

	UT 2 – <i>Práctica Sistema de Ventas</i> Parte IV – Menú Usuarios	CFGS Desarrollo de Aplicaciones Multiplataforma  Módulo: DDI
--	--	---

**Este diseño se corresponde con el video de apoyo número 4.**

Incorporamos estos elementos a nuestro frmUsuarios:

Un label con **autosize** false **dock** left **text** vacio **backcolor** blanco **bordersyle** fixedsingle, que va a contener nuestros campos de usuario que tienen correspondencia con nuestra base de datos.

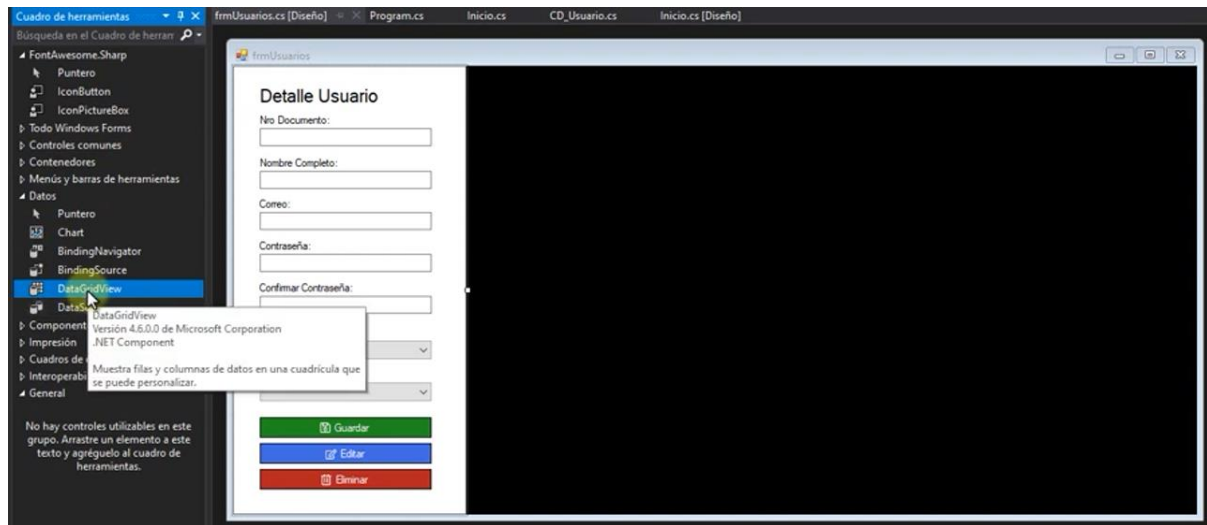
Dentro de este label situamos un **label** que visualizará **Detalle usuario** con tamaño letra 15 y **otros siete** que tienen que tener fondo blanco. El primero tiene que visualizar Usuario el segundo Nombre Completo, el tercero Correo, el cuarto Contraseña, el quinto Confirmar contraseña, el sexto Rol y el séptimo Estado.

Los cinco primeros encabezarán a **cinco textbox** que se llamarán txtdocumento, txtnombrecompleto, txtcorreo, txtclave, txtconfirmarclave.

