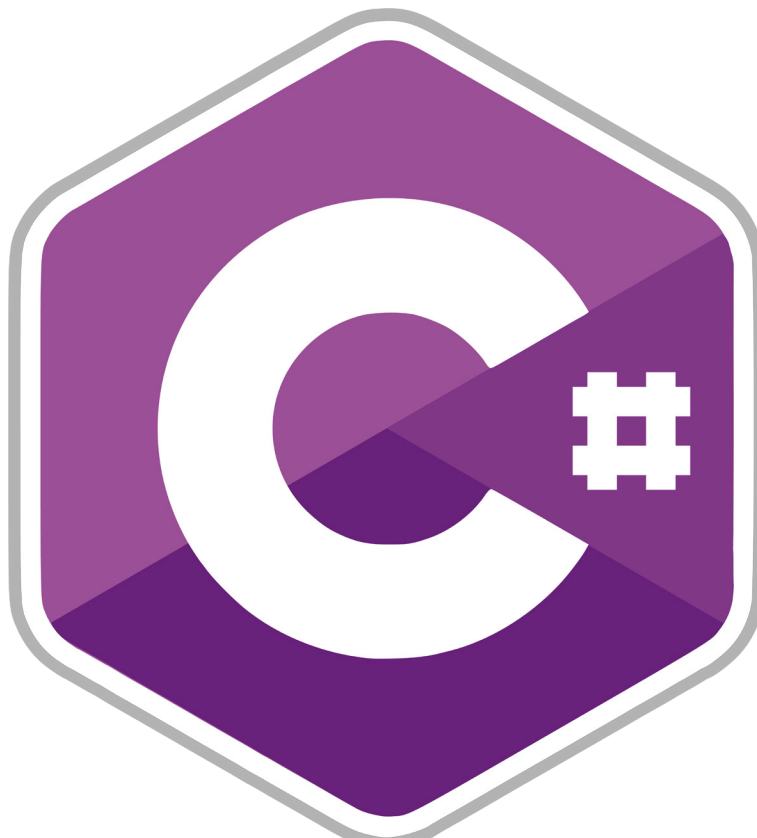


# Práctica 1. Proyecto integrador



20 Noviembre 2019

## Indice

### MiUI 3

1.Estructura del Form Main_menu y elementos .....	3
1.2 Elementos utilizados .....	4
1.3 Elementos utilizados.....	5
2.Estructura del Form inicio y elementos .....	7
3. Estructura del *Form.cs y elementos .....	8
3.1*From.cs .....	9
3.2 Elementos funcionales .....	11
3.3 Comprobación de la información.....	14
4. Bibliografía .....	15

# Desarrollo de Interfaces

## MiUI

En este apartado vamos a realizar el análisis del apartado gráfico, así como se a llevado cabo, cada uno de los elementos así como sus estilos y atributos.

