***CASA DE DEPORTES***

**Grupo 1:**

* González, Mateo (55753)
* Vázquez, Lucas Alejo (55893)
* Yalux, Sergio (55757)

[**Repositorio Git****hub**](https://github.com/VazquezLucasA/ProyectoFinal--Lab-2)

Contenido

[PROPÓSITO Y FUNCIONALIDADES 4](#_Toc120670510)

[BASE DE DATOS 5](#_Toc120670511)

[MODELO ENTIDAD RELACION 5](#_Toc120670512)

[TABLA USUARIO 5](#_Toc120670513)

[TABLA EMPLEADOS 6](#_Toc120670514)

[TABLA VENTAS 6](#_Toc120670515)

[TABLA DETALLES VENTA 6](#_Toc120670516)

[TABLA PRODUCTOS 7](#_Toc120670517)

[FORMULARIOS EN EJECUCIÓN 7](#_Toc120670518)

[frmUsuario 7](#_Toc120670519)

[frmMenu 8](#_Toc120670520)

[frmSucursal 9](#_Toc120670521)

[frmEmpleados 10](#_Toc120670522)

[frmVentas 11](#_Toc120670523)

[frmDetalle 11](#_Toc120670524)

[CÓDIGO DE CADA CLASE 13](#_Toc120670525)

[CAPA DE PRESENTACIÓN 13](#_Toc120670526)

[frmUsuario.cs 13](#_Toc120670527)

[frmMenu.cs 15](#_Toc120670528)

[frmSucursal.cs 17](#_Toc120670529)

[frmEmpleados.cs 21](#_Toc120670530)

[frmVentas.cs 25](#_Toc120670531)

[frmDetalles.cs 28](#_Toc120670532)

[frmProductos.cs 29](#_Toc120670533)

[CAPA DE LÓGICA DEL NEGOCIO 35](#_Toc120670534)

[UsuarioCLN.cs 35](#_Toc120670535)

[SucursalCLN.cs 36](#_Toc120670536)

[EmpleadosCLN.cs 37](#_Toc120670537)

[VentasCLN.cs 39](#_Toc120670538)

[DetallesCLN.cs 41](#_Toc120670539)

[ProductosCLN.cs 42](#_Toc120670540)

[CAPA DE ACCESO A DATOS 44](#_Toc120670541)

[**ConexionCAD.cs** 44](#_Toc120670542)

[UsuarioCAD.cs 45](#_Toc120670543)

[SucursalCAD.cs 46](#_Toc120670544)

[EmpleadoCAD.cs 48](#_Toc120670545)

[VentasCAD.cs 50](#_Toc120670546)

[DetalleCAD.cs 53](#_Toc120670547)

[ProductosCAD.cs 55](#_Toc120670548)

# PROPÓSITO Y FUNCIONALIDADES

La aplicación apunta a brindar soluciones para un comercio de indumentaria deportiva.

Posee la posibilidad de ver, crear, modificar y eliminar:

-Sucursales de la franquicia.

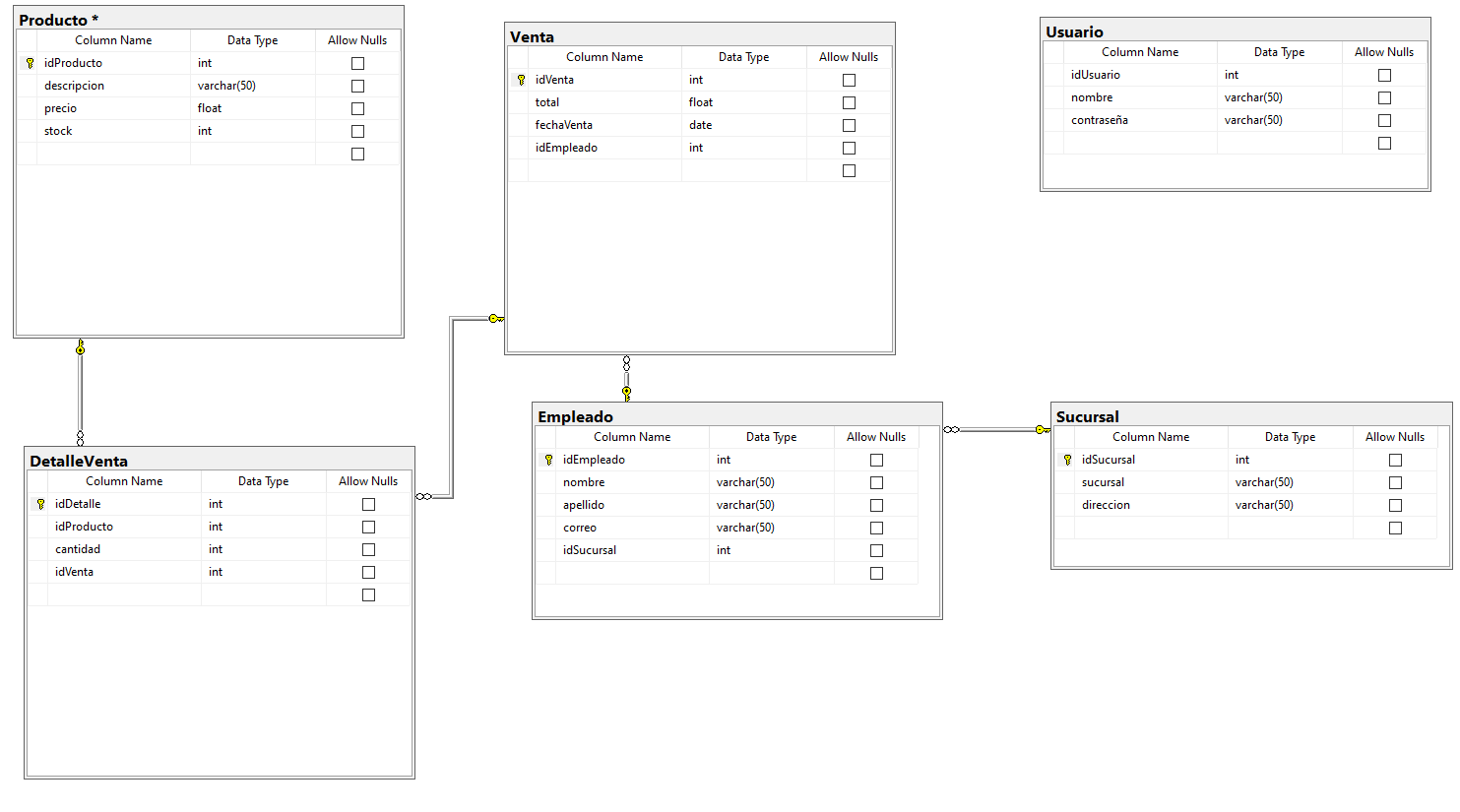
-Empleados de cada sucursal.

-Productos en stock.

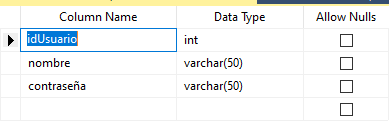
A su vez, también se puede seleccionar productos y realizar una venta la cuál luego figura con su respectivo detalle en la sección de reporte de ventas.

