Progress Bar y Status Strip



Jesus Nieto Leivas

# 

# 

# 

# **Índice**

[**1. ProgressBar 3**](#_iv7g44ffrba3)

[1.1 Casos de uso de el objeto ProgressBar 3](#_bt0vnc75chmf)

[1.2 Cómo crear un ProgressBar 3](#_f6ulljoe0gg1)

[1.3 Principales características del objeto ProgressBar 6](#_c4ezyxu1lp0d)

[**2. StatusStrip 7**](#_zcccpk67csv1)

[2.1 Casos de uso de el objeto StatusStrip 7](#_1i4cletbtt8m)

[2.2 Cómo crear un StatusStrip 7](#_sf22csbnqvap)

[2.3 Principales características del objeto StatusStrip 9](#_mz89svih3epv)

El siguiente programa que vamos a ver se trata de una demostración sencilla de en qué consisten los objetos ProgressBar y Status Strip.

# **1. ProgressBar**

El objeto progress bar se trata de una barra de progreso, no es nada más que un indicador visual para informar al usuario de una situacion en concreto.

## **1.1 Casos de uso de el objeto ProgressBar**

El objeto Progress Bar lo utilizamos sobre todo cuando necesitamos indicar tiempos límite o tiempos de espera. Un ejemplo común de esto sería los tiempos de espera en una instalación o en la carga de un programa, otro ejemplo sería también límite de tiempo en juegos por turnos en el que el usuario tiene un tiempo determinado para realizar una acción y es representado gráficamente en vez de con números.

## 

## **1.2 Cómo crear un ProgressBar**

Primero de nada deberemos seleccionar la ProgressBar de la caja de herramientas y arrastrarla dentro del formulario y crear un timer.

A continuación veremos el código de un ejemplo simple de un progress bar, la función de esto es hacer visualizar cómo se llena la barra poco a poco:

Public Class Form1

Dim i As Integer

Private Sub Form1\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

Timer1.Interval = 100

End Sub

Private Sub bt\_ejemplo\_Click(sender As Object, e As EventArgs) Handles bt\_ejemplo.Click

Timer1.Enabled = True

ProgressBar1.Maximum = 100

i = 1

End Sub

Private Sub bt\_reset\_Click(sender As Object, e As EventArgs) Handles bt\_reset.Click

i = 1

Timer1.Enabled = False

Label1.Text = ""

ProgressBar1.Value = 0

End Sub

Private Sub Timer1\_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick

ProgressBar1.Value = ProgressBar1.Value + 1

Label1.Text = "Cargando " & ProgressBar1.Value & "%"

i += 1

If i > 100 Then

Timer1.Enabled = False

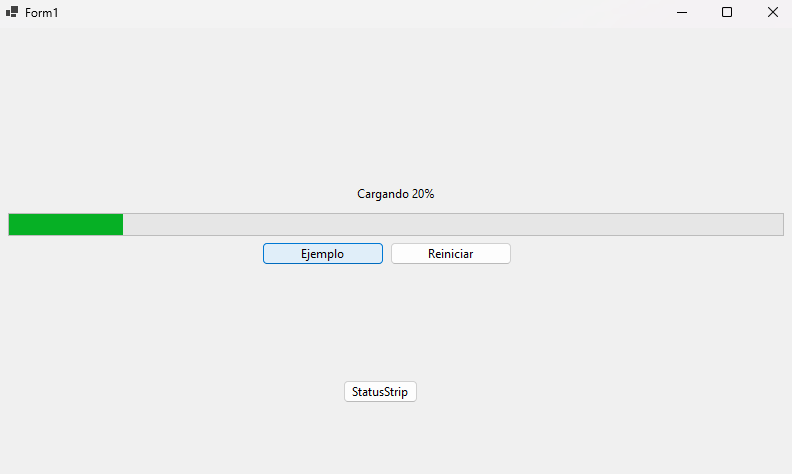
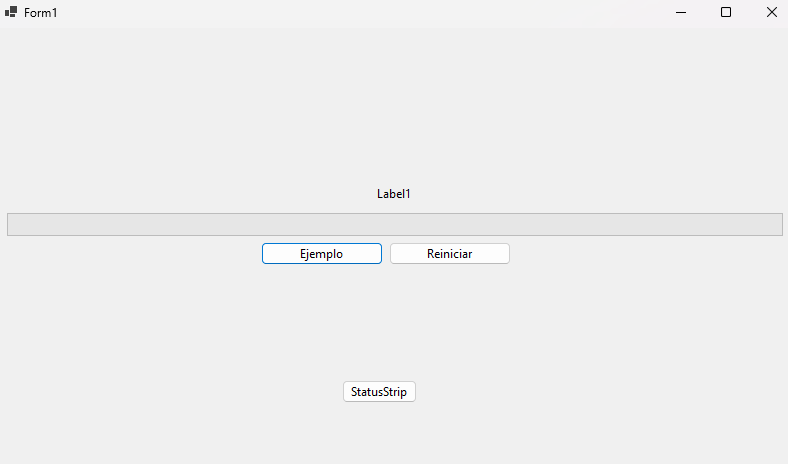
End If

