

WPF

CONEXION

//Es necesario crear una clase conexión

namespace ControlAspirantes

{

class Conexion

{

public static SqlConnection agregaConexion() {

SqlConnection con;

try

{

con = new SqlConnection("Data Source=112SALAS20;Initial

Catalog=baseAspirantes;User ID=sa");

con.Open();

MessageBox.Show("Si pude conectarme");

}

catch (Exception e)

{

con = null;

MessageBox.Show("No se pudo");

}

return con;

}

}

}

LLENAR COMBO BOX → Loaded

//Función en conexion

public static void llenarCombo(ComboBox cb)

{

SqlConnection con;

SqlDataReader rd;

try

{

```

con = Conexion.agregaConexion();
SqlCommand cmd = new SqlCommand("select nombre from programas", con);
rd = cmd.ExecuteReader();
while (rd.Read())
{
    cb.Items.Add(rd.getString(0)); //o rd["nombre"].ToString();
    //El 0 es la columna como lo hayas puesto en el query
    //Solo 1 dato
}
cb.SelectedIndex = 0;
rd.Close();
con.Close();
} catch (Exception e)
{
    MessageBox.Show("No se pudo llenar el combo"+e);
}
}

//Funcion para mandar a llamar el llenar combo en la venta donde se encuentre.Y
PageLoaded->
private void Window_Loaded(object sender, RoutedEventArgs e)
{
    //este combo se llena a manita
    cbGrado.Items.Add(1);
    cbGrado.Items.Add(2);
    cbGrado.Items.Add(0);
    //Abrir la conexion
    Conexion.llenar_combo(cbPrograma);
    //Si no fuera static sería:
    Conexion c= new Conexion();
    c.llenarCombo(cbPrograma);
}

```

MANDAR A LLAMAR UNA VENTANA

```

private void bModificacion_Click(object sender, RoutedEventArgs e)
{
    Modificar m = new Modificar();
    m.Show();
    this.Hide();
}

```

REGRESAR

```

private void btRegresar_Click(object sender, RoutedEventArgs e)
{
    MainWindow main = new MainWindow();
}

```

```

        main.Show();
        this.Hide();
    }

```

QUERY SQL SERVER (crear tabla)

```

create table programas(
clavePrograma int primary key,
nombre varchar (50) not null
)
create table aspirante(
idAspirante int primary key,
nombre varchar (50) not null,
apellidoPaterno varchar (50) not null,
apellidoMaterno varchar (50) not null,
sexo varchar (20) not null,
fechaNacimiento date not null,
correo varchar (50) not null,
grado int not null,
ClavePrograma int references programas
--Foreing key
)

```

CLASE PARA MÉTODOS

```

namespace RegistroAlumno
{
    class Aspirante
    {
        public int idAspirante,grado,clavePrograma;
        public String nombre, apellidoPaterno, apellidoMaterno, sexo, fechaNacimiento, correo;

        //CONSTRUCTOR PARA ALTA
        public Aspirante(int idAspirante, int grado, int clavePrograma, string nombre, string
apellidoPaterno, string apellidoMaterno, string sexo, string fechaNacimiento, string correo)
        {
            this.idAspirante = idAspirante;
            this.grado = grado;
            this.clavePrograma = clavePrograma;
            this.nombre = nombre;
            this.apellidoPaterno = apellidoPaterno;
            this.apellidoMaterno = apellidoMaterno;
            this.sexo = sexo;
            this.fechaNacimiento = fechaNacimiento;
            this.correo = correo;
        }
    }
}

```

```
}
```

ALTA

```
public String AltaAlumno(Aspirantes a) {
    String res;

    try
    {
        SqlConnection con;
        con = Conexion.agregaConexion();
        //creamos el contador
        String queryMax = "select top(1) idAspirante from aspirante order by idAspirante
desc";

        SqlCommand cmd = new SqlCommand(queryMax, con);
        SqlDataReader rd = cmd.ExecuteReader();

        if (rd.HasRows)
        {
            rd.Read();
            int maximo = rd.GetInt32(0) + 1; //Dame la lectura 0
            //Fijate en el tipo de entero (16 o 32)
            rd.Close();
            String query = string.Format("insert into aspirante values
('{0}','{1}','{2}','{3}','{4}','{5}','{6}','{7}','{8}') ", maximo, a.nombre, a.apellidoPaterno,
a.apellidoMaterno, a.sexo, a.fechaNacimiento, a.correo, a.grado, a.clavePrograma);

            SqlCommand cmd2 = new SqlCommand(query, con);
            cmd2.ExecuteNonQuery();
            res = "Alta exitosa" ;
        }
        else
        {
            //f = "NO Tiene filas";
            String query = string.Format("insert into aspirante values
('{0}','{1}','{2}','{3}','{4}','{5}','{6}','{7}','{8}') ", 1, a.nombre, a.apellidoPaterno, a.apellidoMaterno,
a.sexo, a.fechaNacimiento, a.correo, a.grado, a.clavePrograma);
            SqlCommand cmd2 = new SqlCommand(query, con);
            cmd2.ExecuteNonQuery();
            res = "Alta exitosa" ;
        }
    }
}
```

```

    }
    con.Close();
}
catch (Exception e)
{
    res = "Alta NO exitosa" + e.Message;
}
return res;
}

