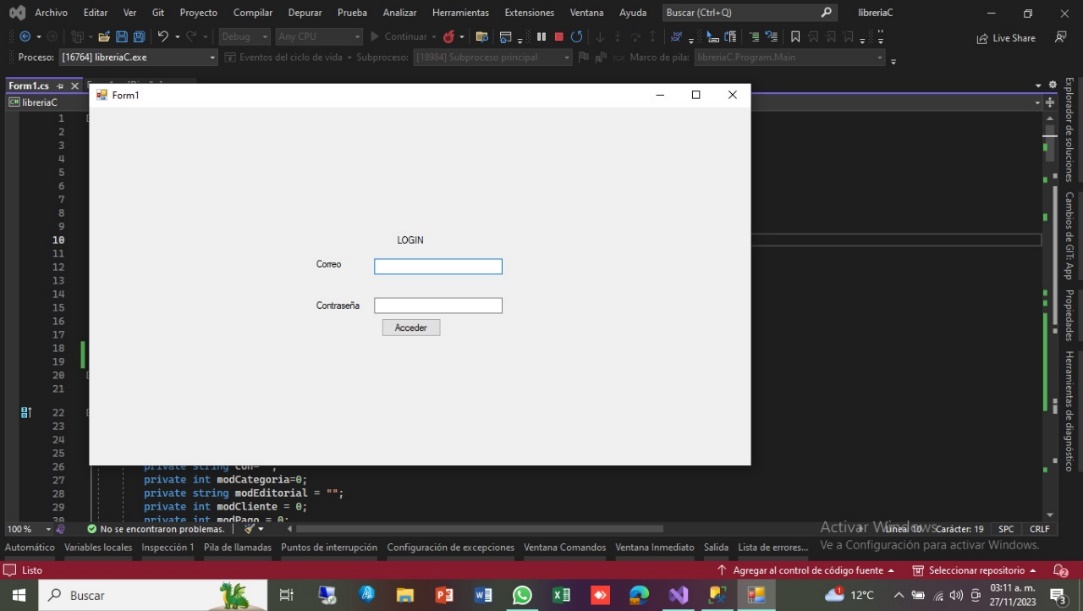
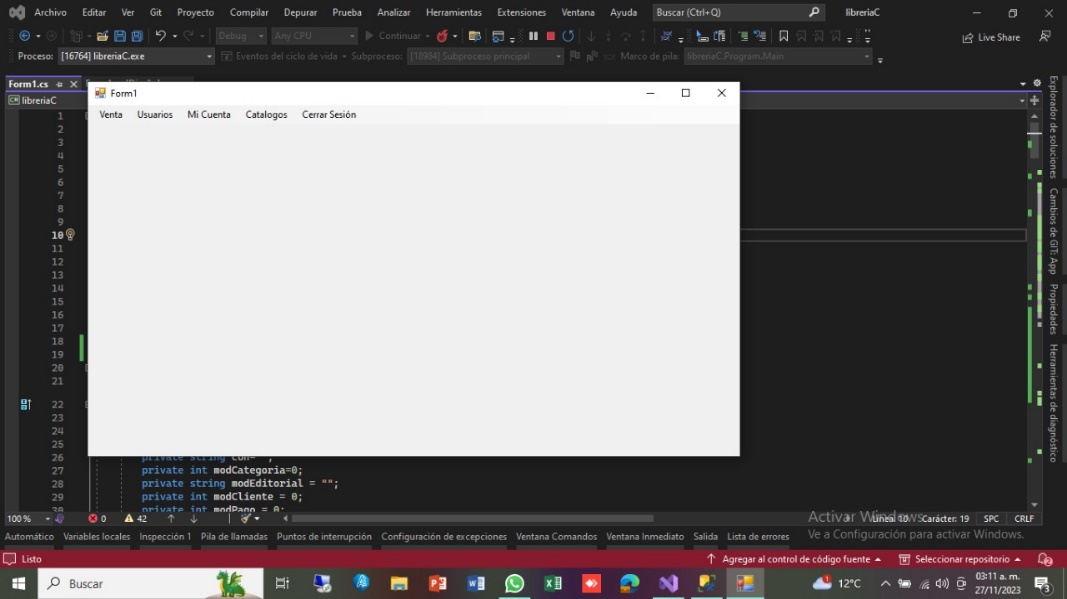
}

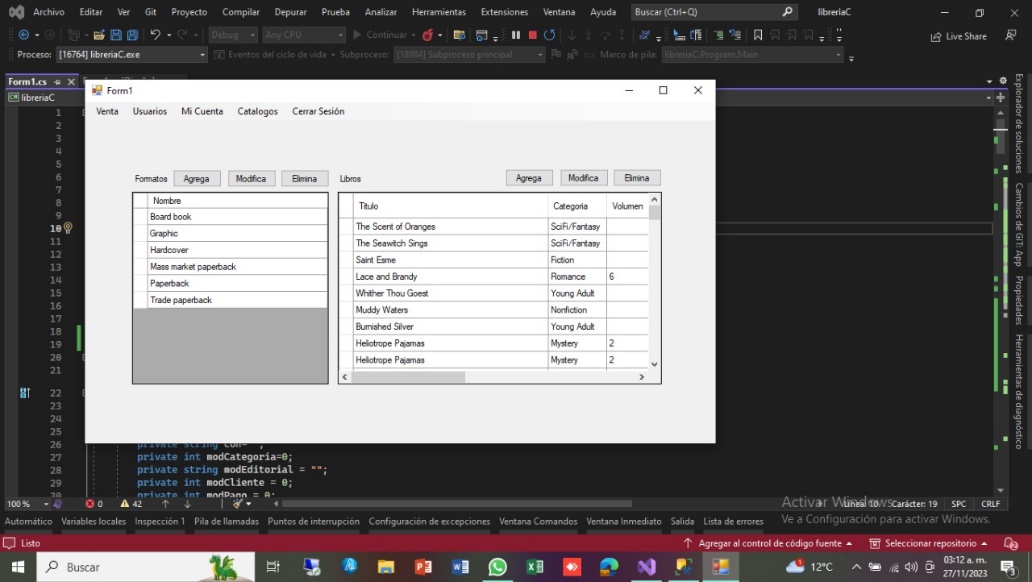
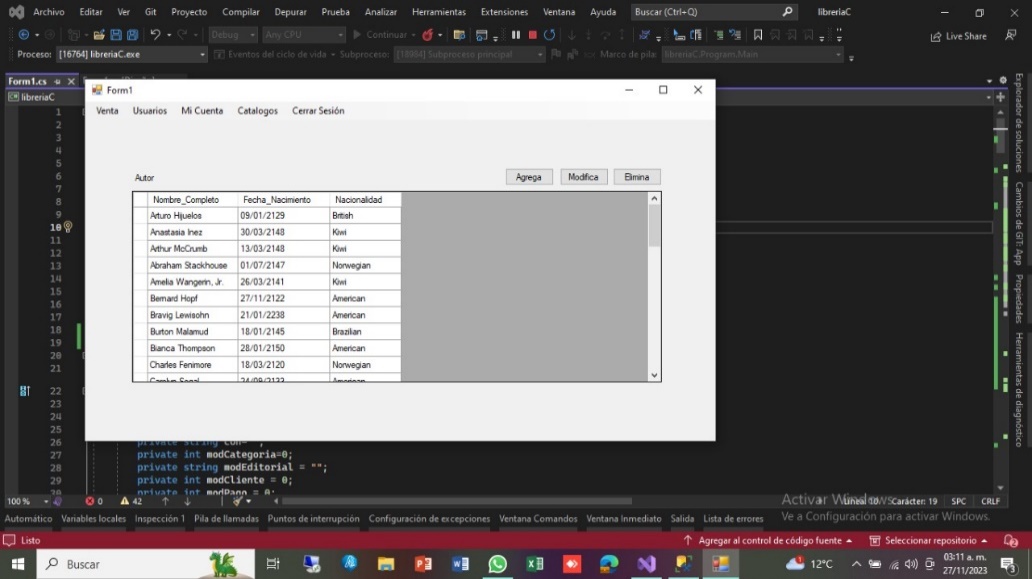
sql.Close();

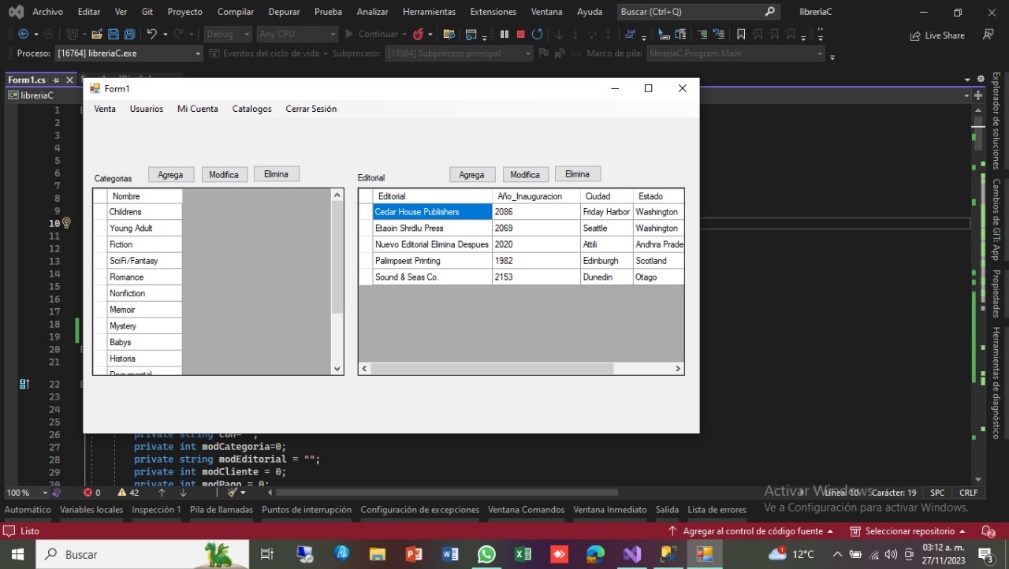
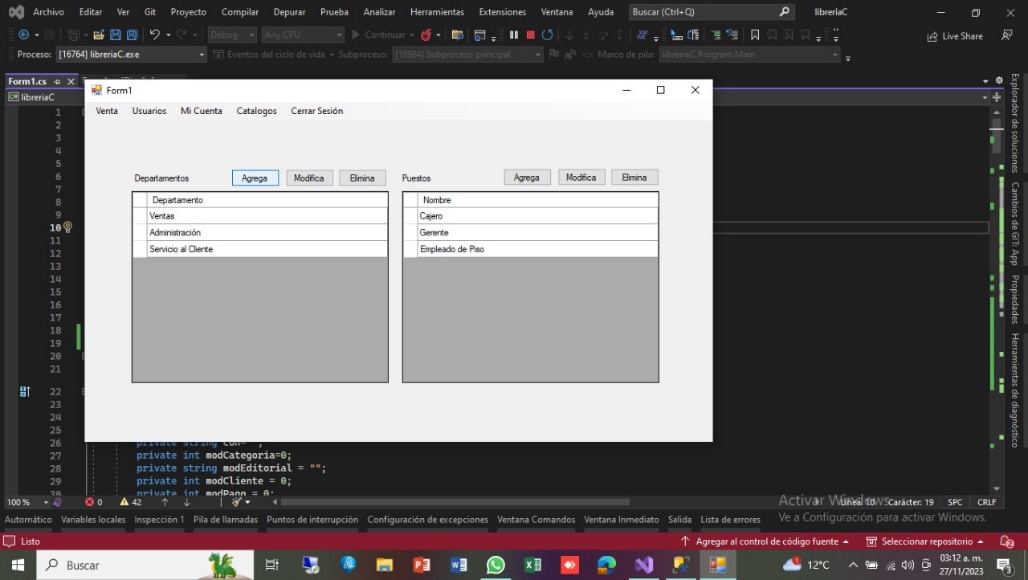
}}

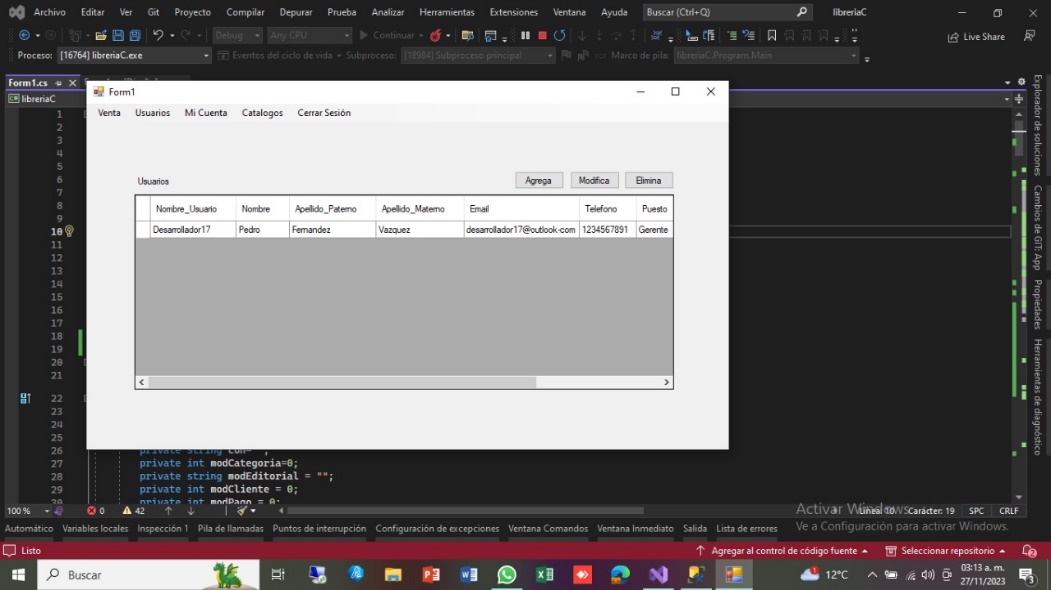
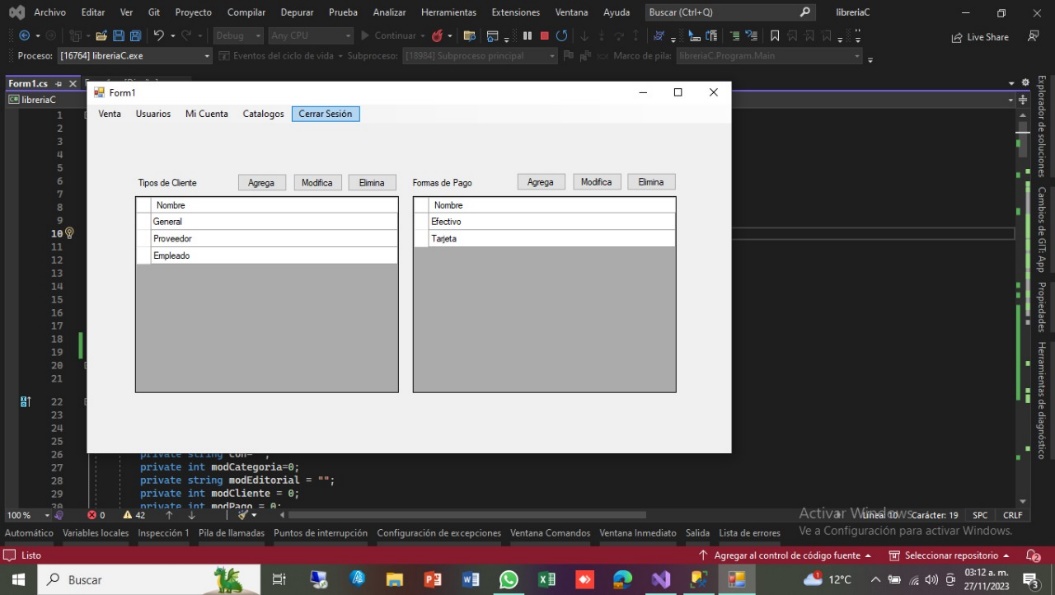
catch (Exception ex) { MessageBox.Show("Error en conexion"); }

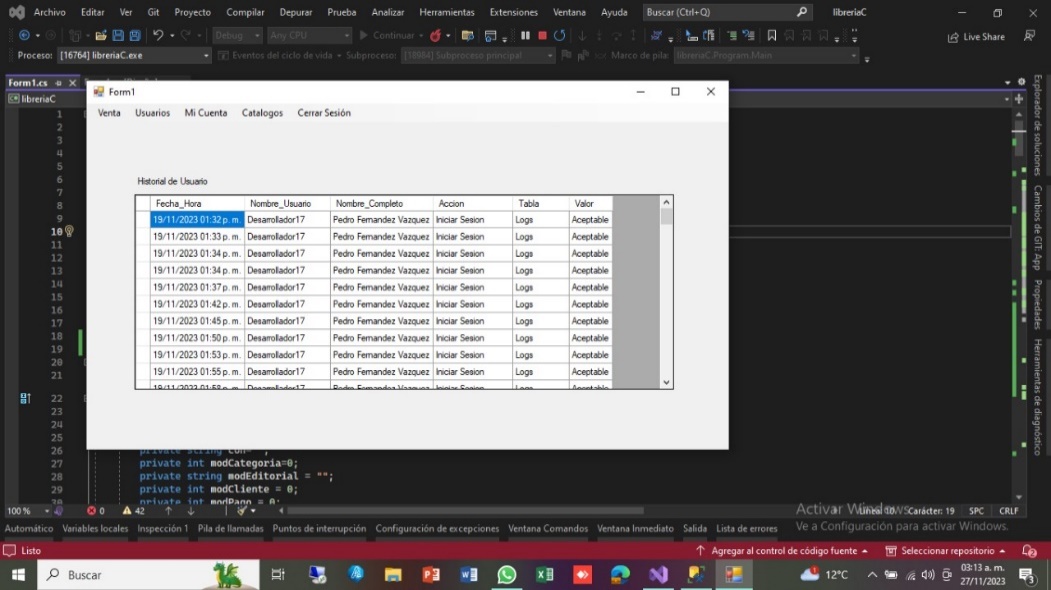
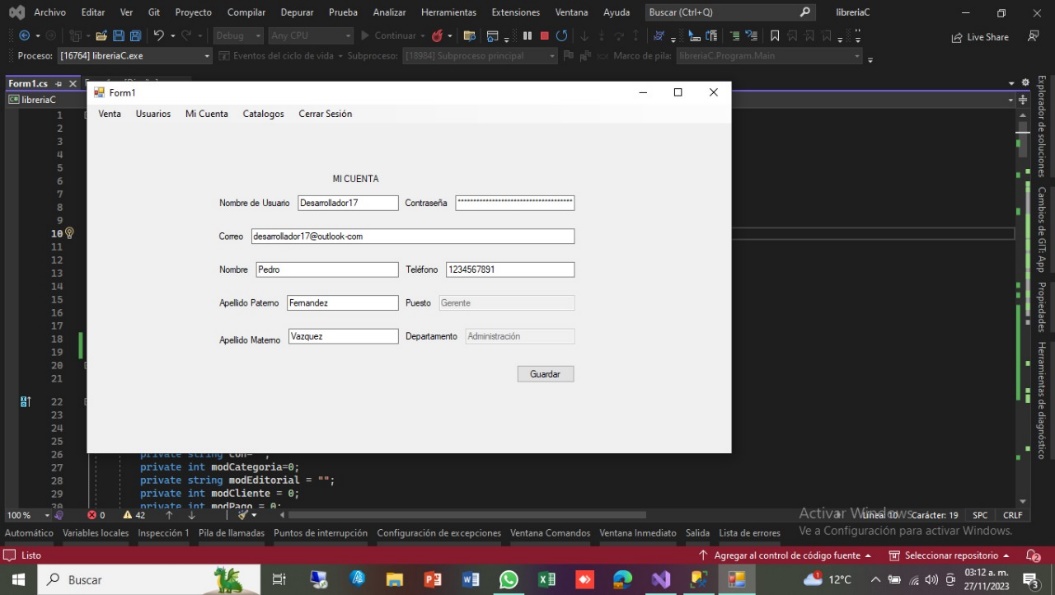
}}}

