UT 3 - Creación de componentes

CREACIÓN DE COMPONENTES WINDOWS FORMS

Visual Studio 2019 - C#

✓ Componentes

Un **componente software** es una *clase* creada para ser reutilizada y que puede ser manipulada por una herramienta de desarrollo de aplicaciones visual. Se define por su **estado** que se almacena en un conjunto de **propiedades**, las cuales pueden ser modificadas para adaptar el componente al programa en el que se inserte.

También tiene un **comportamiento** que se define por los eventos ante los que responde y los **métodos** que ejecuta ante dichos eventos.

Para que pueda ser distribuida, se empaqueta con todo lo necesario para su correcto funcionamiento, quedando independiente de otras bibliotecas o componentes.

Visual Studio 2019 - C#

Un Windows Forms es una clase que deriva directa o indirectamente de System.Windows.Forms.Control .

- ✓ **Combinación de controles existentes para crear un control compuesto.**

  Por eje. un control que consta de un cuadro de texto y un botón de restablecimiento.

- ✓ **Extender un control existente para personalizarlo o agregarlo a su funcionalidad.**

  P.ej. un botón que como propiedad adicional nos dirá cuántas veces se ha clicado

- ✓ **Creación de un control que no combina ni extiende los controles existentes.**

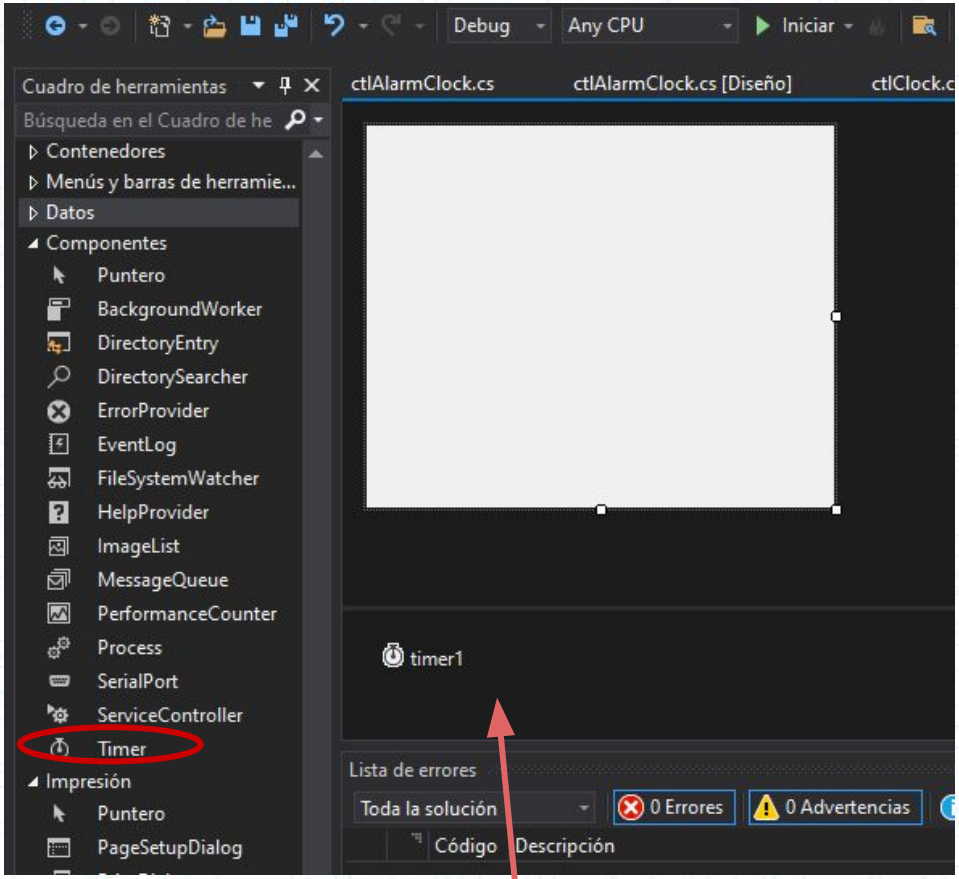
https://docs.microsoft.com/es-es/dotnet/desktop/winforms/controls/how-to-inherit-from-existing-windows-forms-controls?view=netframeworkdesktop-4.8

Visual Studio 2019 - C#

✓ Pasos para crear un nuevo componente

1. Creamos un nuevo proyecto Windows biblioteca de clases .NET Framework (nombre:ctlClockLib)
2. Eliminamos la clase que crea por defecto y agregamos un nuevo ítem (de tipo Control de Usuario) al proyecto (nombre `ctlClock.cs`)
3. Añadiremos una etiqueta y un temporizador al control

✓ Timer



Ojo que el Timer no queda representado en el ‘diseño’ sino que aprece abajo, si lo queremos manipilar

Visual Studio 2019 - C#

✓ Propiedades

❑ Label:

Propiedad	Cambiar a
Nombre	`lblDisplay`
Texto	`(blank space)`
TextAlign	`MiddleCenter`
Font.Size	`14`

❑ Timer:

Propiedad	Cambiar a
Nombre	`timer1`
Enable	`Ture`
Interval	`1000`

La propiedad Interval controla la frecuencia con la que el componente Timer tics. Cuando `timer1` hace tic, se ejecuta el código en el evento `timer1_Tick`. El intervalo => ms entre tics.