### 1.Estructura del Form Main\_menu y elementos.



main\_1



main\_1

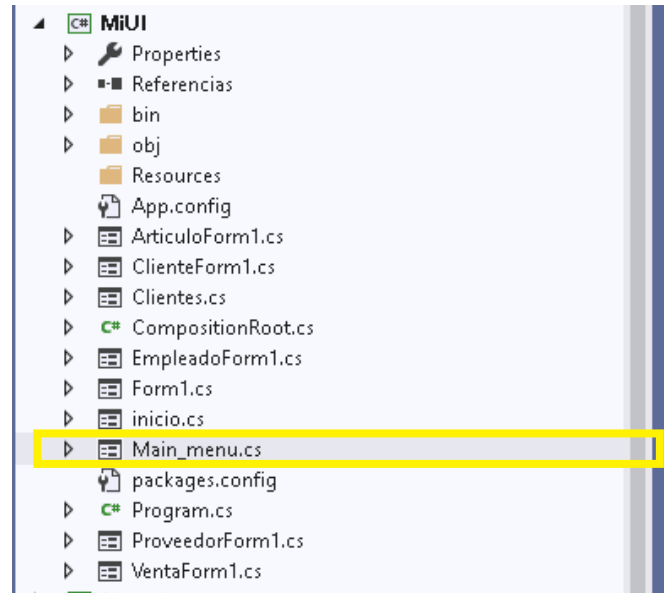
# Desarrollo de Interfaces

## 1.2 Elementos utilizados.

Entre los elementos utilizados en Main\_menu se encuentran:

```
23  |
438 |
439 | private System.Windows.Forms.Panel panel_Titulo;
440 | private System.Windows.Forms.Panel panel_Menu1;
441 | private System.Windows.Forms.Panel contenedor;
442 | private System.Windows.Forms.PictureBox btn_salir;
443 | private System.Windows.Forms.PictureBox btn_min;
444 | private System.Windows.Forms.PictureBox btn_max;
445 | private System.Windows.Forms.PictureBox btn_res;
446 | private System.Windows.Forms.PictureBox pictureBox1;
447 | private System.Windows.Forms.Button btn_articulos;
448 | private System.Windows.Forms.Panel panel1;
449 | private System.Windows.Forms.Panel panel6;
450 | private System.Windows.Forms.Button btn_reportes;
451 | private System.Windows.Forms.Panel panel5;
452 | private System.Windows.Forms.Button btn_proveedores;
453 | private System.Windows.Forms.Panel panel4;
454 | private System.Windows.Forms.Button btn_ventas;
455 | private System.Windows.Forms.Panel panel3;
456 | private System.Windows.Forms.Button btn_clientes;
457 | private System.Windows.Forms.Panel panel2;
458 | private System.Windows.Forms.Button btn_empleado;
459 | private System.Windows.Forms.Panel panel7;
460 | private System.Windows.Forms.Button btn_quejas;
461 | private System.Windows.Forms.PictureBox pictureBox2;
462 | private System.Windows.Forms.Label lfecha;
463 | private System.Windows.Forms.Label lhora;
464 | private System.Windows.Forms.Timer timer1;
465 |
466 | }
```

Captura\_1



Captura\_0

La mayoría del desarrollo se apoyo con material proporcionado en clase.

Elementos con imágenes pictureBox.



main\_1

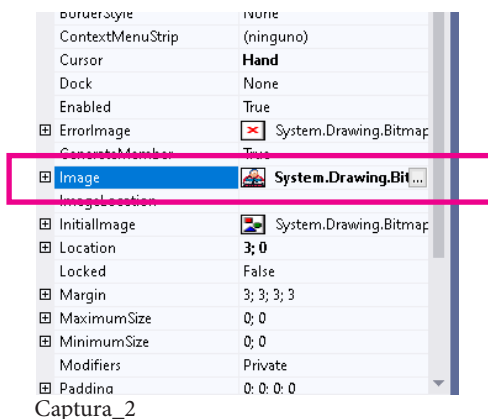


main\_1

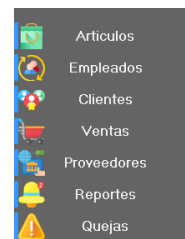


main\_1

Al crearlos ay que exportar la imagen que se desee o con ayuda de la ventana de propiedades.



Captura\_2



main\_1

Los botones también tienen imágenes que se le dio con ayuda de la ventana de propiedades.

# Desarrollo de Interfaces

Todos los elementos se encuentran en una ventana sin ningún tipo de estilo en el borde. Esta ventana contiene 3 paneles principales en donde se va a distribuir todos los elementos.

BackgroundImageLayout	Tile
BorderStyle	None
CausesValidation	True
ContextMenuStrip	(ninguno)

Todos los Form se les a quitado el estilo del borde para la realización de la actividad.

## 1.3 Elementos funcionales

En el panel superior se encuentra 4 imágenes que nos servirán para realizar diferentes acciones, tales como minimizar, maximizar , salir etc.



main\_1

Cada elemento realizar diferentes eventos que se programaron para poder manipular el estado de la venta.

```
1 referencia
private void salir_Click(object sender, EventArgs e)
{
    Application.Exit();
}

1 referencia
private void maximizar_Click(object sender, EventArgs e)
{
    this.WindowState = FormWindowState.Maximized;
    btn_max.Visible = false;
    btn_res.Visible = true;
}

1 referencia
private void pictureBox2_Click(object sender, EventArgs e)
{
    this.WindowState = FormWindowState.Normal;
    btn_res.Visible = false;
    btn_max.Visible = true;
}

1 referencia
private void btn_min_Click(object sender, EventArgs e)
{
    this.WindowState = FormWindowState.Minimized;
}
```