```

EN EL BOTÓN MANDAR A LLAMARLO →

```

private void bAlta_Click(object sender, RoutedEventArgs e)
{
    Aspirantes a = new Aspirantes(tbNombre.Text, tbApellidoPaterno.Text,
tbApellidoMaterno.Text, tbSexo.Text, tbFechaDeNacimiento.Text, tbCorreo.Text,
(int)cbGrado.SelectedItem, cbPrograma.SelectedIndex);
    //SelectedItem jala el valor que esta adentro pero como objeto por eso lo casteo a int
    //Programa solo quiero su clave de programa
    MessageBox.Show(a.AltaAlumno(a));
}

```

OCULTAR VENTANA

```

private void bRegresar_Click(object sender, RoutedEventArgs e)
{
    this.Hide();
    MainWindow m = new MainWindow();
    m.Show();
}

```

LLENAR DATAGRID → CLASE “alumno”, “conexion”

```

public String generarReporte(int clavePrograma, DataGrid dg) {
    String res = " ";
    try
    {
        SqlConnection con = Conexion.agregaConexion();
        String query = String.Format("select * from aspirante where clavePrograma={0}",
clavePrograma);
        SqlCommand com = new SqlCommand(query, con);
    }
}

```

```

        SqlDataReader dr = com.ExecuteReader();
        dg.ItemsSource = dr;
        //Lo igualo al dr porqu eme esta regresando una tabla
        //ES UN Atributo
        //Items son los datos
        res="Éxito";
    }
    catch(Exception e) {
        res="Falló";
    }
    return res;
}

```

EN EL BOTÓN MANDAR A LLAMARLO →

```

private void bGenerar_Click(object sender, RoutedEventArgs e)
{
    Aspirantes a;
    a = new Aspirantes();
    a.generarReporte(cbProgramas.SelectedIndex,dGReporte);
}

```

BAJA

```

public String darDeBaja(String nombre) {
    String res = "";
    try {
        SqlConnection con = Conexion.agregaConexion();
        String query = String.Format("delete from aspirante where nombre='{0}'", nombre);
        SqlCommand cmd = new SqlCommand(query,con);
        cmd.ExecuteNonQuery();
        con.Close();
        res = "Baja exitosa";
    }
    catch (Exception e) {
        res = "No se pudo dar de baja"+e.ToString();
    }
    return res;
}

```

EN EL BOTÓN MANDAR A LLAMARLO →

```

private void btBaja_Click(object sender, RoutedEventArgs e)
{
    Aspirantes a = new Aspirantes();
}

```

```

        MessageBox.Show(a.darDeBaja(cbClaveUica.SelectedItem.ToString()));
    }

```

MODIFICAR

```

public int modificar(int clavePrograma,String nombre)
{
    int res;
    SqlConnection con;
    con = Conexion.agregaConexion();
    SqlCommand cmd = new SqlCommand(String.Format("update aspirante set
clavePrograma={0} where nombre='{1}'", clavePrograma,nombre),con);
    res = cmd.ExecuteNonQuery();
    if (res > 0)
        MessageBox.Show("Se modifiko :D");
    else
        MessageBox.Show("No se modifiko D:");

    return res;
}

```

EN EL BOTÓN MANDAR A LLAMARLO →

```

private void bAceptar_Click(object sender, RoutedEventArgs e)
{
    Aspirantes a = new Aspirantes();
    a.modificar(cbProgramas.SelectedIndex, cbAspirantes.SelectedItem.ToString());
}

```

BUSCAR

```

public static List<Alumno> buscar(String a)
{
    Alumno al;
    List<Alumno> lis = new List<Alumno>();
    SqlConnection con = Conexion.agregarConexion();
    SqlCommand cmd = new SqlCommand(String.Format("select * from alumno where
nombre like '%{0}%", a), con);
    SqlDataReader lec = cmd.ExecuteReader();
    while (lec.Read())
    {
        al = new Alumno();
        al.claveUnica = lec.GetInt16(0);
        al.nombre = lec.GetString(1);
    }
}

```



```

        al.sexo = lec.GetString(2);
        al.correo = lec.GetString(3);
        al.semestre = lec.GetInt16(4);
        al.programa = lec.GetInt16(5);
        lis.Add(al);
    }
    con.Close();
    return lis; }

```

EN EL BOTÓN MANDAR A LLAMARLO →

```

{
    Alumno a = new Alumno(); //constructor vacío
    dgDatos.ItemsSource=a.buscarAlumno(tbNombre.Text);
}