# BASE DE DATOS

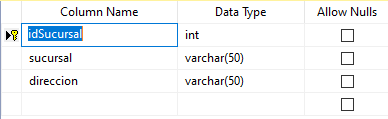
## MODELO ENTIDAD RELACION



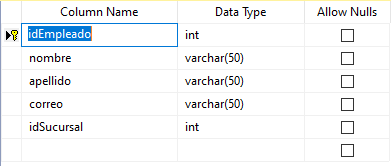
## TABLA USUARIO



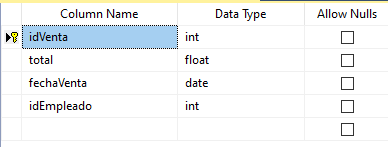
### **TABLA SUCURSALES**



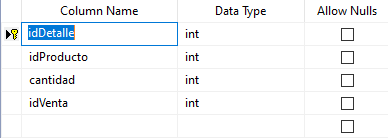
## TABLA EMPLEADOS



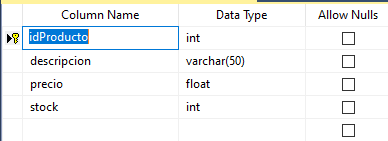
## TABLA VENTAS



## TABLA DETALLES VENTA



## TABLA PRODUCTOS



# FORMULARIOS EN EJECUCIÓN

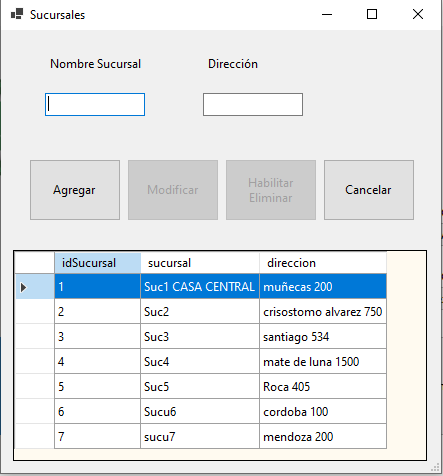
## frmUsuario



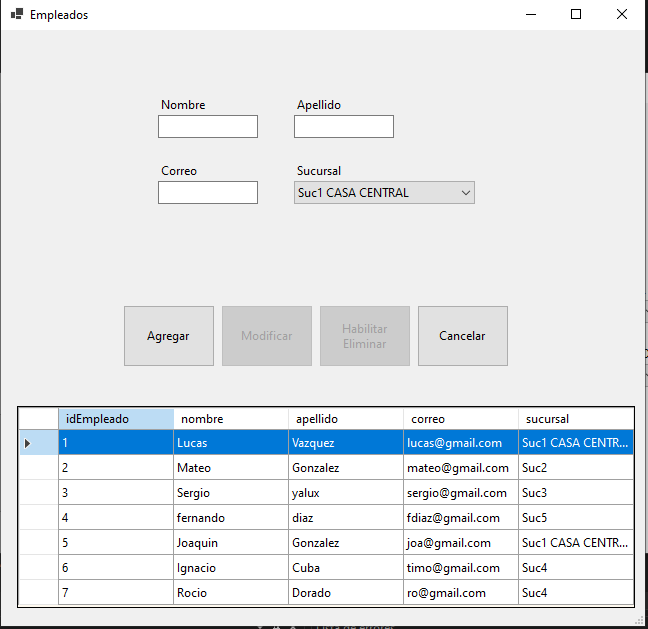
## frmMenu



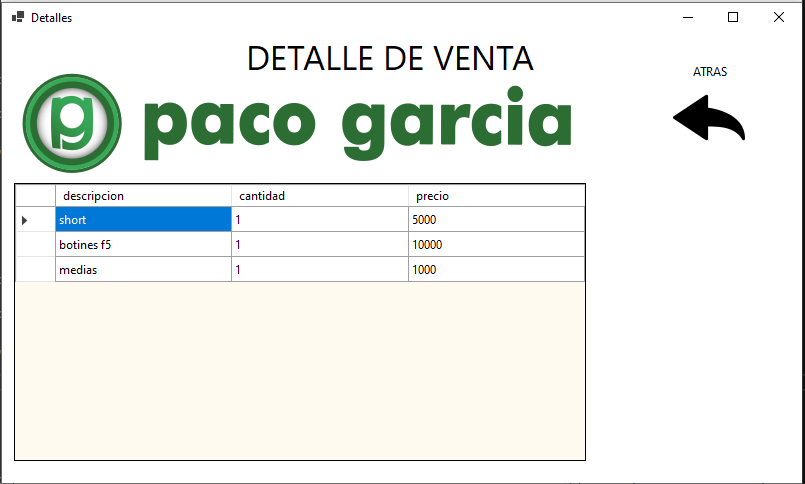
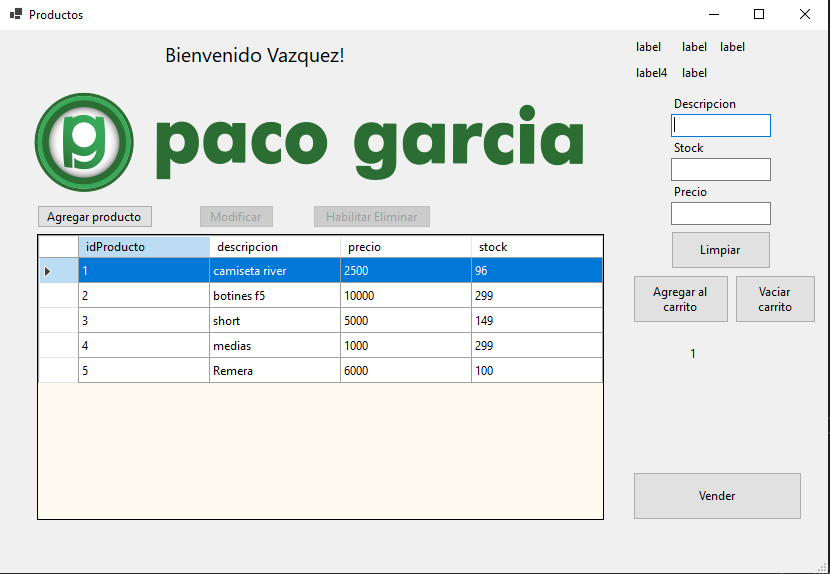
## frmSucursal



## frmEmpleados



frmVentas

frmDetallefrmProductos

# CÓDIGO DE CADA CLASE

## CAPA DE PRESENTACIÓN

### frmUsuario.cs

using CapaLogicaNegocio;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace CapaPresentacion

{

public partial class frmUsuario : Form

{

private UsuarioCLN objUsuarioCLN;

private frmMenu frmMenuInicio;

public frmUsuario()

{

InitializeComponent();

objUsuarioCLN= new UsuarioCLN();

frmMenuInicio = new frmMenu();

llenarCboUsuario();

}

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

{

bool bandera = false;

int idUsuario = Convert.ToInt32(cboUsuarios.SelectedValue);

string clave = txtContraseña.Text;

bandera = objUsuarioCLN.validarUsuario(idUsuario, clave);

if (bandera)

{

MessageBox.Show("Contraseña correcta. Ingresando...");

this.Hide();

frmMenuInicio.ShowDialog();

this.Close();

}

else

{

MessageBox.Show("Contraseña incorrecta. Contacte al administrador.");

cboUsuarios.SelectedIndex = 0;

txtContraseña.ForeColor = Color.DimGray;

txtContraseña.Text = "CONTRASEÑA";

txtContraseña.UseSystemPasswordChar = false;

}

}

private void llenarCboUsuario()

{

//cboUsuarios

DataTable miTabla = new DataTable();

miTabla.Clear();

miTabla = objUsuarioCLN.consultarUsuarios();

cboUsuarios.DataSource = miTabla;

cboUsuarios.ValueMember = "idUsuario";

cboUsuarios.DisplayMember = "nombre";

cboUsuarios.SelectedIndex = 0;

}

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

{

if(txtContraseña.Text == "CONTRASEÑA")

{

txtContraseña.Text = "";

txtContraseña.ForeColor = Color.Red;

txtContraseña.UseSystemPasswordChar= true;

}

}

private void txtContraseña\_Leave(object sender, EventArgs e)

{

if (txtContraseña.Text == "")

{

txtContraseña.Text = "CONTRASEÑA";

txtContraseña.ForeColor = Color.DimGray;

txtContraseña.UseSystemPasswordChar = false;

}

}

}

}

### frmMenu.cs

using CapaLogicaNegocio;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using WinFormsApp1;

namespace CapaPresentacion

{

public partial class frmMenu : Form

{

//MIEMBROS ATRIBUTOS

private frmSucursal frmSuc;

private frmEmpleados frmEmp;

private frmProductos frmProduct;

private frmVentas objFrmVentas;

private SucursalCLN objSucursalCLN;

private EmpleadosCLN objEmpleadosCLN;

private frmCreditos frmCredit;

//CONSTRUCTOR

public frmMenu()

{

InitializeComponent();

frmSuc = new frmSucursal();

frmEmp = new frmEmpleados();

frmCredit = new frmCreditos();

objFrmVentas = new frmVentas();

objSucursalCLN= new SucursalCLN();

objEmpleadosCLN = new EmpleadosCLN();

}

//LOAD

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

{

llenarCbxSucursal();

llenarCbxEmpleado();

}

//BOTON INGRESAR SUCURSAL

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

{

//this.Hide();

frmSuc.ShowDialog();

//this.Close();

}

//BOTON INGRESAR EMPLEADOS

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

{

//this.Hide();

frmEmp.ShowDialog();

//this.Close();

}

//BOTON INGRESAR PRODUCTOS

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

{

//this.Hide();

frmProduct = new frmProductos(cbxEmpleados.Text, Convert.ToInt32(cbxEmpleados.SelectedValue));

frmProduct.ShowDialog();