Captura\_4

El panel que se encuentra en la parte izquierda contiene una imagen del logo , seguida de numeroso botones, todos estos elementos nos ayudan a visualizar en el panel con mayor tamaño cada una de las ventanas creadas para cada Form de cada entidad.

```
private ArtículoForm1 vf1 = CompositionRoot.Resolve<ArtículoForm1>();
private ClienteForm1 cf1 = CompositionRoot.Resolve<ClienteForm1>();
private EmpleadoForm1 vef1 = CompositionRoot.Resolve<EmpleadoForm1>();
private ProveedorForm1 eef1 = CompositionRoot.Resolve<ProveedorForm1>();
private VentaForm1 df1 = CompositionRoot.Resolve<VentaForm1>();
private Form formhija;

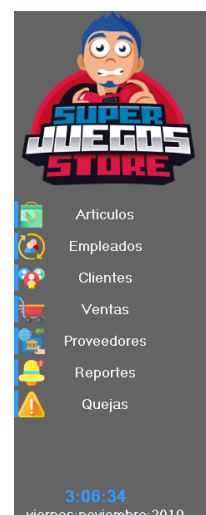
0 referencias
public Titulo()
{
    InitializeComponent();
    vf1.TopLevel = false;
    cf1.TopLevel = false;
    vef1.TopLevel = false;
    eef1.TopLevel = false;
    df1.TopLevel = false;
}

1 referencia
private void btnProductos_Click(object sender, EventArgs e)
{
    if (this.contenedor.Controls.Count > 0)
        this.contenedor.Controls.RemoveAt(0);
    // Form cf1 = formhija as Form;
    // cf1.TopLevel = false;
    vf1.Dock = DockStyle.Fill;
    this.contenedor.Controls.Add(vf1);
    this.contenedor.Tag = vf1;
    vf1.Show();
}

//cambiar de Ventana
```

Captura\_5

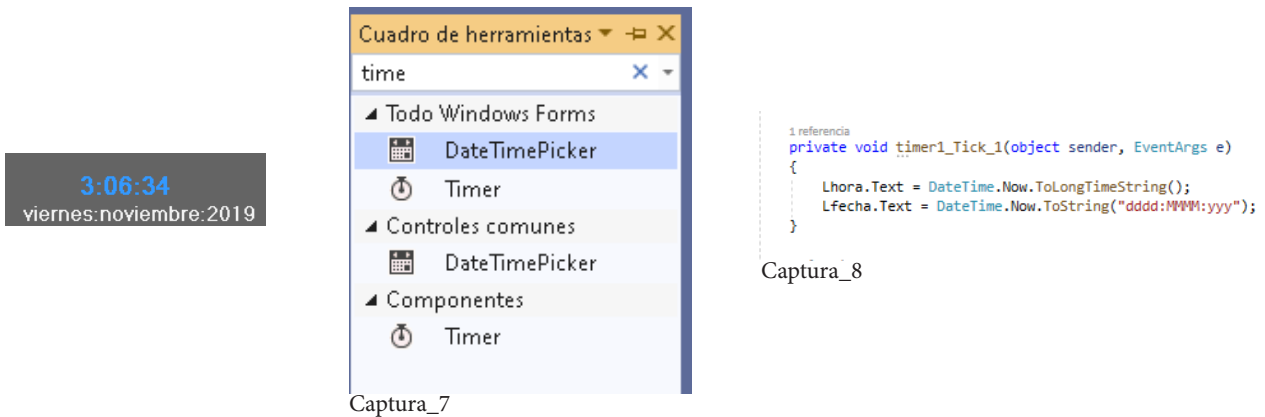
Captura\_6



main\_1

# Desarrollo de Interfaces

Para la realización del Reloj y fecha se a realizado con ayuda de un elemento llamado Timer



Al dar click a uno de los siguientes elementos los llevara a otro form, en cuestión al Form llamado inicio en donde se pueden visualizar una ventana simple con cuatro elementos .



main\_1



main\_1

```
1 referencia
private void pictureBox1_Click(object sender, EventArgs e)
{
    AbrirHija(new inicio());
}
1 referencia
```

Captura\_9

```
2 referencias
private void AbrirHija(object formhija) {
    if (this.contenedor.Controls.Count > 0)
        this.contenedor.Controls.RemoveAt(0);

    Form fh = formhija as Form;
    fh.TopLevel = false;
    fh.Dock = DockStyle.Fill;
    this.contenedor.Controls.Add(fh);
    this.contenedor.Tag = fh;
    fh.Show();
}
```