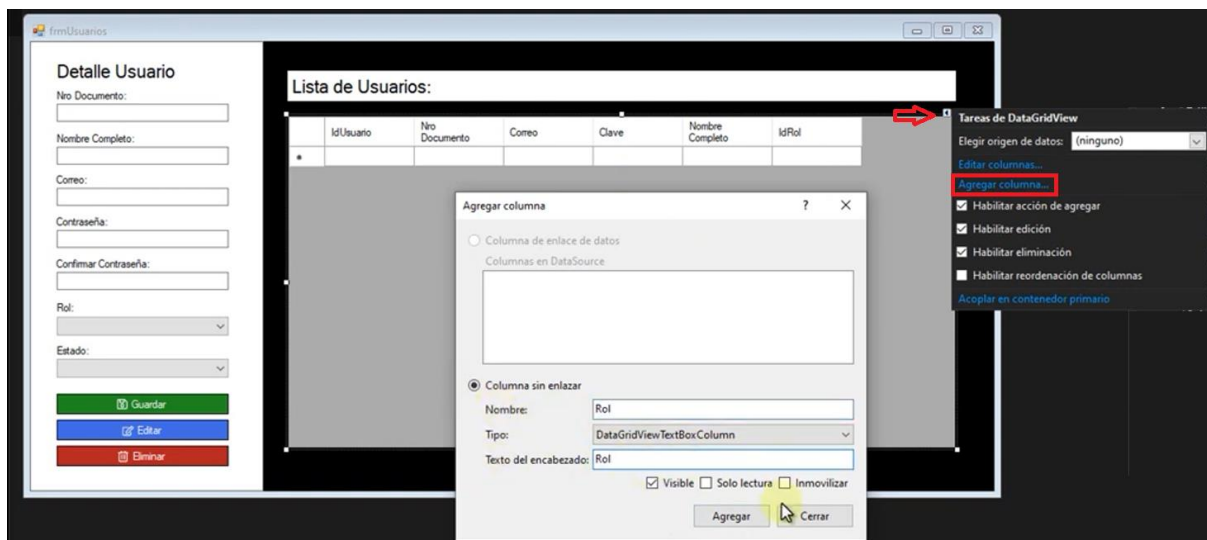
Los dos últimos encabezarán **dos combobox** cborol y cboestado tendrán su propiedad **DropDownStyle** a DropDownList para que no se puedan introducir datos en estos campos ya que visualizaremos una lista de opciones.

Desde herramientas FontAweresome arrastramos **tres botones** (btnguardar btnlimpiar y btneliminar) con **texto** Guardar, Limpiar y Eliminar, **IconChar** Save, Broom y TrashAlt y **backcolor** ForestGreen, RoyalBlue y Firebrick respectivamente con propiedades **IconColor** blanco, **IconSize** 18, **TextAlign** MiddleRight, **TextImageRelation** ImageBeforeText, **Cursor** en forma de mano, **FlatStyle** (apariencia) en modo flat, **ForeColor** (color fuente) blanco, dentro de FlatApparece la propiedad **bordercolor** black, tamaño iconos 18.

Arrastramos un DataGridView que llamaremos dgvdvdata a la parte derecha de nuestro formulario:

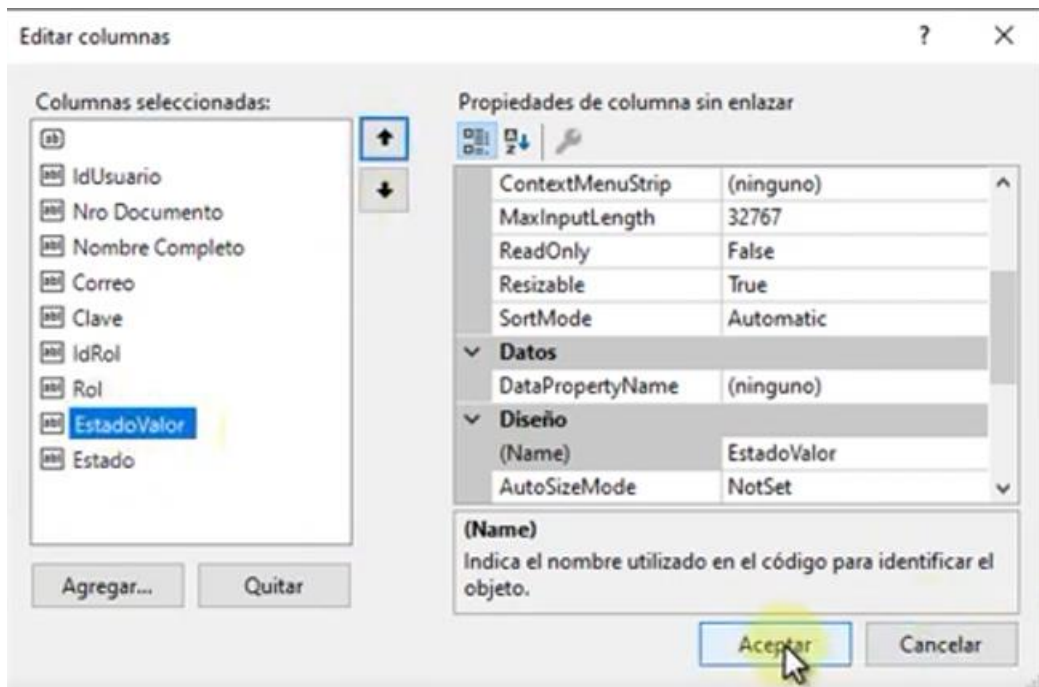


En su cabecera ponemos una etiqueta Lista de Usuarios: con, **autosize=false** para redimensionarlo, **textalign** a la izquierda, **size=15** y **BorderStyle=FixedSingle**  
Agregamos las columnas a nuestro DataGridView,



Y podemos editarlas si alguna propiedad no está bien establecida (IdUsuario y Clave no tiene que ser visible), ordenarlas y añadir nuevas columnas Estado valor y un botón btnSeleccionar (propiedad ColumnType=DataGridViewButtonColumn) que nos va a servir para elegir un usuario.

	UT 2 – <i>Práctica Sistema de Ventas</i> Parte IV – Menú Usuarios	CFGs Desarrollo de Aplicaciones Multiplataforma  Módulo: DDI
--	--	---



Ajustamos el ancho de nuestras columnas visibles a 30, 150, 180, 150, 100, 100.

Cambiamos algunas propiedades de nuestro DataGridView. **AllowUserToAddRows** = false, **ColumnHeaderDefaultCellStyle** => Size=9, Padding 2 (all) **MultiSelect** = false, **ReadOnly** = true. **RowDefaultCellStyle** => SelectionBackColor=blanco SelectionForeColor=negro. Row Template **Height** (alto columna)=28

Por último, incorporamos el textbox txtId en nuestro menú de la derecha que va a contener el Id de usuario seleccionado de forma oculta (visible=false).

## **FUNCIONALIDAD DATA GRID VIEW**

Para dar de alta un nuevo usuario se introducirán los valores en el menú "Detalle Usuario" y al pulsar Guardar si son válidos se almacenarán en nuestra base de datos y se visualizan en nuestro DataGridView.

Para poder obtener los Roles de nuestra BBDD (mismo proceso que recuperar usuarios) creamos las clases correspondientes en nuestra capa de datos y capa de negocio:

```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  using System.Data;
8  using System.Data.SqlClient;
9  using CapaEntidad;
10
11
12 namespace CapaDatos
13 {
14     public class CD_Rol
15     {
16         public List<Rol> Listar()
17         {
18             List<Rol> lista = new List<Rol>();
19
20             using (SqlConnection oconexion = new SqlConnection(Conexion.cadena))
21             {
22                 try
23                 {
24                     StringBuilder query = new StringBuilder();
25                     query.AppendLine("select IdRol,Descripcion from ROL");
26
27                     SqlCommand cmd = new SqlCommand(query.ToString(), oconexion);
28                     cmd.CommandType = CommandType.Text;
29
30                     oconexion.Open();
31
32                     using (SqlDataReader dr = cmd.ExecuteReader())
33                     {
34                         while (dr.Read())
35                         {
36                             lista.Add(new Rol()
37                             {
38                                 IdRol = Convert.ToInt32(dr["IdRol"]),
39                                 Descripcion = dr["Descripcion"].ToString()
40                             });
41                         }
42                     }
43                 }
44                 catch (Exception ex)
45                 {
46
47                     lista = new List<Rol>();
48                 }
49             }
50
51             return lista;
52         }
53     }
54 }
55
56

```

```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  using CapaDatos;
8  using CapaEntidad;
9
10 namespace CapaNegocio
11 {
12     public class CN_Rol
13     {
14         private CD_Rol objcd_rol = new CD_Rol();
15
16         public List<Rol> Listar()
17         {
18             return objcd_rol.Listar();
19         }
20     }
21 }
22
23
24

