

4. Implementación

Ph.D Priscilla Jiménez P.

Inferfaz Gráfica

Eventos y Delegados

WPF

Recursos

Eventos

Menús

Otros Recursos en xaml

String

```
<sys:String x:Key="ComboBoxTitle">Items:</sys:String>
```

```
<Label Content="{StaticResource ComboBoxTitle}" />
```

Array

```
<x:Array x:Key="ComboBoxItems" Type="sys:String">
```

```
<sys:String>Item #1</sys:String>
```

```
<sys:String>Item #2</sys:String>
```

```
<sys:String>Item #3</sys:String>
```

```
</x:Array>
```

```
<ComboBox ItemsSource="{StaticResource ComboBoxItems}" />
```

Para obtener un recurso desde el código de C# se usa `_nombreObjeto_.FindResource("nombre_recurso").ToString()`
`_nombreObjeto_` puede ser `this`, `Application.Current` o el nombre del control donde se definió el recurso.

Programación dirigida por eventos

Paradigma de programación en el que la estructura del programa y la ejecución van determinados por sucesos que ocurren en el sistema:

Una acción del usuario (clic del mouse, presionar una tecla).

Mensaje del sistema operativo u otro programa.



La aplicación está diseñada para detectar eventos a medida que ocurren y luego tratarlos mediante un procedimiento de manejo de eventos apropiado.



Un manejador de eventos es un objeto en el que un elemento delega la tarea de manipular un tipo particular de evento.

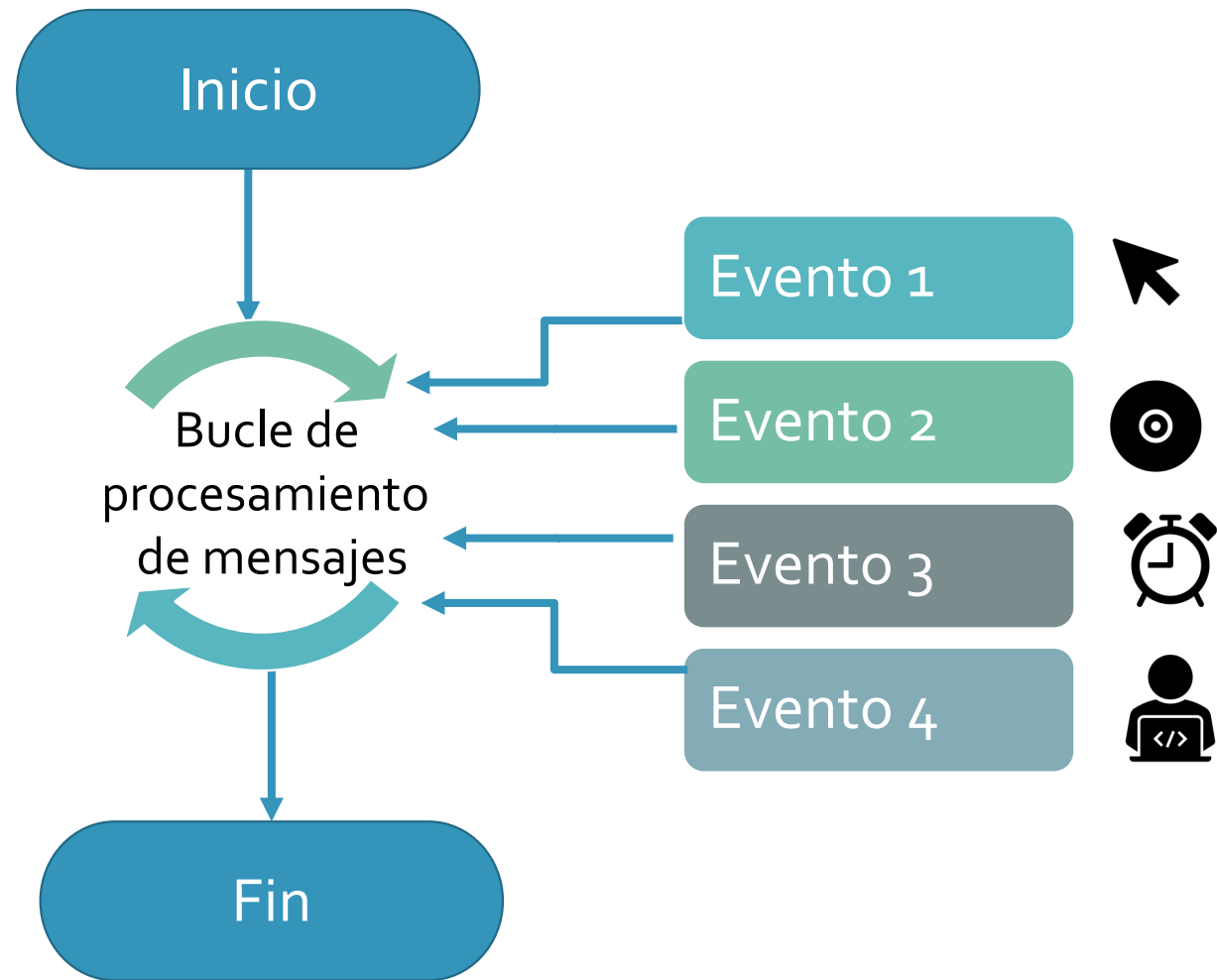
Programación dirigida por eventos

El creador de un programa dirigido por eventos debe definir los eventos que manejarán su programa y las acciones que se realizarán al producirse cada uno de ellos. (administrador de eventos)

Los eventos soportados están determinados por el lenguaje de programación, el sistema operativo, y por eventos creados por el mismo programador.

