

	<b>UT 2 – <i>Práctica Sistema de Ventas</i></b> <b>Parte VII – Formulario Usuarios vs BBDD</b>	CFGS Desarrollo de Aplicaciones Multiplataforma  Módulo: DDI
--	---	--

**Este diseño se corresponde con el video de apoyo número 7**

Para distinguir un alta de una modificación utilizaremos el id de usuario. Si es 0 es nuevo usuario y usaremos el método Registrar de nuestra clase CN\_Usuario pero si es diferente a 0 estaremos editando un usuario ya existente y usaremos el método Editar.

```
private void btnGuardar_Click(object sender, EventArgs e)
{
    string mensaje = string.Empty;

    Usuario objusuario = new Usuario() {
        IdUsuario = Convert.ToInt32(txtid.Text),
        Documento = txtdocumento.Text,
        NombreCompleto = txtnombrecompleto.Text,
        Correo = txtcorreo.Text,
        Clave = txtclave.Text,
        rol = new Rol() { IdRol = Convert.ToInt32(((OpcionCombo)cborol.SelectedItem).Valor) },
        Estado = Convert.ToInt32(((OpcionCombo)cboestado.SelectedItem).Valor) == 1 ? true : false
    };

    if (objusuario.IdUsuario == 0)
    {
        int idusuariogenerado = new CN_Usuario().Registrar(objusuario, out mensaje);

        if (idusuariogenerado != 0)
        {
            dgvdata.Rows.Add(new object[] { "", idusuariogenerado, txtdocumento.Text, txtnombrecompleto.Text, txtcorreo.Text, txtclave.Text,
            ((OpcionCombo)cborol.SelectedItem).Valor.ToString(),
            ((OpcionCombo)cborol.SelectedItem).Texto.ToString(),
            ((OpcionCombo)cboestado.SelectedItem).Valor.ToString(),
            ((OpcionCombo)cboestado.SelectedItem).Texto.ToString()
            });

            Limpiar();
        }
        else
        {
            MessageBox.Show(mensaje);
        }
    }
    else {
        bool resultado = new CN_Usuario().Editar(objusuario, out mensaje);

        if (resultado) {
            DataGridViewRow row = dgvdata.Rows[Convert.ToInt32(txtindice.Text)];
            row.Cells["Id"].Value = txtid.Text;
            row.Cells["Documento"].Value = txtdocumento.Text;
            row.Cells["NombreCompleto"].Value = txtnombrecompleto.Text;
            row.Cells["Correo"].Value = txtcorreo.Text;
            row.Cells["Clave"].Value = txtclave.Text;
            row.Cells["IdRol"].Value = ((OpcionCombo)cborol.SelectedItem).Valor.ToString();
            row.Cells["Rol"].Value = ((OpcionCombo)cborol.SelectedItem).Texto.ToString();
            row.Cells["EstadoValor"].Value = ((OpcionCombo)cboestado.SelectedItem).Valor.ToString();
            row.Cells["Estado"].Value = ((OpcionCombo)cboestado.SelectedItem).Texto.ToString();
        }
    }
}
```

Para eliminar el usuario en el evento click del botón eliminar.

	<b>UT 2 – <i>Práctica Sistema de Ventas</i></b> <b>Parte VII – Formulario Usuarios vs BBDD</b>	CFGS Desarrollo de Aplicaciones Multiplataforma Módulo: DDI
--	---	--

```

156 public bool Eliminar(Usuario obj, out string Mensaje)
157 {
158     bool respuesta = false;
159     Mensaje = string.Empty;
160
161
162     try
163     {
164
165         using (SqlConnection oconexion = new SqlConnection(Conexion.cadena))
166         {
167
168             SqlCommand cmd = new SqlCommand("SP_ELIMINARUSUARIO", oconexion);
169             cmd.Parameters.AddWithValue("IdUsuario", obj.IdUsuario);
170             cmd.Parameters.Add("Respuesta", SqlDbType.Int).Direction = ParameterDirection.Output;
171             cmd.Parameters.Add("Mensaje", SqlDbType.VarChar, 500).Direction = ParameterDirection.Output;
172             cmd.CommandType = CommandType.StoredProcedure;
173
174             oconexion.Open();
175
176             cmd.ExecuteNonQuery();
177
178             respuesta = Convert.ToBoolean(cmd.Parameters["Respuesta"].Value);
179             Mensaje = cmd.Parameters["Mensaje"].Value.ToString();
180
181         }
182     }
183     catch (Exception ex)
184     {
185         respuesta = false;
186         Mensaje = ex.Message;
187     }
188
189     return respuesta;
190 }
191
192
193
194
195
196

```

El evento click del botón limpiar simplemente llamará a la función limpiar.

Lista de Usuarios:

Buscar por:

Quedan por programar los dos botones de búsqueda btnbuscar y btnlimpiarbuscador.

Para el botón buscar btnbuscar la funcionalidad que se busca es que nos visualice todas las filas que en la columna especificada CONTENGAN el valor indicado en el cuadro de texto siguiente. Primero tenemos que obtener la columna por la que queremos realizar el filtro. Si tenemos filas en el grid recorreremos cada fila del grid buscando el valor especificado en el campo txtbusqueda de sea igual que a alguno de los valores de la columna especificada. Para que no evalúe mayúsculas/minúsculas la comparación la hacemos en mayúsculas.

	UT 2 – <i>Práctica Sistema de Ventas</i> Parte VII – Formulario Usuarios vs BBDD	CFGS Desarrollo de Aplicaciones Multiplataforma Módulo: DDI
--	--	---

```
private void btnbuscar_Click(object sender, EventArgs e)
{
    string columnaFiltro = ((OpcionCombo)cbobusqueda.SelectedItem).Valor.ToString();

    if (dgvdata.Rows.Count > 0) {
        foreach (DataGridViewRow row in dgvdata.Rows) {
            if (row.Cells[columnaFiltro].Value.ToString().Trim().ToUpper().Contains(txtbusqueda.Text.Trim().ToUpper()))
                row.Visible = true;
            else
                row.Visible = false;
        }
    }
}
```

btnlimpiarbuscador borrará el contenido de txtbusqueda y visualizará de nuevo todos los datos:

```
private void btnlimpiarbuscador_Click(object sender, EventArgs e)
{
    txtbusqueda.Text = "";
    foreach (DataGridViewRow row in dgvdata.Rows)
    {
        row.Visible = true;
    }
}
```