



# **Instituto Politécnico Nacional**

**Centro de Estudios Científicos y Tecnológicos**

**No.13 “Ricardo Flores Magón”**

**Unidad de Aprendizaje: Programación Orientada a Objetos**

**Profesora: Graciela Sánchez Pérez**

**5IM10**

**Equipo: 1**

**Integrantes:**

**\*Álvarez García Karla Lesly Ivonne**

**\*Aguilar Cardoso Aarón Adrián**

**\*Ortiz Lozano Ana Gabriela**

**\*Portilla Martínez José David**

**\*Urbán González Yatzaryt**

# Índice

Introducción

Contenido

- LOGEO
- Menú principal
- Código de menú principal
- Oficinas principales
- Código de oficinas principales
- Unidad responsable
- Código de unidad responsable
- Historial
- Código de historial

## **Introducción**

Nuestro programa consta principalmente de consultar archivos creados por las oficinas de hacienda, con el único propósito de hacer más fácil una consulta sin tener que acudir a cada oficina por el documento.

Este se visualiza por medio de visual studio donde el usuario tiene que ingresar su contraseña y podrá tener acceso a la información de las áreas, estas se dividen en dos que son manuales de organización y de procedimientos.

Para hacer más sencilla la manera de poder mostrar los archivos, se hizo un menú que tiene 3 opciones.

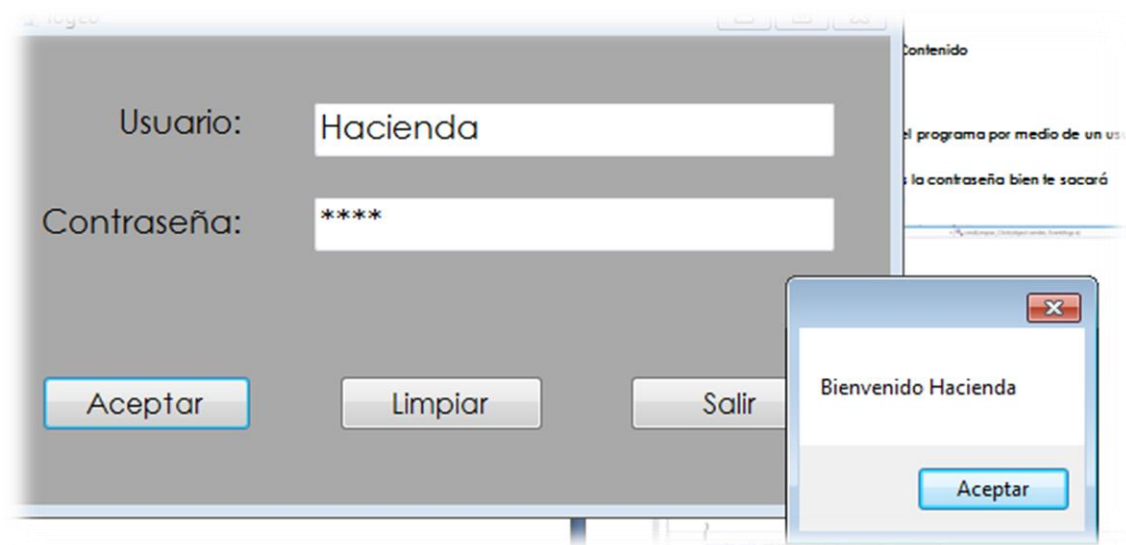
El primero son todas las oficinas principales, la segunda opción son unidades responsables, estas son subdirecciones de las oficinas principales, y la tercera opción es un historial que mostrará la fecha de archivos anteriores de cada oficina y unidad responsable.

# Contenido

## Código de LOGEO:

Dará acceso a la información del programa por medio de un usuario y contraseña.