//this.Close();

}

//COMBOBOX SUCURSAL

private void llenarCbxSucursal()

{

DataTable miTabla = new DataTable();

miTabla = objSucursalCLN.consultarSucursales();

cbxSucursal.DisplayMember = "sucursal";

cbxSucursal.ValueMember = "idSucursal";

cbxSucursal.DataSource = miTabla;

cbxSucursal.SelectedIndex = 0;

}

//COMBOBOX EMPLEADOS

private void llenarCbxEmpleado()

{

DataTable miTabla = new DataTable();

int valor = Convert.ToInt32(cbxSucursal.SelectedValue);

miTabla = objEmpleadosCLN.consultarEmpleadosUnaSucursal(valor);

cbxEmpleados.DataSource = miTabla;

cbxEmpleados.ValueMember = "idEmpleado";

cbxEmpleados.DisplayMember = "apellido";

}

//BOTON INGRESAR VENTAS

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

{

//this.Hide();

objFrmVentas.ShowDialog();

//this.Close();

}

//SELECTED INDEX CHANGED EVENTO

private void cbxSucursal\_SelectedIndexChanged(object sender, EventArgs e)

{

llenarCbxEmpleado();

}

//BTN CREDITOS

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

{

frmCredit.Show();

}

}

}

### frmSucursal.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data;

using CapaLogicaNegocio;

namespace CapaPresentacion

{

public partial class frmSucursal : Form

{

//ATRIBUTOS

private SucursalCLN objSucursalCLN;

private DataTable miTabla;

private int indice;

private string sucursal;

private string direccion;

//CONSTRUCTOR

public frmSucursal()

{

InitializeComponent();

indice = 0;

objSucursalCLN = new SucursalCLN();

miTabla = new DataTable();

}

//BOTON ELIMINAR

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

{

DialogResult opcion = MessageBox.Show("¿Desea eliminar la sucursal?", "Eliminar sucursal", MessageBoxButtons.YesNo);

if (opcion == DialogResult.Yes)

{

objSucursalCLN.eliminarSucursal(Convert.ToInt32(dgvSucursal.Rows[indice].Cells[0].Value));

MessageBox.Show("Se eliminó la sucursal correctamente", "Eliminado", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

LimpiarTextBoxs();

btnEliminar.Visible = false;

btnPreEliminar.Visible = true;

btnPreEliminar.Enabled = true;

btnModificar.Enabled = true;

cargarDgv();

}

else

{ MessageBox.Show("Operación cancelada", "Cancelar", MessageBoxButtons.OK, MessageBoxIcon.Exclamation); }

cargarDgv();

}

//BOTON MODIFICAR

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

{

sucursal= txtSucursal.Text;

direccion = txtDireccion.Text;

if(txtSucursal.Text == string.Empty || txtDireccion.Text == string.Empty)

{

MessageBox.Show("Debe completar todos los campos", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else

{

objSucursalCLN.actualizarSucursal(Convert.ToInt32(dgvSucursal.Rows[indice].Cells[0].Value), sucursal, direccion);

MessageBox.Show("Los datos fueron actualizados");

}

LimpiarTextBoxs();

btnAgregar.Enabled = true;

btnPreEliminar.Enabled = false;

btnModificar.Enabled = false;

cargarDgv();

}

//BOTON AGREGAR

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

{

int indiceUltFila = objSucursalCLN.getIndiceUltFila(dgvSucursal.Rows.Count);

int codigoSucursal = objSucursalCLN.getCodigoSucursal(dgvSucursal.Rows[indiceUltFila].Cells[0].Value);

DialogResult opcion = MessageBox.Show("¿Está seguro que quiere agregar una nueva sucursal?", "Nueva sucursal", MessageBoxButtons.YesNo);

if (opcion == DialogResult.Yes)

{

sucursal = txtSucursal.Text;

direccion = txtDireccion.Text;

if (txtSucursal.Text == string.Empty || txtDireccion.Text == string.Empty)

{

MessageBox.Show("Debe completar todos los campos", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else

{

try

{

objSucursalCLN.agregarSucursal(codigoSucursal, sucursal, direccion);

LimpiarTextBoxs();

}

catch (Exception ex)

{

MessageBox.Show("Hubo un error: ", ex.Message, MessageBoxButtons.OK, MessageBoxIcon.Error);

LimpiarTextBoxs();

}

}

}

else

{

MessageBox.Show("Carga de sucursal cancelada");

LimpiarTextBoxs();

}

cargarDgv();

}

//BOTON CANCELAR

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

{

LimpiarTextBoxs();

btnAgregar.Enabled = true;

btnModificar.Enabled = false;

btnEliminar.Visible = false;

btnEliminar.Enabled = false;

btnPreEliminar.Visible = true;

btnPreEliminar.Enabled = false;

}

//LOAD

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

{

cargarDgv();

btnEliminar.Visible = false;

btnModificar.Enabled = false;

btnEliminar.Enabled = false;

btnPreEliminar.Enabled = false;

dgvSucursal.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.DisplayedCells;

}

//CLICK EN LAS CELDAS

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

{

indice = e.RowIndex;

if (indice == -1)

{

//MessageBox.Show("Seleccione un fila válida");

}

else

{

if (dgvSucursal.Rows[indice].Cells[0].Value == null)

{

MessageBox.Show("Error, haga bien el click");

}

else

{

btnEliminar.Visible = false;

btnAgregar.Enabled = false;

btnPreEliminar.Enabled = true;

btnPreEliminar.Visible = true;

btnModificar.Enabled = true;

txtSucursal.Text = (dgvSucursal.Rows[indice].Cells[1].Value).ToString();

txtDireccion.Text = (dgvSucursal.Rows[indice].Cells[2].Value).ToString();

}

}

}

//SE CARGA EL DGV

private void cargarDgv()

{

miTabla.Clear();

miTabla = objSucursalCLN.consultarSucursales();

dgvSucursal.DataSource = miTabla;

}

//LIMPIAR TXT

private void LimpiarTextBoxs()

{

txtSucursal.Clear();

txtDireccion.Clear();

}

//BOTON PRE ELIMINAR

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

{

btnEliminar.Visible = true;

btnEliminar.Enabled = true;

btnPreEliminar.Visible = false;

btnModificar.Enabled = false;

}

//INTENTOS DE BOTON ATRAS FALLIDO...

private void pictureBox1\_Click(object sender, EventArgs e)

{

// this.Hide();

// frmMen.Show();

// this.Close();

}

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

{

}

}

}

### frmEmpleados.cs

using CapaLogicaNegocio;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace CapaPresentacion

{

public partial class frmEmpleados : Form

{

private EmpleadosCLN objEmpleadosCLN;

private SucursalCLN objSucursalCLN;

private DataTable miTabla;

private int indice;

public frmEmpleados()

{

InitializeComponent();

objEmpleadosCLN = new EmpleadosCLN();

miTabla = new DataTable();

indice = 0;

objSucursalCLN = new SucursalCLN();

}

//BOTON CANCELAR

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

{

LimpiarTextBoxs();

btnAdd.Enabled = true;

btnModify.Enabled = false;

btnEliminate.Visible = false;

btnEliminate.Enabled = false;

btnPreEliminar.Visible = true;

btnPreEliminar.Enabled = false;

}

//BOTON AGREGAR

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

{

int indiceUltFila = objEmpleadosCLN.getIndiceUltFila(dgvEmpleados.Rows.Count);

int idEmpleado = objEmpleadosCLN.getCodigoEmpleado(dgvEmpleados.Rows[indiceUltFila].Cells[0].Value);

lblEmail.Text = idEmpleado.ToString();

DialogResult opcion = MessageBox.Show("¿Está seguro que quiere agregar un nuevo empleado?", "Nuevo empleado", MessageBoxButtons.YesNo);

if (opcion == DialogResult.Yes)

{

try

{

objEmpleadosCLN.agregarEmpleado(idEmpleado, txtName.Text, txtSurname.Text, txtEmail.Text, Convert.ToInt32(cbxSucursal.SelectedValue));

LimpiarTextBoxs();

}

catch (Exception ex)

{

MessageBox.Show("Hubo un error", ex.Message, MessageBoxButtons.OK, MessageBoxIcon.Error);

LimpiarTextBoxs();

}

}

else

{

MessageBox.Show("Carga de empleado cancelada");

LimpiarTextBoxs();

}

cargarDgv();