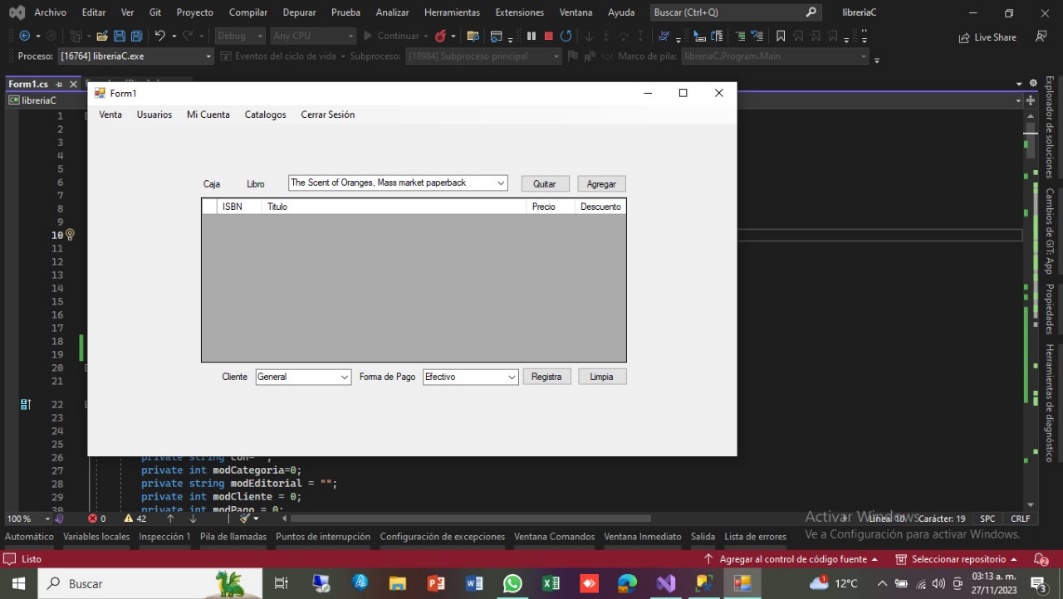
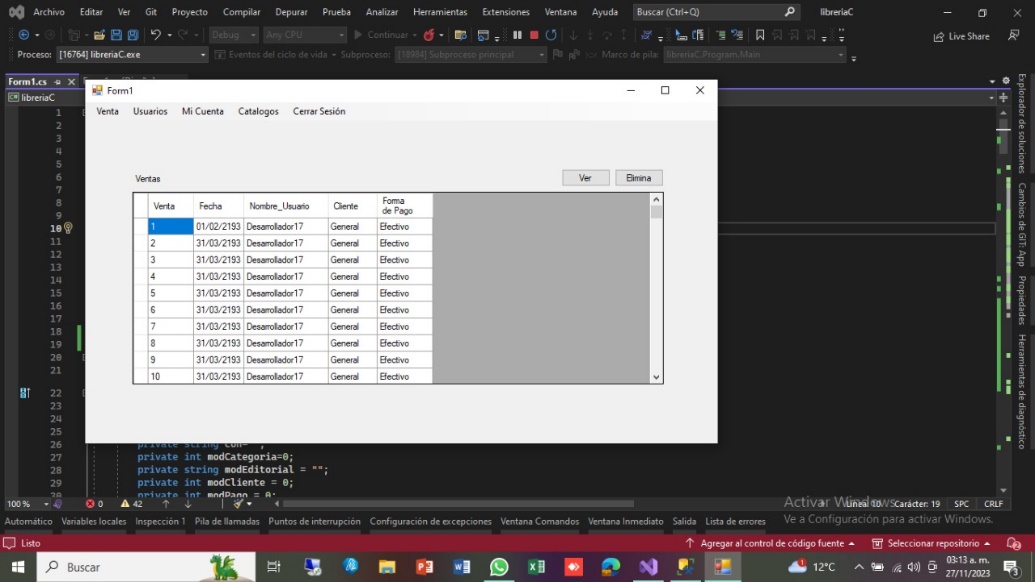


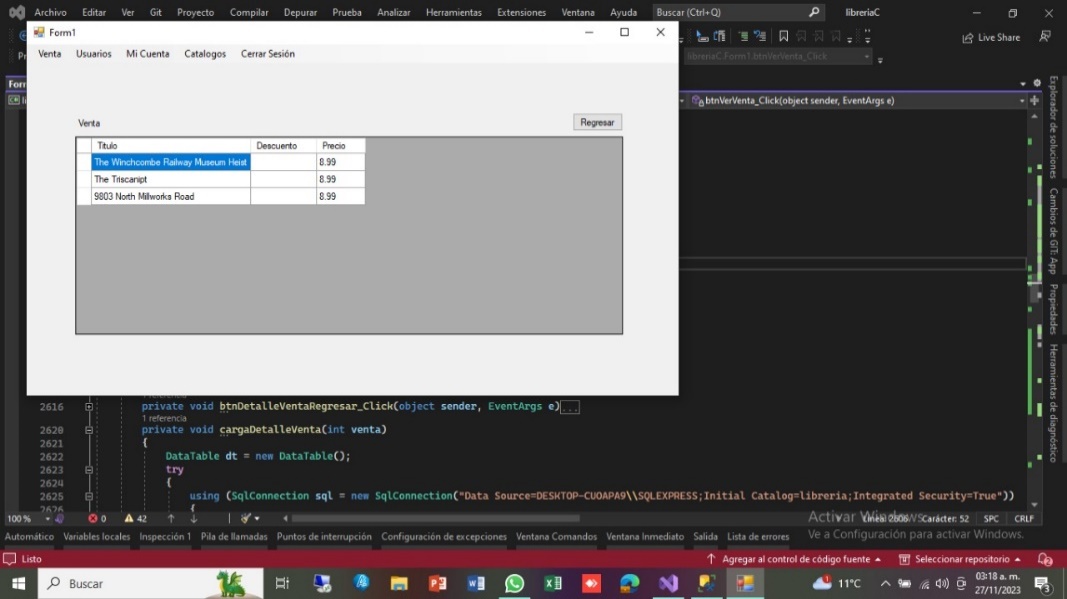
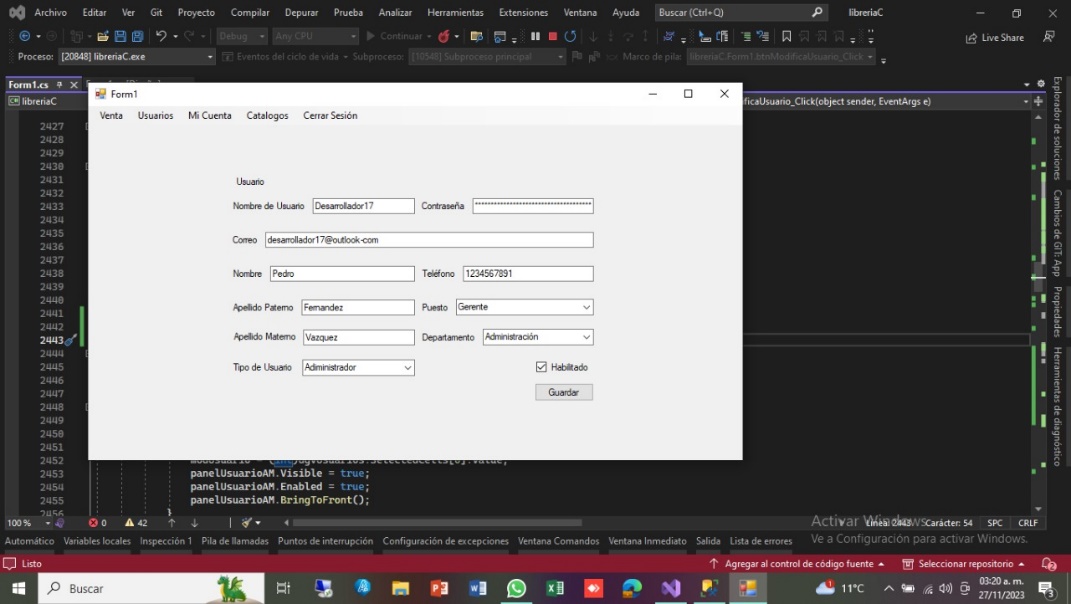


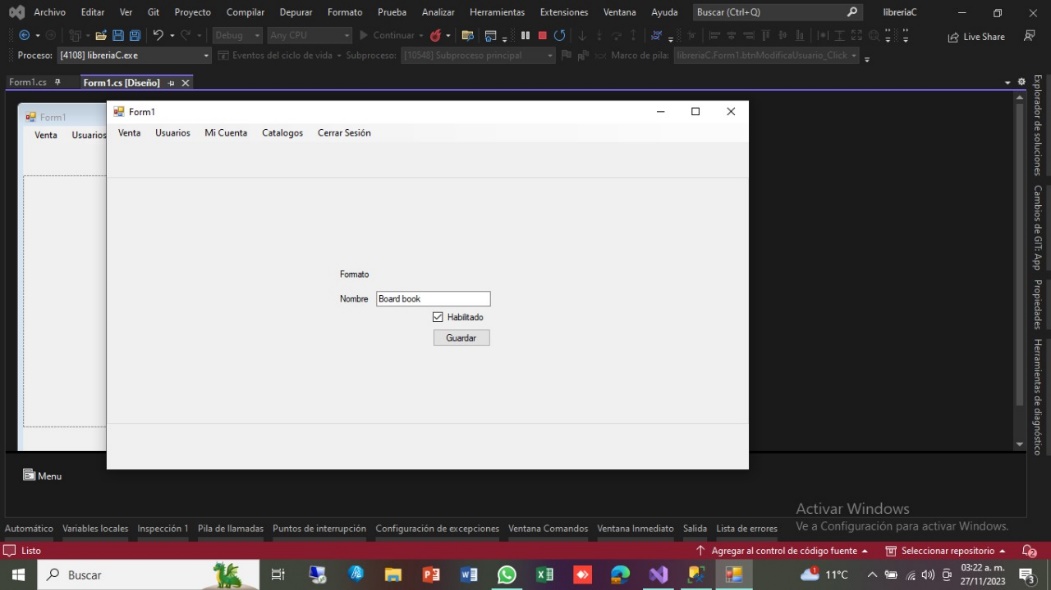
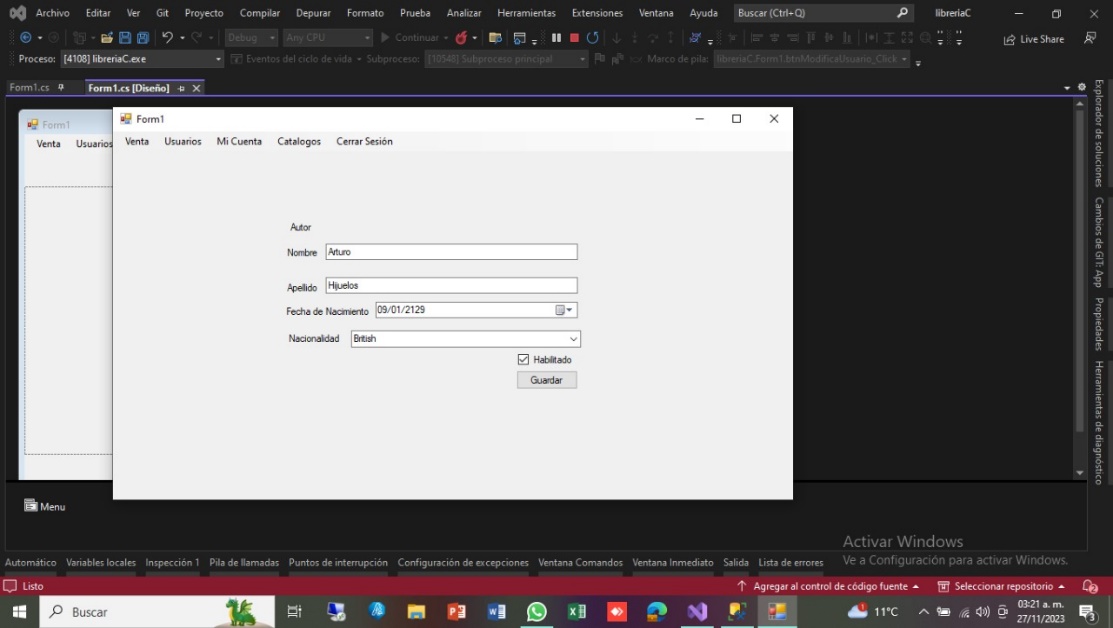


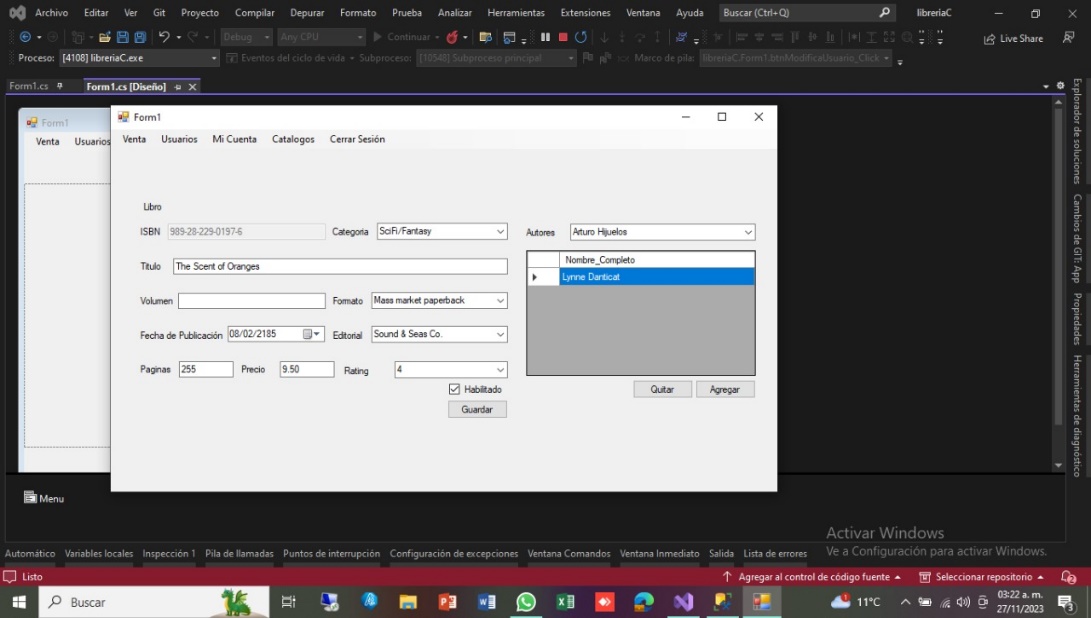
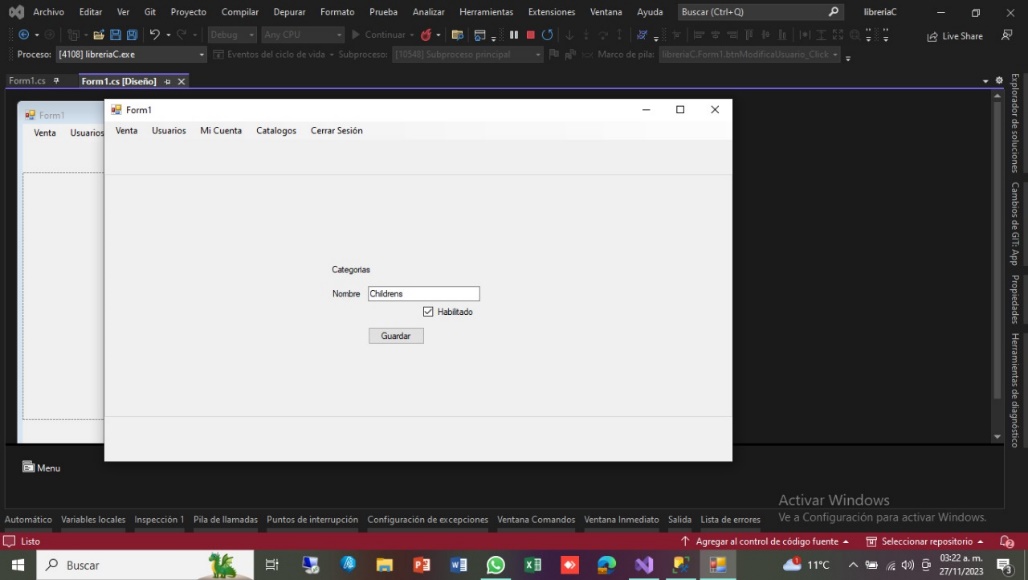


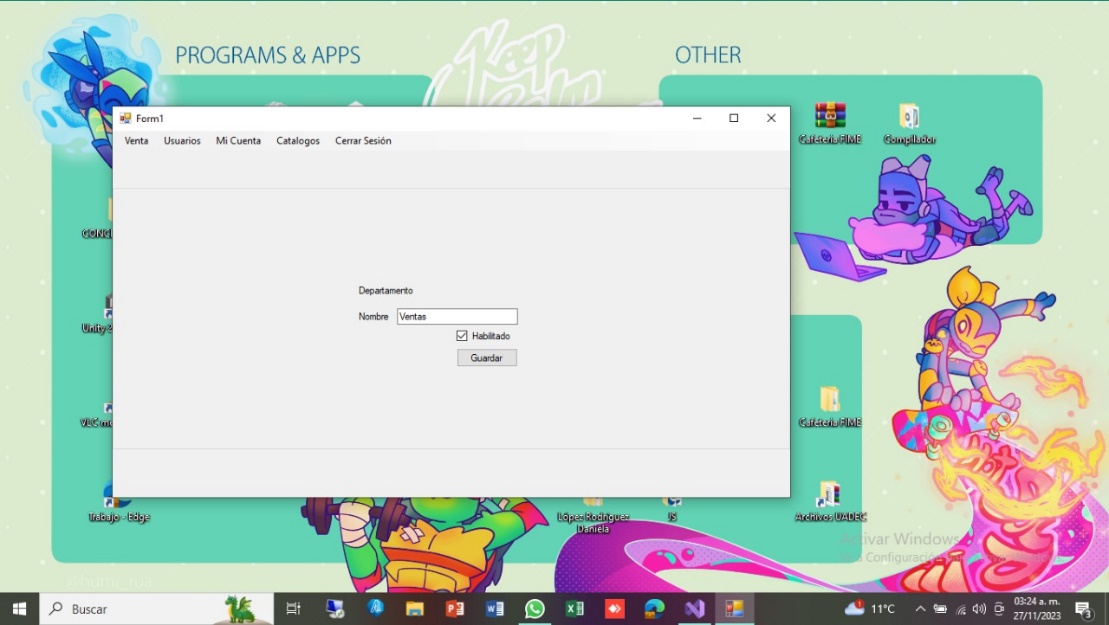


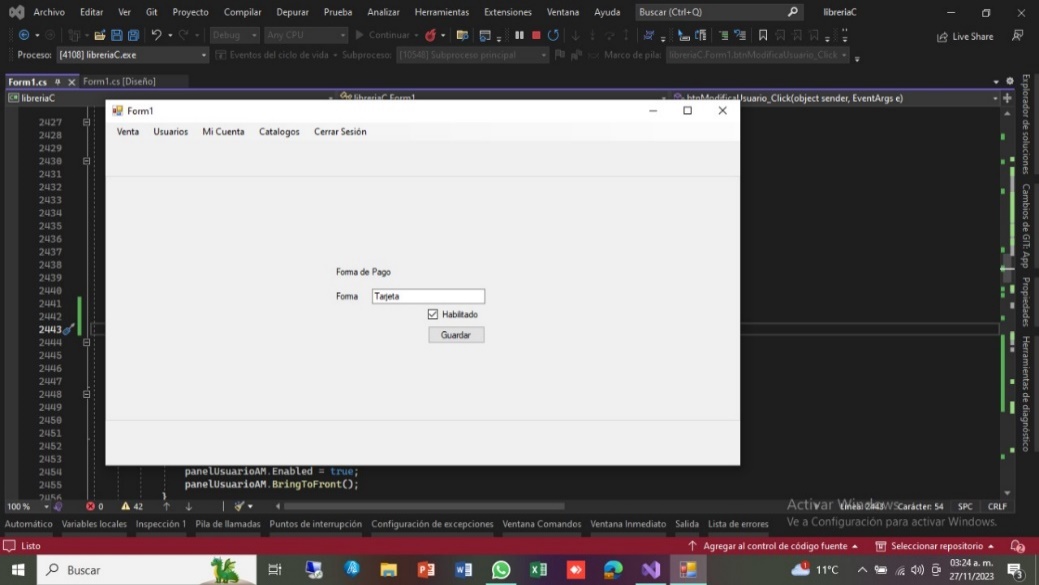
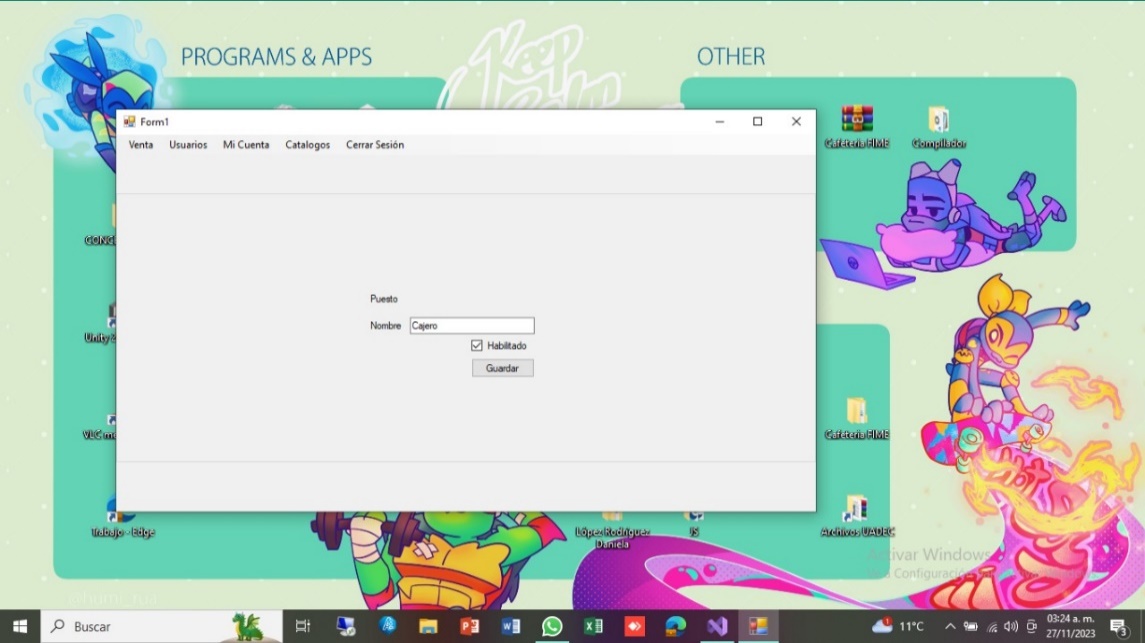