```

EJERCICIO EXAMEN FER Y YO

En clase Conexión:

```

namespace Ahorro_Examen2018 {
    class Conexion {
        public static SqlConnection agregaConexion() {
            SqlConnection con;
            try {

                con = new SqlConnection("Data Source=112SALAS19;Initial
Catalog=Consar;User ID=sa;Password=sqladmin");
                con.Open();
                MessageBox.Show("Si pude conectarme");
            }
            catch (Exception e) {
                con = null;
                MessageBox.Show("No se pudo"+e.ToString());
            }
            return con;
        }

        public static void llenarCombo(ComboBox cb) {
            SqlConnection con;
            SqlDataReader rd;
            try {
                con = Conexion.agregaConexion();
                SqlCommand cmd = new SqlCommand("select distinct persona.NomPersona
from ahorro,persona where persona.IdPersona=ahorro.idPersona", con);
                rd = cmd.ExecuteReader();

```

```

        while (rd.Read()) {
            cb.Items.Add(rd.GetString(0));
        }
        cb.SelectedIndex = 0;
        rd.Close();
        con.Close();
    }
    catch (Exception e) {
        MessageBox.Show("No se pudo llenar el combo" + e);
    }
}
}
}

```

En el Main Window:

```

namespace Ahorro_Examen2018 {

    public partial class MainWindow : Window {
        public MainWindow() {
            InitializeComponent();
        }

        private void Window_Loaded(object sender, RoutedEventArgs e) {
            Conexion.LlenarCombo(cbNombre);
        }

        private void btAlta_Click(object sender, RoutedEventArgs e)

```

```

{

    Cliente a = new
Cliente(Int32.Parse(txMonto.Text),txMes.Text,Int32.Parse(txAnio.Text),cbNombre.Text.To
String());
    MessageBox.Show(a.AltaCliente(a));

}

private void btNext_Click(object sender, RoutedEventArgs e)
{
    Aportaciones a = new Aportaciones();
    this.Close();
    a.Show();
}
}
}

```

En Cliente:

```

namespace Ahorro_Examen2018 {
    class Cliente {
        public int monto { get; set; }
        public String mes{ get; set; }
        public int anio{ get; set; }
        public String nombre{ get; set; }

        public Cliente() {

        }

        public Cliente(int monto, string mes, int anio,string nombre)
        {
            this.monto = monto;
            this.mes = mes;
            this.anio = anio;
            this.nombre = nombre;
        }

        // ALTA

```

```

public String AltaCliente(Cliente a) {
    String res="hola";

    try {
        SqlConnection con;
        con = Conexion.agregaConexion();

        String queryMax = "select max(idPersona) from persona";
        SqlCommand cmd = new SqlCommand(queryMax, con);
        SqlDataReader rd = cmd.ExecuteReader();

        rd.Read();
        int maximo;
        MessageBox.Show("El valor maximo: "+ rd.GetInt16(0));
        maximo =rd.GetInt16(0)+1;
        rd.Close();

        //Query para acceder a ID de la persona
        String queryP = string.Format("select IdPersona from persona where
NomPersona='{0}'",a.nombre.Trim());
        SqlCommand cmd3 = new SqlCommand(queryMax, con);
        SqlDataReader rd2 = cmd3.ExecuteReader();
        rd2.Read();
        int id = rd2.GetInt16(0);
        MessageBox.Show("El id: " + rd2.GetInt16(0));
        rd2.Close();

        String query = string.Format("insert into ahorro values ('{0}','{1}','{2}','{3}','{4}'),",
maximo, a.monto, a.mes, a.anio,id);
        SqlCommand cmd2 = new SqlCommand(query, con);
        cmd2.ExecuteNonQuery();
        res = "Alta exitosa";

        con.Close();
    }
    catch (Exception e) {
        res = "Alta NO exitosa" + e.Message.ToString();
    }
    return res;
}

}

public String generarReportePorMonto(string nombre, DataGridView dg)

```

```

{
    String res = " ";
    try
    {
        SqlConnection con = Conexion.agregaConexion();
        String query = String.Format("select ahorro.monto from ahorro,persona where
ahorro.idPersona=persona.idPersona and persona.nomPersona='{0}'", nombre);
        SqlCommand com = new SqlCommand(query, con);
        SqlDataReader dr = com.ExecuteReader();
        dg.ItemsSource = dr;

        res = "Éxito";
    }
    catch (Exception e)
    {
        res = "Falló";
    }
    return res;
} //consultar

```

```

public String generarReporteTotalMonto(string nombre, DataGridView dg)
{
    String res = " ";
    try
    {
        SqlConnection con = Conexion.agregaConexion();
        String query = String.Format("select sum(monto) as Total from ahorro,persona
where ahorro.idPersona=persona.idPersona and persona.nomPersona='{0}'", nombre);
        SqlCommand com = new SqlCommand(query, con);
        SqlDataReader dr = com.ExecuteReader();
        dg.ItemsSource = dr;

        res = "Éxito";
    }
    catch (Exception e)
    {
        res = "Falló";
    }
    return res;
} //consultar total