}

//BOTON MODIFICAR

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

{

objEmpleadosCLN.actualizarEmpleado(Convert.ToInt32(dgvEmpleados.Rows[indice].Cells[0].Value), txtName.Text, txtSurname.Text, txtEmail.Text, Convert.ToInt32(cbxSucursal.SelectedValue));

MessageBox.Show("Los datos fueron actualizados");

LimpiarTextBoxs();

btnAdd.Enabled = true;

btnPreEliminar.Enabled = false;

btnModify.Enabled = false;

cargarDgv();

}

//BOTON ELIMINAR

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

{

DialogResult opcion = MessageBox.Show("¿Desea eliminar el producto?", "Eliminar producto", MessageBoxButtons.YesNo);

if (opcion == DialogResult.Yes)

{

objEmpleadosCLN.eliminarEmpleado(Convert.ToInt32(dgvEmpleados.Rows[indice].Cells[0].Value));

MessageBox.Show("Se eliminó el producto correctamente", "Eliminado", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

LimpiarTextBoxs();

btnEliminate.Visible = false;

btnPreEliminar.Visible = true;

btnPreEliminar.Enabled = true;

btnModify.Enabled = true;

cargarDgv();

}

else

{ MessageBox.Show("Operación cancelada", "Cancelar", MessageBoxButtons.OK, MessageBoxIcon.Exclamation); }

cargarDgv();

}

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

{

}

//CLICK EN CELDAS

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

{

indice = e.RowIndex;

if (indice == -1)

{ MessageBox.Show("Seleccione un fila válida"); }

else

{

if (dgvEmpleados.Rows[indice].Cells[0].Value == null)

{

MessageBox.Show("Error, haga bien el click");

}

else

{

btnEliminate.Visible = false;

btnAdd.Enabled = false;

btnPreEliminar.Enabled = true;

btnPreEliminar.Visible = true;

btnModify.Enabled = true;

txtName.Text = (dgvEmpleados.Rows[indice].Cells[1].Value).ToString();

txtSurname.Text = (dgvEmpleados.Rows[indice].Cells[2].Value).ToString();

txtEmail.Text = (dgvEmpleados.Rows[indice].Cells[3].Value).ToString();

}

}

}

//SE CARGA EL DGV

private void cargarDgv()

{

miTabla.Clear();

miTabla = objEmpleadosCLN.consultarEmpleados();

dgvEmpleados.DataSource = miTabla;

}

//LIMPIAR TXT

private void LimpiarTextBoxs()

{

txtEmail.Clear();

txtName.Clear();

txtSurname.Clear();

}

//BOTON HABILITAR ELIMINAR

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

{

btnEliminate.Visible = true;

btnEliminate.Enabled = true;

btnPreEliminar.Visible = false;

btnModify.Enabled = false;

}

//EVENTO LOAD

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

{

cargarDgv();

btnEliminate.Visible = false;

btnModify.Enabled = false;

btnEliminate.Enabled = false;

btnPreEliminar.Enabled = false;

llenarCbxSucursal();

dgvEmpleados.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;

}

//LLENAR COMBOBOX SUCUS

private void llenarCbxSucursal()

{

DataTable miTabla = new DataTable();

miTabla.Clear();

cbxSucursal.DataSource = miTabla;

miTabla = objSucursalCLN.consultarSucursales();

cbxSucursal.DataSource = miTabla;

cbxSucursal.ValueMember = "idSucursal";

cbxSucursal.DisplayMember = "sucursal";

cbxSucursal.SelectedIndex = 0;

}

}

}

### frmVentas.cs

using CapaLogicaNegocio;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace CapaPresentacion

{

public partial class frmVentas : Form

{

private VentasCLN objVentasCLN;

private bool banderaSucursal, banderaEmpleados;

private DataTable miTabla;

private DataTable tablaSuc;

private DataTable tablaEmpleados;

private SucursalCLN objSucursalCLN;

private EmpleadosCLN objEmpleadosCLN;

private int indice;

public frmVentas()

{

InitializeComponent();

objVentasCLN = new VentasCLN();

miTabla = new DataTable();

banderaSucursal= false;

banderaEmpleados= false;

objSucursalCLN = new SucursalCLN();

objEmpleadosCLN = new EmpleadosCLN();

tablaSuc = new DataTable();

tablaEmpleados = new DataTable();

indice = 0;

}

private void cargarDgv()

{

miTabla.Clear();

miTabla = objVentasCLN.consultaVentasTodas();

//miTabla = objVentasCLN.consultarVentas();

dgvVentas.DataSource = miTabla;

}

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

{

cargarDgv();

llenarCboSucursales();

llenarCboEmpleados();

dgvVentas.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;

chkEmpleados.Visible=false;

chkSucursal.Visible=false;

}

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

{

int idSucursal, idEmpleado;

idSucursal = Convert.ToInt32(cboSucursal.SelectedValue);

if (chkSucursal.Checked && chkEmpleados.Checked==false)

{

miTabla.Clear();

miTabla = objVentasCLN.consultaVentasTodas();

dgvVentas.DataSource = miTabla;

}

if (chkEmpleados.Checked)

{

lblEmpleados.Text= cboEmpleados.Text;

idSucursal = Convert.ToInt32(cboSucursal.SelectedValue);

lblReporte.Text = idSucursal.ToString();

miTabla.Clear();

miTabla = objVentasCLN.consultarVentasUnaSucursal(idSucursal);

dgvVentas.DataSource = miTabla;

}

if(!(chkEmpleados.Checked && chkSucursal.Checked))

{

idEmpleado = Convert.ToInt32(cboEmpleados.SelectedValue);

miTabla.Clear();

miTabla = objVentasCLN.consultarVentasUnEmpleado(idEmpleado);

dgvVentas.DataSource = miTabla;

}

}

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

{

if(chkSucursal.Checked )

{

cboSucursal.Enabled= false;

cboEmpleados.Enabled= false;

chkEmpleados.Enabled = false;

}

else

{

cboSucursal.Enabled = true;

cboEmpleados.Enabled = true;

chkEmpleados.Enabled = true;

}

}

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

{

if (chkEmpleados.Checked)

{

banderaEmpleados = true;

cboEmpleados.Enabled = false;

}

else

{

banderaEmpleados = false;

cboEmpleados.Enabled = true;

}

}

private void llenarCboSucursales()

{

tablaSuc.Clear();

tablaSuc = objSucursalCLN.consultarSucursales();

cboSucursal.ValueMember = "idSucursal";

cboSucursal.DisplayMember = "sucursal";

cboSucursal.DataSource = tablaSuc;

cboSucursal.SelectedIndex = 0;

}

private void llenarCboEmpleados()

{

tablaEmpleados.Clear();

tablaEmpleados = objEmpleadosCLN.consultarEmpleadosUnaSucursal(Convert.ToInt32(cboSucursal.SelectedValue));

cboEmpleados.DataSource = tablaEmpleados;

cboEmpleados.ValueMember = "idEmpleado";

cboEmpleados.DisplayMember = "apellido";

}

private void cboSucursal\_SelectedIndexChanged(object sender, EventArgs e)

{

llenarCboEmpleados();

}

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

{

frmDetalle objFrmDetalle = new frmDetalle(indice);

objFrmDetalle.Show();

}

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

{

int indiceDgv;

indiceDgv = e.RowIndex;

indice = Convert.ToInt32(dgvVentas.Rows[indiceDgv].Cells[0].Value);

}

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

{

}

}

}

### frmDetalles.cs

using CapaLogicaNegocio;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace CapaPresentacion

{

public partial class frmDetalle : Form

{

private DataTable miTabla;

private DetalleCLN objDetalleCLN;

private int idVenta;

public frmDetalle(int indice)

{

InitializeComponent();

miTabla = new DataTable();

objDetalleCLN = new DetalleCLN();

idVenta = indice;

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

this.Close();

}

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

{

cargarDgv();

dgvDetalles.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;

}

private void cargarDgv()

{

miTabla.Clear();

miTabla = objDetalleCLN.consultarDetalles(idVenta);

dgvDetalles.DataSource = miTabla;

}

}

}

### frmProductos.cs

using CapaLogicaNegocio;

using CapaPresentacion;

using System.Data;

namespace WinFormsApp1

{

public partial class frmProductos : Form

{

/\*

Proyecto Final - Laboratorio de computación 2.

Universidad Tecnologica Nacional. 2022.