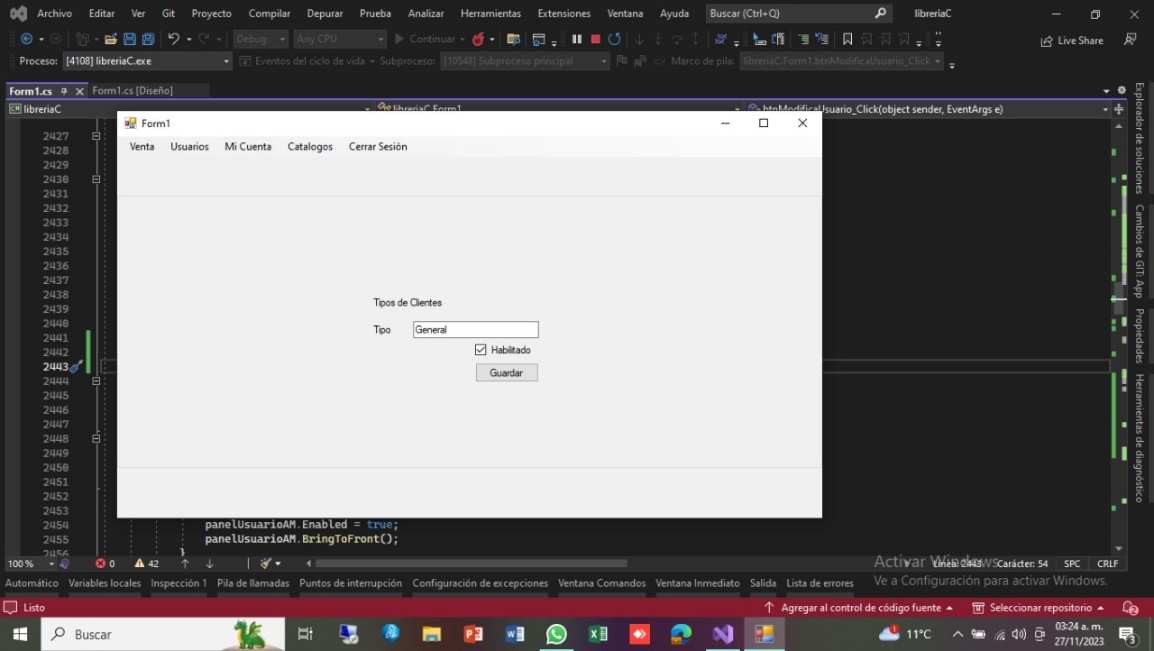












### VB.net

using libreriaC.libreriaDataSetTableAdapters;

using System;

using System.Collections;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Data.SqlTypes;

using System.Drawing;

using System.Linq;

using System.Net.NetworkInformation;

using System.Security.Cryptography;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using static System.Net.Mime.MediaTypeNames;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

using static System.Windows.Forms.VisualStyles.VisualStyleElement.ListView;

namespace libreriaC

{

public partial class Form1 : Form

{

private SqlCommand cmd;

private int IDuser = 0,tipo\_User=0;

private string con="";

private int modCategoria=0;

private string modEditorial = "";

private int modCliente = 0;

private int modPago = 0;

private int modDepartamento = 0;

private int modPuesto = 0;

private int modFormato = 0;

private string modLibro = "";

private string modAutor = "";

private int modUsuario = 0;

private string modcon = "";

private int modVenta = 0;

public Form1()

{

InitializeComponent();

}

private void btnAcceder\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(txtCorreoLogin.Text)) {

txtCorreoLogin.Select();

MessageBox.Show("Favor de llenar todos los campos solicitados");

}

else if (string.IsNullOrEmpty(txtContraLogin.Text)) {

txtContraLogin.Select();

MessageBox.Show("Favor de llenar todos los campos solicitados");

}

else {

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

string consulta = "Select Id\_Usuario,Id\_Tipo\_Usuario from Usuario where Email='" + txtCorreoLogin.Text + "'and Contraseña='" + Encriptado(txtContraLogin.Text) + "'";

cmd = new SqlCommand(consulta, sql);

SqlDataReader lector;

lector = cmd.ExecuteReader();

if (lector.HasRows)

{

while (lector.Read())

{

IDuser = (int)lector[0];

tipo\_User = (int)lector[1];

MessageBox.Show("Se inicio la sesion");

}

if (tipo\_User == 1)

{

Menu.Enabled = true;

Menu.Visible = true;

}

else

{

UsuariosToolStripMenuItem.Visible = false;

UsuariosToolStripMenuItem.Enabled = false;

CatalogosToolStripMenuItem.Visible = false;

CatalogosToolStripMenuItem.Visible = false;

Menu.Enabled = true;

Menu.Visible = true;

}

txtContraLogin.Clear();

PanelLogin.Enabled = false;

PanelLogin.Visible = false;

LogsAceptable("Iniciar Sesion", "Logs");

}

else

{

txtContraLogin.Clear();

txtCorreoLogin.Clear();

MessageBox.Show("El usuario no existe o datos incorrectos");

}

}

}

catch(Exception ex) { MessageBox.Show("Error en conexion"); }

}

}

private string Encriptado(string co)

{

using (var sha256 = new SHA256Managed())

{

return BitConverter.ToString(sha256.ComputeHash(Encoding.UTF8.GetBytes(co))).Replace("-", "");

}

}

private void Form1\_Load(object sender, EventArgs e)

{

cerrarPaneles();

Menu.Visible=false;

Menu.Enabled=false;

PanelLogin.Enabled = true;

PanelLogin.Visible= true;

}

private void LogsAceptable(string accion,string tabla ) {

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("CreateLogs", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Accion", accion));

cmd.Parameters.Add(new SqlParameter("@Tabla", tabla));

cmd.Parameters.Add(new SqlParameter("@Estatus", 1));

cmd.Parameters.Add(new SqlParameter("@Id\_Nivel", 1));

cmd.ExecuteReader();

}

}

catch (Exception ex)

{

}

}

private void Label4\_Click(object sender, EventArgs e)

{

}

private void cerrarPaneles() {

PanelRegistra.Enabled = false;

PanelRegistra.Visible= false;

PanelCateEditor.Visible = false;

PanelCateEditor.Enabled = false;

gbCategoria.Enabled = false;

gbCategoria.Visible = false;

gbEditorial.Visible = false;

gbEditorial.Enabled = false;

pTiClienteTiPago.Enabled = false;

pTiClienteTiPago.Visible = false;

gbClientes.Visible = false;

gbClientes.Enabled = false;

gpPago.Visible = false;

gpPago.Enabled = false;

PanelDepartamentosPuestos.Visible = false;

PanelDepartamentosPuestos.Enabled = false;

gbPuestos.Enabled = false;

gbPuestos.Visible = false;

gbDepartamento.Enabled = false;

gbDepartamento.Visible = false;

panelLibrosFormatos.Enabled = false;

panelLibrosFormatos.Visible = false;

gbFormato.Enabled = false;

gbFormato.Visible = false;

panelLibros.Enabled = false;

panelLibros.Visible = false;

PanelAutor.Enabled = false;

PanelAutor.Visible = false;

panelUsuarioAM.Enabled = false;

panelUsuarioAM.Visible = false;

panelVentaRegistro.Enabled = false;

panelVentaRegistro.Visible = false;

PanelDetalleVenta.Enabled = false;

PanelDetalleVenta.Visible = false;

PanelVentas.Enabled = false;

PanelVentas.Visible = false;

PanelUsuarios.Enabled = false;

PanelUsuarios.Visible = false;

PanelHistorial.Enabled = false;

PanelHistorial.Visible = false;

PanelAutorAM.Enabled = false;

PanelAutorAM.Visible = false;

}

private void cerrarSesiónToolStripMenuItem\_Click(object sender, EventArgs e)

{

cerrarPaneles();

Menu.Visible = false;

Menu.Enabled= false;

MessageBox.Show("Sesión cerrada");

PanelLogin.Enabled = true;

PanelLogin.Visible = true;

txtContraLogin.Clear();

txtCorreoLogin.Clear();

IDuser = 0;

tipo\_User = 0;

}

private void txtContraMiCuenta\_Enter(object sender, EventArgs e)

{

txtContraMiCuenta.Text = "";

}

private void btnGuardarReg\_Click(object sender, EventArgs e)

{

if (!(string.IsNullOrEmpty(txtContraMiCuenta.Text) || string.IsNullOrWhiteSpace(txtContraMiCuenta.Text)) && !(txtContraMiCuenta.Text.Equals(con)))

{

con = Encriptado(txtContraMiCuenta.Text);

}

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

string consulta = "update Usuario set Nombre\_Usuario='" + txtUsuarioMiCuenta.Text + "',Nombre='" + txtNombreMiCuenta.Text + "'," +

"Contraseña='" + con + "',Apellido\_Paterno='" + txtApellidoPMiCuenta.Text + "',Apellido\_Materno='" + txtApellidoMMiCuenta.Text +

"',Email='" + txtCorreoMiCuenta.Text + "',Telefono='" + txtTelefMiCuenta.Text + "',Id\_Usuario\_Modificacion=" + IDuser.ToString() +

",Fecha\_Hora\_Modificacion= '" + DateTime.Now.ToString("yyyy/MM/dd HH:mm:ss.fff") +

"' where Id\_Usuario=" + IDuser.ToString();

cmd = new SqlCommand(consulta, sql);

cmd.ExecuteReader();

PanelRegistra.Visible = true;

PanelRegistra.Enabled = true;

LogsAceptable("Modificar mi perfil", "Usuario");

sql.Close();

MessageBox.Show("Guardado");

}

}

catch (Exception ex) { MessageBox.Show("Error en conexion"); }

}

private void radioButton1\_CheckedChanged(object sender, EventArgs e)

{

}

private void dgvLibros\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void dgvLibros\_CellClick(object sender, DataGridViewCellEventArgs e)

{

}

private void btnAgregCate\_Click(object sender, EventArgs e)

{

gbCategoria.Visible = true;

gbCategoria.Enabled=true;

gbCategoria.BringToFront();

txtUpCreNombreCat.Text = "";

ckbCat.Checked = true;

modCategoria = 0;

}

private void btnUpdCat\_Click(object sender, EventArgs e)

{

if (dgvCategoria.SelectedRows.Count == 1) {

modCategoria =(int) dgvCategoria.SelectedCells[0].Value;

txtUpCreNombreCat.Text = (string)dgvCategoria.SelectedCells[1].Value;

if ((bool)dgvCategoria.SelectedCells[2].Value) {

ckbCat.Checked = true;

}

else {

ckbCat.Checked = false;

}

gbCategoria.Visible = true;

gbCategoria.Enabled = true;

gbCategoria.BringToFront();

}

else { MessageBox.Show("Seleccione una categoria."); }

}

private void recargaCatalogoCategoria() {

DataTable dt = new DataTable();

try {

using (SqlConnection sql= new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd= new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select Id\_Categoria,Nombre,Estatus from Categoria";

using (SqlDataAdapter da =new SqlDataAdapter(cmd))

{

da.Fill(dt);

}

}

sql.Close();

}

}

catch (Exception ex) {

MessageBox.Show(ex.Message);

}

dgvCategoria.DataSource = dt;

//dgvCategoria.Columns["Nombre"].AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;

dgvCategoria.Columns["Id\_Categoria"].Visible = false;

dgvCategoria.Columns["Estatus"].Visible = false;

}

private void recargaCatalogoEditorial()

{

DataTable dt = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "SELECT e.Nombre AS Editorial, e.Año\_Inauguracion, e.Id\_Ciudad, c.Nombre AS Ciudad, es.Nombre AS Estado, p.Nombre AS Pais, c.Id\_Estado, es.Id\_Pais, e.Id\_Editorial, e.Estatus FROM Editorial e INNER JOIN Ciudad c ON e.Id\_Ciudad = c.Id\_Ciudad INNER JOIN Estado es ON c.Id\_Estado = es.Id\_Estado INNER JOIN Pais p ON es.Id\_Pais = p.Id\_Pais";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

dgvEditorial.DataSource = dt;

//dgvEditorial.Columns["Editorial"].AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;

dgvEditorial.Columns["Id\_Ciudad"].Visible = false;

dgvEditorial.Columns["Id\_Estado"].Visible = false;

dgvEditorial.Columns["Id\_Pais"].Visible = false;

dgvEditorial.Columns["Id\_Editorial"].Visible = false;

dgvEditorial.Columns["Estatus"].Visible = false;

}

private void recargaCatalogoTipoCliente()

{

DataTable dt = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = " select Id\_Tipo\_Cliente,Nombre,Estatus from Tipo\_Cliente";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

dgCliente.DataSource = dt;

dgCliente.Columns["Nombre"].AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;

dgCliente.Columns["Id\_Tipo\_Cliente"].Visible = false;

dgCliente.Columns["Estatus"].Visible = false;

}

private void recargaHistorial() {

DataTable dt = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select l.Fecha\_Hora,u.Nombre\_Usuario,concat(u.Nombre, ' ', u.Apellido\_Paterno, ' ', u.Apellido\_Materno) Nombre\_Completo,l.Accion, l.Tabla,n.Valor from Logs l left join Usuario u on l.Id\_Usuario = u.Id\_Usuario left join Nivel n on l.Id\_Nivel = n.Id\_Nivel order by l.Fecha\_Hora";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

dgvHistorial.DataSource = dt;

}

private void recargaCatalogoTipoPago()

{

DataTable dt = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select Id\_Tipo\_Pago,Nombre,Estatus from Tipo\_Pago";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

dgvPago.DataSource = dt;

dgvPago.Columns["Nombre"].AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;

dgvPago.Columns["Id\_Tipo\_Pago"].Visible = false;

dgvPago.Columns["Estatus"].Visible = false;

}

private void recargaCatalogoTipoFormato()

{

DataTable dt = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select Id\_Tipo\_Formato,Nombre,Estatus from Tipo\_Formato";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

dgvFormatos.DataSource = dt;

dgvFormatos.Columns["Nombre"].AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;

dgvFormatos.Columns["Id\_Tipo\_Formato"].Visible = false;

dgvFormatos.Columns["Estatus"].Visible = false;

}

private void recargaCatalogoAutor()

{

DataTable dt = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select a.Id\_Autor,a.Nombre,a.Apellido,CONCAT(a.Nombre,' ',a.Apellido) Nombre\_Completo,a.Fecha\_Nacimiento,a.Id\_Pais,p.Nacionalidad,a.Estatus from Autor a left join Pais p on a.Id\_Pais=p.Id\_Pais";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

dgvAutor.DataSource = dt;

//dgvAutor.Columns["Nombre"].AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;

dgvAutor.Columns["Id\_Autor"].Visible = false;

dgvAutor.Columns["Nombre"].Visible = false;

dgvAutor.Columns["Apellido"].Visible = false;

dgvAutor.Columns["Id\_Pais"].Visible = false;

dgvAutor.Columns["Estatus"].Visible = false;

}

private void recargaCatalogoLibro()

{

DataTable dt = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select l.Id\_Libro\_ISBN,l.Titulo,l.Id\_Categoria,c.Nombre Categoria,isnull(l.Volumen,'') Volumen,l.Id\_Tipo\_Formato,f.Nombre Formato,l.Id\_Editorial, e.Nombre Editorial,isnull(l.Fecha\_Publicacion,'') [Fecha de publicacion],isnull(l.Paginas,'') Paginas,l.Precio,l.Id\_Rating,r.Valor Rating,l.Estatus from Libro l left join Categoria c on l.Id\_Categoria=c.Id\_Categoria left join Tipo\_Formato f on l.Id\_Tipo\_Formato=f.Id\_Tipo\_Formato left join Editorial e on l.Id\_Editorial= e.Id\_Editorial left join Rating r on l.Id\_Rating=r.Id\_Rating\r\n";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

dgvLibros.DataSource = dt;

dgvLibros.Columns["Id\_Libro\_ISBN"].Visible = false;

dgvLibros.Columns["Id\_Categoria"].Visible = false;

dgvLibros.Columns["Id\_Tipo\_Formato"].Visible = false;

dgvLibros.Columns["Id\_Editorial"].Visible = false;

dgvLibros.Columns["Id\_Rating"].Visible = false;

dgvLibros.Columns["Estatus"].Visible = false;

}

private void recargaCatalogoDepartamento()

{

DataTable dt = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select Id\_Departamento,Departamento,Estatus from Departamento";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

dgvDepartamentos.DataSource = dt;

dgvDepartamentos.Columns["Departamento"].AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;

dgvDepartamentos.Columns["Id\_Departamento"].Visible = false;

dgvDepartamentos.Columns["Estatus"].Visible = false;

}

private void recargaCatalogoUsuario()

{

DataTable dt = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select u.Id\_Usuario,u.Nombre\_Usuario,u.Nombre,u.Contraseña,u.Apellido\_Paterno,u.Apellido\_Materno,u.Email,u.Telefono,u.Id\_Puesto,p.Nombre Puesto,u.Id\_Departamento,d.Departamento,u.Id\_Tipo\_Usuario,t.Nombre [Tipo de Usuario],u.Estatus from Usuario u left join Puesto p on u.Id\_Puesto=p.Id\_Puesto left join Departamento d on u.Id\_Departamento=d.Id\_Departamento left join Tipo\_Usuario t on u.Id\_Tipo\_Usuario=t.Id\_Tipo\_Usuario\r\n";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

dgvUsuarios.DataSource = dt;

dgvUsuarios.Columns["Id\_Usuario"].Visible = false;

dgvUsuarios.Columns["Contraseña"].Visible = false;

dgvUsuarios.Columns["Id\_Puesto"].Visible = false;

dgvUsuarios.Columns["Id\_Departamento"].Visible = false;

dgvUsuarios.Columns["Id\_Tipo\_Usuario"].Visible = false;

dgvUsuarios.Columns["Estatus"].Visible = false;

}

private void recargaCatalogoPuestos()

{

DataTable dt = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select Id\_Puesto,Nombre,Estatus from Puesto";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

dgvPuestos.DataSource = dt;

dgvPuestos.Columns["Nombre"].AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;

dgvPuestos.Columns["Id\_Puesto"].Visible = false;

dgvPuestos.Columns["Estatus"].Visible = false;

}

private void cargacombosUsuarios()

{

DataTable t = new DataTable();

DataTable de = new DataTable();

DataTable p = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select Id\_Tipo\_Usuario, Nombre from Tipo\_Usuario";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(t);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

cbTipoUsuarioUAM.DataSource = t;

cbTipoUsuarioUAM.DisplayMember = "Nombre";

cbTipoUsuarioUAM.ValueMember = "Id\_Tipo\_Usuario";

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select Id\_Departamento,Departamento from Departamento where Estatus=1";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(de);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

cbDepartamentoUAM.DataSource = de;

cbDepartamentoUAM.DisplayMember = "Departamento";

cbDepartamentoUAM.ValueMember = "Id\_Departamento";

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select Id\_Puesto,Nombre from Puesto where Estatus=1";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(p);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

cbPuestoUAM.DataSource = p;

cbPuestoUAM.DisplayMember = "Nombre";

cbPuestoUAM.ValueMember = "Id\_Puesto";

}

private void cargaCombosVentaLibros()

{

DataTable l = new DataTable();

DataTable c = new DataTable();

DataTable p = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select l.Id\_Libro\_ISBN,CONCAT(l.Titulo,', ',f.Nombre) Nombre from Libro l left join Tipo\_Formato f on l.Id\_Tipo\_Formato=f.Id\_Tipo\_Formato where l.Estatus=1";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(l);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

cbLibrosCaja.DataSource = l;

cbLibrosCaja.DisplayMember = "Nombre";

cbLibrosCaja.ValueMember = "Id\_Libro\_ISBN";

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select Id\_Tipo\_Cliente,Nombre from Tipo\_Cliente where Estatus=1\r\n";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(c);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

cbClienteCaja.DataSource = c;

cbClienteCaja.DisplayMember = "Nombre";

cbClienteCaja.ValueMember = "Id\_Tipo\_Cliente";

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select Id\_Tipo\_Pago,Nombre from Tipo\_Pago where Estatus=1";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(p);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

cbPagoCaja.DataSource = p;

cbPagoCaja.DisplayMember = "Nombre";

cbPagoCaja.ValueMember = "Id\_Tipo\_Pago";

}

private void cargacombosLibros() {

DataTable cat = new DataTable();

DataTable Ed = new DataTable();

DataTable ra = new DataTable();

DataTable fo = new DataTable();

DataTable au = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select Id\_Categoria,Nombre from Categoria where Estatus=1";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(cat);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

cbCategoriaLibro.DataSource = cat;

cbCategoriaLibro.DisplayMember = "Nombre";

cbCategoriaLibro.ValueMember = "Id\_Categoria";

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select Id\_Editorial,Nombre from Editorial where Estatus=1";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(Ed);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

cbEditorialLibro.DataSource = Ed;

cbEditorialLibro.DisplayMember = "Nombre";

cbEditorialLibro.ValueMember = "Id\_Editorial";

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select Id\_Rating,Valor from Rating where Estatus=1";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(ra);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

cbRatingLibro.DataSource = ra;

cbRatingLibro.DisplayMember = "Valor";

cbRatingLibro.ValueMember = "Id\_Rating";

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select Id\_Tipo\_Formato,Nombre from Tipo\_Formato where Estatus=1";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(fo);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

cbFormatoLibro.DataSource = fo;

cbFormatoLibro.DisplayMember = "Nombre";

cbFormatoLibro.ValueMember = "Id\_Tipo\_Formato";

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select a.Id\_Autor,CONCAT(a.Nombre,' ',a.Apellido) Nombre\_Completo from Autor a where a.Estatus=1\r\n";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(au);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

cbAutoresLibro.DataSource = au;

cbAutoresLibro.DisplayMember = "Nombre\_Completo";

cbAutoresLibro.ValueMember = "Id\_Autor";

}

private void CategoriasYEditorialesToolStripMenuItem\_Click(object sender, EventArgs e)

{

cerrarPaneles();

PanelCateEditor.Visible = true;

PanelCateEditor.Enabled= true;

recargaCatalogoCategoria();

recargaCatalogoEditorial();

}

private void btnElimCate\_Click(object sender, EventArgs e)

