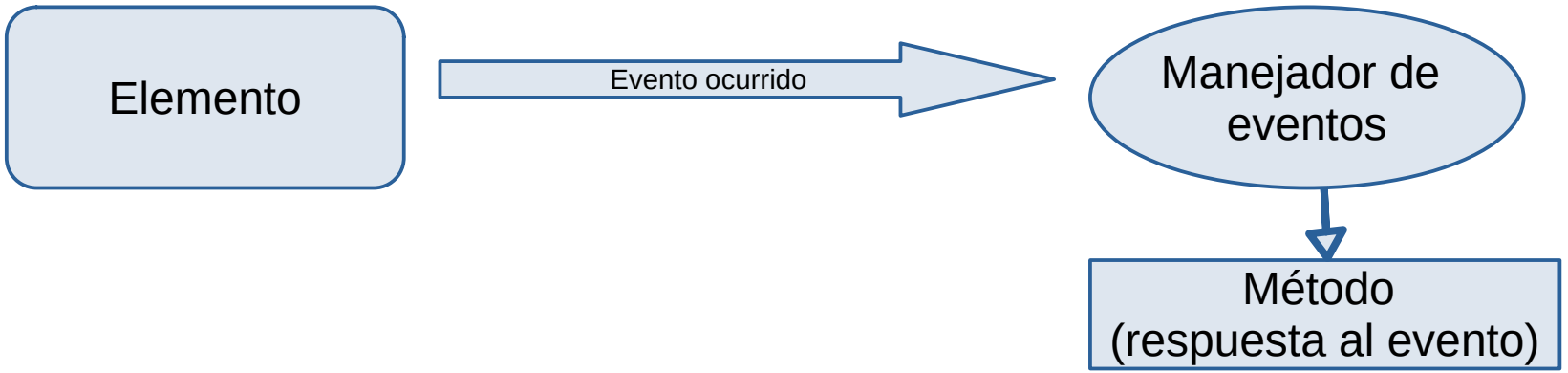
UD3 EVENTOS, VENTANAS Y APLICACIÓN

Ciclo: Desarrollo de Aplicaciones Multiplataforma
Módulo: Desarrollo de interfaces.

MANEJO DE EVENTOS

- Cuando una acción sobre un elemento genera un evento, se espera que suceda algo, entendiendo por evento un mensaje que un objeto envía a algún otro objeto.
- Un manejador de eventos es un objeto en el que un elemento delega la tarea de manipular un tipo particular de eventos.



- Cuando una acción sobre un elemento genera un evento, dicho evento, generalmente, partiendo del elemento donde se originó, se propaga en sentido ascendente por el árbol de elementos hasta que llega a la raíz (normalmente una ventana o una página) en busca de un manejador que responda a dicho evento.
- Estos eventos se denominan eventos enrutados

Asignar manejadores de eventos a un objeto

- Un objeto, generalmente un elemento, puede tener asociados uno o más manejadores para responder a un determinado evento.
- Para agregar un manejador de eventos en XAML, basta con añadir al elemento el nombre del evento como un atributo y asignarle como valor el nombre del método que manejará el evento:

```
<Button Grid.Row="1" Content="Haga _clic aquí" Name="btSaludo"
Height="23" Width="120" ToolTip="Botón de pulsación" Click="btSaludo_Click" />
```

- El método btSaludo_Click contendrá el código que responde al evento Click y debe tener la misma firma o prototipo que el delegado RoutedEventHandler, que es el delegado de los manejadores de eventos enrutados:

```
public delegate void RoutedEventHandlere(Object sender, RoutedEventArgs e);
```

- Un delegado es un objeto de una clase que puede contener una referencia a un método
- Por ejemplo, para que la etiqueta btSaludo muestre el mensaje "ii¡Hola mundo!!!” cuando el usuario de nuestra aplicación haga clic en el botón btSaludo, según lo expuesto, añadiríamos a la clase MainWindow el siguiente método:

```
private void btSaludo_Click(object sender, RoutedEventArgs e)
      {           btSaludo.Content = "Hola mundo!!!";       }
```

EVENTOS ADJUNTOS

- Para permitir que los elementos controlen eventos que se declaran en un elemento diferente, WPF admite lo que denominamos eventos adjuntos.

```
<Grid Button.Click="Boton_Click" Margin="5">
      <Button Name="Botonl" >Acción 1</Button>
      <Button Name="Boton2" >Acción 2</Button>
      <Button Name="Boton3" >Acción 3</Button>
  </Grid>
```

CICLO DE VIDA DE UNA VENTANA

- Como con cualquier objeto de cualquier clase, el ciclo de vida de una ventana comienza cuando se crea por primera vez un objeto de su clase, después se abre se activa, se desactiva y, finalmente, se cierra.
- Cuando se crea una ventana se agrega automáticamente su referencia a la colección de ventanas referenciada por la propiedad Windows del objeto Application y si además se trata de la primera ventana, se establece, de forma predeterminada, como la ventana principal de la aplicación, asignando su referencia a la propiedad MainWindow del objeto Application.

- Cuando se inicia la aplicación, la ventana especificada por el valor de StartupUri se abre de forma no modal
- Una ventana no modal permite a los usuarios activar otras ventanas en la misma aplicación.

- Cuando se abre una ventana, ésta se convierte en la ventana activa y se genera el evento Activated
- A continuación se generan los eventos Loaded y ContentRendered, en este orden.
- Cuando se produce este último evento puede considerarse la ventana abierta.
- WindowStartupLocation: se puede especificar la posición inicial de la ventana la primera vez que se muestra estableciendo esta propiedad .