Grupo:

-González, Mateo

-Vázquez, Lucas Alejo

-Yalux, Sergio

\*/

//MIEMBROS ATRIBUTOS

private int indice;

private ProductoCLN objProductoCLN;

private DataTable miTabla;

private int stock;

private float precio;

private int indiceCarrito;

private string nombreBienvenida;

private string nombreProducto;

private int[] idProductos;

private float precioVenta;

private int idEmpleado;

//CONSTRUCTOR

public frmProductos(string nombreBienvenido, int idEmpleado)

{

InitializeComponent();

indice = 0;

objProductoCLN = new ProductoCLN();

miTabla = new DataTable();

precio = 0;

stock = 0;

indiceCarrito = 0;

nombreBienvenida = nombreBienvenido;

nombreProducto = "";

idProductos = new int[5];

this.idEmpleado= idEmpleado;

}

//MIEMBROS METODOS

//LOAD

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

{

cargarDgv();

btnEliminar.Visible = false;

btnModificar.Enabled = false;

btnEliminar.Enabled = false;

btnPreEliminar.Enabled = false;

lblBienvenida.Text += nombreBienvenida + "!";

}

//CLICK EN LAS CELDAS

private void dgvStock\_CellClick\_1(object sender, DataGridViewCellEventArgs e)

{

indice = e.RowIndex;

nombreProducto = dgvStock.Rows[indice].Cells[1].Value.ToString();

//precioVenta = Convert.ToInt32(dgvStock.Rows[indice].Cells[2].Value);

lblProductoTres.Text = indice.ToString();

if (indice == -1)

{

MessageBox.Show("Seleccione un fila válida");

}

else

{

if (dgvStock.Rows[indice].Cells[0].Value == null)

{

MessageBox.Show("Error, haga bien el click");

}

else

{

btnEliminar.Visible = false;

btnAgregar.Enabled = false;

btnPreEliminar.Enabled = true;

btnPreEliminar.Visible = true;

btnModificar.Enabled = true;

txtDescripcion.Text = (dgvStock.Rows[indice].Cells[1].Value).ToString();

txtPrice.Text = (dgvStock.Rows[indice].Cells[2].Value).ToString();

txtStocki.Text = (dgvStock.Rows[indice].Cells[3].Value).ToString();

}

}

}

//CARGAR DATA GRID

private void cargarDgv()

{

miTabla.Clear();

miTabla = objProductoCLN.consultarProductos();

dgvStock.DataSource = miTabla;

}

//BOTON AGREGAR

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

{

int indiceUltFila = objProductoCLN.getIndiceUltFila(dgvStock.Rows.Count);

int codigoProducto = objProductoCLN.getCodigoProducto(dgvStock.Rows[indiceUltFila].Cells[0].Value);

DialogResult opcion = MessageBox.Show("¿Está seguro que quiere agregar un nuevo producto?", "Nuevo producto", MessageBoxButtons.YesNo);

if (opcion == DialogResult.Yes)

{

//indice = dgvStock.Rows.Add();

if (txtDescripcion.Text == string.Empty || txtPrice.Text == string.Empty || txtStocki.Text == string.Empty)

{

MessageBox.Show("Debe completar todos los campos", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else

{

try

{

precio = float.Parse(txtPrice.Text);

stock = Convert.ToInt32(txtStocki.Text);

objProductoCLN.agregarProducto(codigoProducto, txtDescripcion.Text, precio, stock);

}

catch

{

MessageBox.Show("Debe ingresar los tipos de datos correctos", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

else

{

MessageBox.Show("Carga de producto cancelada");

}

LimpiarTextBoxs();

cargarDgv();

}

//BOTON MODIFICAR

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

{

if (txtDescripcion.Text == string.Empty || txtPrice.Text == string.Empty || txtStocki.Text == string.Empty)

{

MessageBox.Show("Debe completar todos los campos", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else

{

try

{

stock = Convert.ToInt32(txtStocki.Text);

precio = float.Parse(txtPrice.Text);

objProductoCLN.actualizarProducto(Convert.ToInt32(dgvStock.Rows[indice].Cells[0].Value), txtDescripcion.Text, precio, stock);

MessageBox.Show("Los datos fueron actualizados");

}

catch

{

MessageBox.Show("Debe ingresar los tipos de datos correctos", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

LimpiarTextBoxs();

btnAgregar.Enabled = true;

btnPreEliminar.Enabled = false;

btnModificar.Enabled = false;

cargarDgv();

}

//LIMPIAR TXT

private void LimpiarTextBoxs()

{

txtDescripcion.Clear();

txtStocki.Clear();

txtPrice.Clear();

}

//BOTON ELIMINAR

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

{

DialogResult opcion = MessageBox.Show("¿Desea eliminar el producto?", "Eliminar producto", MessageBoxButtons.YesNo);

if (opcion == DialogResult.Yes)

{

objProductoCLN.eliminarProducto(Convert.ToInt32(dgvStock.Rows[indice].Cells[0].Value));

MessageBox.Show("Se eliminó el producto correctamente", "Eliminado", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

LimpiarTextBoxs();

btnEliminar.Visible = false;

btnPreEliminar.Visible = true;

btnPreEliminar.Enabled = true;

btnModificar.Enabled = true;

}

else

{ MessageBox.Show("Operación cancelada", "Cancelar", MessageBoxButtons.OK, MessageBoxIcon.Exclamation); }

cargarDgv();

}

//BOTON VENDER

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

{

DialogResult opcion = MessageBox.Show("¿Desea confirmar la venta del producto?", "Venta de producto", MessageBoxButtons.YesNo);

if (opcion == DialogResult.Yes)

{

try

{

//vender carrito

float venta = objProductoCLN.venderProductos(idProductos, idEmpleado);

MessageBox.Show($"El total a pagar es {venta}", "Venta de producto", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

}

catch

{

MessageBox.Show("Hubo un error, intente nuevamente.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

}

}

else

MessageBox.Show("Operación cancelada", "Cancelar", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

indiceCarrito = 0;

for( int i=0; i<idProductos.Length; i++)

{

idProductos[i] = 0;

}

lblLista.Text = "";

LimpiarTextBoxs();

cargarDgv();

}

//BOTON PARA LIMPIAR

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

{

LimpiarTextBoxs();

btnAgregar.Enabled = true;

btnModificar.Enabled = false;

btnEliminar.Visible = false;

btnEliminar.Enabled = false;

btnPreEliminar.Visible = true;

btnPreEliminar.Enabled = false;

}

//BOTON HABILITAR ELIMINAR

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

{

btnEliminar.Visible = true;

btnEliminar.Enabled = true;

btnPreEliminar.Visible = false;

btnModificar.Enabled = false;

}

//BOTON AGREGAR CARRITO

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

{

if (indiceCarrito < 5)

{

idProductos[indiceCarrito] = Convert.ToInt32(dgvStock.Rows[indice].Cells[0].Value);

lblLista.Text += nombreProducto + " \n";

}

else

{

MessageBox.Show("Alcanzó la cantidad máxima del carrito");

}

indiceCarrito++;

lblProductoCero.Text = idProductos[0].ToString();

lblProductoUno.Text = idProductos[1].ToString();

lblProductoDos.Text = idProductos[2].ToString();

lblProductoTres.Text = idProductos[3].ToString();

lblProductoCuatro.Text = idProductos[4].ToString();

}

//BOTON VACIAR CARRITO

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

{

indiceCarrito = 0;

for (int i = 0; i < idProductos.Length; i++)

{

idProductos[i] = 0;

}

lblLista.Text = "";

cargarDgv();

}

//INTENTO DE AGREGAR AL CARRITO CON DOBLE CLICK FALLIDO...

private void dgvStock\_CellContentDoubleClick(object sender, DataGridViewCellEventArgs e)

{

//int indiceCarrito = 0;

//lblLista.Text += dgvStock.Rows[indice].Cells[1].Value.ToString() + "\n";

string[] listado = new string[5];

int[] idProductos = new int[5];

}

}

}

## CAPA DE LÓGICA DEL NEGOCIO

### UsuarioCLN.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

using CapaAccesoDatos;

using System.Data.SqlClient;

namespace CapaLogicaNegocio

{

public class UsuarioCLN

{

private DataTable miTabla;

private UsuarioCAD objUsuarioCAD;

public UsuarioCLN()

{

miTabla = new DataTable();

objUsuarioCAD = new UsuarioCAD();

}

public bool validarUsuario(int idUsuario, string clave)

{

return objUsuarioCAD.consultarUnaClave(idUsuario, clave);

}

public DataTable consultarUsuarios()

{

miTabla = objUsuarioCAD.consultarUsuarios();

return miTabla;