Tienes 3 intentos, si no introduces la contraseña bien te sacará automáticamente del programa



```
hacienda_proyecto_final\logeo
public partial class logeo : Form
{
    public logeo()
    {
        InitializeComponent();
    }
    string contraseña = "SHCP";
    int v = 3;

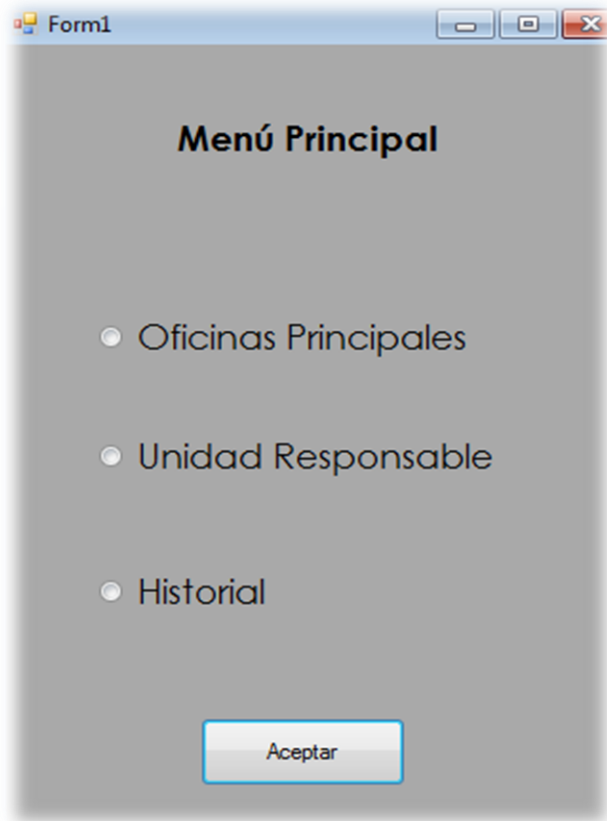
    private void cmdAceptar_Click(object sender, EventArgs e)
    {
        if (txtContraseña.Text == contraseña)
        {
            MessageBox.Show("Bienvenido " + txtUsuario.Text);
            Menu_principal Menu = new Menu_principal();
            Menu.Show();
        }
        else
        {
            MessageBox.Show("Contraseña incorrecta");
            v = v - 1;
        }
        if (v == 0)
        {
            MessageBox.Show("Adios");
            Close();
        }
    }

    private void cmdLimpiar_Click(object sender, EventArgs e)
    {
        txtUsuario.Text = "";
        txtContraseña.Text = "";
    }

    private void cmdSalir_Click(object sender, EventArgs e)
    {
        Close();
    }
}
```

# Menú principal

Este menú nos da tres opciones para poder ingresar a las oficinas principales, unidad responsable e historial



Form1

**Menú Principal**

- ☐ Oficinas Principales
- ☐ Unidad Responsable
- ☐ Historial

Aceptar

# Código Menú principal

Este código es para el menú principal.