- Para verificar si una ventana está activa se puede inspeccionar la propiedad IsActive.

- Cuando se desactiva una ventana puede que la aplicación continúe ejecutando código en segundo plano.
- En este caso, cuando se complete la tarea en segundo plano es posible avisar al usuario invocando al método Activate, lo cual hará parpadear el botón de la barra de tareas de la ventana en el supuesto de que el usuario esté interactuando con otra aplicación o traerá la ventana al primer plano si el usuario está interactuando con la aplicación actual.
- Una ventana minimizada se contrae en un botón en la barra de tareas si su propiedad ShowInTaskbar vale true.

- Cuando se cierra una ventana se generan los eventos Closing, antes de que la ventana se cierre y con la posibilidad de detener esta acción, Deactivated y Closed

PROPIEDADES BÁSICAS DE LA VENTANA

- Una ventana WPF se compone de dos áreas distintas: área no cliente y área cliente.
- El área no cliente comprende los elementos gráficos comunes a todas las ventanas: menú del sistema, icono, título, botón para minimizar, botón para maximizar, botón para cerrar y un borde.
- El área cliente es la parte de la ventana destinada a ubicar los controles que formarán la interfaz gráfica.

- La clase Window proporciona servicios para administrar la duración de la ventana, para administrarla y para establecer su apariencia y comportamiento.

Administración de la duración

- Método Activate. Permite activar la ventana situándola en primer plano.
- Método Close. Permite cerrar la ventana.
- Métodos Show y Hide. Show muestra una ventana sin impedir que los usuarios interactúen con otras ventanas de la aplicación y Hide la oculta.
- Propiedad IsActive. Esta propiedad vale true si la ventana está activa y false en caso contrario.

Administración de ventanas

- Método static GetWindow. Devuelve la referencia al objeto Window que define el objeto de dependencia (objeto de la clase DependencyObject) pasado como argumento.
- Propiedad Owned Windows. Referencia la colección WindowCollection de ventanas de las que esta ventana es la propietaria.
- Propiedad Owner. Obtiene o establece la referencia a la ventana propietaria de este objeto Window.

Apariencia y comportamiento

- Propiedad AllowsTransparency. Esta propiedad vale true si la ventana admite la transparencia y false en caso contrario.
- Método DragMove. Permite arrastrar una ventana haciendo clic con el ratón en cualquier parte de la ventana, no necesariamente en la barra de título.
- Propiedad Icon. Permite establecer el icono de una ventana.

Apariencia y comportamiento

- Propiedades Top y Left. Estas propiedades permiten obtener o establecer la posición de los bordes superior e izquierdo de la ventana, respectivamente, con respecto al escritorio. Por lo tanto, podrían ser utilizadas para establecer la posición inicial de la ventana.
- Evento Location Changed. Se produce cuando una ventana cambia de posición.

Apariencia y comportamiento

- Propiedad ResizeMode. El tamaño de todas las ventanas se puede cambiar. Sin embargo, este comportamiento puede ser modificado por medio de la propiedad Resize Mode que puede tomar estos valores:
  - NoResize, no se puede cambiar el tamaño de la ventana.
  - Can Minimize, la ventana sólo se puede minimizar y restaurar desde la barra de tareas.
  - CanResize, se tiene total capacidad para cambiar el tamaño de la ventana.
  - CanResizeWithGrip que ofrece la misma funcionalidad que CanResize, pero agrega un "control de ajuste de tamaño" en la esquina inferior derecha de la ventana.

Apariencia y comportamiento

- Propiedad RestoreBounds. Esta propiedad se puede usar para guardar el tamaño y la ubicación (objeto Rect) de una ventana antes de que se cierre la aplicación y recuperar esos valores la próxima vez que se inicie la aplicación para que la ventana se muestre con esos valores.
- Propiedad ShowActivated. Cuando se abre una ventana que tiene esta propiedad establecida en false, la ventana no se activa hasta que se active manualmente al seleccionarla.

Apariencia y comportamiento

- Propiedad ShowinTaskbar. Si el valor de esta propiedad es true, valor predeterminado, y se minimiza la ventana, ésta se muestra en la barra de tareas.
- Propiedad Size ToContent. El valor de esta propiedad indica si una ventana ajustará automáticamente su tamaño al de su contenido. Su valor predeterminado es Manual, esto es, el tamaño se ajustará manualmente; este valor se establece automáticamente siempre que el usuario cambie el tamaño de la ventana.

Apariencia y comportamiento

- Propiedad Topmost. Orden Z de las ventanas. Cuando esta propiedad vale true, la ventana correspondiente aparece encima de todas las ventanas cuya propiedad Topmost valga false y dentro de las ventanas que tienen esta propiedad a true, la ventana actualmente activada es la ventana de nivel superior.
- Propiedad WindowStartupLocation. Permite obtener o establecer la posición de una ventana cuando se muestra por primera vez. Si su valor es Manual, valor predeterminado, la posición de la ventana queda definida por sus propiedades Top y Left

Apariencia y comportamiento

- Propiedad WindowState. El valor de esta propiedad indica si una ventana está restaurada (Normal), minimizada (Minimized) o maximizada (Maximized).
- Evento StateChanged. Se produce cuando cambia la propiedad WindowState de la ventana.
- Propiedad WindowStyle. Indica el estilo del borde de una ventana. Su valor puede ser: None, sólo está visible el área cliente pero tiene un pequeño borde para poder cambiar de tamaño, SingleBorderWindow, borde único (es el valor predeterminado), ThreeDBorderWindow, borde 3D, y ToolWindow, ventana de herramientas fija.