IDEs facilitan gran parte del código para detectar eventos cuando se crea una aplicación basada en formularios.

Programación dirigida por eventos



El programa queda bloqueado hasta que se produzca un evento. Cuando un evento ocurra, el programa pasará a ejecutar el código del correspondiente administrador o manejador de evento.



Paradigma de programación dirigida por evento

Eventos más comunes -> cuando el usuario interactúa con un componente de la GUI. Un manejador de eventos vinculado con el elemento se encarga de responder al mismo ejecutando el método programado para ello.

Delegados

- Para indicar al emisor de eventos que método responderá a sus eventos se utilizan delegados.
- Un objeto delegado es un objeto que puede contener una referencia a un método.

*modificador **delegate** tipo-retorno identificador (parámetros)*

```
public delegate void RoutedEventHandler( object sender,  
                                         RoutedEventArgs e )
```

sender, especifica el objeto al que está asociado el manejador de eventos.

e, los datos para el evento

El control que genera un evento se conoce como el emisor del evento.

El método manejador es conocido como el receptor del evento.

El método manejador del evento en su mayoría tiene dos parámetros (Object sender, RoutedEventArgs e)

```
<Button x:Name="btnSaludar" Content="Saludar" Click="btnSaludar_Click"/>  
btnSaludar.Click += new RoutedEventHandler (btnSaludar_Click);
```

```
private void btnSaludar_Click(object sender, RoutedEventArgs e)
```

El mecanismo de manejo de eventos en WPF

Utilizar Delegados

PASO 1 : Declaración

- Los delegados son punteros a funciones y al objeto sobre el cual el método es llamado.
- La firma del delegado debe ser igual a la del método y debe incluir la palabra delegate.

```
public delegate void EventHandler( object sender, EventArgs e );  
public delegate void MensajeDelegado(string mensaje);
```

- Para crear una instancia de un delegado se debe pasar el nombre del método (al que se apuntará) como parámetro al constructor del delegado.

Se crea el
método.

Se
indica el
método que
debe llamar un
delegado.

```
void ImprimirEnConsola(string mensaje){}  
void MostrarMensajeBox(string mensaje){}
```

```
btnSaludar.Click += btnSaludar_Click;  
MensajeDelegado msjDelegado = ImprimirEnConsola;
```

¿Cómo llamar a un delegado?

- Se puede llamar al método con `.Invoke(parámetros)` o simplemente como se llama a una función.

Invocación

```
MensajeDelegado.Invoke("Hola");  
MensajeDelegado("Hola");
```

Multidifusión

- Un delegado mantiene una lista de invocación, permiten que varios métodos se llamen en respuesta a un solo evento.
- Los delegados de eventos son multidifusión, representan un conjunto de objetos delegados, en donde todos tienen la misma firma.
- Para agregar un método adicional se utiliza el operador +=
- Para eliminar un método de la lista se utiliza el operador -=
- Los métodos se llamarán en el orden de asignación.
- Los delegados con más de un método en su lista de invocación se derivan de MulticastDelegate que es una subclase de System.Delegate.

¿Por qué son útiles los delegados?

1. Soportan eventos
2. Dan a un programa un medio para ejecutar métodos en tiempo de ejecución sin necesidad de especificar qué método se trate en tiempo de compilación.
 - Ej: aplicación de filtros a imágenes.

Eventos en WPF

Teclado

- Delegado: `KeyboardEventHandler(Object sender, KeyboardEventArgs e)` o `KeyEventHandler(Object sender, KeyEventArgs e)`
- Evento: `PreviewKeyDown`, `KeyDown`, `PreviewKeyUp`, `KeyUp`,