```

```

public void ahorroRep(string nombre)
{

```

```

        SqlConnection con = Conexion.agregaConexion();
        String query = String.Format("select sueldoBase.monto from sueldoBase, persona
where sueldoBase.idPersona=persona.idPersona and persona.nomPersona='{0}'",
nombre);
        SqlCommand com = new SqlCommand(query, con);
        SqlDataReader dr = com.ExecuteReader();
        dr.Read();
        int sueldoBase = dr.GetInt32(0);
        dr.Close();
        double porSueBase = 0;
        porSueBase = sueldoBase * (.13);

        String query2 = String.Format("select avg(monto) from ahorro, persona where
ahorro.idPersona=persona.idPersona and persona.nomPersona='{0}'", nombre);
        SqlCommand com2 = new SqlCommand(query, con);
        SqlDataReader dr1 = com2.ExecuteReader();
        dr1.Read();
        int avgMonto = dr1.GetInt32(0);
        dr1.Close();

        if (porSueBase > avgMonto)
            MessageBox.Show("Continúe ahorrando así!");
        else
            MessageBox.Show("Le falta ahorrar " + (avgMonto - porSueBase));
    }
}
}

```

En la otra ventana:

The screenshot shows a Windows application window titled "Aportaciones". Inside the window, there is a dropdown menu at the top left. Below the dropdown menu is a large, empty rectangular area, which appears to be a placeholder for a list or a data grid. To the right of this area, there are three buttons stacked vertically: "% de ahorro", "Total de montos", and "Consultar cada monto".

namespace Ahorro_Examen2018

```

{
    public partial class Aportaciones : Window
    {
        public Aportaciones()
        {
            InitializeComponent();

            private void Window_Loaded(object sender, RoutedEventArgs e)
            {
                Conexion.llenarCombo(cbNombre);
            }

            private void btMonto_Click(object sender, RoutedEventArgs e)
            {
                Cliente a;
                a = new Cliente();
                a.generarReportePorMonto(cbNombre.Text.ToString(), dgAport);
            }

            private void btTotMontos_Click(object sender, RoutedEventArgs e)
            {
                Cliente a;
                a = new Cliente();
                a.generarReporteTotalMonto(cbNombre.Text.ToString(), dgAport);
            }

            private void btAhorro_Click(object sender, RoutedEventArgs e)
            {
                Cliente a;
                a = new Cliente();
                a.ahorroRep(cbNombre.Text.ToString());
            }
        }
    }
}

```

Examen DAI2918 SOLUCIÓN DEL OTRO TIPIN (+ rápida no hace clase de cliente)

En conexión:

```

class Conexion
{
    public static SqlConnection agregarConexion()

```

```

    {
        SqlConnection conexion = new SqlConnection("Data Source=localhost;Initial
Catalog=consar;Persist Security Info=True;User ID=sa;Password=sqladmin");
        conexion.Open();
        return conexion;
    }//agregar Conexion

    public static void llenarComboNombre(ComboBox cb)
    {
        try
        {
            SqlConnection con = agregarConexion();
            string q = "select NomPersona from persona";
            SqlCommand cmd = new SqlCommand(q, con);
            SqlDataReader lec = cmd.ExecuteReader();
            while (lec.Read())
            {
                cb.Items.Add(lec.GetSqlString(0));
            }
            cb.SelectedIndex = 0;
            con.Close();
        }
        catch (Exception ex)
        {
            MessageBox.Show("No se puede llenar el combo: " + ex);
        }
    }//llenar combo
} //Class Conexion

```

En el Main Activity:

```

public MainWindow()
{
    InitializeComponent();
}

private void Window_Loaded(object sender, RoutedEventArgs e)
{
    Conexion.llenarComboNombre(comboBox);
}

private void btAlta_Click(object sender, RoutedEventArgs e)
{
    try

```



```

    {
        int res;
        SqlConnection con = Conexion.agregarConexion();
        string s = "insert into ahorro(idAhorro, monto, mes, año, idPersona) values
('"+txClave.Text+ "', '"+txMonto.Text+ "', '"+txMes.Text+ "', '"+txAño.Text+ "',
 '"+txClavePersona.Text+"'");
        SqlCommand cmd = new SqlCommand(s, con);
        res = cmd.ExecuteNonQuery();
        if (res != 0)
            MessageBox.Show("Se dio de alta el ahorro");
        else
            MessageBox.Show("No se dio de alta el ahorro");
    }catch(Exception ex)
    {
        MessageBox.Show("Error: "+ex);
    }
}

private void btSiguiente_Click(object sender, RoutedEventArgs e)
{
    Reporte r = new Reporte();
    r.Show();
    this.Hide();
}