{

if (dgvCategoria.SelectedRows.Count == 1)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("DeleteCategoria", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Id", dgvCategoria.SelectedCells[0].Value));

cmd.ExecuteReader();

LogsAceptable("Eliminar Categoria", "Categoria");

sql.Close();

MessageBox.Show("Eliminada");

recargaCatalogoCategoria();

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else { MessageBox.Show("Seleccione una categoria."); }

}

private void btnUpdEdit\_Click(object sender, EventArgs e)

{

if (dgvEditorial.SelectedRows.Count == 1)

{

DataTable dt = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "SELECT c.Id\_Ciudad, concat(c.Nombre,', ',es.Nombre,', ',p.Nombre) as localizacion FROM Ciudad c INNER JOIN Estado es ON c.Id\_Estado = es.Id\_Estado INNER JOIN Pais p ON es.Id\_Pais = p.Id\_Pais";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

string año = "2023";

if (!string.IsNullOrEmpty((string)dgvEditorial.SelectedCells[1].Value)) {

año = (string)dgvEditorial.SelectedCells[1].Value;

}

cbCiudadEditorial.DataSource = dt;

cbCiudadEditorial.DisplayMember = "localizacion";

cbCiudadEditorial.ValueMember = "Id\_Ciudad";

cbCiudadEditorial.SelectedValue= (int)dgvEditorial.SelectedCells[2].Value;

modEditorial = (string)dgvEditorial.SelectedCells[8].Value;

txtNombreEditorial.Text = (string)dgvEditorial.SelectedCells[0].Value;

dtañoinaguracionEditorial.Value=new DateTime(Convert.ToInt32(año),1,1);

if ((bool)dgvEditorial.SelectedCells[9].Value)

{

chkEditorial.Checked = true;

}

else

{

chkEditorial.Checked = false;

}

gbEditorial.Visible = true;

gbEditorial.Enabled = true;

gbEditorial.BringToFront();

}

else { MessageBox.Show("Seleccione una Editorial."); }

}

private string SiglaEditExis(string nombre) {

string siglas = "";

for (int i=0; i<nombre.Length-1;i++) {

if (i == 0) {

siglas = siglas + nombre.Substring(i, 1);

}

if (nombre.Substring(i,1).Equals(" ")){

siglas = siglas + nombre.Substring(i+1, 1);

}

}

if (siglas.Length > 3){

siglas = siglas.Substring(0, 3);

}

return siglas.ToUpper();

}

private void btnElimEdit\_Click(object sender, EventArgs e)

{

if (dgvEditorial.SelectedRows.Count == 1)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("DeleteEditorial", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Id", dgvEditorial.SelectedCells[8].Value));

cmd.ExecuteReader();

LogsAceptable("Eliminar Editorial", "Editorial");

sql.Close();

MessageBox.Show("Eliminada");

recargaCatalogoEditorial();

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else { MessageBox.Show("Seleccione una Editorial."); }

}

private void btnAgregaEdit\_Click(object sender, EventArgs e)

{

gbEditorial.Visible = true;

gbEditorial.Enabled = true;

gbEditorial.BringToFront();

txtNombreEditorial.Text = "";

dtañoinaguracionEditorial.Value = DateTime.Now;

DataTable dt = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "SELECT c.Id\_Ciudad, concat(c.Nombre,', ',es.Nombre,', ',p.Nombre) as localizacion FROM Ciudad c INNER JOIN Estado es ON c.Id\_Estado = es.Id\_Estado INNER JOIN Pais p ON es.Id\_Pais = p.Id\_Pais";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

cbCiudadEditorial.DataSource = dt;

cbCiudadEditorial.DisplayMember = "localizacion";

cbCiudadEditorial.ValueMember = "Id\_Ciudad";

chkEditorial.Checked = true;

modEditorial = "0";

}

private void categ0ToolStripButton\_Click(object sender, EventArgs e)

{

}

private void btnGuarModCat\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(txtUpCreNombreCat.Text) || string.IsNullOrWhiteSpace(txtUpCreNombreCat.Text)) {

MessageBox.Show("Ingrese los datos solicitados");

}

else

{

string a = "0";

if (ckbCat.Checked)

{

a = "1";

}

if (modCategoria == 0)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("CreateCategoria", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Nombre", txtUpCreNombreCat.Text));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Creacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Creacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Modificacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Modificacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Estatus", a));

cmd.ExecuteReader();

LogsAceptable("Insertar Categoria", "Categoria");

sql.Close();

MessageBox.Show("Guardado");

gbCategoria.Visible = false;

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

string consulta = "Update Categoria set Nombre='" + txtUpCreNombreCat.Text + "',Estatus=" + a + ",Id\_Usuario\_Modificacion=" + IDuser.ToString() +

",Fecha\_Hora\_Modificacion= '" + DateTime.Now.ToString("yyyy/MM/dd HH:mm:ss.fff") + "' where id\_Categoria=" + modCategoria.ToString();

cmd = new SqlCommand(consulta, sql);

cmd.ExecuteReader();

LogsAceptable("Modificar categorias", "Categoria");

sql.Close();

MessageBox.Show("Guardado");

gbCategoria.Visible = false;

}

}

catch (Exception ex) { MessageBox.Show("Error en conexion"); }

}

recargaCatalogoCategoria();

}

}

private void btnModInsEditorial\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(txtNombreEditorial.Text) || string.IsNullOrWhiteSpace(txtNombreEditorial.Text)||string.IsNullOrEmpty(cbCiudadEditorial.Text)||string.IsNullOrWhiteSpace(cbCiudadEditorial.Text) || string.IsNullOrEmpty(dtañoinaguracionEditorial.Text) || string.IsNullOrWhiteSpace(dtañoinaguracionEditorial.Text))

{

MessageBox.Show("Ingrese los datos solicitados");

}

else

{

string a = "0";

if (chkEditorial.Checked)

{

a = "1";

}

if (modEditorial.Equals("0"))

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

//

sql.Open();

cmd = new SqlCommand("CreateEditorial", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Id\_Editorial", SiglaEditExis(txtNombreEditorial.Text)));

cmd.Parameters.Add(new SqlParameter("@Nombre", txtNombreEditorial.Text));

cmd.Parameters.Add(new SqlParameter("@Id\_Ciudad", cbCiudadEditorial.SelectedValue));

cmd.Parameters.Add(new SqlParameter("@Año\_Inaguracion", dtañoinaguracionEditorial.Text));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Creacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Creacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Modificacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Modificacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Estatus", a));

cmd.ExecuteReader();

LogsAceptable("Insertar Editorial", "Editorial");

sql.Close();

MessageBox.Show("Guardado");

gbEditorial.Visible = false;

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

//ciudad año inaugura

string consulta = "Update Editorial set Nombre='" + txtNombreEditorial.Text + "',Estatus=" + a + ",Id\_Usuario\_Modificacion=" + IDuser.ToString() + ",Año\_Inauguracion='" +dtañoinaguracionEditorial.Text+ "', Id\_Ciudad=" + cbCiudadEditorial.SelectedValue.ToString() + ",Fecha\_Hora\_Modificacion= '" + DateTime.Now.ToString("yyyy/MM/dd HH:mm:ss.fff") + "' where Id\_Editorial='" + modEditorial+"'";

cmd = new SqlCommand(consulta, sql);

cmd.ExecuteReader();

LogsAceptable("Modificar Editorial", "Editorial");

sql.Close();

MessageBox.Show("Guardado");

gbEditorial.Visible = false;

}

}

catch (Exception ex) { MessageBox.Show("Error en conexion"); }

}

recargaCatalogoEditorial();

}

}

private void btnAgregaCliente\_Click(object sender, EventArgs e)

{

gbClientes.Visible = true;

gbClientes.Enabled = true;

gbClientes.BringToFront();

txtNombreCliente.Text = "";

chkTiCliente.Checked = true;

modCliente = 0;

}

private void btnModCliente\_Click(object sender, EventArgs e)

{

if (dgCliente.SelectedRows.Count == 1)

{

txtNombreCliente.Text = (string)dgCliente.SelectedCells[1].Value;

if ((bool)dgCliente.SelectedCells[2].Value)

{

chkTiCliente.Checked = true;

}

else

{

chkTiCliente.Checked = false;

}

modCliente = (int)dgCliente.SelectedCells[0].Value;

gbClientes.Visible = true;

gbClientes.Enabled = true;

gbClientes.BringToFront();

}

else { MessageBox.Show("Seleccione un Tipo de Cliente."); }

}

private void btnEliminaCliente\_Click(object sender, EventArgs e)

{

if (dgCliente.SelectedRows.Count == 1)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("DeleteTipo\_Cliente", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Id", dgCliente.SelectedCells[0].Value));

cmd.ExecuteReader();

LogsAceptable("Eliminar Tipo Cliente", "Tipo\_Cliente");

sql.Close();

MessageBox.Show("Eliminada");

recargaCatalogoTipoCliente();

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else { MessageBox.Show("Seleccione un Tipo de Cliente."); }

}

private void btnAgregaPago\_Click(object sender, EventArgs e)

{

gpPago.Visible = true;

gpPago.Enabled = true;

gpPago.BringToFront();

txtNombrePago.Text = "";

chkPago.Checked = true;

modPago = 0;

}

private void btnModificaPago\_Click(object sender, EventArgs e)

{

if (dgvPago.SelectedRows.Count == 1)

{

txtNombrePago.Text = (string)dgvPago.SelectedCells[1].Value;

if ((bool)dgvPago.SelectedCells[2].Value)

{

chkPago.Checked = true;

}

else

{

chkPago.Checked = false;

}

modPago = (int)dgvPago.SelectedCells[0].Value;

gpPago.Visible = true;

gpPago.Enabled = true;

gpPago.BringToFront();

}

else { MessageBox.Show("Seleccione una Forma de Pago."); }

}

private void btnEliminaPago\_Click(object sender, EventArgs e)

{

if (dgvPago.SelectedRows.Count == 1)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("DeleteTipo\_Pago", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Id", dgvPago.SelectedCells[0].Value));

cmd.ExecuteReader();

LogsAceptable("Eliminar Tipo de Pago", "Tipo\_Pago");

sql.Close();

MessageBox.Show("Eliminada");

recargaCatalogoTipoPago();

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else { MessageBox.Show("Seleccione un Tipo de Pago."); }

}

private void TiposDeClientesYTiposDePagosToolStripMenuItem\_Click(object sender, EventArgs e)

{

cerrarPaneles();

pTiClienteTiPago.Visible = true;

pTiClienteTiPago.Enabled = true;

recargaCatalogoTipoPago();

recargaCatalogoTipoCliente();

}

private void btnAgregaModificaPago\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(txtNombrePago.Text) || string.IsNullOrWhiteSpace(txtNombrePago.Text))

{

MessageBox.Show("Ingrese los datos solicitados");

}

else

{

string a = "0";

if (chkPago.Checked)

{

a = "1";

}

if (modPago == 0)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("CreateTipoPago", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Nombre", txtNombrePago.Text));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Creacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Creacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Modificacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Modificacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Estatus", a));

cmd.ExecuteReader();

LogsAceptable("Insertar Forma de Pago", "Tipo\_Pago");

sql.Close();

MessageBox.Show("Guardado");

gbCategoria.Visible = false;

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

string consulta = "Update Tipo\_Pago set Nombre='" + txtNombrePago.Text + "',Estatus=" + a + ",Id\_Usuario\_Modificacion=" + IDuser.ToString() +

",Fecha\_Hora\_Modificacion= '" + DateTime.Now.ToString("yyyy/MM/dd HH:mm:ss.fff") + "' where Id\_Tipo\_Pago=" + modPago.ToString();

cmd = new SqlCommand(consulta, sql);

cmd.ExecuteReader();

LogsAceptable("Modificar Forma de Pago", "Tipo\_Pago");

sql.Close();

MessageBox.Show("Guardado");

gbCategoria.Visible = false;

}

}

catch (Exception ex) { MessageBox.Show("Error en conexion"); }

}

recargaCatalogoTipoPago();

}

}

private void btnAgregaModificaTiCliente\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(txtNombreCliente.Text) || string.IsNullOrWhiteSpace(txtNombreCliente.Text))

{

MessageBox.Show("Ingrese los datos solicitados");

}

else

{

string a = "0";

if (chkTiCliente.Checked)

{

a = "1";

}

if (modCliente == 0)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("CreateTipoCliente", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Nombre", txtNombreCliente.Text));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Creacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Creacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Modificacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Modificacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Estatus", a));

cmd.ExecuteReader();

LogsAceptable("Insertar Tipo de Cliente", "Tipo\_Cliente");

sql.Close();

MessageBox.Show("Guardado");

gbCategoria.Visible = false;

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

string consulta = "Update Tipo\_Cliente set Nombre='" + txtNombreCliente.Text + "',Estatus=" + a + ",Id\_Usuario\_Modificacion=" + IDuser.ToString() +

",Fecha\_Hora\_Modificacion= '" + DateTime.Now.ToString("yyyy/MM/dd HH:mm:ss.fff") + "' where id\_Tipo\_Cliente=" + modCliente.ToString();

cmd = new SqlCommand(consulta, sql);

cmd.ExecuteReader();

LogsAceptable("Modificar Tipo de Cliente", "Tipo\_Cliente");

sql.Close();

MessageBox.Show("Guardado");

gbCategoria.Visible = false;

}

}

catch (Exception ex) { MessageBox.Show("Error en conexion"); }

}

recargaCatalogoTipoCliente();

}

}

private void LibrosToolStripMenuItem\_Click(object sender, EventArgs e)

{

cerrarPaneles();

panelLibrosFormatos.Visible = true;

panelLibrosFormatos.Enabled = true;

recargaCatalogoTipoFormato();

recargaCatalogoLibro();

}

private void DepartamentosYPuestosToolStripMenuItem\_Click(object sender, EventArgs e)

{

cerrarPaneles();

PanelDepartamentosPuestos.Visible = true;

PanelDepartamentosPuestos.Enabled = true;

recargaCatalogoPuestos();

recargaCatalogoDepartamento();

}

private void btnAgregaDepartamento\_Click(object sender, EventArgs e)

{

gbDepartamento.Visible = true;

gbDepartamento.Enabled = true;

gbDepartamento.BringToFront();

txtNombreDepartamento.Text = "";

chkDepartamento.Checked = true;

modDepartamento = 0;

}

private void btnAgregaPuestos\_Click(object sender, EventArgs e)

{

gbPuestos.Visible = true;

gbPuestos.Enabled = true;

gbPuestos.BringToFront();

txtNombrePuestos.Text = "";

chkPuesto.Checked = true;

modPuesto = 0;

}

private void btnAgregaModificaDepartamento\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(txtNombreDepartamento.Text) || string.IsNullOrWhiteSpace(txtNombreDepartamento.Text))

{

MessageBox.Show("Ingrese los datos solicitados");

}

else

{

string a = "0";

if (chkDepartamento.Checked)

{

a = "1";

}

if (modDepartamento == 0)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("CreateDepartamento", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Departamento", txtNombreDepartamento.Text));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Creacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Creacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Modificacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Modificacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Estatus", a));

cmd.ExecuteReader();

LogsAceptable("Insertar Departamento", "Departamento");

sql.Close();

MessageBox.Show("Guardado");

gbDepartamento.Visible = false;

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

string consulta = "Update Departamento set Departamento='" + txtNombreDepartamento.Text + "',Estatus=" + a + ",Id\_Usuario\_Modificacion=" + IDuser.ToString() +

",Fecha\_Hora\_Modificacion= '" + DateTime.Now.ToString("yyyy/MM/dd HH:mm:ss.fff") + "' where Id\_Departamento=" + modDepartamento.ToString();

cmd = new SqlCommand(consulta, sql);

cmd.ExecuteReader();

LogsAceptable("Modificar Departamento", "Departamento");

sql.Close();

MessageBox.Show("Guardado");

gbDepartamento.Visible = false;

}

}

catch (Exception ex) { MessageBox.Show("Error en conexion"); }

}

recargaCatalogoDepartamento();

}

}

private void btnAgregaModificaPuesto\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(txtNombrePuestos.Text) || string.IsNullOrWhiteSpace(txtNombrePuestos.Text))

{

MessageBox.Show("Ingrese los datos solicitados");

}

else

{

string a = "0";

if (chkPuesto.Checked)

{

a = "1";

}

if (modPuesto == 0)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("CreatePuesto", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Nombre", txtNombrePuestos.Text));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Creacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Creacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Modificacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Modificacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Estatus", a));

cmd.ExecuteReader();

LogsAceptable("Insertar Puesto", "Puesto");

sql.Close();

MessageBox.Show("Guardado");

gbPuestos.Visible = false;

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

string consulta = "Update Puesto set Nombre='" + txtNombrePuestos.Text + "',Estatus=" + a + ",Id\_Usuario\_Modificacion=" + IDuser.ToString() +

",Fecha\_Hora\_Modificacion= '" + DateTime.Now.ToString("yyyy/MM/dd HH:mm:ss.fff") + "' where Id\_Puesto=" + modPuesto.ToString();

cmd = new SqlCommand(consulta, sql);

cmd.ExecuteReader();

LogsAceptable("Modificar Puesto", "Puesto");

sql.Close();

MessageBox.Show("Guardado");

gbPuestos.Visible = false;

}

}

catch (Exception ex) { MessageBox.Show("Error en conexion"); }

}

recargaCatalogoPuestos();

}

}

private void btnModificaDepartamento\_Click(object sender, EventArgs e)

{

if (dgvDepartamentos.SelectedRows.Count == 1)

{

txtNombreDepartamento.Text = (string)dgvDepartamentos.SelectedCells[1].Value;

if ((bool)dgvDepartamentos.SelectedCells[2].Value)

{

chkDepartamento.Checked = true;

}

else

{

chkDepartamento.Checked = false;

}

modDepartamento = (int)dgvDepartamentos.SelectedCells[0].Value;

gbDepartamento.Visible = true;

gbDepartamento.Enabled = true;

gbDepartamento.BringToFront();

}

else { MessageBox.Show("Seleccione un Departamento."); }

}

private void btnModificaPuestos\_Click(object sender, EventArgs e)

{

if (dgvPuestos.SelectedRows.Count == 1)

{

txtNombrePuestos.Text = (string)dgvPuestos.SelectedCells[1].Value;

if ((bool)dgvPuestos.SelectedCells[2].Value)

{

chkPuesto.Checked = true;

}

else

{

chkPuesto.Checked = false;

}

modPuesto = (int)dgvPuestos.SelectedCells[0].Value;

gbPuestos.Visible = true;

gbPuestos.Enabled = true;

gbPuestos.BringToFront();

}

else { MessageBox.Show("Seleccione un Puesto."); }

}

private void btnEliminaDepartamento\_Click(object sender, EventArgs e)

{

if (dgvDepartamentos.SelectedRows.Count == 1)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("DeleteDepartamento", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Id", dgvDepartamentos.SelectedCells[0].Value));

cmd.ExecuteReader();

LogsAceptable("Eliminar Departamento", "Departamento");

sql.Close();

MessageBox.Show("Eliminada");

recargaCatalogoDepartamento();

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else { MessageBox.Show("Seleccione un Departamento."); }

}

private void btnEliminaPuestos\_Click(object sender, EventArgs e)

{

if (dgvPuestos.SelectedRows.Count == 1)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("DeletePuesto", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Id", dgvPuestos.SelectedCells[0].Value));

cmd.ExecuteReader();

LogsAceptable("Eliminar Puesto", "Puesto");

sql.Close();

MessageBox.Show("Eliminada");

recargaCatalogoPuestos();

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else { MessageBox.Show("Seleccione un Puesto."); }

}

private void btnAgregaFormatos\_Click(object sender, EventArgs e)

{

gbFormato.Visible = true;

gbFormato.Enabled = true;

gbFormato.BringToFront();

txtNombreFormato.Text = "";

chkFormato.Checked = true;

modFormato = 0;

}

private void btnAgregarLibro\_Click(object sender, EventArgs e)

{

dgvAutoresLibro.Rows.Clear();

panelLibros.Visible = true;

panelLibros.Enabled = true;

panelLibros.BringToFront();

txtISBN.Text = "";

txtISBN.Enabled =true;

txtTitulo.Text = "";

cargacombosLibros();

chkLibro.Checked = true;

modLibro = "0";

txtVolumenLibro.Text = "";

dtFechaLibro.Value = DateTime.Now;

txtPaginasLibro.Text = "";

txtPrecioLibro.Text = "";

dgvAutoresLibro.DataSource = null;

}

private void btnModificaFormato\_Click(object sender, EventArgs e)

{

if (dgvFormatos.SelectedRows.Count == 1)

{

txtNombreFormato.Text = (string)dgvFormatos.SelectedCells[1].Value;

if ((bool)dgvFormatos.SelectedCells[2].Value)

{

chkFormato.Checked = true;

}

else

{

chkFormato.Checked = false;

}

modFormato = (int)dgvFormatos.SelectedCells[0].Value;

gbFormato.Visible = true;

gbFormato.Enabled = true;

gbFormato.BringToFront();

}

else { MessageBox.Show("Seleccione un Formato."); }

}

private void cargaLibrosAutoresTabla(string isbn) {

dgvAutoresLibro.Rows.Clear();

DataTable dt = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select l.Id\_Autor,CONCAT(a.Nombre,' ',a.Apellido) Nombre\_Completo from Libro\_Autores l left join Autor a on l.Id\_Autor=a.Id\_Autor where l.Id\_Libro\_ISBN='"+isbn+"'";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

for (int i = 0; i < dt.Rows.Count; i++)

{

dgvAutoresLibro.Rows.Add(dt.Rows[i]["Id\_Autor"].ToString(), dt.Rows[i]["Nombre\_Completo"].ToString());

}

}

private void btnModificaLibro\_Click(object sender, EventArgs e)

