# 1 – Introducción

## Repositorio

Creamos el repositorio en GitHub y lo llamamos AppComercial.

Debe tener el gitignore de VisualStudio.

## Solución

Copiamos la ruta del repositorio y vamos a Visual Studio y clonamos el Repositorio en la Carpeta C:\AppComercial

Dentro de la Solución AppComercial creamos los Proyectos:

* Aplicación de Windows Forms (.NET Framework), y lo llamamos AppComercial
* Biblioteca de Clases (.NET Framework), y lo llamamos CADAppCOmercial

## Formulario Login

En el Proyecto AppComercial cambiamos el nombre del formulario Form1 a frmLogin

Le ponemos estas Propiedades y Controles:

|  |  |  |
| --- | --- | --- |
| **Elemento** | **Propiedad** | **Valor** |
| Formulario | (Name) | frmLogin |
|  | AcceptButton | btnAceptar |
|  | BackColor | SkyBlue |
|  | CancelButton | btnCancelar |
|  | FormBorderStyle | FixedDialog |
|  | MaximizeBox | False |
|  | MinimizeBox | False |
|  | StartPosition | CenterScreen |
|  | Text | Ingreso al Sistema |
|  | Size | 413,280 |
|  |  |  |
| Label | (Name) | label1 |
|  | Text | &Usuario; |
|  |  |  |
| TextBox | (Name) | txtUsuario |
|  |  |  |
| Label | (Name) | Label2 |
|  | Text | &Clave; |
|  | PasswordChar | \* |
|  |  |  |
| PictureBox | Image | Cargamos la imagen como recurso local |
|  |  |  |
| Button | (Name) | btnAceptar |
|  | Text | &Aceptar |
|  | Image | Cargamos la imagen como recurso local |
|  | ImageAlign | TopCenter |
|  | TextAlign | BottomCenter |
|  |  |  |
| Button | (Name) | btnCancelar |
|  | Text | &Cancelar |
|  | Image | Cargamos la imagen como recurso local |
|  | ImageAlign | TopCenter |
|  | TextAlign | BottomCenter |
|  |  |  |
| ErrorProvider | (Name) | errorProvider1 |

# 2 - Iniciando

## Código del frmLogin

En el código del frmLogin ponemos:

|  |
| --- |
| using System;  using System.Windows.Forms;  namespace AppComercial  {  public partial class frmLogin : Form  {  public frmLogin()  {  InitializeComponent();  }  private void frmLogin\_Load(object sender, EventArgs e)  {  }  private void btnCancelar\_Click(object sender, EventArgs e)  {  this.Close();  }  private void btnAceptar\_Click(object sender, EventArgs e)  {  errorProvider1.Clear();  if (txtUsuario.Text==string.Empty)  {  errorProvider1.SetError(txtUsuario, "Debe ingresar un Usuario");  txtUsuario.Focus();  return;  }  if (txtClave.Text == string.Empty)  {  errorProvider1.SetError(txtClave, "Debe ingresar una Clave");  txtClave.Focus();  return;  }  }  }  } |

## Usuario TableAdapter

En el Proyecto **CADAppComercial** hacemos clic derecho – Agregar - Nuevo Elemento – Datos – Conjunto de Datos y lo llamamos **DSAppComercial**.

Agregamos en el explorador de servidores la Base de Datos.

Arrastramos las Tabla **Usuario** y **Rol**.

En **UsuarioTableAdapter** hacemos clic derecho, agregar Consulta, Crear un nuevo procedimiento almacenado, SELECT que devuelve un solo valor

|  |
| --- |
| SELECT 1 FROM Usuario WHERE IDUsuario=@IDUsuario AND Clave=@Clave |

Como nombre le ponemos **ValidaUsuario.**

Hay que crear una Clase que permita utilizar esa consulta.

## Clase CADUsuario

En el Proyecto **CADAppComercial** creamos la Clase **CADUsuario**.

# 3 - Login

|  |
| --- |
| using CADAppComercial.DSAppComercialTableAdapters;  namespace CADAppComercial  {  public class CADUsuario  {  private static UsuarioTableAdapter adaptador = new UsuarioTableAdapter();  public static bool ValidaUsuario(string IDUsuario, string Clave)  {  if (adaptador.ValidaUsuario(IDUsuario, Clave) == null)  {  return false;  }  else  {  return true;  }  }  }  } |

Compilamos la solución.

En el Proyecto **AppComercial** agregamos la Referencia del proyecto **CADAppComercial**

## Verificando Usuario y Clave contra la Base de Datos

En **frmLogin** hacemos:

|  |
| --- |
| using CADAppComercial;  using System;  using System.Windows.Forms;  namespace AppComercial  {  public partial class frmLogin : Form  {  public frmLogin()  {  InitializeComponent();  }  private void frmLogin\_Load(object sender, EventArgs e)  {  }  private void btnCancelar\_Click(object sender, EventArgs e)  {  this.Close();  }  private void btnAceptar\_Click(object sender, EventArgs e)  {  errorProvider1.Clear();    if (txtUsuario.Text==string.Empty)  {  errorProvider1.SetError(txtUsuario, "Debe ingresar un Usuario");  txtUsuario.Focus();  return;  }    if (txtClave.Text == string.Empty)  {  errorProvider1.SetError(txtClave, "Debe ingresar una Clave");  txtClave.Focus();  return;  }  if(!CADUsuario.ValidaUsuario(txtUsuario.Text,txtClave.Text))  {  MessageBox.Show("Usuario o Clave no válidos", "Error", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);  txtUsuario.Text = string.Empty;  txtClave.Text = string.Empty;  txtUsuario.Focus();  return;  }  frmPrincipal miForm = new frmPrincipal();  miForm.Show();  this.Hide();  }  }  } |

## Formulario Principal

Creamos el **frmPrincipal**:

|  |  |  |  |
| --- | --- | --- | --- |
| **Elemento** |  | **Propiedad** | **Valor** |
| Formulario |  | (Name) | frmPrincipal |
|  |  | BackCOlor |  |
|  |  | IsMdiContainer | True |
|  |  | WindowsState | Maximized |
|  |  | Text | Sistema Comercial |

# 4-Menú

En el **frmPrincipal** agregamos desde el Cuadro de Herramientas un **MenuStrip**.