Hicimos 3 condiciones que al seleccionar un área podrán ingresar a la información.

```
racienda_proyecto_final.Menu_principal | % cmdAceptar_Click(object sender, EventArgs e)
public Menu_principal()
{
    InitializeComponent();
}

private void Form1_Load(object sender, EventArgs e)
{
}

private void rboficinasprincipales_CheckedChanged(object sender, EventArgs e)
{
}

private void rbunidadresponsable_CheckedChanged(object sender, EventArgs e)
{
}

private void rbhistorial_CheckedChanged(object sender, EventArgs e)
{
}

private void cmdAceptar_Click(object sender, EventArgs e)
{
    if (rboficinasprincipales.Checked == true)
    {
        Oficinas_Principales OP = new Oficinas_Principales();
        OP.Show();
    }
    if (rbunidadresponsable.Checked == true)
    {
        Historial UR = new Historial();
    }
}
```

```
racienda_proyecto_final.Menu_principal | % cmdAceptar_Click(object sender, EventArgs e)
{
}

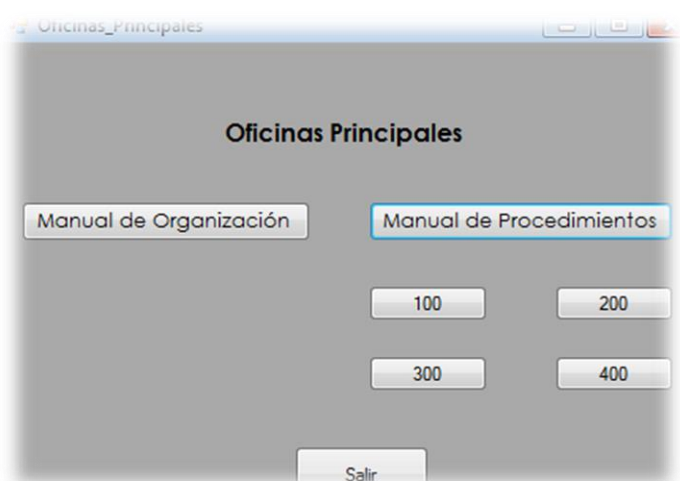
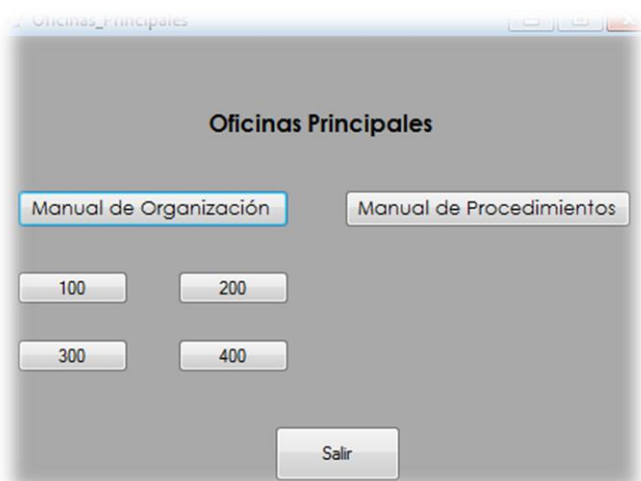
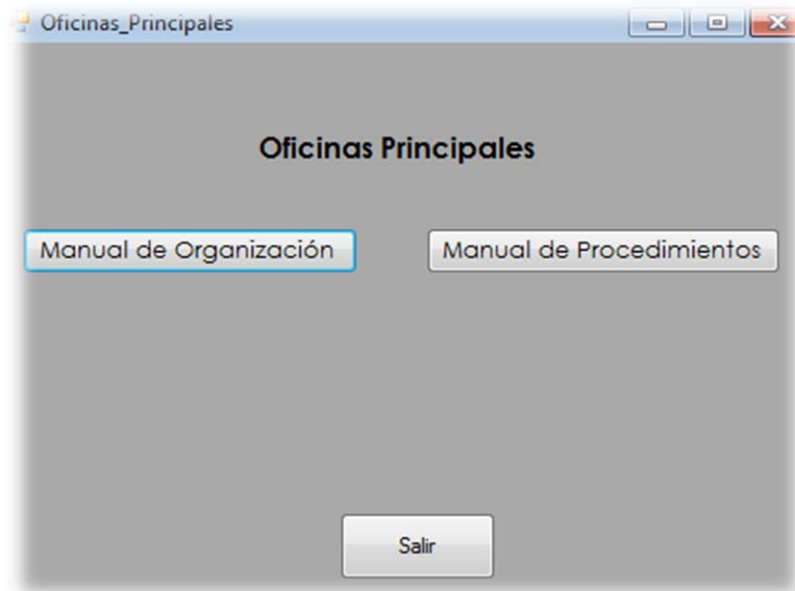
private void rbunidadresponsable_CheckedChanged(object sender, EventArgs e)
{
}

private void rbhistorial_CheckedChanged(object sender, EventArgs e)
{
}

private void cmdAceptar_Click(object sender, EventArgs e)
{
    if (rboficinasprincipales.Checked == true)
    {
        Oficinas_Principales OP = new Oficinas_Principales();
        OP.Show();
    }
    if (rbunidadresponsable.Checked == true)
    {
        Historial UR = new Historial();
        UR.Show();
    }
    if (rbhistorial.Checked == true)
    {
        HistorialG Historial = new HistorialG();
        Historial.Show();
    }
}
}
```

# Oficinas principales

Cuando ingresas a la opción de Oficinas principales y de esta mostrará los manuales de organización y de procedimientos más actual de cada oficina



# Código Oficinas Principales

```
namespace Hacienda_proyecto_final.Oficinas_Principales
{
    public partial class Oficinas_Principales : Form
    {
        public Oficinas_Principales()
        {
            InitializeComponent();
        }
        private string _producto;
        public string producto
        {
            get { return _producto; }
            set { _producto = value; }
        }
        Archivo pdf = new Archivo();

        private void label1_Click(object sender, EventArgs e)
        {
        }

        private void Oficinas_Principales_Load(object sender, EventArgs e)
        {
            cmdno.Visible = true;
            cmd100opmo.Visible = false;
            cmd200opmo.Visible = false;
            cmd300opmo.Visible = false;
            cmd400opmo.Visible = false;

            cmdmp.Visible = true;
            cmd100opmp.Visible = false;
            cmd200opmp.Visible = false;
        }
    }
}
```

Este código es para las oficinas principales, se puede ser que existen cuatro, pero se dividen en Manuales de Organización y Manuales de Procedimientos

Mientras selecciones un área te mostrará el archivo en PDF.