```

En ventana Reporte:

```

{
    InitializeComponent();
}

private void Window_Loaded(object sender, RoutedEventArgs e)
{
    Conexion.llenarComboNombre(comboBox1);
}

private void button1_Click(object sender, RoutedEventArgs e)
{
    try
    {
        SqlConnection con = Conexion.agregarConexion();
        string s = "select * from ahorro, persona where ahorro.idPersona =
persona.idPersona and persona.NomPersona = '" + comboBox1.SelectedItem.ToString()
+ "'";
    }
}

```

```

        SqlCommand cmd = new SqlCommand(s, con);
        SqlDataReader lector = cmd.ExecuteReader();
        dgReporte.ItemsSource = lector;
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error al generar el reporte: " + ex.Message);
    }
    try
    {
        int total = 0;
        SqlConnection con = Conexion.agregarConexion();
        string q = "select monto from ahorro, persona where ahorro.idPersona =
persona.idPersona and persona.NomPersona = '" + comboBox1.SelectedItem.ToString()
+ "'";

        SqlCommand otro = new SqlCommand(q, con);
        SqlDataReader lee = otro.ExecuteReader();
        while (lee.Read())
        {
            total = total + lee.GetInt32(0);
        }
        MessageBox.Show("El monto total ahorrado es: " + total);
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error al sumar: " + ex.Message);
    }
}

private void button2_Click(object sender, RoutedEventArgs e)
{
    try
    {
        int m;
        SqlConnection con = Conexion.agregarConexion();
        string s = "select sueldoBase.monto from sueldobase, persona where
sueldobase.idPersona = persona.idPersona and persona.NomPersona = '" +
comboBox1.SelectedItem.ToString() + "'";

        SqlCommand cmd = new SqlCommand(s, con);
        SqlDataReader l = cmd.ExecuteReader();
        l.Read();
        m = l.GetInt32(0);
        l.Close();

        string q = "select monto from ahorro, persona where ahorro.idPersona =
persona.idPersona and persona.NomPersona = '" + comboBox1.SelectedItem.ToString()
+ "'";
    }
}

```

```

        SqlCommand cm = new SqlCommand(q, con);
        int total = 0, cuenta=1;
        SqlDataReader lec = cm.ExecuteReader();
        while (lec.Read())
        {
            total = total + lec.GetInt32(0);
            cuenta = cuenta + 1;
        }
        double promedio;
        promedio = total / cuenta;
        double trece = m * 0.13;
        if (promedio > trece)
            MessageBox.Show("Has ahorrado correctamente, sigue así");
        else
        {
            MessageBox.Show("Te hace falta ahorrar mensualmente:
" + (trece - promedio));
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error: " + ex);
    }
}

private void button3_Click(object sender, RoutedEventArgs e)
{
    MainWindow m = new MainWindow();
    m.Show();
    this.Hide();
}

```

ANDROID TIPIN:

Crear new java class: llamada Conexión en vez de SQLiteOpenHelper

```

public class Conexion extends SQLiteOpenHelper{

    public Conexion(Context context, String name, SQLiteDatabase.CursorFactory factory,
int version) {
        super(context, name, factory, version);
    }

    @Override

```

```

    public void onCreate(SQLiteDatabase sqLiteDatabase) {
        sqLiteDatabase.execSQL("create table persona(idPersona integer primary key,
NomPersona text, edad, integer, idEmpresa integer)");
        sqLiteDatabase.execSQL("create table ahorro(idAhorro integer primary key, monto
integer, mes text, año integer, idPersona integer)");
    }

    @Override
    public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {
        sqLiteDatabase.execSQL("drop table if exists persona");
        sqLiteDatabase.execSQL("drop table if exists ahorro");
    }
}

```

En el MAIN ACTIVITY

```

public class MainActivity extends AppCompatActivity {

    EditText id, nombre, edad, empresa;
    TextView vista;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        id = (EditText)findViewById(R.id.etPersona);
        nombre = (EditText)findViewById(R.id.etNombre);
        edad = (EditText)findViewById(R.id.etEdad);
        empresa = (EditText)findViewById(R.id.etEmpresa);
        vista = (TextView)findViewById(R.id.tvAhorro);
    }

    public void sigue(View view){
        Intent i = new Intent(this, Main2Activity.class);
        Bundle bundle = new Bundle();
        bundle.putString("id", id.getText().toString());
        i.putExtras(bundle);
        startActivity(i);
    }

    public void limpia(View view){
        id.setText("");
        nombre.setText("");
        edad.setText("");
        empresa.setText("");
    }

    public void alta(View view){
        Conexion con = new Conexion(this, "Ahorros", null, 1);
        SQLiteDatabase db = con.getWritableDatabase();
        ContentValues registro = new ContentValues();