```

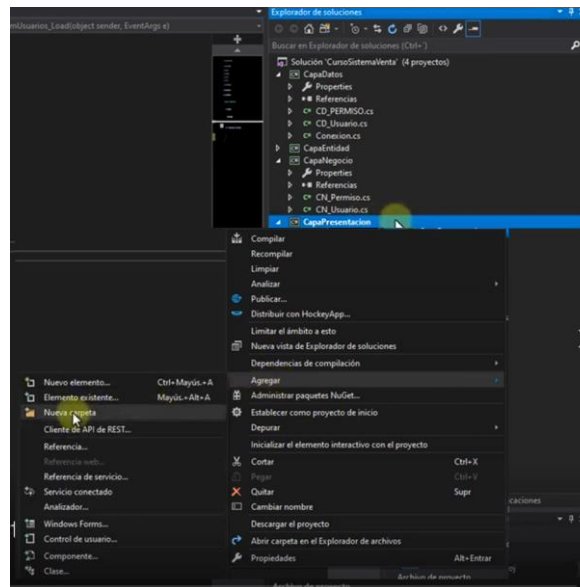
En los combos de Rol y Estado se deben visualizar los valores permitidos. Este proceso se deber realizar en el evento Load de nuestro formulario inicio.

	<b>UT 2 – <i>Práctica Sistema de Ventas</i></b> <b>Parte IV – Menú Usuarios</b>	CFGS Desarrollo de Aplicaciones Multiplataforma <b>Módulo: DDI</b>
--	--	--

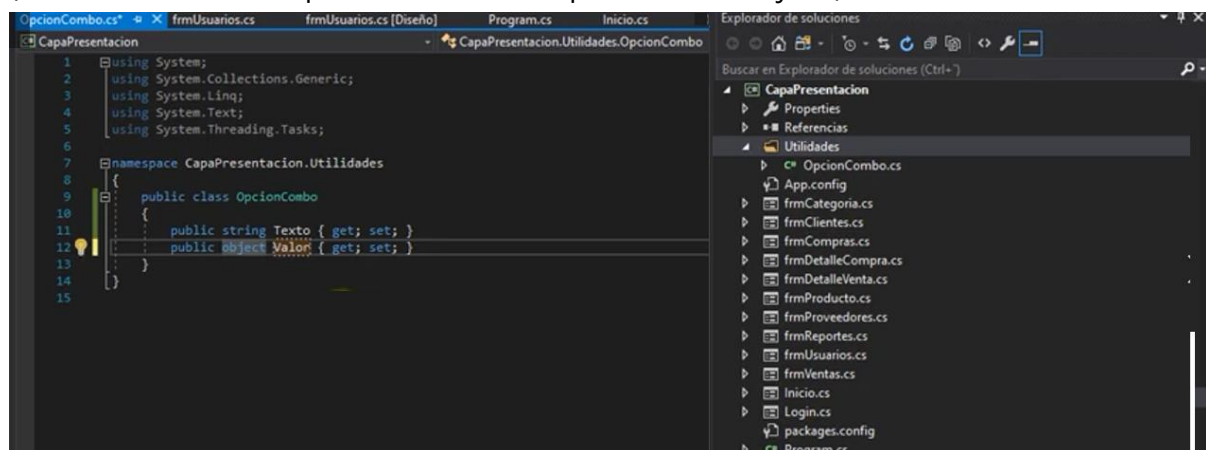
Los combobox trabajan con ítems. Podemos cargar estos combobox de diferentes maneras: manualmente, con arrays, diccionarios o con objetos.

Vamos a cargar nuestros ComboBox con objetos que van a tener dos propiedades texto y valor, el texto se va a visualizar en el combo y el valor va a ser un dato interno. Para ello crearemos una nueva clase que nos servirá para este propósito.

En nuestra capa presentación creamos una carpeta llamada Utilidades y dentro de ella creamos una nueva clase OpcionCombo que nos va a servir para manejar los ítems del combo



Dentro de la carpeta agregamos una nueva clase pública OpcionCombo que va a tener dos valores, uno visible en el combo con un literal y otro interno con un valor asociado al literal (misma funcionalidad que los diccionarios que hemos trabajado).