No se pueden ver las dos juntas (manuales de organización y procedimientos), se debe seleccionar solo una, o para ver

otra debes presionar el botón que deseas

```
private void cmdSalir_Click(object sender, EventArgs e)
{
    Close();
}

private void cmd100opmo_Click(object sender, EventArgs e)
{
    producto = "1001.pdf";
    pdf.Show();
}

private void cmd200opmo_Click(object sender, EventArgs e)
{
    producto = "2001.pdf";
    pdf.Show();
}

private void cmd300opmo_Click(object sender, EventArgs e)
{
    producto = "3001.pdf";
    pdf.Show();
}

private void cmd400opmo_Click(object sender, EventArgs e)
{
    producto = "4001.pdf";
    pdf.Show();
}

private void cmd100opmp_Click(object sender, EventArgs e)
{
    producto = "1002.pdf";
    pdf.Show();
}
```

```
private void cmd200opmp_Click(object sender, EventArgs e)
{
    producto = "2002.pdf";
    pdf.Show();
}

private void cmd300opmp_Click(object sender, EventArgs e)
{
    producto = "3002.pdf";
    pdf.Show();
}

private void cmd400opmp_Click(object sender, EventArgs e)
{
    producto = "4002.pdf";
    pdf.Show();
}
}
```

## Unidad responsable



La segunda opción que tiene el programa es la unidad responsable que de igual manera mostrará los manuales de organización y procedimientos más actuales pero ahora será de las subdirecciones de cada área.

Unidad\_responsable

Unidad Responsable

Manual de Organización

100

200

300

400

Manual de Procedimientos

100

200

300

400

Aceptar

Salir

# Código Unidad Responsable

```
namespace Hacienda_proyecto_final
{
    public partial class Historial : Form
    {
        public Historial()
        {
            InitializeComponent();
        }
        string nombre, nombre1, nombre2, nombre3, nombre4, nombre5, nombre6, nombre7;
        private string _producto2;
        public string producto2
        {
            get { return _producto2; }
            set { _producto2 = value; }
        }
        Archivo pdf = new Archivo();

        private void Unidad_responsable_Load(object sender, EventArgs e)
        {
        }

        private void cmdAceptar_Click(object sender, EventArgs e)
        {
            nombre = cbo100mo.Text;
            nombre1 = cbo100mp.Text;
            nombre2 = cbo200mo.Text;
            nombre3 = cbo200mp.Text;
            nombre4 = cbo300mo.Text;
            nombre5 = cbo300mp.Text;
            nombre6 = cbo400mo.Text;
            nombre7 = cbo400mp.Text;
        }
    }
}
```

En esta Parte del programa tenemos las oficinas principales pero de estas mismas se derivan cuatro subaeras que te igual manera cada una tiene su manual de organización y de procedimiento.

Te mostrará el archivo más actual

```
nombre = cbo100mo.Text;
nombre1 = cbo100mp.Text;
nombre2 = cbo200mo.Text;
nombre3 = cbo200mp.Text;
nombre4 = cbo300mo.Text;
nombre5 = cbo300mp.Text;
nombre6 = cbo400mo.Text;
nombre7 = cbo400mp.Text;

if (nombre == "1101")
{
    producto2 = "1101.pdf";
    pdf.Show();
}
if (nombre == "1111")
{
    producto2 = "1111.pdf";
    pdf.Show();
}
if (nombre == "1121")
{
    producto2 = "1121.pdf";
    pdf.Show();
}
if (nombre1 == "2101")
{
    producto2 = "2101.pdf";
    pdf.Show();
}
if (nombre1 == "2111")
{
    producto2 = "2111.pdf";
    pdf.Show();
}

private void cmdAceptar_Click(object sender, EventArgs e)
{
    producto2 = "zzzz.pdf";
    pdf.Show();
}
if (nombre6 == "3102")
{
    producto2 = "3102.pdf";
    pdf.Show();
}
if (nombre6 == "3112")
{
    producto2 = "3112.pdf";
    pdf.Show();
}
if (nombre6 == "3122")
{
    producto2 = "3122.pdf";
    pdf.Show();
}
if (nombre7 == "4102")
{
    producto2 = "4102.pdf";
    pdf.Show();
}
if (nombre7 == "4112")
{
    producto2 = "4112.pdf";
    pdf.Show();
}
if (nombre7 == "4122")
{
    producto2 = "4122.pdf";
    pdf.Show();
}
}

private void cmdSalir_Click(object sender, EventArgs e)
{
    Close();
}
```

# Historial

En el historial nos va a mostrar los manuales de organización y de procedimientos de cada unidad tanto oficina principal como subdirecciones, ahí podrán ver los cambios que han tendido las oficinas.

También da la opción de buscar algún documento por fecha.