{

if (dgvLibros.SelectedRows.Count == 1)

{

cargacombosLibros();

txtISBN.Text = (string)dgvLibros.SelectedCells[0].Value;

txtISBN.Enabled =false;

cargaLibrosAutoresTabla(txtISBN.Text);

txtTitulo.Text= (string)dgvLibros.SelectedCells[1].Value;

cbCategoriaLibro.SelectedValue= (int)dgvLibros.SelectedCells[2].Value;

txtVolumenLibro.Text= (string)dgvLibros.SelectedCells[4].Value;

cbFormatoLibro.SelectedValue = (int)dgvLibros.SelectedCells[5].Value;

cbEditorialLibro.SelectedValue = (string)dgvLibros.SelectedCells[7].Value;

dtFechaLibro.Value=(DateTime)dgvLibros.SelectedCells[9].Value;

txtPaginasLibro.Text= ((int)dgvLibros.SelectedCells[10].Value).ToString();

txtPrecioLibro.Text = ((decimal)dgvLibros.SelectedCells[11].Value).ToString();

cbRatingLibro.SelectedValue = (int)dgvLibros.SelectedCells[12].Value;

if ((bool)dgvLibros.SelectedCells[14].Value)

{

chkLibro.Checked = true;

}

else

{

chkLibro.Checked = false;

}

modLibro = (string)dgvLibros.SelectedCells[0].Value;

panelLibros.Visible = true;

panelLibros.Enabled = true;

panelLibros.BringToFront();

}

else { MessageBox.Show("Seleccione un Libro."); }

}

private void btnEliminaFormato\_Click(object sender, EventArgs e)

{

if (dgvFormatos.SelectedRows.Count == 1)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("DeleteTipo\_Formato", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Id", dgvFormatos.SelectedCells[0].Value));

cmd.ExecuteReader();

LogsAceptable("Eliminar Formato", "Tipo\_Formato");

sql.Close();

MessageBox.Show("Eliminada");

recargaCatalogoTipoFormato();

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else { MessageBox.Show("Seleccione un Formato."); }

}

private void btnEliminaLibro\_Click(object sender, EventArgs e)

{

if (dgvLibros.SelectedRows.Count == 1)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("DeleteLibro", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Id", dgvLibros.SelectedCells[0].Value));

cmd.ExecuteReader();

LogsAceptable("Eliminar Libro", "Libro");

sql.Close();

MessageBox.Show("Eliminada");

recargaCatalogoLibro();

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else { MessageBox.Show("Seleccione un Libro."); }

}

private void btnAgregarModificarFormato\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(txtNombreFormato.Text) || string.IsNullOrWhiteSpace(txtNombreFormato.Text))

{

MessageBox.Show("Ingrese los datos solicitados");

}

else

{

string a = "0";

if (chkFormato.Checked)

{

a = "1";

}

if (modFormato == 0)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("CreateTipoFormato", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Nombre", txtNombreFormato.Text));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Creacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Creacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Modificacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Modificacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Estatus", a));

cmd.ExecuteReader();

LogsAceptable("Insertar Formato", "Tipo\_Formato");

sql.Close();

MessageBox.Show("Guardado");

gbFormato.Visible = false;

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

string consulta = "Update Tipo\_Formato set Nombre='" + txtNombreFormato.Text + "',Estatus=" + a + ",Id\_Usuario\_Modificacion=" + IDuser.ToString() +

",Fecha\_Hora\_Modificacion= '" + DateTime.Now.ToString("yyyy/MM/dd HH:mm:ss.fff") + "' where Id\_Tipo\_Formato=" + modFormato.ToString();

cmd = new SqlCommand(consulta, sql);

cmd.ExecuteReader();

LogsAceptable("Modificar Formato", "Tipo\_Formato");

sql.Close();

MessageBox.Show("Guardado");

gbFormato.Visible = false;

}

}

catch (Exception ex) { MessageBox.Show("Error en conexion"); }

}

recargaCatalogoTipoFormato();

}

}

private void btnAgregarModificarLibro\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(txtISBN.Text) || string.IsNullOrWhiteSpace(txtISBN.Text)

|| string.IsNullOrEmpty(txtTitulo.Text) || string.IsNullOrWhiteSpace(txtTitulo.Text) ||

string.IsNullOrEmpty(txtPrecioLibro.Text) || string.IsNullOrWhiteSpace(txtPrecioLibro.Text) ||

string.IsNullOrEmpty(cbCategoriaLibro.Text) || string.IsNullOrWhiteSpace(cbCategoriaLibro.Text) ||

string.IsNullOrEmpty(cbFormatoLibro.Text) || string.IsNullOrWhiteSpace(cbFormatoLibro.Text) ||

string.IsNullOrEmpty(cbEditorialLibro.Text) || string.IsNullOrWhiteSpace(cbEditorialLibro.Text) ||

string.IsNullOrEmpty(cbRatingLibro.Text) || string.IsNullOrWhiteSpace(cbRatingLibro.Text))

{

MessageBox.Show("Ingrese los datos solicitados");

}

else

{

string vol = null;

string f = null;

string pag = null;

if (!(string.IsNullOrEmpty(txtVolumenLibro.Text) || string.IsNullOrWhiteSpace(txtVolumenLibro.Text))) {

vol = txtVolumenLibro.Text;

}

if (!(dtFechaLibro.Text.Equals(""))) {

f = dtFechaLibro.Value.ToShortDateString();

}

if (!(txtPaginasLibro.Text.Equals(""))) {

pag = txtPaginasLibro.Text;

}

string a = "0";

if (chkLibro.Checked)

{

a = "1";

}

if (modLibro.Equals("0"))

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("CreateLibro", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Id\_Libro\_ISBN", txtISBN.Text));

cmd.Parameters.Add(new SqlParameter("@Titulo", txtTitulo.Text));

cmd.Parameters.Add(new SqlParameter("@Id\_Categoria", cbCategoriaLibro.SelectedValue));

cmd.Parameters.Add(new SqlParameter("@Volumen",vol ));

cmd.Parameters.Add(new SqlParameter("@Id\_Tipo\_Formato", cbFormatoLibro.SelectedValue));

cmd.Parameters.Add(new SqlParameter("@Id\_Editorial", cbEditorialLibro.SelectedValue));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Publicacion",f));

cmd.Parameters.Add(new SqlParameter("@Paginas",pag));

cmd.Parameters.Add(new SqlParameter("@Precio",txtPrecioLibro.Text));

cmd.Parameters.Add(new SqlParameter("@Id\_Rating", cbRatingLibro.SelectedValue));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Creacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Creacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Modificacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Modificacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Estatus", a));

cmd.ExecuteReader();

EliminaTodosAutoresLibro(txtISBN.Text);

for (int i = 0; i < dgvAutoresLibro.Rows.Count; i++)

{

if (!(dgvAutoresLibro.Rows[i].Cells["Id\_Autor"].Value.ToString()).Equals(""))

{

agregarAutoresLibro(txtISBN.Text, dgvAutoresLibro.Rows[i].Cells["Id\_Autor"].Value.ToString());

}

}

LogsAceptable("Insertar Libro", "Libro");

sql.Close();

MessageBox.Show("Guardado");

panelLibros.Visible = false;

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

if (!(string.IsNullOrEmpty(txtVolumenLibro.Text) || string.IsNullOrWhiteSpace(txtVolumenLibro.Text)))

{

vol ="'"+txtVolumenLibro.Text+"'";

}

else {

vol = "NULL";

}

if (!(dtFechaLibro.Text.Equals("")))

{

f = "'" + dtFechaLibro.Value.ToShortDateString() + "'";

}

else {

f = "NULL";

}

if (!(txtPaginasLibro.Text.Equals("")))

{

pag = txtPaginasLibro.Text;

}

else {

pag = "NULL";

}

string consulta = "Update Libro set Titulo='" + txtTitulo.Text + "',Estatus=" + a + ",Id\_Usuario\_Modificacion=" + IDuser.ToString() +

",Fecha\_Hora\_Modificacion= '" + DateTime.Now.ToString("yyyy/MM/dd HH:mm:ss.fff") + "',"+

"Id\_Categoria="+ cbCategoriaLibro.SelectedValue.ToString()+",Volumen="+vol+","+

"Paginas="+pag+ ",Id\_Tipo\_Formato="+cbEditorialLibro.SelectedValue.ToString() + ",Id\_Editorial="+cbEditorialLibro.SelectedValue.ToString() +

",Id\_Rating="+cbRatingLibro.SelectedValue.ToString()+ ",Precio="+txtPrecioLibro.Text+ ",Fecha\_Publicacion="+f +

" where Id\_Libro\_ISBN='" + modLibro.ToString()+"'";

cmd = new SqlCommand(consulta, sql);

cmd.ExecuteReader();

LogsAceptable("Modificar Libro", "Libro");

EliminaTodosAutoresLibro(txtISBN.Text);

for (int i = 0; i < dgvAutoresLibro.Rows.Count; i++)

{

if (!(dgvAutoresLibro.Rows[i].Cells["Id\_Autor"].Value.ToString()).Equals(""))

{

agregarAutoresLibro(txtISBN.Text, dgvAutoresLibro.Rows[i].Cells["Id\_Autor"].Value.ToString());

}

}

sql.Close();

MessageBox.Show("Guardado");

panelLibros.Visible = false;

}

}

catch (Exception ex) { MessageBox.Show("Error en conexion"); }

}

recargaCatalogoLibro();

}

}

private void AutoresToolStripMenuItem\_Click(object sender, EventArgs e)

{

cerrarPaneles();

PanelAutor.Visible = true;

PanelAutor.Enabled = true;

recargaCatalogoAutor();

}

private void btnagregaAutorLibro\_Click(object sender, EventArgs e)

{ bool existe = false;

string id=cbAutoresLibro.SelectedValue.ToString();

for (int i=0;i< dgvAutoresLibro.Rows.Count;i++) {

if ((dgvAutoresLibro.Rows[i].Cells["Id\_Autor"].Value.ToString()).Equals(id)) {

existe = true;

}

}

if (!existe){

dgvAutoresLibro.Rows.Add(id, cbAutoresLibro.Text);

}

}

private void agregarAutoresLibro(string isbn, string autor) {

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("CreateLibroAutores", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Id\_Libro\_ISBN", isbn));

cmd.Parameters.Add(new SqlParameter("@Id\_Autor", autor));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Creacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Creacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Modificacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Modificacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Estatus", "1"));

cmd.ExecuteReader();

sql.Close();

}

}

catch (Exception ex)

{

throw;

}

}

private void btnQuitarLibroAutores\_Click(object sender, EventArgs e)

{

if (dgvAutoresLibro.SelectedRows.Count == 1)

{

dgvAutoresLibro.Rows.Remove(dgvAutoresLibro.SelectedRows[0]);

}

else { MessageBox.Show("Seleccione un Autor."); }

}

private void EliminaTodosAutoresLibro(string isbn)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

string consulta = "delete from Libro\_Autores where Id\_Libro\_ISBN='" + isbn + "'";

cmd = new SqlCommand(consulta, sql);

cmd.ExecuteReader();

sql.Close();

}

}

catch (Exception ex) { throw; }

}

private void HistorialToolStripMenuItem\_Click(object sender, EventArgs e)

{

cerrarPaneles();

PanelHistorial.Visible = true;

PanelHistorial.Enabled = true;

recargaHistorial();

}

private void btnAgregaAutor\_Click(object sender, EventArgs e)

{

PanelAutorAM.Visible = true;

PanelAutorAM.Enabled = true;

PanelAutorAM.BringToFront();

txtNombreAutor.Text = "";

txtApellidoAutor.Text = "";

dtNacioAutor.Value = DateTime.Now;

DataTable dt = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select Id\_Pais,Nacionalidad from Pais";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

cbNacionalidadAutor.DataSource = dt;

cbNacionalidadAutor.DisplayMember = "Nacionalidad";

cbNacionalidadAutor.ValueMember = "Id\_Pais";

chkAutor.Checked = true;

modAutor = "0";

}

private void btnModificaAutor\_Click(object sender, EventArgs e)

{

if (dgvAutor.SelectedRows.Count == 1)

{

DataTable dt = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select Id\_Pais,Nacionalidad from Pais";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

cbNacionalidadAutor.DataSource = dt;

cbNacionalidadAutor.DisplayMember = "Nacionalidad";

cbNacionalidadAutor.ValueMember = "Id\_Pais";

cbNacionalidadAutor.SelectedValue=(int)dgvAutor.SelectedCells[5].Value;

if ((bool)dgvAutor.SelectedCells[7].Value)

{

chkAutor.Checked = true;

}

else

{

chkAutor.Checked = false;

}

txtNombreAutor.Text = (string)dgvAutor.SelectedCells[1].Value; ;

txtApellidoAutor.Text = (string)dgvAutor.SelectedCells[2].Value;

dtNacioAutor.Value = (DateTime)dgvAutor.SelectedCells[4].Value;

modAutor = (string)dgvAutor.SelectedCells[0].Value;

PanelAutorAM.Visible = true;

PanelAutorAM.Enabled = true;

PanelAutorAM.BringToFront();

}

else { MessageBox.Show("Seleccione un Autor."); }

}

private void btnEliminaAutor\_Click(object sender, EventArgs e)

{

if (dgvAutor.SelectedRows.Count == 1)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("DeleteAutor", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Id", dgvAutor.SelectedCells[0].Value));

cmd.ExecuteReader();

LogsAceptable("Eliminar Autor", "Autor");

sql.Close();

MessageBox.Show("Eliminada");

recargaCatalogoAutor();

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else { MessageBox.Show("Seleccione un Autor."); }

}

private void btnAgregarModificarAutor\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(txtNombreAutor.Text) || string.IsNullOrWhiteSpace(txtNombreAutor.Text) || string.IsNullOrEmpty(txtApellidoAutor.Text) || string.IsNullOrWhiteSpace(txtApellidoAutor.Text) || string.IsNullOrEmpty(dtNacioAutor.Text) || string.IsNullOrWhiteSpace(dtNacioAutor.Text))

{

MessageBox.Show("Ingrese los datos solicitados");

}

else

{

string a = "0";

if (chkAutor.Checked)

{

a = "1";

}

if (modAutor.Equals("0"))

{

string f = null;

string p = null;

if (!(cbNacionalidadAutor.Text.Equals(""))) {

p = cbNacionalidadAutor.SelectedValue.ToString();

}

if (!(dtNacioAutor.Text.Equals(""))) {

f = dtNacioAutor.Value.ToShortDateString();

}

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("CreateAutor", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Id\_Autor", SiglaAutor(txtNombreAutor.Text,txtApellidoAutor.Text, dtNacioAutor.Value.ToString("ddMMyyyy"))));

cmd.Parameters.Add(new SqlParameter("@Nombre", txtNombreAutor.Text));

cmd.Parameters.Add(new SqlParameter("@Apellido", txtApellidoAutor.Text));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Nacimiento", f));

cmd.Parameters.Add(new SqlParameter("@Id\_Pais",p ));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Creacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Creacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Modificacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Modificacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Estatus", a));

cmd.ExecuteReader();

LogsAceptable("Insertar Autor", "Autor");

sql.Close();

MessageBox.Show("Guardado");

PanelAutorAM.Visible = false;

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else

{

try

{

string f = "NULL";

string p = "NULL";

if (!(cbNacionalidadAutor.Text.Equals("")))

{

p = cbNacionalidadAutor.SelectedValue.ToString();

}

if (!(dtNacioAutor.Text.Equals("")))

{

f ="'"+ dtNacioAutor.Value.ToShortDateString()+ "'";

}

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

//ciudad año inaugura

string consulta = "Update Autor set Nombre='" + txtNombreAutor.Text + "',Estatus=" + a + ",Id\_Usuario\_Modificacion=" + IDuser.ToString() + ",Apellido='" + txtApellidoAutor.Text + "',Fecha\_Nacimiento="+f+", Id\_Pais=" +p + ",Fecha\_Hora\_Modificacion= '" + DateTime.Now.ToString("yyyy/MM/dd HH:mm:ss.fff") + "' where Id\_Autor='" + modAutor + "'";

cmd = new SqlCommand(consulta, sql);

cmd.ExecuteReader();

LogsAceptable("Modificar Autor", "Autor");

sql.Close();

MessageBox.Show("Guardado");

PanelAutorAM.Visible = false;

}

}

catch (Exception ex) { MessageBox.Show("Error en conexion"); }

}

recargaCatalogoAutor();

}

}

private string SiglaAutor(string nombre,string apellido, string fecha\_nacimiento) {

string resul="";

resul = nombre.Substring(0, 1) + apellido.Substring(0,1) + fecha\_nacimiento;

return resul.ToUpper();

}

private void UsuariosToolStripMenuItem1\_Click(object sender, EventArgs e)

{

cerrarPaneles();

PanelUsuarios.Visible = true;

PanelUsuarios.Enabled = true;

recargaCatalogoUsuario();

}

private void ventasToolStripMenuItem\_Click(object sender, EventArgs e)

{

cerrarPaneles();

PanelVentas.Visible = true;

PanelVentas.Enabled = true;

recargaCatalogoVentas();

}

private void cajaToolStripMenuItem\_Click(object sender, EventArgs e)

{

cerrarPaneles();

panelVentaRegistro.Visible = true;

panelVentaRegistro.Enabled = true;

cargaCombosVentaLibros();

}

private void btnAgregaUsuario\_Click(object sender, EventArgs e)

{

cargacombosUsuarios();

panelUsuarioAM.Visible = true;

panelUsuarioAM.Enabled = true;

panelUsuarioAM.BringToFront();

txtNombreUsuarioUAM.Text = "";

txtNombreUAM.Text = "";

txtContraseñaUAM.Text = "";

txtApellidoPatUAM.Text = "";

txtApellidoMatUAM.Text = "";

txtCorreoUAM.Text = "";

txtTelefonoUAM.Text = "";

chkUsuarioUAM.Checked = true;

modUsuario = 0;

modcon = "";

}

private void btnModificaUsuario\_Click(object sender, EventArgs e)

{

if (dgvUsuarios.SelectedRows.Count == 1)

{

cargacombosUsuarios();

txtNombreUsuarioUAM.Text = (string)dgvUsuarios.SelectedCells[1].Value;

txtNombreUAM.Text = (string)dgvUsuarios.SelectedCells[2].Value;

txtContraseñaUAM.Text = (string)dgvUsuarios.SelectedCells[3].Value;

modcon = txtContraseñaUAM.Text;

txtApellidoPatUAM.Text = (string)dgvUsuarios.SelectedCells[4].Value;

txtApellidoMatUAM.Text = (string)dgvUsuarios.SelectedCells[5].Value;

txtCorreoUAM.Text = (string)dgvUsuarios.SelectedCells[6].Value;

txtTelefonoUAM.Text = (string)dgvUsuarios.SelectedCells[7].Value;

cbPuestoUAM.SelectedValue = (int)dgvUsuarios.SelectedCells[8].Value;

cbDepartamentoUAM.SelectedValue = (int)dgvUsuarios.SelectedCells[10].Value;

cbTipoUsuarioUAM.SelectedValue = (int)dgvUsuarios.SelectedCells[12].Value;

if ((bool)dgvUsuarios.SelectedCells[14].Value)

{

chkUsuarioUAM.Checked = true;

}

else

{

chkUsuarioUAM.Checked = false;

}

modUsuario = (int)dgvUsuarios.SelectedCells[0].Value;

panelUsuarioAM.Visible = true;

panelUsuarioAM.Enabled = true;

panelUsuarioAM.BringToFront();

}

else { MessageBox.Show("Seleccione un Usuario."); }

}

private void btnEliminaUsuario\_Click(object sender, EventArgs e)

{

if (dgvUsuarios.SelectedRows.Count == 1)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("DeleteUsuario", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Id", dgvUsuarios.SelectedCells[0].Value));

cmd.ExecuteReader();

LogsAceptable("Eliminar Usuario", "Usuario");

sql.Close();

MessageBox.Show("Eliminada");

recargaCatalogoUsuario();

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else { MessageBox.Show("Seleccione un Usuario."); }

}

private void btnAgregarModificarUsuario\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(txtNombreUAM.Text) || string.IsNullOrWhiteSpace(txtNombreUAM.Text)

|| string.IsNullOrEmpty(txtNombreUsuarioUAM.Text) || string.IsNullOrWhiteSpace(txtNombreUsuarioUAM.Text)

|| string.IsNullOrEmpty(txtApellidoPatUAM.Text) || string.IsNullOrWhiteSpace(txtApellidoPatUAM.Text)

|| string.IsNullOrEmpty(txtCorreoUAM.Text) || string.IsNullOrWhiteSpace(txtCorreoUAM.Text)

|| string.IsNullOrEmpty(cbPuestoUAM.Text) || string.IsNullOrWhiteSpace(cbPuestoUAM.Text)

|| string.IsNullOrEmpty(cbDepartamentoUAM.Text) || string.IsNullOrWhiteSpace(cbDepartamentoUAM.Text)

|| string.IsNullOrEmpty(cbTipoUsuarioUAM.Text) || string.IsNullOrWhiteSpace(cbTipoUsuarioUAM.Text)

|| ((string.IsNullOrEmpty(txtContraseñaUAM.Text) || string.IsNullOrWhiteSpace(txtContraseñaUAM.Text))

&& (string.IsNullOrEmpty(modcon) || string.IsNullOrWhiteSpace(modcon))))

{

MessageBox.Show("Ingrese los datos solicitados");

}

else

{

if (!(string.IsNullOrEmpty(txtContraseñaUAM.Text) || string.IsNullOrWhiteSpace(txtContraseñaUAM.Text)) && !(txtContraseñaUAM.Text.Equals(modcon)))

{

modcon = Encriptado(txtContraseñaUAM.Text);

}

string mater = "NULL";

string telef = "NULL";

string a = "0";

if (chkUsuarioUAM.Checked)