}

}

}

### SucursalCLN.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

using CapaAccesoDatos;

namespace CapaLogicaNegocio

{

public class SucursalCLN

{

private DataTable miTabla;

private SucursalCAD objSucursalCAD;

public SucursalCLN()

{

miTabla = new DataTable();

objSucursalCAD = new SucursalCAD();

}

public DataTable consultarSucursales()

{

miTabla = objSucursalCAD.consultarSucursales();

return miTabla;

}

public void agregarSucursal(int idSucursal, string nombre, string direccion)

{

idSucursal++;

objSucursalCAD.agregarSucursal(idSucursal, nombre, direccion);

}

public void eliminarSucursal(int idSucursal)

{

objSucursalCAD.eliminarSucursal(idSucursal);

}

public void actualizarSucursal(int idSucursal, string nombre, string direccion)

{

objSucursalCAD.actualizarSucursal(idSucursal, nombre, direccion);

}

public int getIndiceUltFila(int cantFilas)

{

return cantFilas - 2;

}

public int getCodigoSucursal(object a)

{

int numero = Convert.ToInt32(a);

int codigo = numero + 1;

return codigo;

}

}

}

### EmpleadosCLN.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

using CapaAccesoDatos;

namespace CapaLogicaNegocio

{

public class EmpleadosCLN

{

private DataTable miTabla;

private EmpleadosCAD objEmpleadosCAD;

public EmpleadosCLN()

{

miTabla = new DataTable();

objEmpleadosCAD = new EmpleadosCAD();

}

public DataTable consultarEmpleados()

{

miTabla.Clear();

miTabla = objEmpleadosCAD.consultarEmpleados();

return miTabla;

}

public void agregarEmpleado(int idEmpleado, string nombre, string apellido, string correo, int idSucursal)

{

objEmpleadosCAD.agregarEmpleado(idEmpleado, nombre, apellido, correo, idSucursal);

}

public void eliminarEmpleado(int idEmpleado)

{

objEmpleadosCAD.eliminarEmpleado(idEmpleado);

}

public void actualizarEmpleado(int idEmpleado, string nombre, string apellido, string correo, int idSucursal)

{

objEmpleadosCAD.actualizarEmpleado(idEmpleado, nombre, apellido, correo, idSucursal);

}

public int getIndiceUltFila(int cantFilas)

{

return cantFilas - 1;

}

public int getCodigoEmpleado(object a)

{

int numero = Convert.ToInt32(a);

int codigo = numero + 1;

return codigo;

}

public DataTable consultarEmpleadosUnaSucursal(int idSucursal)

{

miTabla.Clear();

miTabla = objEmpleadosCAD.consultarEmpleadosUnaSucursal(idSucursal);

return miTabla;

}

}

}

### VentasCLN.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

using CapaAccesoDatos;

namespace CapaLogicaNegocio

{

public class VentasCLN

{

private DataTable miTabla;

private VentasCAD objVentasCAD;

public VentasCLN()

{

miTabla = new DataTable();

objVentasCAD = new VentasCAD();

}

public DataTable consultarVentas()

{

miTabla = objVentasCAD.consultarVentas();

return miTabla;

}

public void agregarVenta(int idVenta, float total, DateTime fechaVenta, int idEmpleado)

{

objVentasCAD.agregarVenta(idVenta, total, fechaVenta, idEmpleado);

}

public void eliminarVenta(int idVenta)

{

objVentasCAD.eliminarVenta(idVenta);

}

public void actualizarVenta(int idVenta, float total, DateTime fechaVenta, int idEmpleado)

{

objVentasCAD.actualizarVenta(idVenta, total, fechaVenta, idEmpleado);

}

public int getIndiceUltFila(int cantFilas)

{

return cantFilas - 2;

}

public int getCodigoVenta(object a)

{

int numero = Convert.ToInt32(a);

int codigo = numero + 1;

return codigo;

}

public DataTable consultarVentasUnEmpleado(int idEmpleado)

{

miTabla = objVentasCAD.consultarVentasUnEmpleado(idEmpleado);

return miTabla;

}

public DataTable consultarVentasUnaSucursal(int idSucursal)

{

miTabla = objVentasCAD.consultarVentasUnaSucursal(idSucursal);

return miTabla;

}

public DataTable consultaVentasTodas()

{

miTabla = objVentasCAD.consultaVentasTodas();

return miTabla;

}

}

}

### DetallesCLN.cs

using CapaAccesoDatos;

using System;

using System.Collections.Generic;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace CapaLogicaNegocio

{

public class DetalleCLN

{

private DataTable miTabla;

private DetalleCAD objDetalleCAD;

public DetalleCLN()

{

miTabla = new DataTable();

objDetalleCAD = new DetalleCAD();

}

public DataTable consultarDetalles(int idVenta)

{

miTabla.Clear();

miTabla = objDetalleCAD.consultarDetalles(idVenta);

return miTabla;

}

}

}

### ProductosCLN.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

using CapaAccesoDatos;

namespace CapaLogicaNegocio

{

public class ProductoCLN

{

//miembros atributos

private DataTable miTabla;

private ProductoCAD objProductoCAD;

private VentasCAD objVentasCAD;

private DetalleCAD objDetalleCAD;

//miembros metodos

public ProductoCLN()

{

miTabla = new DataTable();

objProductoCAD = new ProductoCAD();

objVentasCAD = new VentasCAD();

objDetalleCAD = new DetalleCAD();

}

public DataTable consultarProductos()

{

miTabla = objProductoCAD.consultarProductos();

return miTabla;

}

public void agregarProducto(int idProducto, string descripcion, float precio, int stock)

{

idProducto++;

objProductoCAD.agregarProducto(idProducto, descripcion, precio, stock);

}

public void actualizarProducto(int idProducto, string descripcion, float precio, int stock)

{

objProductoCAD.actualizarProducto(idProducto, descripcion, precio, stock);

}

public void eliminarProducto(int idProducto)

{

objProductoCAD.eliminarProducto(idProducto);

}

public int getIndiceUltFila(int cantFilas)

{

return cantFilas - 2;

}

public int getCodigoProducto(object a)

{

int numero = Convert.ToInt32(a);

int codigo = numero + 1;

return codigo;

}

public int convertirINT()

{

return 0;

}

public float convertirFloat(string numero)

{

float jesus = float.Parse(numero);

jesus = (float)Math.Truncate((double)jesus \* 100.0 / 100.0);

return jesus;

}

public float venderProductos(int[] idProducto, int idEmpleado)

{

/\*se necesita:

venta: ultimo idVenta +1, total, fecha, idEmpleado

detalle: ultimo idDetalle +1 por cada producto, id de cada producto, cantidad siempre 1, mismo idVenta

producto: where idProducto, stock -1

\*/

//DataTable idVenta, idDetalle, precio= new DataTable();

int idVentaUlt = objVentasCAD.getIdVenta() +1;

float total = 0;

DateTime fechaVenta = DateTime.Now;

int idDetalle = objDetalleCAD.getIdDetalle();

int cantidad = 1;

for (int i=0; i< idProducto.Length; i++)

{

if (idProducto[i] != 0)

{

int precio = objProductoCAD.getPrecio(idProducto[i]);

total = total + precio;

objProductoCAD.venderProducto(idProducto[i]);

}

}

objVentasCAD.agregarVenta(idVentaUlt, total, fechaVenta, idEmpleado);

for (int i = 0; i < idProducto.Length; i++)

{

idDetalle = idDetalle + 1;

if (idProducto[i] != 0)

{

objDetalleCAD.agregarDetalles(idDetalle, idProducto[i], cantidad, idVentaUlt);

}

}

return total;

}

}

}

## CAPA DE ACCESO A DATOS

### [**ConexionCAD.cs**](../../Nueva%20carpeta%20(2)/capa%20datos/1ConexionCAD.txt)

using System;

using System.Data.SqlClient;

using System.Windows;

//listo

namespace CapaAccesoDatos

{

public class ConexionCAD

{

//private string cadena;

private string caden;

private SqlConnection conectarDB;

public ConexionCAD()

{

caden = "Data Source=DESKTOP-SHOETUK\\SQLEXPRESS;Initial Catalog=ProyectoFinalDB;Integrated Security=True";

//cadena = "Data Source=DESKTOP-30DTIQQ\\MSSQLEXPRESS;Initial Catalog = Cuspide SA; User ID = sa; Password = 654321";

conectarDB = new SqlConnection();

conectarDB.ConnectionString = caden;

}

public SqlConnection abrirConeccion()

{

conectarDB.Open();

return conectarDB;

}

public void cerrarConexion()