End Sub

Private Sub bt\_statusStrip\_Click(sender As Object, e As EventArgs) Handles bt\_statusStrip.Click

End Sub

End Class



Como podemos observar para la creación de una progress bar es indispensable el uso de un objeto timer para indicar cada cuanto tiempo queremos rellenar la barra, además de eso debemos indicar con el método value cuánto queremos llenarla.

Cada vez que demos clic al botón bt\_ejemplo la barra comenzará a llenarse en un 1% cada 100 ms siendo indicado esto también por el label encima de la barra de progreso.

Cada vez que demos clic al botón bt\_reset reinicia la barra de progreso y borra el contenido del label.

El ejemplo podría haberse hecho al revés si el valor inicial de la barra fuese 100 en vez de 0 y se hubiese ido restando en vez de sumando valor.

## **1.3 Principales características del objeto ProgressBar**

* Propiedades:
  + **BackgroundImage:** Obtiene o establece la imagen de fondo para el control ProgressBar.
  + **BackgroundImageLayout:** Obtiene o establece el diseño de la imagen de fondo de la barra de progreso.
  + **CausesValidation:** Obtiene o establece un valor que indica si el control, cuando recibe el enfoque, hace que la validación se realice en cualquier control que requiera validación.
  + **Font:** Obtiene o establece la fuente de texto en la Barra de progreso.
  + **MarqueeAnimationSpeed:** Obtiene o establece el período de tiempo, en milisegundos, que toma el bloque de progreso para desplazarse a través de la barra de progreso.
  + **Maximum:**Obtiene o establece el valor máximo del rango del control.
  + **Minimum:** Obtiene o establece el valor mínimo del rango del control.
  + **Value:** Obtiene o establece la posición actual de la barra de progreso.
* Métodos:
  + **Increment:** Aumenta la posición actual del control de ProgressBar en una cantidad especificada.
  + **PerformStep:** Aumenta el valor por el paso especificado.
  + **ResetText:** Restablece la propiedad Texto a su valor predeterminado.
  + **ToString:** Devuelve una cadena que representa el control de la barra de progreso.

# **2. StatusStrip**

El objeto StatusStrip es otro indicador visual al igual que el objeto ProgressBar, la diferencia está en que es algo más complejo visualmente hablando. La diferencia está en que se pueden añadir elementos como podrían ser labels y botones que puedan hacer acciones (es decir, puede actuar como un menu strip también).

## **2.1 Casos de uso de el objeto StatusStrip**

Tiene los mismos casos de uso que el objeto hablado anteriormente, de hecho, podría ser hasta mejor debido a que podemos indicar por ejemplo el día y la hora actual, es decir, indicar más datos de una forma más limpia y ordenada que en el anterior objeto a parte de dar opciones al usuario. Pero destaca sobre todo a la hora de hacer instalaciones.

## **2.2 Cómo crear un StatusStrip**

Para crear un status strip necesitaremos seleccionar en la caja de herramientas tanto un timer como el objeto status bar cual se pondrá automáticamente en la parte de abajo (podemos cambiarlo a cualquier lateral del programa). Podemos agregar labels, progress bars, etc. tan solo dando click en el botón de agregar de la propia StatusStrip. Una vez creado todo podremos comenzar a programarlo. Ejemplo:

Public Class Form2

Private Sub Form2\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

ToolStripStatusLabel1.Text = Today

ToolStripStatusLabel2.Text = TimeOfDay

End Sub

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

Timer1.Enabled = True

End Sub

Private Sub Timer1\_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick

If ToolStripProgressBar1.Value < ToolStripProgressBar1.Maximum - 1 Then

ToolStripProgressBar1.Value += 1

Else

Timer1.Enabled = False

End If

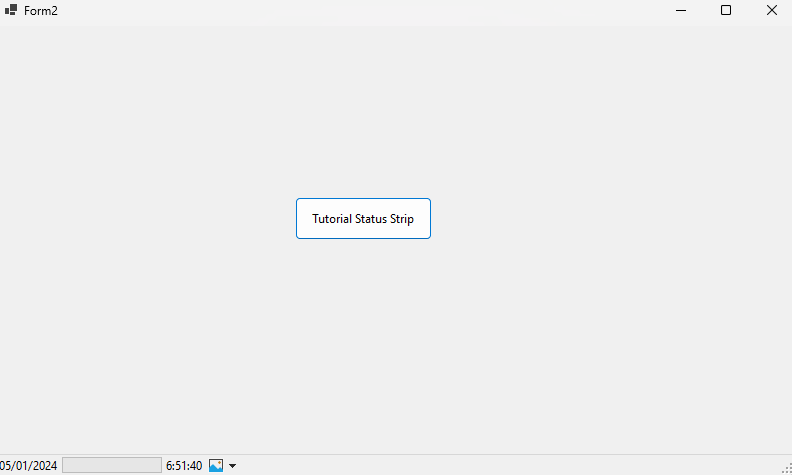
End Sub

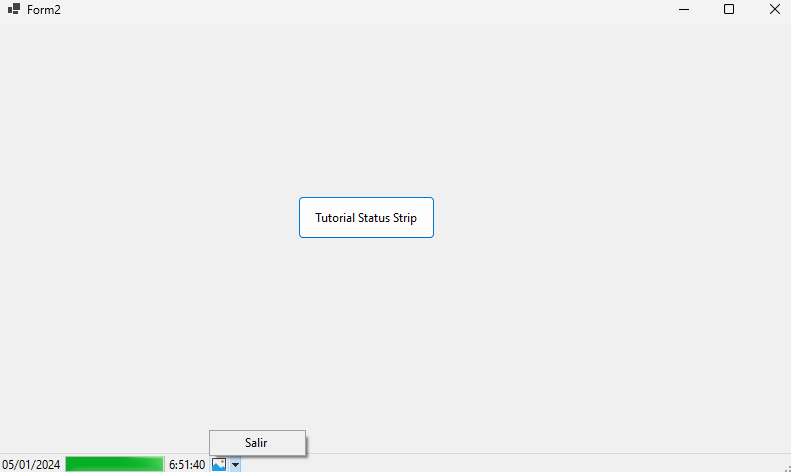
Private Sub SalirToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles SalirToolStripMenuItem.Click

End

End Sub

End Class





Como podemos ver, cuando le damos clic al botón tutorial status strip se comienza a rellenar la barra, aparte de que nada más cargar el formulario están puestos tanto la fecha como la hora exacta y un botón completamente funcional que llevaría al fin de la ejecución del programa.

## **2.3 Principales características del objeto StatusStrip**

| Nombre | Descripción |
| --- | --- |
| [ToolStripStatusLabel](https://learn.microsoft.com/es-es/dotnet/api/system.windows.forms.toolstripstatuslabel) | Representa un panel de control [StatusStrip](https://learn.microsoft.com/es-es/dotnet/api/system.windows.forms.statusstrip). |
| [ToolStripDropDownButton](https://learn.microsoft.com/es-es/dotnet/api/system.windows.forms.toolstripdropdownbutton) | Muestra un [ToolStripDropDown](https://learn.microsoft.com/es-es/dotnet/api/system.windows.forms.toolstripdropdown) asociado en el que el usuario puede seleccionar un elemento único. |
| [ToolStripSplitButton](https://learn.microsoft.com/es-es/dotnet/api/system.windows.forms.toolstripsplitbutton) | Representa un control de dos elementos que son un botón estándar y un menú desplegable. |
| [ToolStripProgressBar](https://learn.microsoft.com/es-es/dotnet/api/system.windows.forms.toolstripprogressbar) | Muestra el estado de finalización de un proceso. |

En el caso del toolStripProgressBar tiene los mismos métodos y propiedades nombrados anteriormente en el objeto ProgressBar.