{

a = "1";

}

if (modUsuario == 0)

{

try

{

if (!(string.IsNullOrEmpty(txtApellidoMatUAM.Text) || string.IsNullOrWhiteSpace(txtApellidoMatUAM.Text)))

{

mater =txtApellidoMatUAM.Text;

}

if (!(string.IsNullOrEmpty(txtTelefonoUAM.Text) || string.IsNullOrWhiteSpace(txtTelefonoUAM.Text)))

{

telef = txtTelefonoUAM.Text;

}

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("CreateTipoCliente", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Nombre", txtNombreUAM.Text));

cmd.Parameters.Add(new SqlParameter("@Contraseña", modcon));

cmd.Parameters.Add(new SqlParameter("@Nombre\_Usuario", txtNombreUsuarioUAM.Text));

cmd.Parameters.Add(new SqlParameter("@Apellido\_Paterno", txtApellidoPatUAM.Text));

cmd.Parameters.Add(new SqlParameter("@Apellido\_Materno", mater));

cmd.Parameters.Add(new SqlParameter("@Email", txtCorreoUAM.Text));

cmd.Parameters.Add(new SqlParameter("@Telefono",telef ));

cmd.Parameters.Add(new SqlParameter("@Id\_Puesto", cbPuestoUAM.SelectedValue));

cmd.Parameters.Add(new SqlParameter("@Id\_Departamento", cbDepartamentoUAM.SelectedValue));

cmd.Parameters.Add(new SqlParameter("@Id\_Tipo\_usuario", cbTipoUsuarioUAM.SelectedValue));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Creacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Creacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Modificacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Modificacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Estatus", a));

cmd.ExecuteReader();

LogsAceptable("Insertar Usuario", "Usuario");

sql.Close();

MessageBox.Show("Guardado");

panelUsuarioAM.Visible = false;

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else

{

try

{

if (!(string.IsNullOrEmpty(txtApellidoMatUAM.Text) || string.IsNullOrWhiteSpace(txtApellidoMatUAM.Text)))

{

mater = "'" + txtApellidoMatUAM.Text + "'";

}

if (!(string.IsNullOrEmpty(txtTelefonoUAM.Text) || string.IsNullOrWhiteSpace(txtTelefonoUAM.Text)))

{

telef = "'" + txtTelefonoUAM.Text + "'";

}

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

string consulta = "update Usuario set Nombre\_Usuario='" + txtNombreUsuarioUAM.Text + "',Nombre='" + txtNombreUAM.Text + "'," +

"Contraseña='" + modcon + "',Apellido\_Paterno='" + txtApellidoPatUAM.Text + "',Apellido\_Materno=" + mater +

",Email='" + txtCorreoUAM.Text + "',Telefono=" + telef + ",Id\_Usuario\_Modificacion=" + IDuser.ToString() +

",Fecha\_Hora\_Modificacion= '" + DateTime.Now.ToString("yyyy/MM/dd HH:mm:ss.fff") +

"',Estatus="+a+ ",Id\_Puesto="+ cbPuestoUAM.SelectedValue.ToString() + ",Id\_Departamento="+ cbDepartamentoUAM.SelectedValue.ToString()+

",Id\_Tipo\_Usuario="+ cbTipoUsuarioUAM.SelectedValue.ToString()+ " where Id\_Usuario=" + modUsuario.ToString();

cmd = new SqlCommand(consulta, sql);

cmd.ExecuteReader();

LogsAceptable("Modificar Usuario", "Usuario");

sql.Close();

MessageBox.Show("Guardado");

panelUsuarioAM.Visible = false;

}

}

catch (Exception ex) { MessageBox.Show("Error en conexion"); }

}

recargaCatalogoUsuario();

}

}

private void btnVerVenta\_Click(object sender, EventArgs e)

{

if (dgvVerVentas.SelectedRows.Count == 1)

{

cargaDetalleVenta((int)dgvVerVentas.SelectedCells[0].Value);

PanelDetalleVenta.Visible = true;

PanelDetalleVenta.Enabled = true;

PanelDetalleVenta.BringToFront();

}

else { MessageBox.Show("Seleccione una Venta."); }

}

private void btnDetalleVentaRegresar\_Click(object sender, EventArgs e)

{

PanelDetalleVenta.Visible = false;

}

private void cargaDetalleVenta(int venta)

{

DataTable dt = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select l.Titulo,(de.Porcentaje\*100) Descuento,l.Precio from Detalle\_Venta d left join Libro l on d.Id\_Libro\_ISBN=l.Id\_Libro\_ISBN left join Descuento de on d.Id\_Descuento= de.id\_Descuento where d.Id\_Venta="+venta.ToString();

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

dgvDetalleVenta.DataSource = dt;

}

private void recargaCatalogoVentas()

{

DataTable dt = new DataTable();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

using (SqlCommand cmd = new SqlCommand())

{

cmd.Connection = sql;

cmd.CommandType = CommandType.Text;

cmd.CommandText = "select v.Id\_Venta Venta,v.Fecha\_Hora\_Creacion Fecha, u.Nombre\_Usuario,c.Nombre Cliente,p.Nombre [Forma de Pago] from venta v left join Tipo\_Cliente c on v.Id\_Tipo\_Cliente=c.Id\_Tipo\_Cliente left join Usuario u on v.Id\_Usuario\_Creacion=u.Id\_Usuario left join Tipo\_Pago p on v.Id\_Tipo\_Pago=p.Id\_Tipo\_Pago where v.Estatus=1";

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt);

}

}

sql.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

dgvVerVentas.DataSource = dt;

}

private void btnEliminaVenta\_Click(object sender, EventArgs e)

{

if (dgvVerVentas.SelectedRows.Count == 1)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("DeleteVenta", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Id", dgvVerVentas.SelectedCells[0].Value));

cmd.ExecuteReader();

LogsAceptable("Eliminar Venta", "Venta");

sql.Close();

MessageBox.Show("Eliminada");

recargaCatalogoVentas();

}

}

catch (Exception ex)

{

MessageBox.Show("Error de conexion");

}

}

else { MessageBox.Show("Seleccione una Venta."); }

}

private void btnBuscaCaja\_Click(object sender, EventArgs e)

{

bool existe = false;

string id = cbLibrosCaja.SelectedValue.ToString();

for (int i = 0; i < dgvCaja.Rows.Count; i++)

{

if ((dgvCaja.Rows[i].Cells["ISBN"].Value.ToString()).Equals(id))

{

existe = true;

}

}

if (!existe)

{

string ti = "";

decimal pre = 0;

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

string consulta = "select l.Id\_Libro\_ISBN ISBN,CONCAT(l.Titulo,', ',f.Nombre) Titulo,l.Precio from Libro l left join Tipo\_Formato f on l.Id\_Tipo\_Formato=f.Id\_Tipo\_Formato where l.Id\_Libro\_ISBN='" + id+"'";

cmd = new SqlCommand(consulta, sql);

SqlDataReader lector;

lector = cmd.ExecuteReader();

if (lector.HasRows)

{

while (lector.Read())

{

ti = (string)lector[1];

pre = (decimal)lector[2];

}

}

}

}

catch (Exception ex) { MessageBox.Show("Error en conexion"); }

dgvCaja.Rows.Add(id, ti,pre,"0");

}

}

private void button1\_Click(object sender, EventArgs e)

{

if (dgvCaja.SelectedRows.Count == 1)

{

dgvCaja.Rows.Remove(dgvCaja.SelectedRows[0]);

}

else { MessageBox.Show("Seleccione un Libro."); }

}

private void btnLimpiaCaja\_Click(object sender, EventArgs e)

{

dgvCaja.Rows.Clear();

}

private void btnRegistraVentaCaja\_Click(object sender, EventArgs e)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("CreateVenta", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Id\_Tipo\_Cliente", cbClienteCaja.SelectedValue));

cmd.Parameters.Add(new SqlParameter("@Id\_Tipo\_Pago", cbPagoCaja.SelectedValue));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Creacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Creacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Modificacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Modificacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Estatus", "1"));

cmd.ExecuteReader();

sql.Close();

}

LogsAceptable("Insertar Venta", "Venta");

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

string consulta = "select top(1) id\_Venta from Venta order by Id\_Venta desc";

cmd = new SqlCommand(consulta, sql);

SqlDataReader lector;

lector = cmd.ExecuteReader();

if (lector.HasRows)

{

while (lector.Read())

{

modVenta = (int)lector[0];

}

}

}

}

catch (Exception ex) { MessageBox.Show("Error en conexion"); }

for (int i = 0; i < dgvCaja.Rows.Count; i++)

{

if (!(dgvCaja.Rows[i].Cells["ISBN"].Value.ToString()).Equals(""))

{

string d = "0";

if (string.IsNullOrEmpty(dgvCaja.Rows[i].Cells["Descuento"].Value.ToString()) || string.IsNullOrWhiteSpace(dgvCaja.Rows[i].Cells["Descuento"].Value.ToString()))

{

d = "0";

}

else {

d = dgvCaja.Rows[i].Cells["Descuento"].Value.ToString();

}

agregarDetalleVentaLibros(dgvCaja.Rows[i].Cells["ISBN"].Value.ToString(),d);

LogsAceptable("Insertar Detalle", "Detalle\_Venta");

}

}

MessageBox.Show("Guardado");

}

catch (Exception ex)

{

MessageBox.Show("Error en conexion");

}

}

private void agregarDetalleVentaLibros(string isbn, string descuento)

{

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

cmd = new SqlCommand("CreateDetalleVenta", sql);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter("@Id\_Venta", modVenta));

cmd.Parameters.Add(new SqlParameter("@Id\_Descuento", descuento));

cmd.Parameters.Add(new SqlParameter("@Id\_Libro\_ISBN", isbn));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Creacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Creacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Id\_Usuario\_Modificacion", IDuser));

cmd.Parameters.Add(new SqlParameter("@Fecha\_Hora\_Modificacion", DateTime.Now));

cmd.Parameters.Add(new SqlParameter("@Estatus", "1"));

cmd.ExecuteReader();

sql.Close();

}

}

catch (Exception ex)

{

throw;

}

}

private void MiCuentaToolStripMenuItem\_Click(object sender, EventArgs e)

{

int id = 0;

cerrarPaneles();

try

{

using (SqlConnection sql = new SqlConnection("Data Source=DESKTOP-CUOAPA9\\SQLEXPRESS;Initial Catalog=libreria;Integrated Security=True"))

{

sql.Open();

string consulta = "select Nombre\_Usuario,Nombre,Contraseña,Apellido\_Paterno,Apellido\_Materno,Email,Telefono,Puesto,Departamento from VRegistroUsuarios where Id\_Usuario=" + IDuser.ToString();

cmd = new SqlCommand(consulta, sql);

SqlDataReader lector;

lector = cmd.ExecuteReader();

if (lector.HasRows)

{

while (lector.Read())

{

txtUsuarioMiCuenta.Text = (string)lector[0];

txtNombreMiCuenta.Text = (string)lector[1];

txtContraMiCuenta.Text = (string)lector[2];

txtApellidoPMiCuenta.Text = (string)lector[3];

txtApellidoMMiCuenta.Text = (string)lector[4];

txtCorreoMiCuenta.Text = (string)lector[5];

txtTelefMiCuenta.Text = (string)lector[6];

txtPuestoMiCuenta.Text = (string)lector[7];

txtDepartMiCuenta.Text = (string)lector[8];

con = (string)lector[2];

}

PanelRegistra.Visible = true;

PanelRegistra.Enabled = true;

}

else

{

PanelRegistra.Visible = false;

PanelRegistra.Enabled = false;

MessageBox.Show("El usuario no existe o datos incorrectos");

}

sql.Close();

}

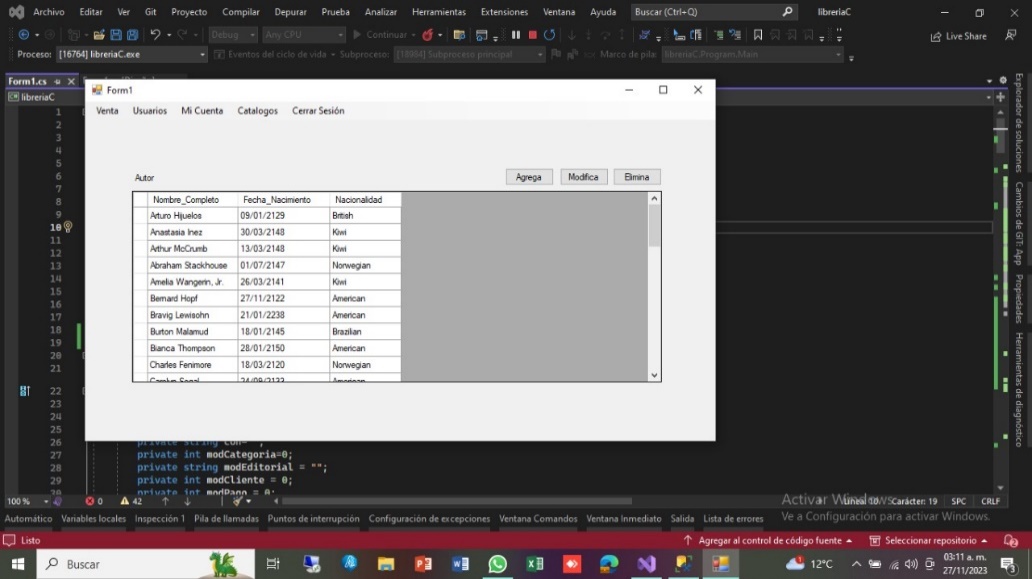
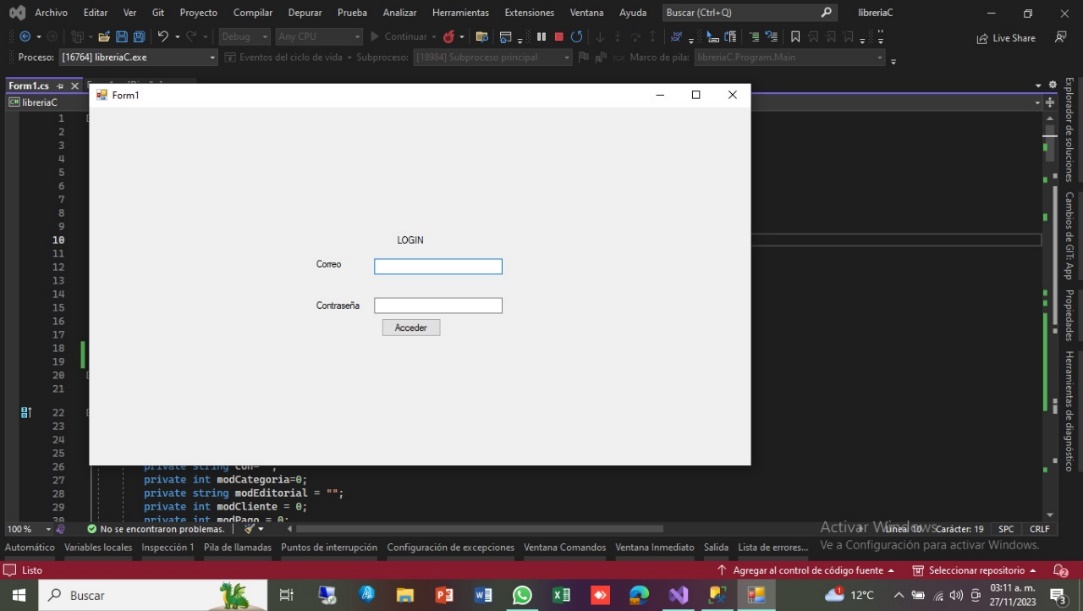
}

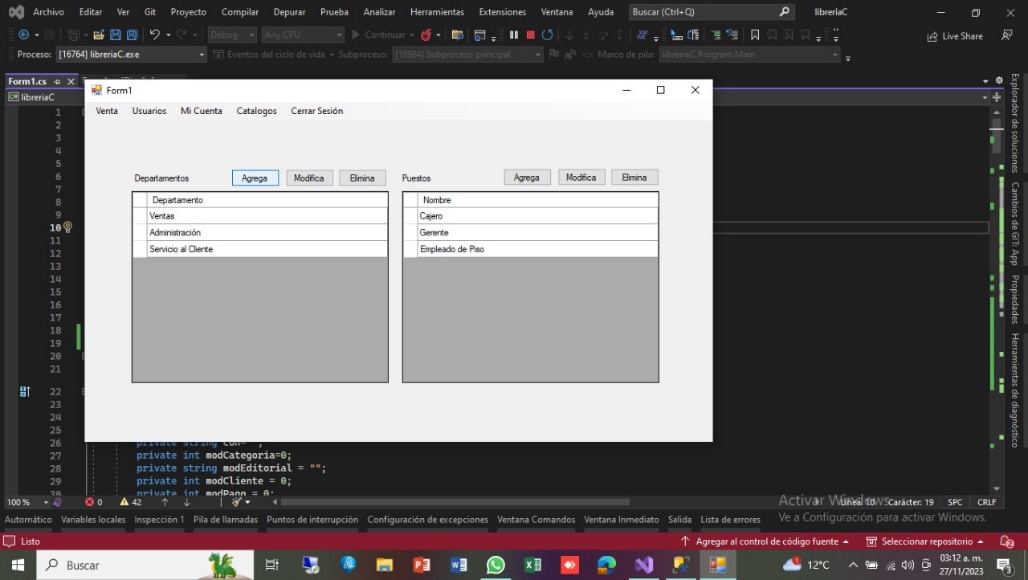
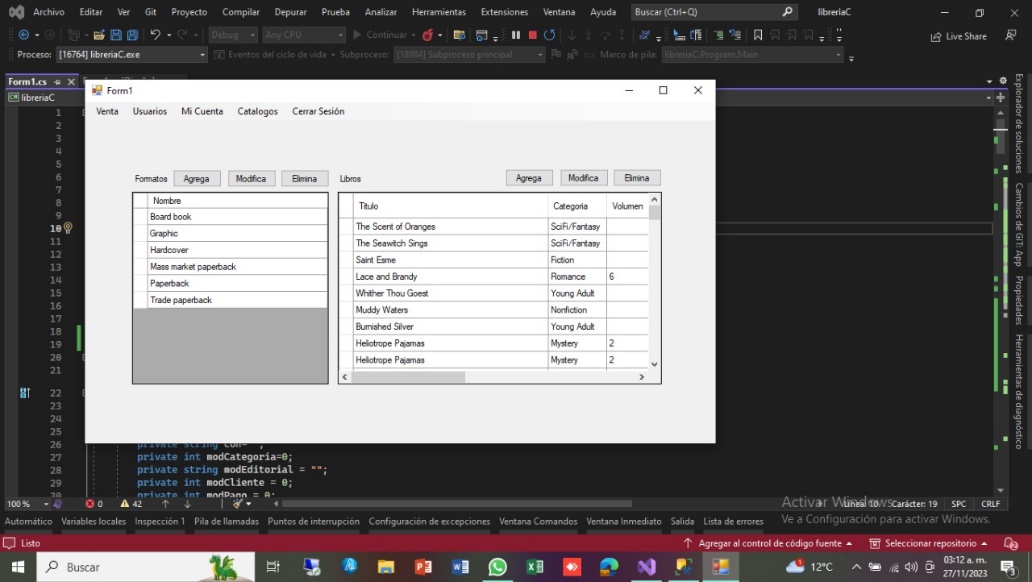
catch (Exception ex) { MessageBox.Show("Error en conexion"); }

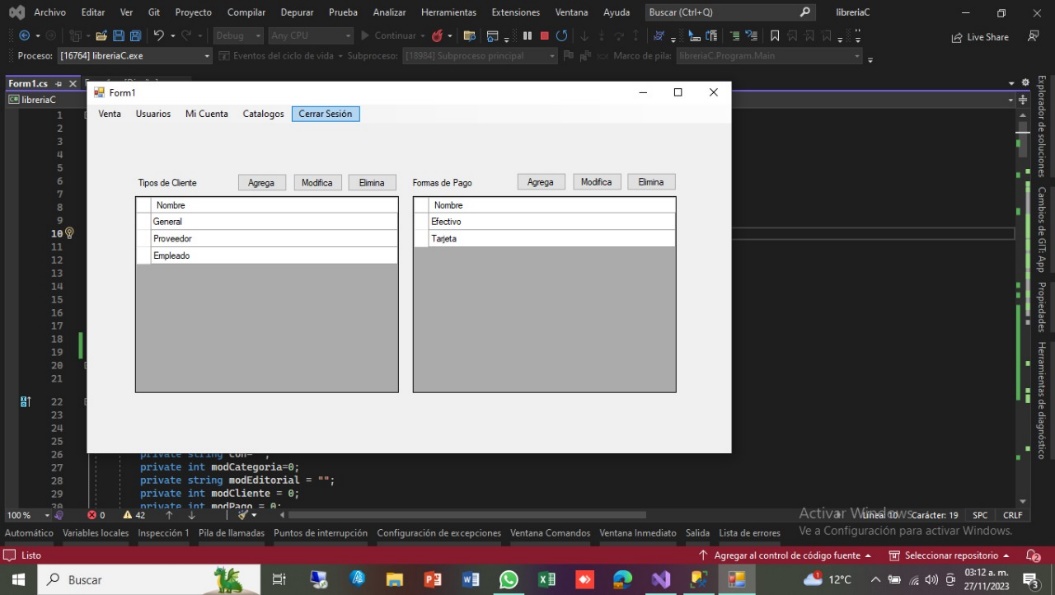
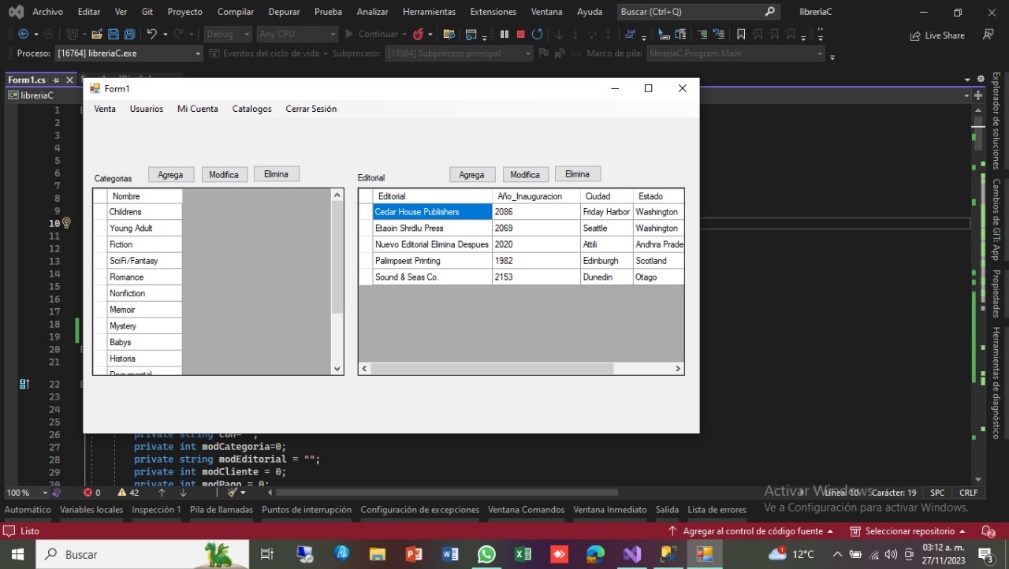
}