Captura\_10

# Desarrollo de Interfaces

## 2.Estructura del Form inicio y elementos.

Esta ventana no tiene ninguna funcionalidad, solo muestra información sobre la empresa , y los nombre de los desarrolladores del programa.



inicio\_1

```
Windows Form Designer generated code

private System.Windows.Forms.PictureBox pictureBox1;
private System.Windows.Forms.Label label12;
private System.Windows.Forms.Label label11;
private System.Windows.Forms.Label label13;
}
```

Captura\_11

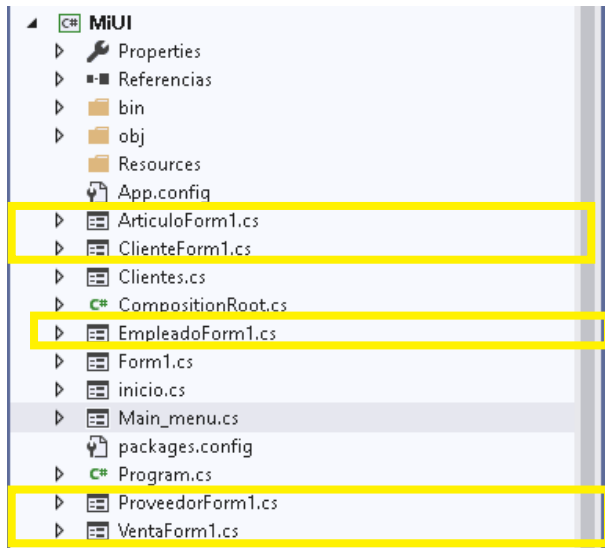


inicio\_1

# Desarrollo de Interfaces

## 3.Estructura del \*Form.cs y elementos.

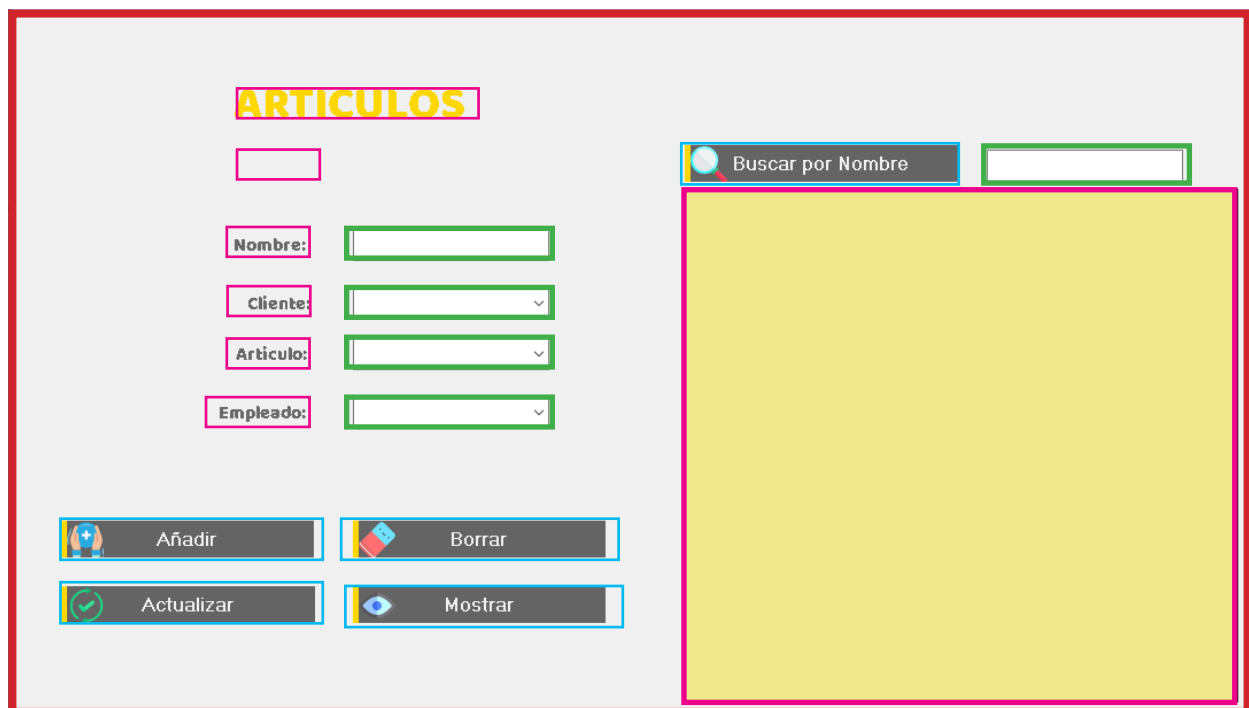
Todos los Form restantes tienen la misma forma base con pequeños cambios de color y iconos así como funcionalidades parecidas.