```

```

registro.put("idPersona", id.getText().toString());
registro.put("NomPersona", nombre.getText().toString());
registro.put("edad", edad.getText().toString());
registro.put("idEmpresa", empresa.getText().toString());
if(db.insert("persona", null, registro)>0){
    Toast.makeText(this, "Se agregó la persona", Toast.LENGTH_LONG).show();
}
else{
    Toast.makeText(this, "No se agregó la persona", Toast.LENGTH_LONG).show();
}
}

public void ver(View view){
    Conexion con = new Conexion(this, "Ahorros", null, 1);
    SQLiteDatabase db = con.getWritableDatabase();
    Cursor cur = db.rawQuery("select NomPersona, edad, idEmpresa from persona where
idPersona = '"+id.getText().toString()+"'", null);
    cur.moveToNext();
    nombre.setText(cur.getString(0));
    edad.setText(cur.getString(1));
    empresa.setText(cur.getString(2));
    cur.close();
}

public void ahorros(View view){
    Conexion con = new Conexion(this, "Ahorros", null, 1);
    SQLiteDatabase db = con.getWritableDatabase();
    int suma = 0;
    Cursor cur = db.rawQuery("select monto from ahorro where idPersona =
'"+id.getText().toString()+"'", null);
    while(cur.moveToNext()){
        suma = suma + cur.getInt(0);
    }
    cur.close();
    vista.setText("El ahorro total es: "+suma);
}
}

```

Interface del Main Activity

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/etPersona"
        android:hint="ID Persona"/>

```

```
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/etNombre"
    android:hint="Nombre"/>
```

```
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/etEdad"
    android:hint="Edad"/>
```

```
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/etEmpresa"
    android:hint="ID Empresa"/>
```

```
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Dar de alta persona"
    android:id="@+id/btAltaPersona"
    android:onClick="alta"/>
```

```
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Siguiente"
    android:id="@+id/btSiguiente"
    android:onClick="sigue"/>
```

```
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Limpiar"
    android:id="@+id/btLimpia"
    android:onClick="limpia"/>
```

```
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Ver datos"
    android:id="@+id/btVer"
    android:onClick="ver"/>
```

```
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Suma de ahorros"
```

```

        android:id="@+id/btAhorro"
        android:onClick="ahorros"/>

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/tvAhorro"/>

</LinearLayout>

```



En el MAIN ACTIVITY 2 (SEGUNDA ACTIVITY)

```

public class Main2Activity extends AppCompatActivity {

    EditText ahorro1, monto1, mes, año, persona;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);
        ahorro1 = (EditText) findViewById(R.id.etAhorro);
        monto1 = (EditText) findViewById(R.id.etMonto);
        mes = (EditText) findViewById(R.id.etMes);
        año = (EditText) findViewById(R.id.etAño);
    }
}

```

```

        persona = (EditText)findViewById(R.id.etIdPersona);
        Bundle datos = this.getIntent().getExtras();
        String este = datos.getString("id");
        persona.setText(este);
    }

    public void limpia(View view){
        ahorrol.setText("");
        montol.setText("");
        mes.setText("");
        año.setText("");
        persona.setText("");
    }

    public void alta(View view){
        Conexion con = new Conexion(this, "Ahorros", null, 1);
        SQLiteDatabase db = con.getWritableDatabase();
        ContentValues registro = new ContentValues();
        registro.put("idAhorro", ahorrol.getText().toString());
        registro.put("monto", montol.getText().toString());
        registro.put("mes", mes.getText().toString());
        registro.put("año", año.getText().toString());
        registro.put("idPersona", persona.getText().toString());
        if(db.insert("ahorro", null, registro)>0){
            Toast.makeText(this, "Se agregó el ahorro", Toast.LENGTH_LONG).show();
        }
        else{
            Toast.makeText(this, "No se agregó el ahorro", Toast.LENGTH_LONG).show();
        }
    }
}

```

INTERFACE MAIN ACTIVITY 2

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="ID Ahorro"
        android:id="@+id/etAhorro"/>

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Monto"
        android:id="@+id/etMonto"/>

```



```
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Mes"
    android:id="@+id/etMes"/>
```

```
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Año"
    android:id="@+id/etAño"/>
```

```
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="ID Persona"
    android:id="@+id/etIdPersona"/>
```

```
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Dar de alta ahorro"
    android:id="@+id/btAltaAhorro"
    android:onClick="alta"/>
```

```
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Limpiar"
    android:id="@+id/btLimpia"
    android:onClick="limpia"/>
```

```
</LinearLayout>
```

My Application

ID Ahorro

Monto

Mes

Año

ID Persona

DAR DE ALTA AHORRO

LIMPIAR

WEB

TURBO DE CONEXIÓN

```
protected OdbcConnection conectarBD() {
    String stringConexion = "Driver={SQL Server Native Client 11.0};
    Server=DESKTOP-OFMNE4A;Uid=sa;Pwd=sqladmin;Database=holis";
    try {
        OdbcConnection conexion = new OdbcConnection(stringConexion);
        conexion.Open();
        lbResultados.Text = "conexion exitosa";
        return conexion;
    }
    catch (Exception ex) {
        lbResultados.Text = ex.StackTrace.ToString();
    }
}
```

```

        return null;
    }
}