{

conectarDB.Close();

}

}

}

### UsuarioCAD.cs

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace CapaAccesoDatos

{

public class UsuarioCAD

{

private ConexionCAD objConexionCAD;

private SqlDataReader leerTabla;

private DataTable miTabla;

private SqlCommand comando;

public UsuarioCAD()

{

objConexionCAD = new ConexionCAD();

miTabla = new DataTable();

comando = new SqlCommand();

}

public DataTable consultarUsuarios()

{

comando.Connection = objConexionCAD.abrirConeccion();

comando.CommandText = "consultarUsuarios";

comando.CommandType = CommandType.StoredProcedure;

comando.Parameters.Clear();

leerTabla = comando.ExecuteReader();

miTabla.Load(leerTabla);

objConexionCAD.cerrarConexion();

return miTabla;

}

public bool consultarUnaClave(int idUsuario, string clave)

{

bool bandera = false;

comando.Connection = objConexionCAD.abrirConeccion();

comando.CommandText = "recuperarClave";

comando.CommandType = CommandType.StoredProcedure;

comando.Parameters.Clear();

comando.Parameters.AddWithValue("@idUsuario", idUsuario);

comando.Parameters.AddWithValue("@contraseña", clave);

leerTabla = comando.ExecuteReader();

if (leerTabla.HasRows)

{

bandera = true;

}

else

{

bandera = false;

}

objConexionCAD.cerrarConexion();

return bandera;

}

}}

### SucursalCAD.cs

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

//listo

namespace CapaAccesoDatos

{

public class SucursalCAD

{

//miembros atributos

private ConexionCAD objConexionCAD;

private SqlDataReader leerTabla;

private DataTable miTabla;

private SqlCommand comando;

public SucursalCAD()

{

objConexionCAD = new ConexionCAD();

miTabla = new DataTable();

comando = new SqlCommand();

}

public DataTable consultarSucursales()

{

comando.Connection = objConexionCAD.abrirConeccion();

comando.CommandText = "consultarSucursales";

comando.CommandType = CommandType.StoredProcedure;

comando.Parameters.Clear();

leerTabla = comando.ExecuteReader();

miTabla.Load(leerTabla);

objConexionCAD.cerrarConexion();

return miTabla;

}

public void agregarSucursal(int idSucursal, string sucursal, string direccion)

{

comando.Connection = objConexionCAD.abrirConeccion();

comando.CommandText = "agregarSucursal";

comando.CommandType = CommandType.StoredProcedure;

comando.Parameters.Clear();

comando.Parameters.AddWithValue("@idSucursal", idSucursal);

comando.Parameters.AddWithValue("@sucursal", sucursal);

comando.Parameters.AddWithValue("@direccion", direccion);

comando.ExecuteNonQuery();

objConexionCAD.cerrarConexion();

}

public void actualizarSucursal(int idSucursal, string sucursal, string direccion)

{

comando.Connection = objConexionCAD.abrirConeccion();

comando.CommandText = "actualizarSucursal";

comando.CommandType = CommandType.StoredProcedure;

comando.Parameters.Clear();

comando.Parameters.AddWithValue("@idSucursal", idSucursal);

comando.Parameters.AddWithValue("@sucursal", sucursal);

comando.Parameters.AddWithValue("@direccion", direccion);

comando.ExecuteNonQuery();

objConexionCAD.cerrarConexion();

}

public void eliminarSucursal(int idSucursal)

{

comando.Connection = objConexionCAD.abrirConeccion();

comando.CommandText = "eliminarSucursal";

comando.CommandType = CommandType.StoredProcedure;

comando.Parameters.Clear();

comando.Parameters.AddWithValue("@idSucursal", idSucursal);

comando.ExecuteNonQuery();

objConexionCAD.cerrarConexion();

}

}

}

### EmpleadoCAD.cs

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

//LISTO

namespace CapaAccesoDatos

{

public class EmpleadosCAD

{

//miembros atributos

private ConexionCAD objConexionCAD;

private SqlDataReader leerTabla;

private DataTable miTabla;

private SqlCommand comando;

//metodos para el CRUD y constructor

public EmpleadosCAD()

{

objConexionCAD = new ConexionCAD();

miTabla = new DataTable();

comando = new SqlCommand();

}

public DataTable consultarEmpleados()

{

comando.Connection = objConexionCAD.abrirConeccion();

comando.CommandText = "consultaEmpleadoSuc";

comando.CommandType = CommandType.StoredProcedure;

comando.Parameters.Clear();

leerTabla = comando.ExecuteReader();

miTabla.Load(leerTabla);

objConexionCAD.cerrarConexion();

return miTabla;

}

public void agregarEmpleado(int idEmpleado, string nombre, string apellido, string correo, int idSucursal)

{

comando.Connection = objConexionCAD.abrirConeccion();

comando.CommandText = "agregarEmpleado";

comando.CommandType = CommandType.StoredProcedure;

comando.Parameters.Clear();

comando.Parameters.AddWithValue("@idEmpleado", idEmpleado);

comando.Parameters.AddWithValue("@nombre", nombre);

comando.Parameters.AddWithValue("@apellido", apellido);

comando.Parameters.AddWithValue("@correo", correo);

comando.Parameters.AddWithValue("@idSucursal", idSucursal);

comando.ExecuteNonQuery();

objConexionCAD.cerrarConexion();

}

public void actualizarEmpleado(int idEmpleado, string nombre, string apellido, string correo, int idSucursal)

{

comando.Connection = objConexionCAD.abrirConeccion();

comando.CommandText = "actualizarEmpleado";

comando.CommandType = CommandType.StoredProcedure;

comando.Parameters.Clear();

comando.Parameters.AddWithValue("@idEmpleado", idEmpleado);

comando.Parameters.AddWithValue("@nombre", nombre);

comando.Parameters.AddWithValue("@apellido", apellido);

comando.Parameters.AddWithValue("@correo", correo);

comando.Parameters.AddWithValue("@idSucursal", idSucursal);

comando.ExecuteNonQuery();

objConexionCAD.cerrarConexion();

}

public void eliminarEmpleado(int idEmpleado)

{

comando.Connection = objConexionCAD.abrirConeccion();

comando.CommandText = "eliminarEmpleado";

comando.CommandType = CommandType.StoredProcedure;

comando.Parameters.Clear();

comando.Parameters.AddWithValue("@idEmpleado", idEmpleado);

comando.ExecuteNonQuery();

objConexionCAD.cerrarConexion();

}

public DataTable consultarEmpleadosUnaSucursal(int idSucursal)

{

comando.Connection = objConexionCAD.abrirConeccion();

comando.CommandText = "consultarEmpleadosUnaSucursal";

comando.CommandType = CommandType.StoredProcedure;

comando.Parameters.Clear();

comando.Parameters.AddWithValue("@idSucursal", idSucursal);

leerTabla = comando.ExecuteReader();

miTabla.Load(leerTabla);

objConexionCAD.cerrarConexion();

return miTabla;

}

}

}

### VentasCAD.cs

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace CapaAccesoDatos

{

public class VentasCAD

{

//miembros atributos

private ConexionCAD objConexionCAD;

private SqlDataReader leerTabla;

private DataTable miTabla;

private SqlCommand comando;

public VentasCAD()

{

objConexionCAD = new ConexionCAD();

miTabla = new DataTable();

comando = new SqlCommand();

}

public DataTable consultarVentas()

{

comando.Connection = objConexionCAD.abrirConeccion();

comando.CommandText = "consultarVentas";

comando.CommandType = CommandType.StoredProcedure;

comando.Parameters.Clear();

leerTabla = comando.ExecuteReader();

miTabla.Load(leerTabla);

objConexionCAD.cerrarConexion();

return miTabla;

}

public DataTable consultarVentasUnEmpleado(int idEmpleado)

{

comando.Connection = objConexionCAD.abrirConeccion();

comando.CommandText = "consultarVentasUnEmpleado";

comando.CommandType = CommandType.StoredProcedure;

comando.Parameters.Clear();

comando.Parameters.AddWithValue("@idEmpleado", idEmpleado);

leerTabla = comando.ExecuteReader();

miTabla.Load(leerTabla);

objConexionCAD.cerrarConexion();

return miTabla;

}

public DataTable consultarVentasUnaSucursal(int idSucursal)

{

comando.Connection = objConexionCAD.abrirConeccion();

comando.CommandText = "consultarVentasUnaSucursal";

comando.CommandType = CommandType.StoredProcedure;

comando.Parameters.Clear();

comando.Parameters.AddWithValue("@idSucursal", idSucursal);

leerTabla = comando.ExecuteReader();

miTabla.Load(leerTabla);

objConexionCAD.cerrarConexion();

return miTabla;