Captura\_12



form\_articulos



form\_articulos



# Desarrollo de Interfaces



\*From.cs



**Proveedores**

Nombre:


Telefono:

Email:

 Añadir  Borrar

 Actualizar  Mostrar

Buscar por Nombre





form\_proveedores



**EMPLEADOS**

Nombre:


DNI-NIF-CIF:

Numero telefono:

 Añadir  Borrar

 Actualizar  Mostrar

Buscar por Nombre



form\_empleados

# Desarrollo de Interfaces

CLIENTES

Nombre:

DNI-NIF-CIF:

Numero telefono:

Email:

Añadir

Borrar

Actualizar

Mostrar

Buscar por Nombre

form\_clientes

ARTICULOS

Nombre:

Precio:

Stock:

Categoría:

Añadir

Borrar

Actualizar

Mostrar

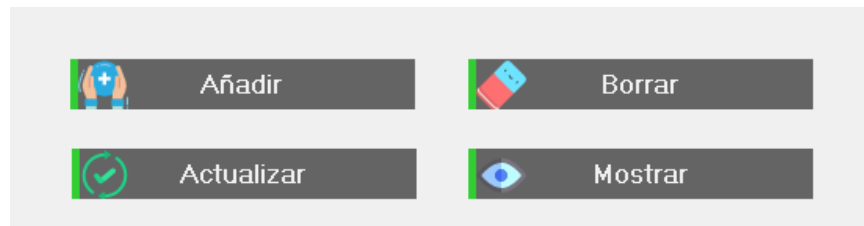
Buscar por Nombre

form\_articulos

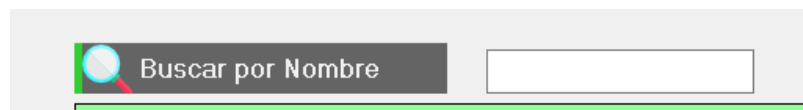
# Desarrollo de Interfaces

## Elementos funcionales

Las Funcionalidades Principales de los Form se centran principalmente la manipulación de los datos, tanto creando nuevos registros, actualización, borrando, buscar, mostrar, además de verificar la información que se desea introducir.

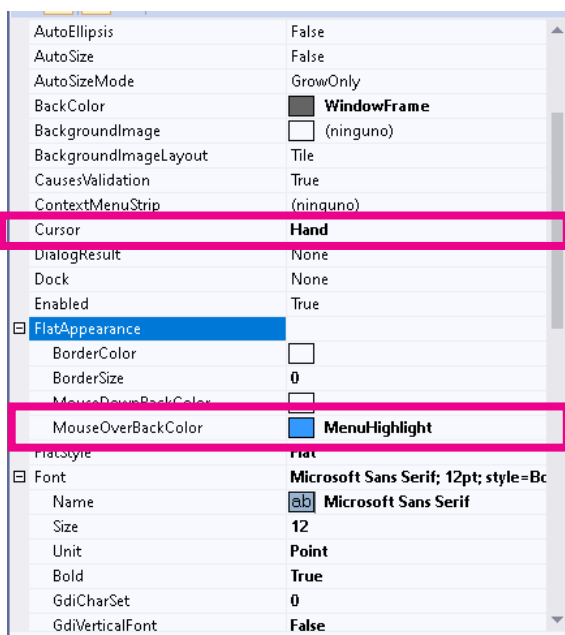


form\_articulos

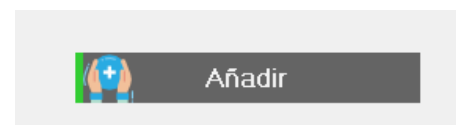


form\_articulos

Añadir: este botón nos crea un registro nuevo dependiendo de la ventana en la que nos encontremos. A todos los botones se les a dado una interacción de posicionamiento cuando en mouse se posa encima, esto hace que cambie de color, esto se activa con ayuda de la ventana de propiedades.