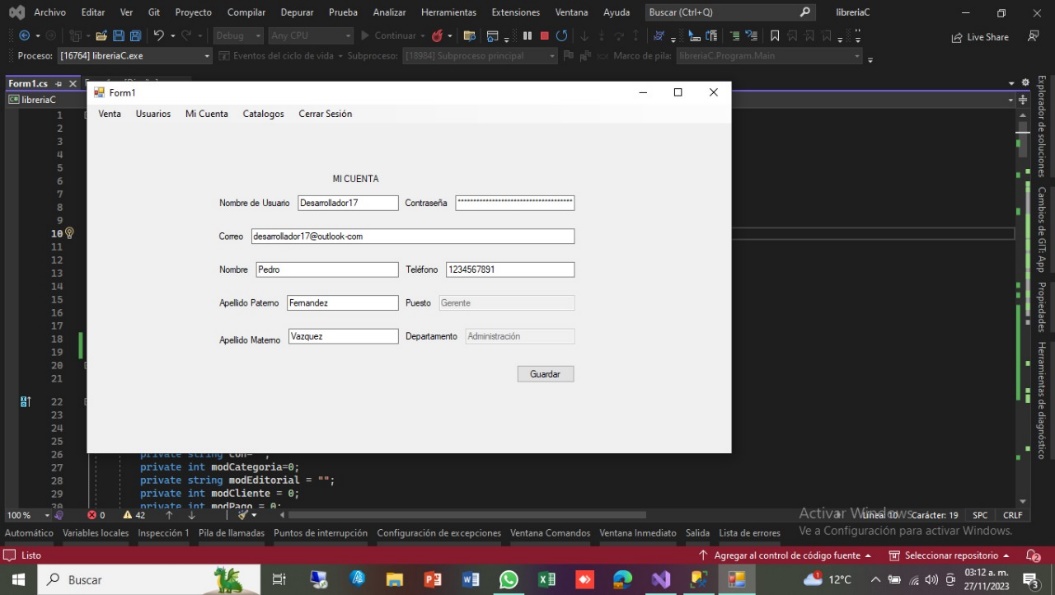
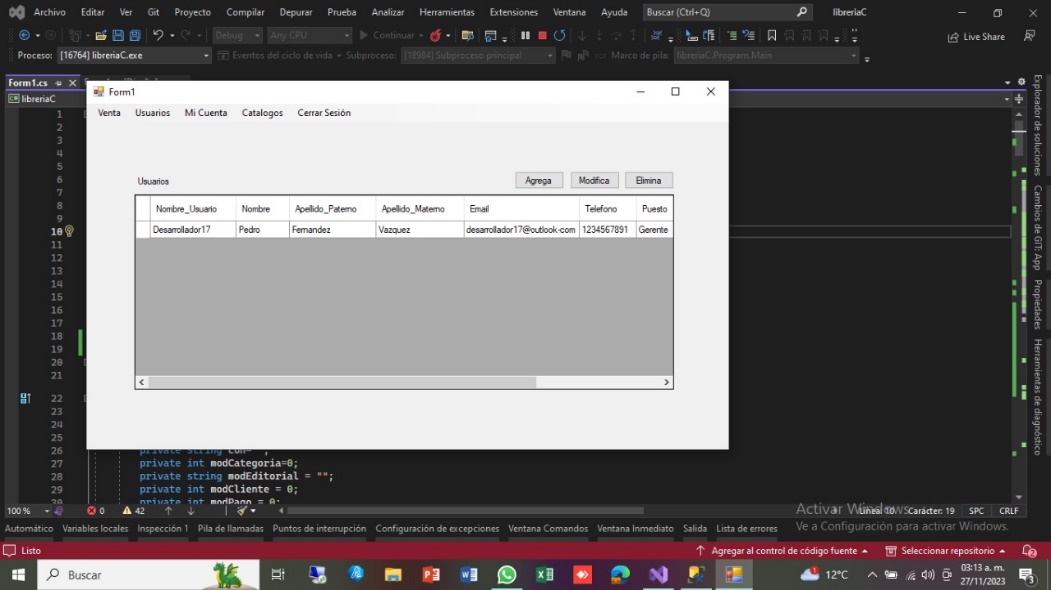
}

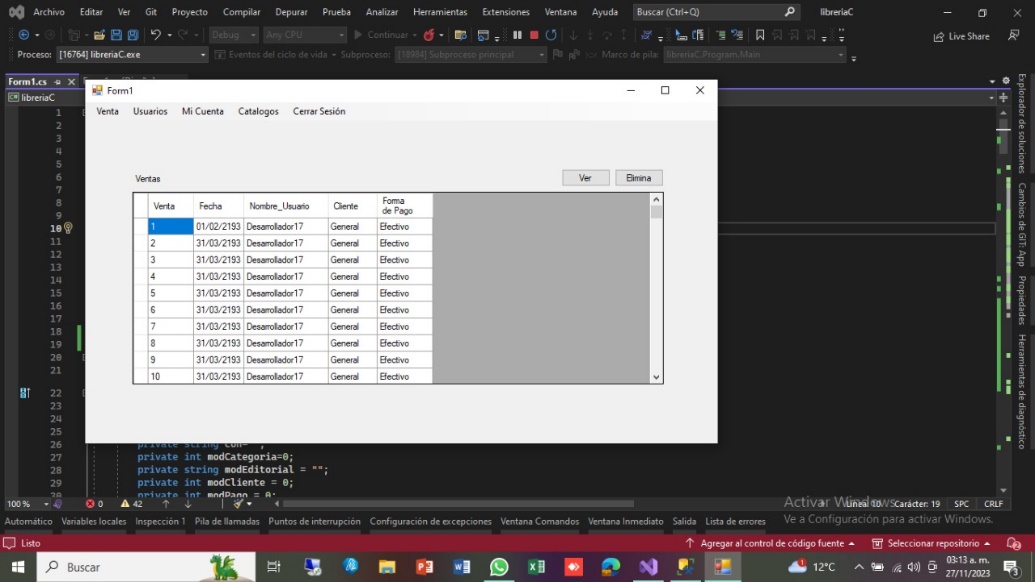
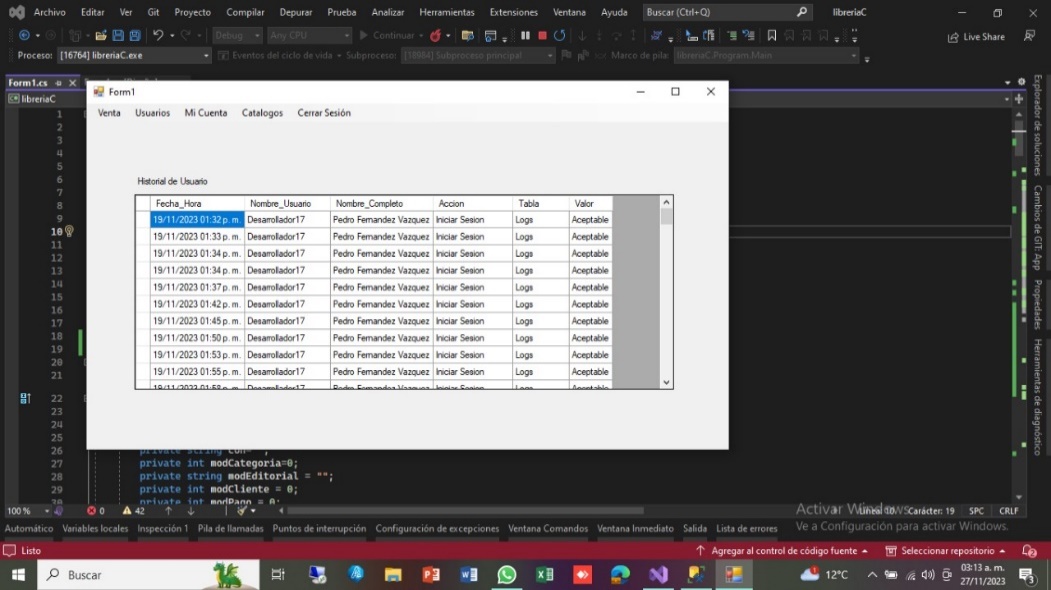
}

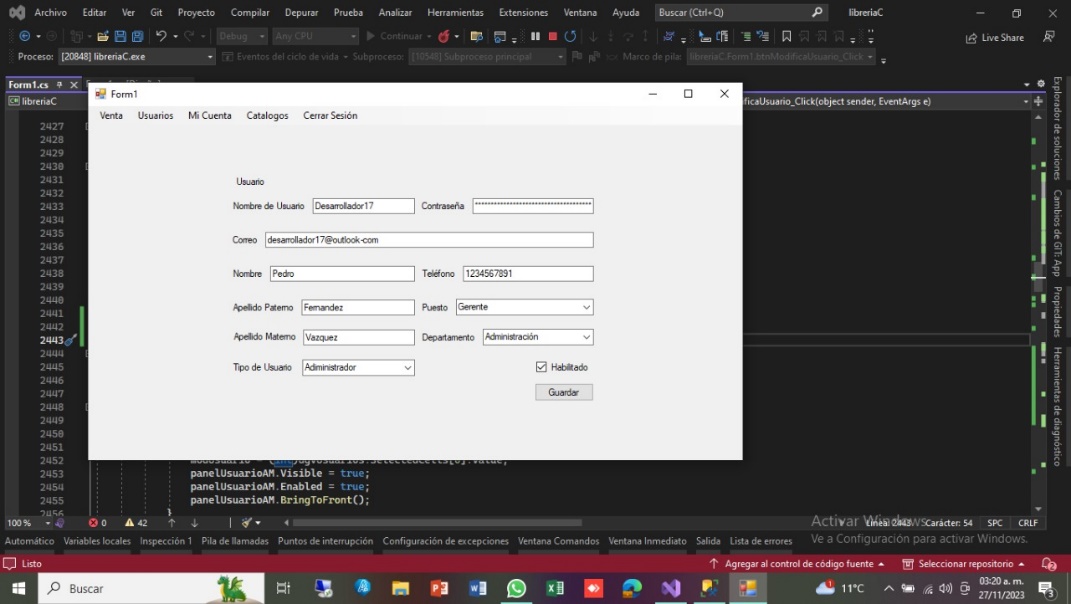
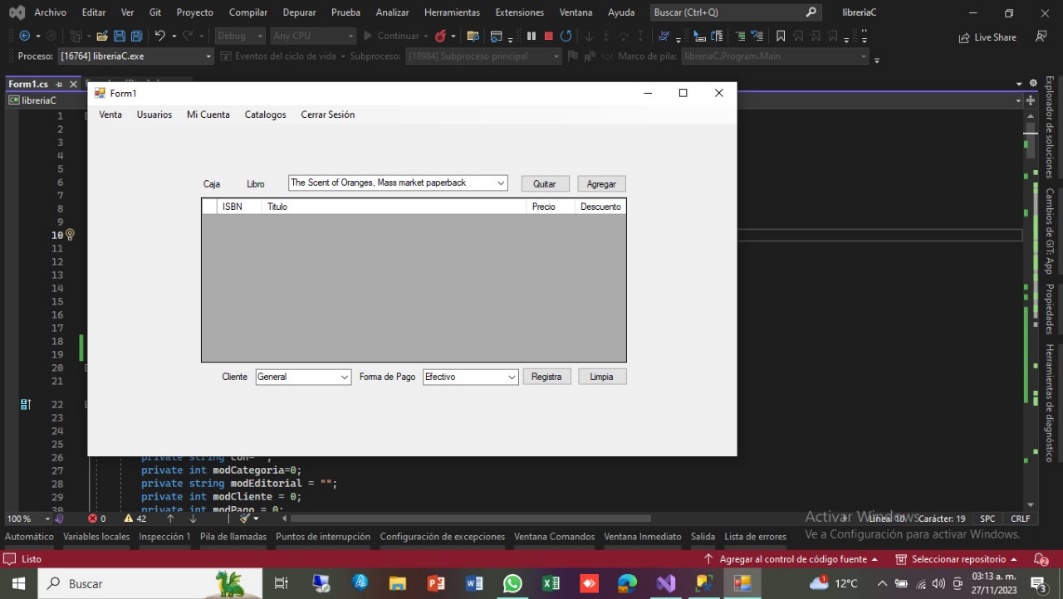


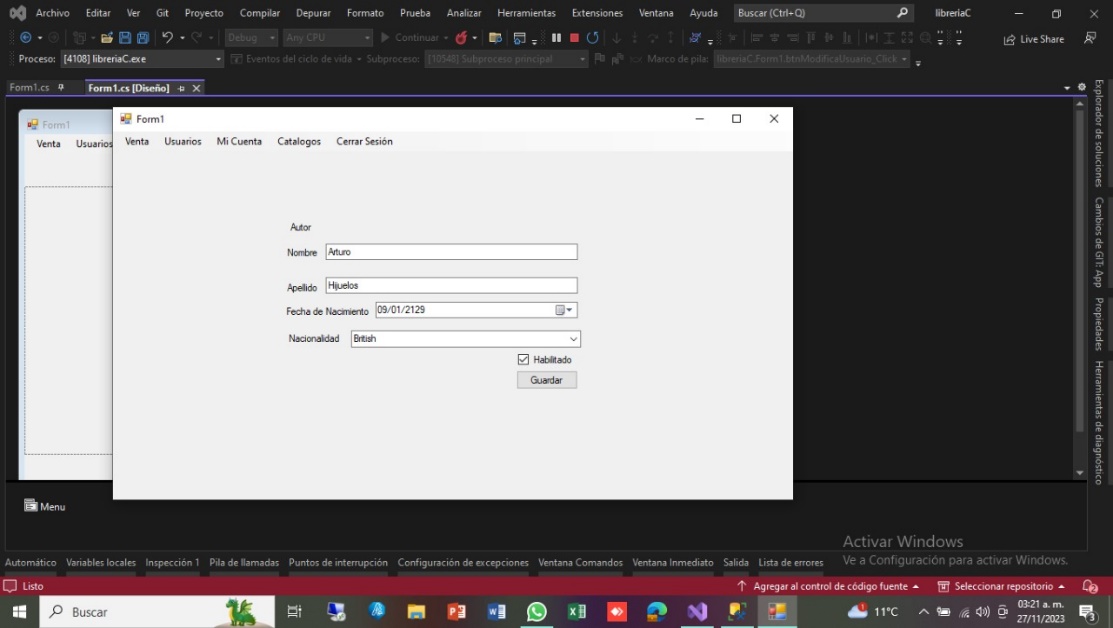
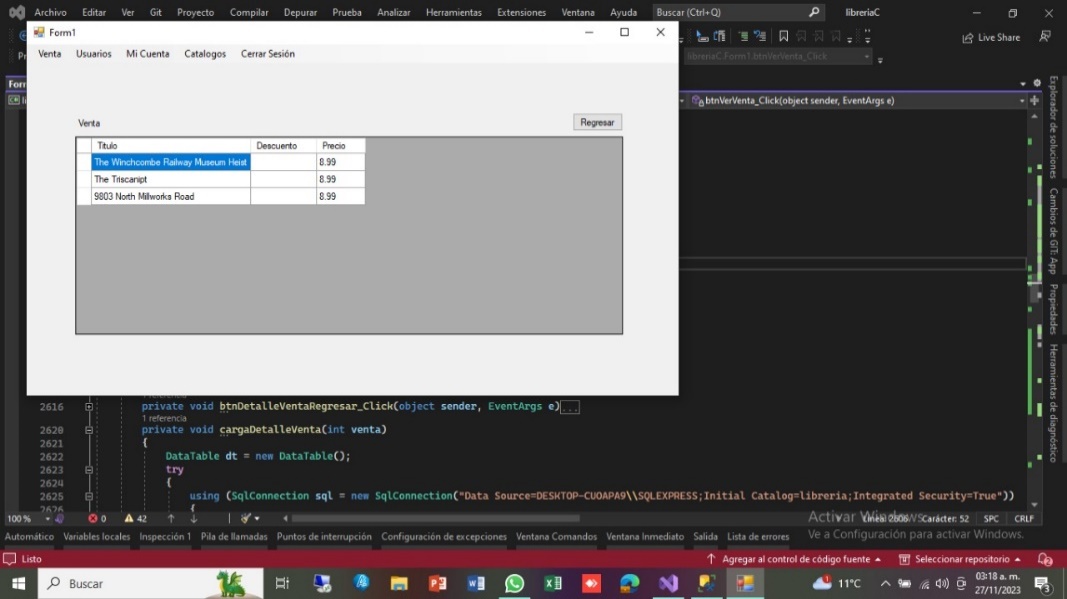


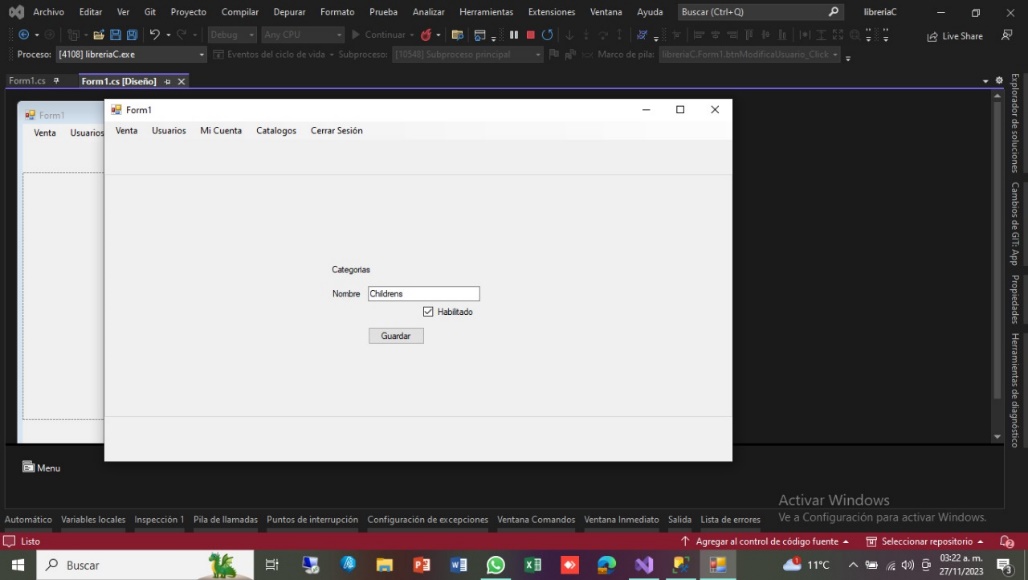
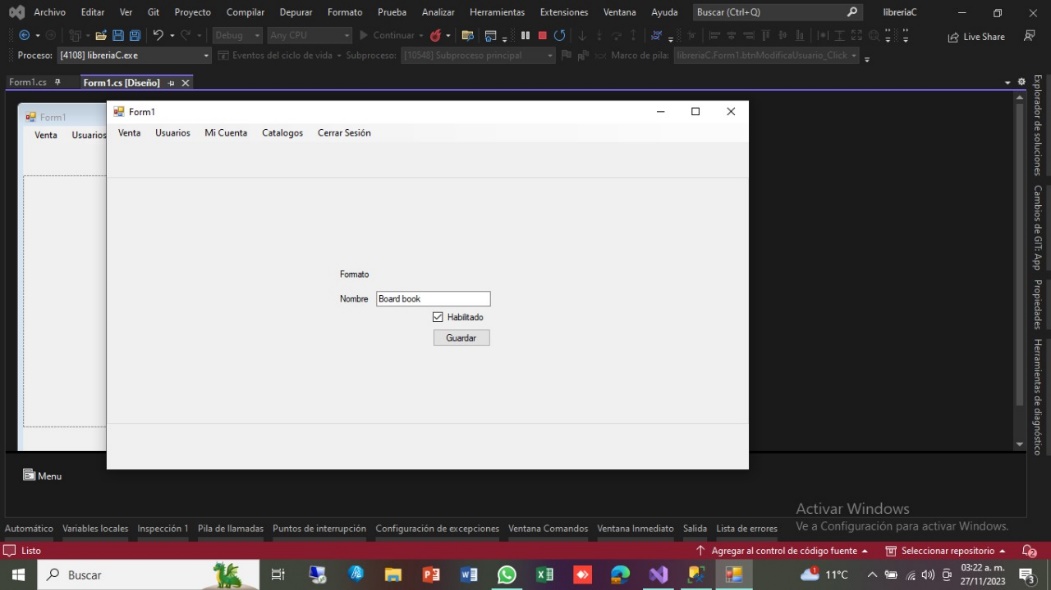


