

Interfaz gráfica

Unidad Didáctica 2

Objetivos de aprendizaje

- Crear un interfaz gráfico, utilizando los asistentes de un editor visual.
- Utilizar las funciones del editor para ubicar los componentes del interfaz.
- Modificar las propiedades de los componentes para adecuarlas a las necesidades de la aplicación.
- Analizar el código generado por el editor visual.
- Modificar el código generado por el editor visual.
- Asociar a los eventos las acciones correspondientes.
- Desarrollar una aplicación que incluye el interfaz gráfico obtenido.
- Crear menús que se ajustan a los estándares.
- Crear menús contextuales cuya estructura y contenido siguen los estándares establecidos.
- Distribuir las acciones en menús, barras de herramientas, botones de comando, entre otros, siguiendo un criterio coherente.
- Distribuir adecuadamente los controles en la interfaz de usuario.
- Utilizar el tipo de control más apropiado en cada caso.
- Diseñar el aspecto de la interfaz de usuario (colores y fuentes, entre otros) atendiendo a su legibilidad.
- Verificar que los mensajes generados por la aplicación son adecuados en extensión y claridad.

Material

- Drive **UD2**
 - Apuntes:
 - Introducción a POO
 - Introducción a Formularios
 - Teoría
 - Ejemplos Pruebas
 - Ejercicios

Aplicaciones con interfaz gráfica

Dentro de .NET, las ventanas clásicas Windows, reciben la denominación de **Windows Forms Applications** para diferenciarlas de los formularios Web o WebForms.

Un **Windows Forms Application** representa la conocida ventana, que se utiliza en las aplicaciones ejecutadas bajo alguno de los sistemas operativos de la familia Windows: Windows95/98, NT, ME, 2000, XP, Windows7, etc.

Ventanas

El (IDE) Entorno de Desarrollo Integrado de Visual .Net, nos va a permitir crear aplicaciones que utilice interfaz gráfica. Para realizar una aplicación que muestre un interfaz gráfico, se crean **objetos** que dan lugar a ventanas (o formularios), y sobre esas ventanas se dibujan otros objetos llamados controles.

Un **control**, por otra parte, es aquel elemento situado dentro de una ventana o formulario, y que permite al usuario de la aplicación Windows, interactuar con la misma, para introducir datos o recuperar información.

Se escribirá el código fuente relacionado con la función que tiene que realizar cada objeto del interfaz. Cada objeto estará ligado a un código que permanece inactivo hasta que se produzca el evento que lo active.

Se dice entonces que la Programación es conducida por eventos y orientada a **objetos**

Primeros pasos

- <https://msdn.microsoft.com/es-es/library/dd492132.aspx>
- <https://msdn.microsoft.com/es-es/library/dd492173.aspx>

Programación Orientada a Objetos

Clases y Objetos

Los términos clase y objeto se usan a veces indistintamente pero, en realidad, las clases describen el tipo de los objetos, mientras que los objetos son instancias de clases que se pueden usar. Así, la acción de crear un objeto se denomina creación de instancias.

Una clase es un plano y un objeto es un edificio construido a partir de ese plano.


```

Class CCuenta
  'Atributos
  Private nombre As String
  Private cuenta As String
  Private saldo As Double
  Private tipoDeInterés As Double
  ...
  ...
  'Métodos
  Public Sub New()
  End Sub
  Public Sub New(nom As String, cue As String, _
                  sal As Double, tipo As Double)

    //Code
  End Sub
  Public Sub aumentarSaldo (aumento As Double)
  //code
  End Sub

End Class

```

El **CONSTRUCTOR**, es un método esencia y especial de la clase que es llamado automáticamente siempre que se crea un objeto de esa clase. Su función es iniciar el objeto. Se distingue fácilmente porque tiene el nombre **New** y no puede retornar un valor

```
Dim obj As CCuenta = New CCuenta( )
```

```
obj.aumentarSaldo( xxx )
```

Property

```
Private nombre As String
Public Property Dnombre() As String
    Get
        ' Aquí se devuelve el valor del atributo
        Return Dnombre
    End Get
    Set(Dato As String)
        ' Aquí se asigna el valor del atributo
        Dnombre = Dato
    End Set
End Property
```

```
obj.Dnombre = "ANE"                Invoca al SET
System.Console.WriteLine(obj.Dnombre() ) Invoca al GET
```

Ejercicio opcional

Ejercicio 1: Realiza un programa que lea una cadena de caracteres e imprima el resultado que se obtiene cada vez que se realice una rotación de un carácter a la derecha sobre dicha cadena. El proceso finalizará cuando se haya obtenido nuevamente la cadena de caracteres original.

Por ejemplo,

HOLA AHOL LAHO OLAH HOLA