 

 



# 5 – CRUD Clientes, Barra de herramientas

## Para que al cerrar el frmPrincipal se cierre la Aplicación

En el código del **frmPrincipal** ponemos:

|  |
| --- |
| using System.Windows.Forms;  namespace AppComercial  {  public partial class frmPrincipal : Form  {  public frmPrincipal()  {  InitializeComponent();  }  private void frmPrincipal\_FormClosing(object sender, FormClosingEventArgs e)  {  Application.Exit();  }  private void salirToolStripMenuItem\_Click(object sender, System.EventArgs e)  {  this.Close();  }  }  } |

## Formulario Clientes (primera versión)

Creamos el formulario **frmClientesTemp.** Le ponemos estas Propiedades y Controles:

|  |  |  |
| --- | --- | --- |
| **Elemento** | **Propiedad** | **Valor** |
| Formulario | (Name) | frmClientesTemp |
|  | Text | Clientes |
|  | BackColor | SkyBlue |
|  | MaximizeBox | False |
|  | MinimizeBox | False |
|  | StartPosition | CenterScreen |
|  | FormBorderStyle | FixedDialog |
|  |  |  |
| ToolStrip | (Name) | toolStrip1 |
|  | Agregamos 4 botones (para navegar entre registros), un separador, y 6 botones más (Nuevo, Editar, Guardar, Cancelar, Borrar y Buscar).  Le ponemos íconos en la propiedad Image.    Les ponemos nombres:   * tsbPrimero * tsbAnterior * tsbSiguiente * tsbUltimo * tsbEditar * tsbNuevo * tsbBorrar * tsbGuardar * tsbCancelar * tsbBuscar   Agregamos los ToolStripText | |

# 6 – CRUD Clientes, diseño

## Abrir el Formulario de Cliente

Para poder llamar el formulario de Clientes desde el Menú Principal, ponemos en el código del **frmPrincipal**:

|  |
| --- |
| private void clientesToolStripMenuItem\_Click(object sender, System.EventArgs e)  {  frmClientesTemp miForm = new frmClientesTemp();  miForm.MdiParent = this;  miForm.Show();  } |

## DataSet DSAppComercial

En el Proyecto **AppComercial** creamos el DataSet **DSAppComercial** y le agregamos la Tabla **TiposDocumento**



## Formulario Clientes

Agregamos al formulario **frmClientesTemp:**

|  |  |  |
| --- | --- | --- |
| **Elemento** | **Propiedad** | **Valor** |
| Label | (Name) | label1 |
|  | Text | ID Cliente: |
|  |  |  |
| TextBox | (Name) | txtIDCliente |
|  | ReadOnly | True |
|  |  |  |
| Label | (Name) | label2 |
|  | Text | Tipo Documento: |
|  |  |  |
| ComboBox | (Name) | cbTipoDocumento |
|  | Enabled | False |
|  | BackColor | Aquamarine |
|  | Datos de origen |  |
|  |  |  |
| Label | (Name) | Label3 |
|  | Text | Documento: |
|  |  |  |
| TextBox | (Name) | txtDocumento: |
|  | ReadOnly | True |
|  | BackColor | Aquamarine |
|  |  |  |
| Label | (Name) | Label4 |
|  | Text | Nombre Comercial: |
|  |  |  |
| TextBox | (Name) | txtNombreComercial |
|  | ReadOnly | True |
|  | BackColor | Aquamarine |
|  |  |  |
| Label | (Name) | Label5 |
|  | Text | Nombre Contacto: |
|  |  |  |
| TextBox | (Name) | txtNombreContacto |
|  | ReadOnly | True |
|  | BackColor | Aquamarine |
|  |  |  |
| Label | (Name) | Label6 |
|  | Text | Apellido Contacto: |
|  |  |  |
| TextBox | (Name) | txtApellidoContacto |
|  | ReadOnly | True |
|  | BackColor | Aquamarine |
|  |  |  |
| Label | (Name) | Label7 |
|  | Text | Dirección: |
|  |  |  |
| TextBox | (Name) | txtDireccion |
|  | ReadOnly | True |
|  | BackColor | White |
|  |  |  |
| Label | (Name) | Label8 |
|  | Text | Teléfono: |
|  |  |  |
| TextBox | (Name) | txtTeléfono |
|  | ReadOnly | True |
|  | BackColor | White |
|  |  |  |
| Label | (Name) | Label9 |
|  | Text | Celular: |
|  |  |  |
| TextBox | (Name) | txtCelular |
|  | ReadOnly | True |
|  | BackColor | White |

# 7 – CRUD Clientes, cargar datos en Grid

|  |  |  |
| --- | --- | --- |
| **Elemento** | **Propiedad** | **Valor** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# PROCEDIMIENTOS ALMACENADOS

|  |  |  |
| --- | --- | --- |
| **Video** | **Nombre** | **Sirve para** |
| 2 | ValidaUsuario | Validar Usuario y Clave contra la Base de Datos |
|  |  |  |
|  |  |  |