```

Para las otras páginas: quitar label. Ejemplo:

```

OdbcConnection conexion = new OdbcConnection(stringConexion);
conexion.Open();
//lbError.Text = "conexion exitosa";
return conexion;

```

ALTA

```

protected void btRegistrar_Click(object sender, EventArgs e) {

```

```

    OdbcConnection con = conectarBD();

```

```

    if (con != null) {
        //
        String query2 = "select max(id) from datosaAlumno18";
        OdbcCommand sql2 = new OdbcCommand(query2, con);
        int folio = Int16.Parse(sql2.ExecuteScalar().ToString()) + 1;
        //

```

```

        String query = "insert into datosaAlumno18(id, nombre, contra, email) values ('" + folio +
        "', '" + txNombre.Text + "', '" + txContra.Text + "', '" + txCorreo.Text + "')";
        OdbcCommand sql = new OdbcCommand(query, con);
        sql.ExecuteNonQuery();

```

```

        Response.Redirect("Pagina2.aspx");
        con.Close();

```

```

    }
}

```

PARA LOS LINKS

Usar  LinkButton

```

protected void LinkBaja_Click(object sender, EventArgs e)
{
    Response.Redirect("Pagina2.aspx");
}

```

DROP DOWN LIST

```

protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack) //Recargue la pagina
    {
        OdbcConnection miConexion = conectarBD();
        if (miConexion != null)
        {
            String query = "select id,nombre from datosaAlumno18 except (select id,nombre from datosaAlumno18 where id=0)";
            OdbcCommand cmd = new OdbcCommand(query,miConexion);
            OdbcDataReader rd;
            rd = cmd.ExecuteReader();
            //Todo lo que tengo que hacer como "metodo"
            DDLista.DataSource = rd; //Siempre en una herramienta debe de venir un Data
Source
            DDLista.DataTextField = "nombre"; //Busca por nombre
            DDLista.DataValueField = "id"; // Busca por id
            DDLista.DataBind(); //Siempre cierro con este, liga los datos al DropDownList
            //
        }
        miConexion.Close();
    }
}

```

BAJA

```

protected void Button1_Click(object sender, EventArgs e)
{
    OdbcConnection con = conectarBD();
    if (con != null) {
        String query = "delete from datosaAlumno18 where id=" + DDLista.SelectedValue; //
        encuentro el valor seleccionado
        OdbcCommand com = new OdbcCommand(query, con);
        com.ExecuteNonQuery();
    }
    Response.Redirect("Pagina2.aspx");
}

```

MODIFICAR

```

protected void Button1_Click(object sender, EventArgs e)
{
    OdbcConnection con = conectarBD();
    if(con != null)

```

```

{
    OdbcCommand com;
    if (CheckBox1.Checked)
    {
        if (CheckBox2.Checked)
        {
            if (CheckBox3.Checked)
            {
                String query = "update datosaAlumno18 set email=" + TextBox1.Text + ",
contra=" + TextBox2.Text + ", nombre =" + TextBox3.Text + " where id=" +
DropDownList1.SelectedValue;
                com = new OdbcCommand(query, con);
                com.ExecuteNonQuery();
            }
            else
            {
                String query = "update datosaAlumno18 set email=" + TextBox1.Text + ",
contra=" + TextBox2.Text + " where id=" + DropDownList1.SelectedValue;
                com = new OdbcCommand(query, con);
                com.ExecuteNonQuery();
            }
        }
        else if (CheckBox3.Checked)
        {
            String query = "update datosaAlumno18 set email=" + TextBox1.Text + ", nombre
=" + TextBox3.Text + " where id=" + DropDownList1.SelectedValue;
            com = new OdbcCommand(query, con);
            com.ExecuteNonQuery();
        }
        else
        {
            String query = "update datosaAlumno18 set email=" + TextBox1.Text + " where
id=" + DropDownList1.SelectedValue;
            com = new OdbcCommand(query, con);
            com.ExecuteNonQuery();
        }
    }
    else if (CheckBox2.Checked)
    {
        if (CheckBox3.Checked)
        {
            String query = "update datosaAlumno18 set contra=" + TextBox2.Text + ", nombre
=" + TextBox3.Text + " where id=" + DropDownList1.SelectedValue;

```

```

        com = new OdbcCommand(query, con);
        com.ExecuteNonQuery();
    }
    else
    {
        String query = "update datosaAlumno18 set contra='" + TextBox2.Text + "' where
id=" + DropDownList1.SelectedValue;
        com = new OdbcCommand(query, con);
        com.ExecuteNonQuery();
    }
    }else if (CheckBox3.Checked)
    {
        String query = "update datosaAlumno18 set nombre =" + TextBox3.Text + " where
id=" + DropDownList1.SelectedValue;
        com = new OdbcCommand(query, con);
        com.ExecuteNonQuery();
    }
    Response.Redirect("Pagina3.aspx");
}
}

```

[Alta](#) | [Baja](#)

Clase DAI-Cambios informacion del alumno

Alumno

Correo Electronico ☐ [CheckBox1]