}

public DataTable consultaVentasTodas()

{

comando.Connection = objConexionCAD.abrirConeccion();

comando.CommandText = "consultaVentasTodas";

comando.CommandType = CommandType.StoredProcedure;

comando.Parameters.Clear();

leerTabla = comando.ExecuteReader();

miTabla.Load(leerTabla);

objConexionCAD.cerrarConexion();

return miTabla;

}

public void agregarVenta(int idVenta,float total, DateTime fecha, int idEmpleado)

{

comando.Connection = objConexionCAD.abrirConeccion();

comando.CommandText = "agregarVenta";

comando.CommandType = CommandType.StoredProcedure;

comando.Parameters.Clear();

comando.Parameters.AddWithValue("@idVenta", idVenta);

comando.Parameters.AddWithValue("@total", total);

comando.Parameters.AddWithValue("@fechaVenta", fecha);

comando.Parameters.AddWithValue("@idEmpleado", idEmpleado);

comando.ExecuteNonQuery();

objConexionCAD.cerrarConexion();

}

public void actualizarVenta(int idVenta, float total, DateTime fecha, int idEmpleado)

{

comando.Connection = objConexionCAD.abrirConeccion();

comando.CommandText = "actualizarVenta";

comando.CommandType = CommandType.StoredProcedure;

comando.Parameters.Clear();

comando.Parameters.AddWithValue("@idVenta", idVenta);

comando.Parameters.AddWithValue("@total", total);

comando.Parameters.AddWithValue("@fechaVenta", fecha);

comando.Parameters.AddWithValue("@idEmpleado", idEmpleado);

comando.ExecuteNonQuery();

objConexionCAD.cerrarConexion();

}

public void eliminarVenta(int idVenta)

{

comando.Connection = objConexionCAD.abrirConeccion();

comando.CommandText = "eliminarVenta";

comando.CommandType = CommandType.StoredProcedure;

comando.Parameters.Clear();

comando.Parameters.AddWithValue("@idVenta", idVenta);

comando.ExecuteNonQuery();

objConexionCAD.cerrarConexion();

}

public int getIdVenta()

{

comando.Connection = objConexionCAD.abrirConeccion();

comando.CommandText = "select top 1 idVenta from Venta order by idVenta desc";

comando.CommandType = CommandType.Text;

comando.Parameters.Clear();

object objIdVenta = comando.ExecuteScalar();

objConexionCAD.cerrarConexion();

int idVenta = Convert.ToInt32(objIdVenta);

return idVenta;

}

}

}

### DetalleCAD.cs

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace CapaAccesoDatos

{

public class DetalleCAD

{

//miembros atributos

private ConexionCAD objConexionCAD;

private SqlDataReader leerTabla;

private DataTable miTabla;

private SqlCommand comando;

//metodos para el CRUD y constructor

public DetalleCAD()

{

objConexionCAD = new ConexionCAD();

miTabla = new DataTable();

comando = new SqlCommand();

}

public DataTable consultarDetalles(int idVenta)

{

comando.Connection = objConexionCAD.abrirConeccion();

//comando.CommandText = "consultarDetalles";

//comando.CommandType = CommandType.StoredProcedure;}

comando.CommandText = $"Select descripcion, cantidad, precio from DetalleVenta inner join Producto on DetalleVenta.idProducto = Producto.idProducto where idVenta = {idVenta}";

comando.Parameters.Clear();

comando.Parameters.AddWithValue("@idVenta", idVenta);

leerTabla = comando.ExecuteReader();

miTabla.Load(leerTabla);

objConexionCAD.cerrarConexion();

return miTabla;

}

public void agregarDetalles(int idDetalle, int idProducto, int cantidad, int idVenta)

{

comando.Connection = objConexionCAD.abrirConeccion();

comando.CommandText = "agregarDetalle";

comando.CommandType = CommandType.StoredProcedure;

comando.Parameters.Clear();

comando.Parameters.AddWithValue("@idDetalle", idDetalle);

comando.Parameters.AddWithValue("@idProducto", idProducto);

comando.Parameters.AddWithValue("@cantidad", cantidad);

comando.Parameters.AddWithValue("@idVenta", idVenta);

comando.ExecuteNonQuery();

objConexionCAD.cerrarConexion();

}

public int getIdDetalle()

{

comando.Connection = objConexionCAD.abrirConeccion();

comando.CommandText = "select top 1 idDetalle from DetalleVenta order by idDetalle desc";

comando.CommandType = CommandType.Text;

comando.Parameters.Clear();

object objIdDetalle = comando.ExecuteScalar();

objConexionCAD.cerrarConexion();

int idDetalle = Convert.ToInt32(objIdDetalle);

return idDetalle;

}

}

}

## ProductosCAD.cs

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

//listo

namespace CapaAccesoDatos

{

public class ProductoCAD

{

//miembros atributos

private ConexionCAD objConexionCAD;

private SqlDataReader leerTabla;

private DataTable miTabla;

private SqlCommand comando;

public ProductoCAD()

{

objConexionCAD = new ConexionCAD();

miTabla = new DataTable();

comando = new SqlCommand();

}

public DataTable consultarProductos()

{

comando.Connection = objConexionCAD.abrirConeccion();

comando.CommandText = "consultarProductos";

comando.CommandType = CommandType.StoredProcedure;

comando.Parameters.Clear();

leerTabla = comando.ExecuteReader();

miTabla.Load(leerTabla);

objConexionCAD.cerrarConexion();

return miTabla;

}

public string consultarUnProducto(int idProducto)

{

comando.Connection = objConexionCAD.abrirConeccion();

comando.CommandText = "consultarUnProducto";

comando.CommandType = CommandType.StoredProcedure;

comando.Parameters.Clear();

comando.Parameters.AddWithValue("@idProducto", idProducto);

leerTabla = comando.ExecuteReader();

miTabla.Load(leerTabla);

string resultado = miTabla.ToString();

objConexionCAD.cerrarConexion();

return resultado;

}

public void agregarProducto(int idProducto, string descripcion, float precio, int stock)

{

comando.Connection = objConexionCAD.abrirConeccion();

comando.CommandText = "agregarProducto";

comando.CommandType = CommandType.StoredProcedure;

comando.Parameters.Clear();

comando.Parameters.AddWithValue("@idProducto", idProducto);

comando.Parameters.AddWithValue("@descripcion", descripcion);

comando.Parameters.AddWithValue("@precio", precio);

comando.Parameters.AddWithValue("@stock", stock);

comando.ExecuteNonQuery();

objConexionCAD.cerrarConexion();

}

public void actualizarProducto(int idProducto, string descripcion, float precio, int stock)

{

comando.Connection = objConexionCAD.abrirConeccion();

comando.CommandText = "actualizarProducto";

comando.CommandType = CommandType.StoredProcedure;

comando.Parameters.Clear();

comando.Parameters.AddWithValue("@idProducto", idProducto);

comando.Parameters.AddWithValue("@descripcion", descripcion);

comando.Parameters.AddWithValue("@precio", precio);

comando.Parameters.AddWithValue("@stock", stock);

comando.ExecuteNonQuery();

objConexionCAD.cerrarConexion();

}

public void eliminarProducto(int idProducto)

{

comando.Connection = objConexionCAD.abrirConeccion();

comando.CommandText = "eliminarProducto";

comando.CommandType = CommandType.StoredProcedure;

comando.Parameters.Clear();

comando.Parameters.AddWithValue("@idProducto", idProducto);

comando.ExecuteNonQuery();

objConexionCAD.cerrarConexion();

}

public void venderProducto(int idProducto)

{

comando.Connection = objConexionCAD.abrirConeccion();

comando.CommandText = "venderUnProducto";

comando.CommandType = CommandType.StoredProcedure;

comando.Parameters.Clear();

comando.Parameters.AddWithValue("@idProducto", idProducto);

comando.ExecuteNonQuery();

objConexionCAD.cerrarConexion();

}

public int getPrecio(int idProducto)

{

comando.Connection = objConexionCAD.abrirConeccion();

comando.CommandText = $"select precio from Producto where idProducto = {idProducto}";

comando.CommandType = CommandType.Text;

comando.Parameters.Clear();

object objPrecio = comando.ExecuteScalar();

objConexionCAD.cerrarConexion();

int precio = Convert.ToInt32(objPrecio);

return precio;

}

}

}