Visual Studio 2019 - C#

- ✓ **Añadimos un evento a timer1**

```
    protected virtual void timer1_Tick(object sender, EventArgs e)
    {
        // Causes the label to display the current time.
        lblDisplay.Text = DateTime.Now.ToLongTimeString();
    }
```

- ✓ **Compilamos:** y vemos que se ha generado dll en debug

Visual Studio 2019 - C#

✓ Añadimos propiedades al control:

```
//Atributos
   private Color colFColor;
   private Color colBColor;
       //--------Aquí el controlador-----
   //Métodos
   public Color ClockBackColor
   {
      // Retrieves the value of the private variable colBColor.
      get  {              return colBColor;          }
      // Stores the selected value in the private variable colBColor, and
      // updates the background color of the label control lblDisplay.
      set  {
         colBColor = value;
         lblDisplay.BackColor = colBColor;
      }
   }
   // Provides a similar set of instructions for the foreground color.
   public Color ClockForeColor
   {
      get   {              return colFColor;          }
      set
      {
         colFColor = value;
         lblDisplay.ForeColor = colFColor;
      }
   }
```

Visual Studio 2019 - C#

- ✓ **Como este nuevo componente es en sí una clase lo podemos heredar:**

  - ❑ En *Explorador de soluciones*, haga clic con el botón derecho en ctlClockLib, seleccione Agregar y, a continuación, haga clic en Control de usuario. ---> Agregar nuevo elemento.
  - ❑ Seleccione la plantilla Control de usuario heredado. (nombre: `ctlAlarmClock.cs`)
  - ❑ Aparece el cuadro de diálogo Selector de herencia donde seleccionamos nuestro .

Visual Studio 2019 - C#

✓ Añadimos propiedades al control nuevo:

```
//Atributos
  private DateTime dteAlarmTime;
  private bool blnAlarmSet;
  private bool blnColorTicker;

  // Getters y setters
  public DateTime AlarmTime
  {
     get {
        return dteAlarmTime;
     }
     set  {
        dteAlarmTime = value;
     }
  }
  public bool AlarmSet
  {
     get{
        return blnAlarmSet;
     }
     set {
        blnAlarmSet = value;
     }
  }
```

Visual Studio 2019 - C#

✓ En la interfaz gráfica del control

- Añadimos un label

Propiedad	Cambiar a
Nombre	`lblAlarm`
Texto	¡Alarma!
TextAlign	`MiddleCenter`
Visible	`false`

- y un botón

Propiedad	Cambiar a
Nombre	`btnAlarmOff`
Texto	Deshabilitar alarma

Visual Studio 2019 - C#

✓ En la interfaz gráfica del control ‘Alarm’

- ❑ Añadimos un label

Propiedad	Cambiar a
Nombre	`lblAlarm`
Texto	¡Alarma!
TextAlign	`MiddleCenter`
Visible	`false`

- ❑ y un botón

Propiedad	Cambiar a
Nombre	`btnAlarmOff`
Texto	Deshabilitar alarma

```
private void btnAlarmOff_Click(object sender, System.EventArgs e)
{
    // Turns off the alarm.
    AlarmSet = false;
    // Hides the flashing label.
    lblAlarm.Visible = false;
}
```

✓ Sobreescribiremos el método timer1_Tick de ctlClock:

```
protected override void timer1_Tick(object sender, System.EventArgs e)
 {
    // Calls the Timer1_Tick method of ctlClock.
    base.timer1_Tick(sender, e);
    // Checks to see if the alarm is set.
    if (AlarmSet == false)
       return;
    else          {
    // If the date, hour, and minute of the alarm time are the same as/ the current time, flash an alarm.

       if (AlarmTime.Date == DateTime.Now.Date && AlarmTime.Hour ==
          DateTime.Now.Hour && AlarmTime.Minute == DateTime.Now.Minute)   {
          // Sets lblAlarmVisible to true, and changes the background color based on the value of
          // blnColorTicker. The background color of the label  will flash once per tick of the clock.
          lblAlarm.Visible = true;
          if (blnColorTicker == false) {
             lblAlarm.BackColor = Color.Red;
             blnColorTicker = true;
          }
          else  {
             lblAlarm.BackColor = Color.Blue;
             blnColorTicker = false;
          }
       }
       else {
          // Once the alarm has sounded for a minute, the label is made          // invisible again.
          lblAlarm.Visible = false;
       }            }        }
```

Visual Studio 2019 - C#

✓ **Y ya estamos!**

Compilamos para generar la dll

- y en el formulario donde lo queremos usar:

  Creamos:

  -label

  -dateTimePicker

Propiedad	Propiedad	Cambiar a
label1	Texto	(blank space)
	Nombre	lblTest
dateTimePicker1	Nombre	dtpTest
	Format	Time

Visual Studio 2019 - C#

- **Ahora hacer doble clic en dtpTest.**
  (el datetimePicker)

  Y programamos en evento

```
private void dtpTest_ValueChanged(object sender, System.EventArgs e)
{

    ctlAlarmClock1.AlarmTime = dtpTest.Value;
    ctlAlarmClock1.AlarmSet = true;
    Console.WriteLine(Controls.Count);
    for (int i = 0; i < Controls.Count; i++) {
        if(Controls[i].Name == "ctlAlarmClockLaia")
        {
            ctlClockLib.ctlAlarmClock all = (ctlClockLib.ctlAlarmClock)Controls[i];
            all.AlarmTime = dtpTest.Value;
            all.AlarmSet = true;
        }
    }
    //
    lblTest.Text = "Alarm Time is " + ctlAlarmClock1.AlarmTime.ToShortTimeString();

}
```