Contraseña ☐ [CheckBox2]

Nombre ☐ [CheckBox3]

GRIDVIEW (TABLA)

```

protected void Page_Load(object sender, EventArgs e)
{

```

```

if (!IsPostBack) //Recargue la pagina
{
    OdbcConnection miConexion = conectarBD();
    if (miConexion != null) {
        String query = "select * from datosaAlumno18 except (select * from datosaAlumno18
where id=0)";
        OdbcCommand com = new OdbcCommand(query, miConexion);
        OdbcDataReader rd=com.ExecuteReader();
        GridView1.DataSource = rd;
        GridView1.DataBind();
        miConexion.Close();
    }
}
}
}

```

INICIO DE SESIÓN

a. Iniciar sesión

```

protected void btPagina2_Click(object sender, EventArgs e)
{
    OdbcConnection miConexion = conectarBD();
    if (miConexion != null)
    {
        string query = "select claveU from usuario where email="
+ txUsuario.Text + "and password = " + txContraseña.Text + """;
        OdbcCommand sql = new OdbcCommand(query, miConexion);
        OdbcDataReader lector = sql.ExecuteReader();
    }
}

```

```
        if (lector.HasRows)
        {
            lector.Read();
            Session["CU"] = lector.GetInt16(0).ToString();
            Response.Redirect("Pagina2.aspx");
        }
    }
    else
    {
        lbContador.Text = "El usuario no se encuentra";
    }
}
```


g. Llenar radioButton

Abrir con ▼

```
protected void Page_Load(object sender, EventArgs e)
{
    if (RadioButtonList1.Items.Count == 0)
    {
        RadioButtonList1.Items.Add("Vainilla");
        RadioButtonList1.Items.Add("Fresa");
        RadioButtonList1.Items.Add("Limón");
        RadioButtonList1.Items.Add("Chocolate");
    }
}
```

h. Llenar checkBox

```
protected void Page_Load(object sender, EventArgs e)
{
    if (CheckBoxList1.Items.Count == 0)
    {
        CheckBoxList1.Items.Add("Americano");
        CheckBoxList1.Items.Add("Capuchino");
        CheckBoxList1.Items.Add("Late");
        CheckBoxList1.Items.Add("Mocha");
    }
}
```

4. Session

```
protected void btPag2_Click(object sender, EventArgs e)
{
    Session["sesion"] = txUsuario.Text;
}

protected void Page_Load(object sender, EventArgs e)
{
    lbSession.Text = Session["sesion"].ToString();
}
```

-----COCHINADAS-----

LOCAL HOST

"Driver={SQL Server Native Client
11.0};Server=localhost;Uid=sa;Pwd=SQLadmin18;Database=ejercicio2;"

CALCULAR

```
protected void bCalcular_Click(object sender, EventArgs e)
{
```

```

OdbcConnection con=conectarBD();

String nombreVendedor = Session["nombre"].ToString();
if (con != null)
{
    String query = "select sueldoBase from informacion$ where vendedor='" +
nombreVendedor + "'";
    OdbcCommand cmd = new OdbcCommand(query, con);
    try
    {
        OdbcDataReader rd = cmd.ExecuteReader();
        if (rd.HasRows)
        {
            String value = rd.GetString(0);
            float sueldoBase = Convert.ToSingle(value);
            int montoVenta = int.Parse(tbVentas.Text);

            double comision = 0;
            if (montoVenta >= 100 && montoVenta <= 1000)
            {
                comision = montoVenta * 0.1;
            }
            else if (montoVenta >= 1000 && montoVenta <= 2000)
            {
                comision = montoVenta * 0.2;
            }
            else if (montoVenta >= 2000 && montoVenta <= 4000)
            {
                comision = montoVenta * 0.25;
            }
            else
            {
                comision = montoVenta * 0.35;
            }
            double sueldoFinal = sueldoBase + comision;

            Label1.Text = ":) tu sueldo base es:" + sueldoBase.ToString() + " y tu sueldo final
es: " + sueldoFinal.ToString();

```

```

String queryfinal = String.Format("update informacion$ set sueldoFinal= " +
sueldoFinal.ToString() + " where vendedor= '" + nombreVendedor + "'");
OdbcCommand cmd2 = new OdbcCommand(queryfinal, con);
cmd2.ExecuteNonQuery();

```

```

        con.Close();

    }
    else {

    }

} catch(Exception ex)
{
    Label1.Text = ex.StackTrace.ToString();
}
}

```

SESIÓN

```

protected void bSiguiente_Click(object sender, EventArgs e)
{
    Session["nombre"] = tbNombre.Text;
    Response.Redirect("Pagina2t.aspx");
}

protected void bReporte_Click(object sender, EventArgs e)
{
    Response.Redirect("Pagina3.aspx");
}
}

```

<https://support.wix.com/es/article/agregando-una-caja-de-comentarios-a-tu-p%C3%A1gina-web>

Te quiero mucho bonita <3