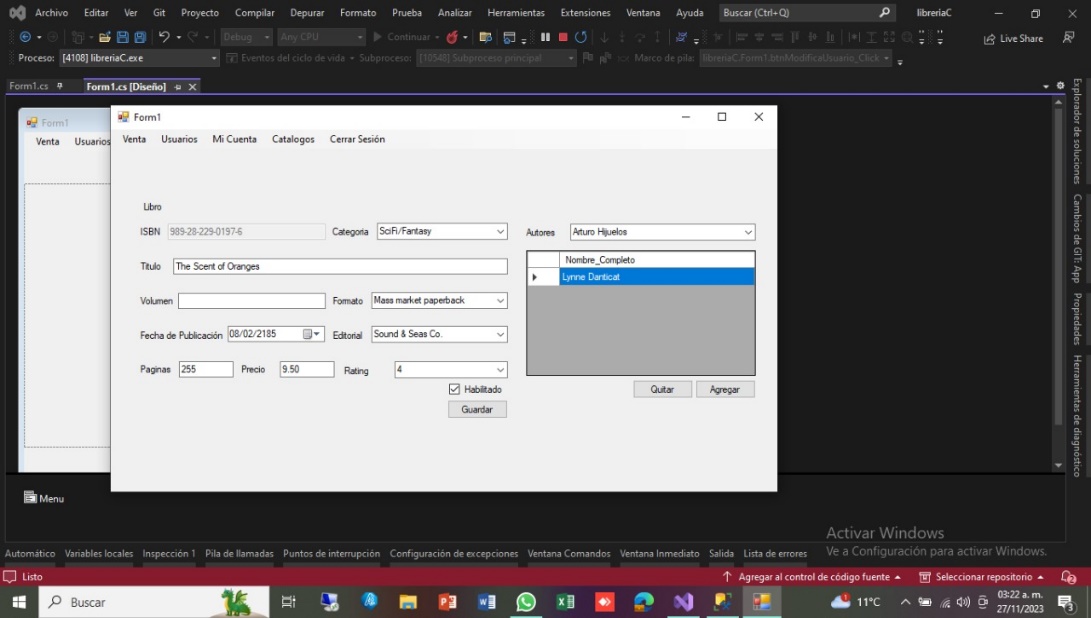


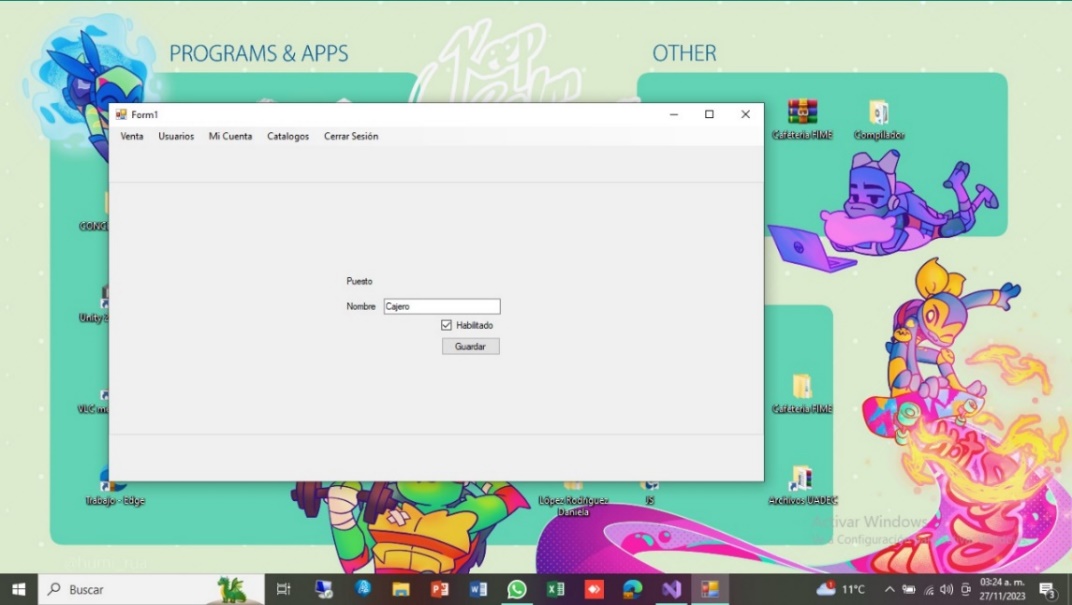
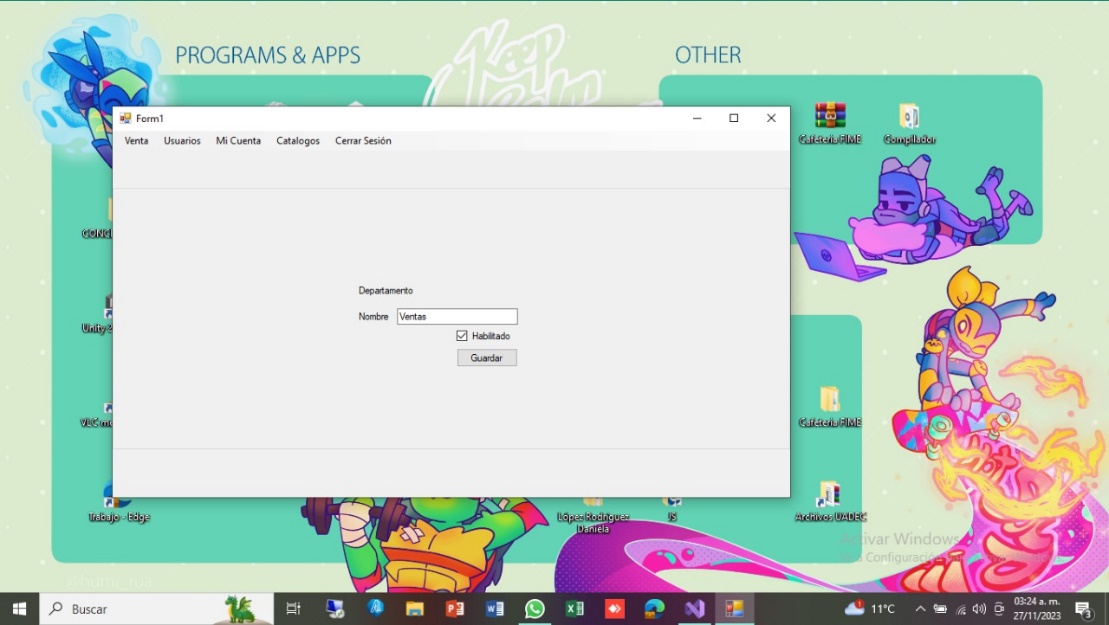


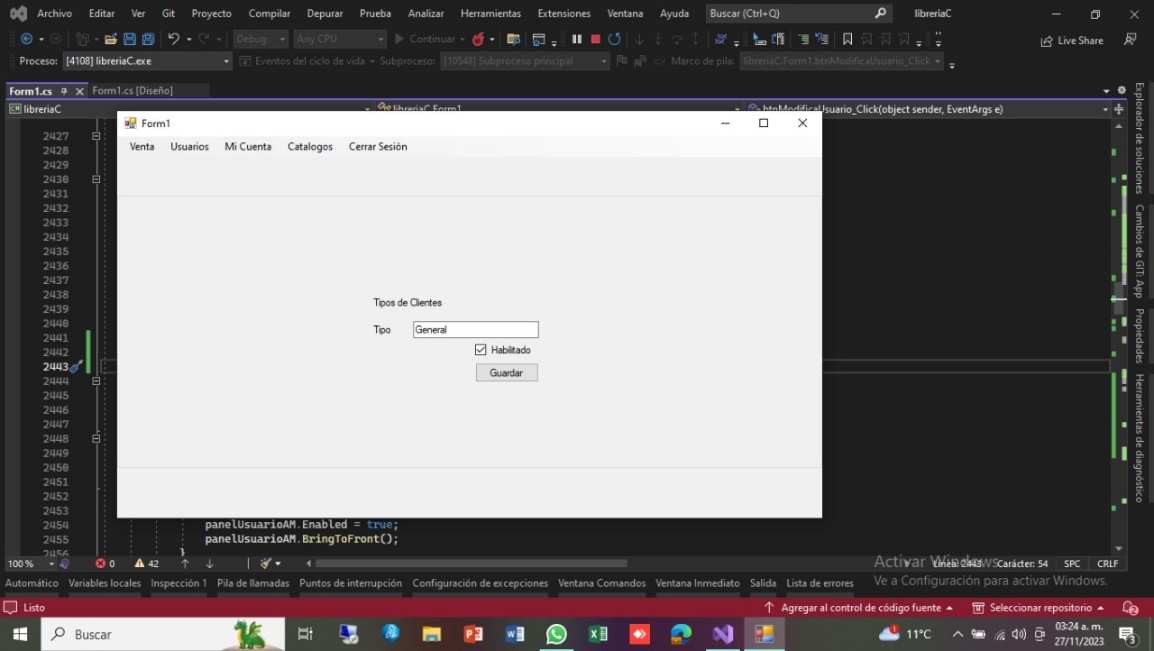
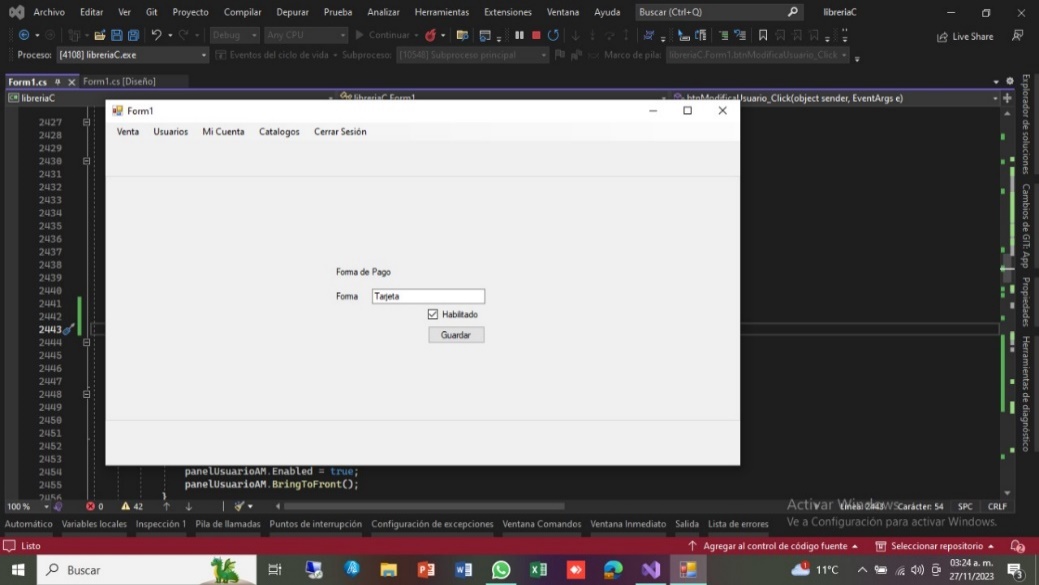












### HTML

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Libreria</title>

<style>

\*{

font-family: Arial, Helvetica, sans-serif;

}

.registro{

background-color: #333;

width: 30%;

height: 450px;

margin: auto;

margin-top: 50px;

padding: 30px;

border: 20px solid #222;

display: flex;

flex-direction: column;

justify-content: space-between;

}

.inicio\_sesion{

background-color: #333;

width: 30%;

height: 300px;

margin: auto;

margin-top: 50px;

padding: 30px;

border: 20px solid #222;

display: flex;

flex-direction: column;

justify-content: space-between;

}

.two-columns{

display: flex;

justify-content: space-between;

}

.fields{

display: flex;

flex-direction: column;

}

label{

margin-bottom: 10px;

text-transform: uppercase;

font-weight: 700;

color: #dedede;

font-size: 14px;

}

input,textarea{

background-color: transparent;

border: none;

border-bottom: 2px solid rgba(255, 255, 255, 0.5);

padding: 10px;

color: #fff;

transition: all 0.3s;

}

input:focus,textarea:focus{

outline: none;

}

input:focus{

border-bottom: 2px solid rgba(255, 255, 255, 1);

}

textarea{

resize: none;

border: 2px solid rgba(255, 255, 255, 0.5);

flex-basis: 100px;

transition: all 0.3s;

}

textarea:focus{

border: 2px solid rgba(255, 255, 255, 1);

}

.btn{

display: flex;

justify-content: center;

}

button{

padding: 15px 40px;

background: #1baea990;

border: none;

font-size: 15px;

font-weight: 700px;

text-transform: uppercase;

color: #fff;

cursor: pointer;

transition: all 0.3;

}

button:hover{

padding: 15px 70px;

background: #1baea9;

}

.tablas{

color: black;

margin-left: auto;

margin-right: auto;

}

button .edicion{

justify-content: center;

text-align:center;

padding: 10px 30px;

background: #1baea990;

border: none;

font-size: 15px;

font-weight: 100px;

text-transform: uppercase;

color: #fff;

}

</style>

</head>

<body>

<div class="two-columns">

<form id="inicio\_sesion" action="" class="inicio\_sesion">

<label >Inicio Sesion</label>

<div >

<div class="fields">

<label for="correo">Correo</label>

<input id="correo" type="email" name="correo" autocomplete="off">

</div>

<div class="fields">

<label for="pasword">Contraseña</label>

<input id="pasword" type="text" name="pasword" autocomplete="off" >

</div>

</div>

<div class="two-columns">

<div class="btn">

<button>Entrar</button>

</div>

<div class="btn">

<button>Registro</button>

</div>

</div>

</form>

<form id="registro" action="" class="registro">

<label >Registro Usuario</label>

<div class="two-columns">

<div class="fields">

<label for="name\_user">Nombre de Usuario</label>

<input id="name\_uset" type="text" name="name\_user" autocomplete="off">

</div>

<div class="fields">

<label for="pasword">Contraseña</label>

<input id="pasword" type="text" name="pasword" autocomplete="off" >

</div>

</div>

<div class="two-columns">

<div class="fields">

<label for="name\_user">Nombre</label>

<input id="name" type="text" name="name" autocomplete="off">

</div>

<div class="fields">

<label for="lastname1">Apellido Paterno</label>

<input id="lastname1" type="text" name="lastname1" autocomplete="off">

</div>

</div>

<div class="two-columns">

<div class="fields">

<label for="lastname2">Apellido Materno</label>

<input id="lastname2" type="text" name="lastname2" autocomplete="off">

</div>

<div class="fields">

<label for="email">Email</label>

<input id="email" type="text" name="email" autocomplete="off">

</div>

</div>

<div class="two-columns">

<div class="fields">

<label for="tel">Telefono</label>

<input id="tel" type="text" name="tel" autocomplete="off">

</div>

<div class="fields">

<label for="puesto">Puesto</label>

<input id="puesto" type="text" name="puesto" autocomplete="off">

</div>

</div>

<div class="two-columns">

<div class="fields">

<label for="tel">Departamento</label>

<input id="dep" type="text" name="dep" autocomplete="off">

</div>

<div class="fields">

<label for="tipo\_User">Tipo Usuario</label>

<input id="tipo\_User" type="text" name="tipo\_User" autocomplete="off">

</div>

</div>

<div class="btn">

<button>Registrar</button>

</div>

</form>

</div>

<br>

<br>

<br>

<div >

<label class="tablas">Ventas</label>

<table border="1" class="tablas">

<tr>

<th>Id\_Venta</th>

<th>Id\_Tipo\_Cliente</th>

<th>Id\_Tipo\_Pago</th>

</tr>

<tr>

<td><br></td>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

<td><br></td>

</tr>

</table>

<br>

<button class="edicion">Crear</button>

<button class="edicion">Editar</button>

<button class="edicion">Eliminar</button>

</div>

<br>

<br>

<br>

<div>

<label class="tablas">Detalle Ventas</label>

<table border="1" class="tablas">

<tr>

<th>Id\_Detalle</th>

<th>Id\_Tipo\_Cliente</th>

<th>Id\_Descuento</th>

<th>Id\_Libro\_ISBN</th>

</tr>

<tr>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

</tr>

</table>

<br>

<button class="edicion">Crear</button>

<button class="edicion">Editar</button>

<button class="edicion">Eliminar</button>

</div>

<br>

<br>

<br>

<div>

<label class="tablas">Puesto</label>

<table border="1" class="tablas">

<tr>

<th>Id\_Puesto</th>

<th>Id\_Nombre</th>

</tr>

<tr>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

</tr>

</table>

<br>

<button class="edicion">Crear</button>

<button class="edicion">Editar</button>

<button class="edicion">Eliminar</button>

</div>

<br>

<br>

<br>

<div>

<label class="tablas">Usuario</label>

<table border="1" class="tablas">

<tr>

<th>Id\_Usuario</th>

<th>Nombre\_Usuario</th>

<th>Contraseña</th>

<th>Nombre</th>

<th>Apellido\_Paterno</th>

<th>Apellido\_Materno</th>

<th>Email</th>

<th>Telefono</th>

<th>Email</th>

<th>Id\_Puesto</th>

<th>Id\_Departamento</th>

<th>Id\_Tipo Usuario</th>

</tr>

<tr>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

</tr>

</table>

<br>

<button class="edicion">Crear</button>

<button class="edicion">Editar</button>

<button class="edicion">Eliminar</button>

</div>

<br>

<br>

<br>

<div>

<label class="tablas">Tipo\_Usuario</label>

<table border="1" class="tablas">

<tr>

<th>Id\_Tipo\_Usuario</th>

<th>Nombre</th>

</tr>

<tr>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

</tr>

</table>

<br>

<button class="edicion">Crear</button>

<button class="edicion">Editar</button>

<button class="edicion">Eliminar</button>

</div>

<br>

<br>

<br>

<div>

<label class="tablas">Tipo\_Pago</label>

<table border="1" class="tablas">

<tr>

<th>Id\_Tipo\_Pago</th>

<th>Nombre</th>

</tr>

<tr>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

</tr>

</table>

<br>

<button class="edicion">Crear</button>

<button class="edicion">Editar</button>

<button class="edicion">Eliminar</button>

</div>

<br>

<br>

<br>

<div>

<label class="tablas">Categoria</label>

<table border="1" class="tablas">

<tr>

<th>Id\_Categoria</th>

<th>Nombre</th>

</tr>

<tr>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

</tr>

</table>

<br>

<button class="edicion">Crear</button>

<button class="edicion">Editar</button>

<button class="edicion">Eliminar</button>

</div>

<br>

<br>

<br>

<div>

<label class="tablas">Libro</label>

<table border="1" class="tablas">

<tr>

<th>Id\_Libro\_ISBN</th>

<th>Titulo</th>

<th>Id\_Categoria</th>

<th>Volumen</th>

<th>Id\_Tipo\_Formato</th>

<th>Id\_Editorial</th>

<th>Fecha\_Publicacion</th>

<th>Paginas</th>

<th>Precio</th>

<th>Id\_Rating</th>

</tr>

<tr>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

</tr>

</table>

<br>

<button class="edicion">Crear</button>

<button class="edicion">Editar</button>

<button class="edicion">Eliminar</button>

</div>

<br>

<br>

<br>

<div>

<label class="tablas">Tipo\_Formato</label>

<table border="1" class="tablas">

<tr>

<th>Id\_Tipo\_Formato</th>

<th>Nombre</th>

</tr>

<tr>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

</tr>

</table>

<br>

<button class="edicion">Crear</button>

<button class="edicion">Editar</button>

<button class="edicion">Eliminar</button>

</div>

<br>

<br>

<br>

<div>

<label class="tablas">Departamento</label>

<table border="1" class="tablas">

<tr>

<th>Id\_Departamento</th>

<th>Departamento</th>

</tr>

<tr>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

</tr>

</table>

<br>

<button class="edicion">Crear</button>

<button class="edicion">Editar</button>

<button class="edicion">Eliminar</button>

</div>

<br>

<br>

<br>

<div>

<label class="tablas">Autor</label>

<table border="1" class="tablas">

<tr>

<th>Id\_Autor</th>

<th>Nombre</th>

<th>Apellido</th>

<th>Fecha\_Nacimiento</th>

<th>Id\_Pais</th>

</tr>

<tr>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

</tr>

</table>

<br>

<button class="edicion">Crear</button>

<button class="edicion">Editar</button>

<button class="edicion">Eliminar</button>

</div>

<br>

<br>

<br>

<div>

<label class="tablas">Editorial</label>

<table border="1" class="tablas">

<tr>

<th>Id\_Editorial</th>

<th>Nombre</th>

<th>Id\_Ciudad</th>

<th>Año\_Inaguracion</th>

</tr>

<tr>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

<td><br></td>

<td><br></td>

</tr>

</table>

<br>

<button class="edicion">Crear</button>

<button class="edicion">Editar</button>

<button class="edicion">Eliminar</button>

</div>

<br>

<br>

<br>

<div>

<label class="tablas">Libro\_Autores</label>

<table border="1" class="tablas">

<tr>

<th>Id\_Libro\_Autores</th>

<th>Id\_Libro\_ISBN</th>

<th>Id\_Autor</th>

</tr>

<tr>

<td><br></td>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

<td><br></td>

</tr>

<tr>

<td><br></td>

<td><br></td>

<td><br></td>

</tr>

</table>

<br>

<button class="edicion">Crear</button>

<button class="edicion">Editar</button>

<button class="edicion">Eliminar</button>

</div>

</body>

</html>