The screenshot shows a web application window titled "Historial". It features a search interface with three dropdown menus: "Area", "Tipo de Manual", and "Disponibilidad:". Below these is a date input field set to "24/02/2015" with a calendar icon. There are three buttons: "Limpiar", "Aceptar", and "Salir". Below the search area is a table with the following columns: ID, Area, Tipo\_Manual, Fecha\_adicion, Fecha\_replazo, and Estado. The first row of the table is highlighted with a mouse cursor. The rest of the table area is empty.

	ID	Area	Tipo_Manual	Fecha_adicion	Fecha_replazo	Estado
➡						

# Código Historial

```
namespace Hacienda_proyecto_final
{
    public partial class HistorialG : Form
    {
        private DataGridView dataGridView1 = new DataGridView();
        private BindingSource bindingSource1 = new BindingSource();
        private SqlDataAdapter dataAdapter = new SqlDataAdapter();

        SqlConnection Cn;
        SqlCommand Cm;
        SqlDataReader Dr;
        String Str;

        public HistorialG()
        {
            InitializeComponent();
        }
        private void GetData(string selectCommand)
        {
            try
            {
                String connectionString =
                    "Integrated Security=SSPI;Persist Security Info=False;" +
                    "Initial Catalog=BaseHacienda;Data Source=";

                dataAdapter = new SqlDataAdapter(selectCommand, connectionString);
                SqlCommandBuilder commandBuilder = new SqlCommandBuilder(dataAdapter);

                DataTable table = new DataTable();
                table.Locale = System.Globalization.CultureInfo.InvariantCulture;
                dataAdapter.Fill(table);
                bindingSource1.DataSource = table;
            }
            catch (SqlException)
            {
                MessageBox.Show("To run this example, replace the value of the " +
                    "connectionString variable with a connection string that is " +
                    "valid for your system.");
            }
        }
    }
}
```

Código de Historial estaremos conectados con una base de datos que en esa base están los archivos de cada área, cada cambio que han tenido.

Esto se hace para tener sus antecedentes.

En esta parte mostrará oficinas principales como unidad responsable, manuales de organización y de procedimientos de cada una.

```
private void cmdAceptar_Click(object sender, EventArgs e)
{
    dataAdapter.Fill(table);
    bindingSource1.DataSource = table;

    //*****
    this.Controls.AddRange(new Control[] { dataGridView1, panel });
    this.Load += new System.EventHandler(HistorialG_Load);
    this.Text = "Historial";
    this.dataGridView1.Location = new System.Drawing.Point(50, 220);
    this.dataGridView1.Size = new System.Drawing.Size(700, 400);
    //*****

    /*dataGridView1.AutoSizeColumnsMode(
        DataGridViewAutoSizeColumnsMode.AllCellsExceptHeader);*/
}
catch (SqlException)
{
    MessageBox.Show("To run this example, replace the value of the " +
        "connectionString variable with a connection string that is " +
        "valid for your system.");
}
}

private void HistorialG_Load(object sender, EventArgs e)
{
    dataGridView1.DataSource = bindingSource1;
    GetData("Select * From HistorialH where Area = '" + cmbArea.Text + "' and Tipo_Manual = '" + cmbTipoM.Text + "'");
}
}
```

```
private void cmdAceptar_Click(object sender, EventArgs e)
{
    Str = global.Str;
    Cn = new SqlConnection();
    Cn.ConnectionString = Str;
    System.Text.StringBuilder Sel = new StringBuilder();
    Sel.Append("Select * From HistorialH where Area = '" + cmbArea.Text + "' and Tipo_Manual = '" + cmbTipoM.Text + "' and Estado = '" +
        cmbDisponibilidad.Text + "' or Fecha_adicion = '" + datePHistorial.Text + "'");
    Cm = new SqlCommand(Sel.ToString(), Cn);
    Cn.Open();
    Dr = Cm.ExecuteReader();
    if (Dr.Read())
    {
        dataGridView1.DataSource = bindingSource1;
        GetData("Select * From HistorialH where Area = '" + cmbArea.Text + "' and Tipo_Manual = '" + cmbTipoM.Text + "' and Estado = '" +
            cmbDisponibilidad.Text + "' or Fecha_adicion = '" + datePHistorial.Text + "'");
    }
    else
    {
        MessageBox.Show("El registro no existe");
    }
    Cn.Close();
    Dr.Close();
}

private void cmdLimpiar_Click(object sender, EventArgs e)
{
    Historial historial = new HistorialG();
    historial.ResetHouseEvents();
    cmbArea.Text = "";
    cmbTipoM.Text = "";
    cmbDisponibilidad.Text = "";
}

private void cmdSalir_Click(object sender, EventArgs e)
{
    Close();
}

public Control panel { get; set; }
```