Introducimos en el evento de Load el código para visualizar los valores correspondientes.

```

1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10
11  using CapaPresentacion.Utilidades;
12
13  using CapaEntidad;
14  using CapaNegocio;
15
16  namespace CapaPresentacion
17  {
18      public partial class frmUsuarios : Form
19      {
20          public frmUsuarios()
21          {
22              InitializeComponent();
23          }
24
25          private void frmUsuarios_Load(object sender, EventArgs e)
26          {
27              cboestado.Items.Add(new OpcionCombo() { Valor = 1, Texto = "Activo" });
28              cboestado.Items.Add(new OpcionCombo() { Valor = 0, Texto = "No Activo" });
29              cboestado.DisplayMember = "Texto";
30              cboestado.ValueMember = "Valor";
31              cboestado.SelectedIndex = 0;
32
33              List<Rol> listaRol = new OL_Rol().Listar();
34
35              foreach (Rol item in listaRol) {
36                  cborol.Items.Add(new OpcionCombo() { Valor = item.IdRol, Texto = item.Descripcion });
37              }
38              cborol.DisplayMember = "Texto";
39              cborol.ValueMember = "Valor";
40              cborol.SelectedIndex = 0;
41
42          }
43      }
44  }
45
46
47
48
49
50
51

```

Para que nos pinte los datos introducidos en el DataGridView al pinchar el botón Guardar:

```

46
47  private void btnGuardar_Click(object sender, EventArgs e)
48  {
49      dgvdata.Rows.Add(new object[] { "", txtid.Text, txtdocumento.Text, txtnombrecompleto.Text, txtcorreo.Text, txtclave.Text,
50          ((OpcionCombo)cborol.SelectedItem).Valor.ToString(),
51          ((OpcionCombo)cborol.SelectedItem).Texto.ToString(),
52          ((OpcionCombo)cboestado.SelectedItem).Valor.ToString(),
53          ((OpcionCombo)cboestado.SelectedItem).Texto.ToString()
54      });
55
56      Limpiar();
57  }
58
59
60

```

```

private void Limpiar() {
    txtindice.Text = "-1";
    txtid.Text = "0";
    txtdocumento.Text = "";
    txtnombrecompleto.Text = "";
    txtcorreo.Text = "";
    txtclave.Text = "";
    txtconfirmarclave.Text = "";
    cborol.SelectedIndex = 0;
    cboestado.SelectedIndex = 0;

    txtdocumento.Select();
}

```



	<b>UT 2 – <i>Práctica Sistema de Ventas</i></b> <b>Parte IV – Menú Usuarios</b>	CFGS Desarrollo de Aplicaciones Multiplataforma  <b>Módulo: DDI</b>
--	--	--

Para poder filtrar dentro del data grid vamos a incorporar dos elementos nuevos a nuestro formulario de usuario, un combo llamado cbobusqueda que va a contener los nombres de nuestras columnas (Headertext) del DataGridView y un textbox txtbusqueda y dos botones btnbuscar y btnlimpiarbuscador con IconChar Search y Broom, TextImageRelation=Overlay, back color=blanco, IconColor=black, Flat Style=flat).

Cargamos el combo cbobusqueda:

```

26 private void frmUsuarios_Load(object sender, EventArgs e)
27 {
28     cboestado.Items.Add(new OpcionCombo() { Valor = 1, Texto = "Activo" });
29     cboestado.Items.Add(new OpcionCombo() { Valor = 0, Texto = "No Activo" });
30     cboestado.DisplayMember = "Texto";
31     cboestado.ValueMember = "Valor";
32     cboestado.SelectedIndex = 0;
33
34
35     List<Rol> listaRol = new CN_Rol().Listar();
36
37     foreach (Rol item in listaRol) {
38         cborol.Items.Add(new OpcionCombo() { Valor = item.IdRol, Texto = item.Descripcion });
39     }
40     cborol.DisplayMember = "Texto";
41     cborol.ValueMember = "Valor";
42     cborol.SelectedIndex = 0;
43
44
45     foreach (DataGridViewColumn columna in dgvdata.Columns) {
46         if (columna.Visible == true && columna.Name != "btnseleccionar") {
47             cbobusqueda.Items.Add(new OpcionCombo() { Valor = columna.Name, Texto = columna.HeaderText});
48         }
49     }
50     cbobusqueda.DisplayMember = "Texto";
51     cbobusqueda.ValueMember = "Valor";
52     cbobusqueda.SelectedIndex = 0;
53
54
55 }
56
57

```