Mouse

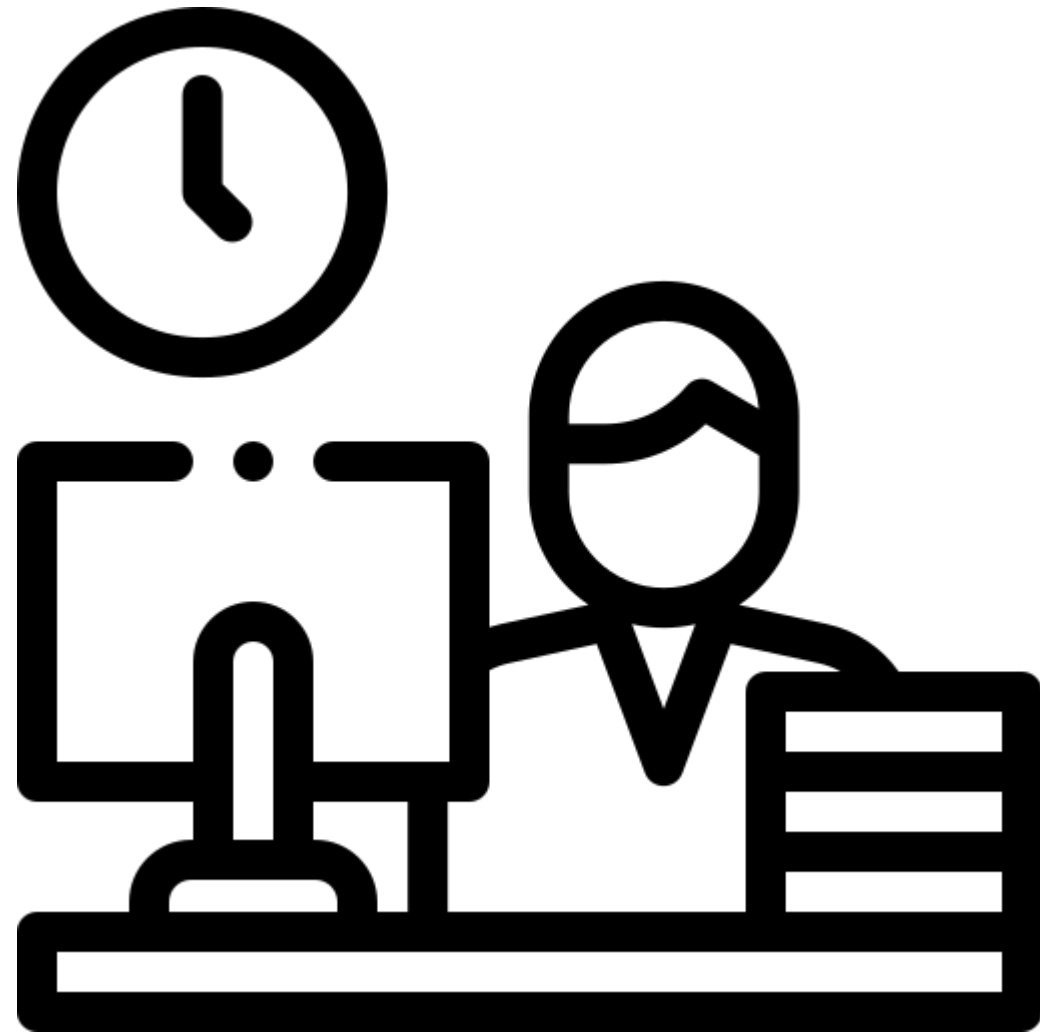
- Delegado `MouseButtonEventHandler(object, System.Windows.Input.MouseEventArgs)`
- Evento: `MouseDown`, `MouseUp`, `MouseMove`

Menu

- <Menu>
 - isMainMenu
 - <MenuItem>
 - Header
 - isCheckedable
 - <MenuItem.Icon>
 - <Image>
 - Source

Práctica 15

Eventos y delegados





Bibliografía y Recursos útiles

- Colmenar, Santos, Antonio. *Visual C#: interfaces gráficas y aplicaciones para Internet con WPF, WCF y Silverlight*, RA-MA Editorial, 2014. ProQuest Ebook Central, Cap 1 – 5.
<https://bvirtual.epn.edu.ec:2057/lib/epnsp/detail.action?docID=5759069>.



Bibliografía y Recursos útiles

- <https://docs.microsoft.com/es-es/dotnet/csharp/programming-guide/delegates/>
- <https://docs.microsoft.com/en-us/dotnet/framework/wpf/controls/menu-overview>
- <https://docs.microsoft.com/en-us/dotnet/framework/wpf/controls/menu-overview#menu-control>
- <https://dotnettutorials.net/lesson/delegates-csharp/>
- <https://www.completecsharp tutorial.com/basic/c-events-tutorial-with-programming-example.php>
- <https://docs.microsoft.com/es-es/dotnet/desktop-wpf/fundamentals/xaml-resources-define>
- <https://blogs.msmvps.com/otelis/2015/07/21/eventos-del-teclado-en-wpf/>
- <https://docs.microsoft.com/en-us/dotnet/api/system.windows.input.keyboard.keydown?view=netcore-3.1>



Bibliografía y Recursos útiles

1. Videos:

- Delegados

https://www.youtube.com/watch?v=8e_z_4SwrRc

<https://www.youtube.com/watch?v=GQbJH6HRenE>

- Eventos WinForms:

<https://www.youtube.com/watch?v=INFcXyDQsVg>

- Eventos WPF:

https://www.youtube.com/watch?v=_JgLcsdTWC8