Captura\_13



```
1 referencia
private void btn_anadir_Click(object sender, EventArgs e)
{
    if (ComprobarCampoVacio())
    {
        MessageBox.Show("Rellena todos los campos.");
    }
    else if (ComprobarEstadoArticulo())
    {
        MessageBox.Show("El articulo ya existe.");
    }
    else
    {
        Articulo nuevo = new Articulo();
        nuevo.Id = Guid.NewGuid().ToString();
        nuevo.Nombre = tb_nombre.Text;
        nuevo.Precio = tb_precio.Text;
        nuevo.Stock = tb_stock.Text;
        nuevo.Categoria = cbCategoria.Text;
        _servicioArticulo.Add(nuevo);
        MessageBox.Show("Se ha Añadido un Articulo. "+tb_nombre.Text);
    }
    dataGridView1.DataSource = _servicioArticulo.GetAll();
}
```

Captura\_14

```
2 referencias
private bool ComprobarCampoVacio()
{
    espaciosNombre = tb_nombre.Text.Replace(" ", "");
    espaciosPrecio = tb_precio.Text.Replace(" ", "");
    espaciosStock = tb_stock.Text.Replace(" ", "");
    espaciosCategoria = cbCategoria.Text.Replace(" ", "");
    if (espaciosNombre.Equals("") ||
        espaciosPrecio.Equals("") ||
        espaciosStock.Equals("") || espaciosCategoria.Equals(""))
    {
        return true;
    }
    else
    {
        return false;
    }
}
```

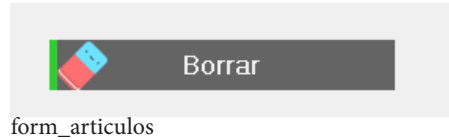
Captura\_15

El evento utiliza una función externa llamada comprobarCapoVacio()

# Desarrollo de Interfaces

## Elementos funcionales

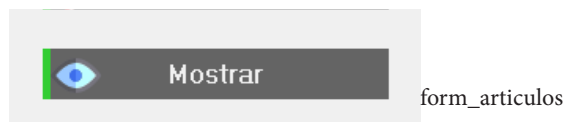
Borrar: para borrar se tiene que seleccionar con el mouse el registro que se visualiza en dataGridView1. Antes de realizar la acción se debe mostrar para poder seleccionar.



```
private void btn_borrar_Click(object sender, EventArgs e)
{
    try
    {
        _servicioArticulo.EliminarArticulo(dataGridView1.CurrentRow.Value.ToString());
        dataGridView1.DataSource = _servicioArticulo.GetAll();
        MessageBox.Show("Se ha Borrado un registro ");
        // _servicioCliente.EliminarCliente(tbNombre.Text);
        // Re
    }
    catch (Exception)
    {
        MessageBox.Show("No se ha seleccionado nada para borrar " + MessageBoxIcon.Exclamation);
        dataGridView1.DataSource = _servicioArticulo.GetAll();
    }
}
```

Captura\_16

Mostrar: en Botón mostrar nos ayudara a visualizar toda la información de cada uno de los registros en dataGridView1.



```
1 referencia
private void btn_mostrar_Click(object sender, EventArgs e)
{
    dataGridView1.DataSource = _servicioArticulo.GetAll();
}
```

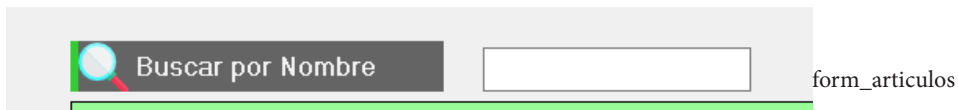
Captura\_17



# Desarrollo de Interfaces

## Elementos funcionales

Buscar: para poder buscar se debe utilizar la caja de texto que se encuentra a su lado derecho.



```
f
1 referencia
private void btn_buscar_Click(object sender, EventArgs e)
{
    dataGridView1.DataSource = _servicioArticulo.GetAll(x => x.Nombre.Contains(txt_buscar.Text));
}
Captura_18
```



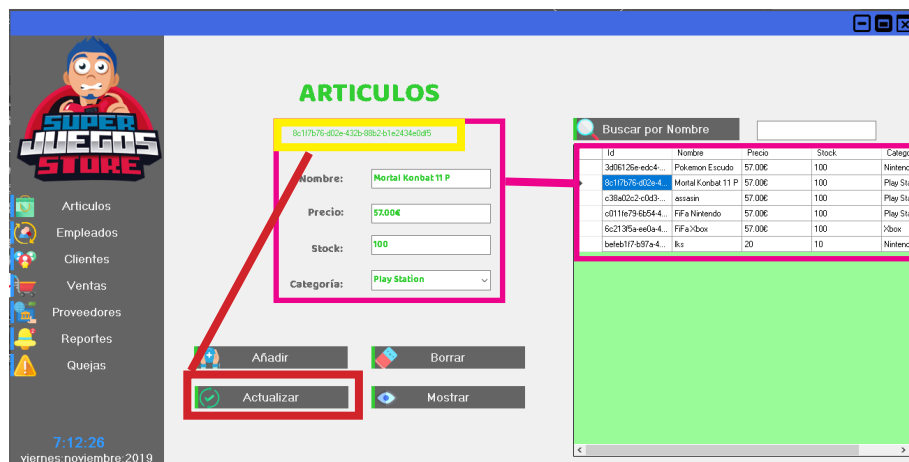
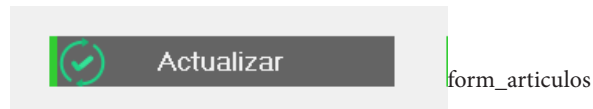
Evento de apoyo para el botón actualizar.

```
1 referencia
private void dataGridView1_CellClick(object sender, DataGridViewCellEventArgs e)
{
    try
    {
        int index = e.RowIndex;
        DataGridViewRow selectedRow = dataGridView1.Rows[index];
        label_ID.Text = selectedRow.Cells[0].Value.ToString();
        tb_nombre.Text = selectedRow.Cells[1].Value.ToString();
        tb_precio.Text = selectedRow.Cells[2].Value.ToString();
        tb_stock.Text = selectedRow.Cells[3].Value.ToString();
        cbCategoria.Text = selectedRow.Cells[4].Value.ToString();
    }
    catch (Exception)
    {
        MessageBox.Show("Seleccionen un Registro " + MessageBoxIcon.Exclamation);
    }
}
```

# Desarrollo de Interfaces

## Elementos funcionales

Actualizar: el botón actualizar se apoya de un evento cellclick , este evento lleva la ID aun label para que luego la actualización se realice sobre el registro con la ID .



## Comprobación de la información.

Con estas lineas se intenta filtrar los datos que se quieren introducir.

Comprueba si se introduce datos de tipo int.

```
1 referencia
private void tb_stock_KeyPress(object sender, KeyPressEventArgs e)
{
    if (char.IsNumber(e.KeyChar))
    {
        e.Handled = false;
    }
    else if (char.IsControl(e.KeyChar))
    {
        e.Handled = false;
    }
    else if (tb_stock.Text.Trim().Length != 4 || tb_stock.Text.Trim().Length < 4)
    {
        MessageBox.Show("El telefono debe tener 4 diguitos");
        e.Handled = false;
    }
    else
    {
        e.Handled = true;
    }
}
```

# Desarrollo de Interfaces

## Comprobación de la información.

Comprueba si es valido la Documentación.

```
//Metodo para validar Documentacion
1 referencia
public static Boolean valida_NIFCIFNIE(string data)[...]

1 referencia
private static string getLetra(int id)[...]

4 referencias
private static bool validarNIF(string data)[...]

1 referencia
private static bool validadCIF(string data)[...]

1 referencia
```

Comprueba si es valido el Email.

```
//Metodo para validar un Email
1 referencia
public bool ValidarEmail(String email)
{
    if (email == "")
    {
        // MessageBox.Show("No se ha introducido nada " + MessageBoxIcon.Exclamation);
        return true;
    }
    try
    {
        new MailAddress(email);
        return true;
    }
    catch (FormatException)
    {
        return false;
    }
}
1 referencia
```

## Bibliografía

<https://www.youtube.com/watch?v=eCSbUCL4teE>  
<https://www.youtube.com/watch?v=68NGvO9ZdRQ>  
[https://www.youtube.com/watch?v=FRK4V9c9A\\_Q&t=529s](https://www.youtube.com/watch?v=FRK4V9c9A_Q&t=529s)  
[https://www.youtube.com/watch?v=FRK4V9c9A\\_Q](https://www.youtube.com/watch?v=FRK4V9c9A_Q)  
<https://www.youtube.com/watch?v=zXVV0Ihgmus>  
<https://geeks.ms/jvelasco/2011/01/03/c-validar-cif-nif-nie/>