

# Programación basado en eventos

Pedro O. Pérez M., PhD.

Desarrollo de videojuegos con Java  
Tecnológico de Monterrey

*pperezm@tec.mx*

10-2023

## Programación basada en eventos

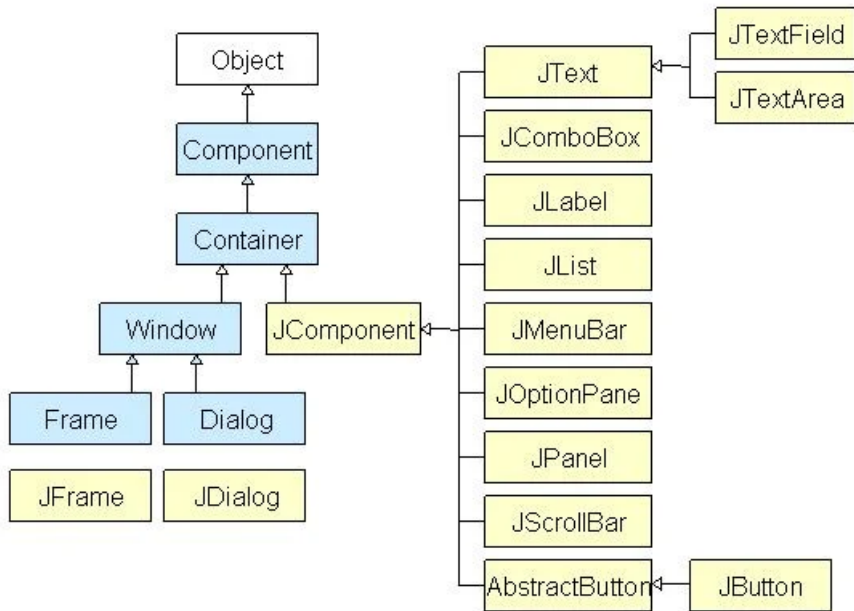
- Interfaz gráfica

- Eventos

  - Tipos de detectores (Listener)

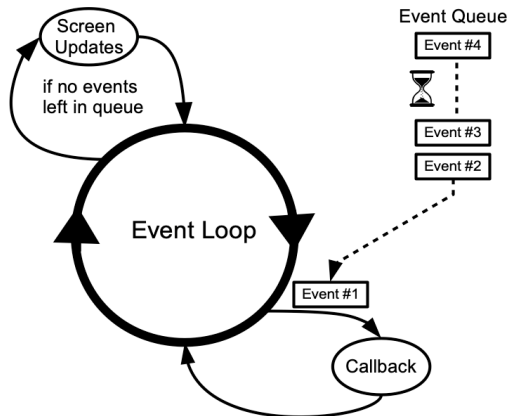
## Examen Rápido 5

- ▶ Las interfaces gráficas de usuario (GUI, Graphicsl User Interface) ofrecen al usuario ventanas, cuadros de diálogo, barras de herramientas, botones, listas desplegables y muchos otros elementos con los que ya estamos muy acostumbrados a tratar.
- ▶ La estructura básica de la biblioteca gira en torno a **componentes** y **contenedores**. Los contenedores contienen componentes y son componentes a su vez, de forma que los eventos pueden tratarse tanto en contenedores como en componentes.



- ▶ La programación basada en eventos de Java es un paradigma poderoso que permite a los desarrolladores crear aplicaciones reactivas y responsivas. Al aprovechar los eventos, los desarrolladores pueden escribir código que responda automáticamente a los cambios en el estado de una aplicación, garantizando que la interfaz de usuario permanezca actualizada y responda.
- ▶ En Java, un evento es un objeto que representa un cambio en el estado de una aplicación. Es una forma de que una parte de un programa se comuniquen con otra parte, a menudo en respuesta a la entrada del usuario u otros estímulos externos. Los eventos se utilizan normalmente en interfaces gráficas de usuario (GUI) para manejar las interacciones del usuario, como clics de botones, pulsaciones de teclas y movimientos del ratón.

- ▶ Cuando ocurre un evento, generalmente se pasa a un detector de eventos, que es un objeto que se ha registrado para recibir y manejar un tipo específico de evento. El oyente procesa el evento y puede realizar cualquier acción necesaria, como actualizar la interfaz de usuario o realizar cálculos.



- ▶ En la programación basada en eventos de Java, hay dos componentes principales: el detector de eventos y el origen del evento. El origen del evento es el objeto que genera el evento, mientras que el detector de eventos es el objeto que recibe y procesa el evento.
- ▶ Para crear un detector de eventos, debe implementar la interfaz adecuada para el evento que desea manejar. Por ejemplo, para manejar eventos de acción (como clics en botones), implementaría la interfaz `ActionListener`.

## Tipos de detectores (Listener)

- ▶ Los detectores de eventos están representados por diferentes interfaces, cada una de las cuales está diseñada para procesar un evento equivalente.
- ▶ Ten en cuenta que los detectores de eventos son flexibles en el sentido de que se puede registrar un único detector para `.escuchar` varios tipos de eventos. Esto significa que, para un conjunto similar de componentes que realizan el mismo tipo de acción, un detector de eventos puede manejar todos los eventos.



- ▶ Éstos son algunos de los tipos más comunes:
  - ▶ **ActionListener:** Escucha un `ActionEvent`, es decir, cuando se hace clic en un elemento gráfico, como un botón o un elemento de una lista.
  - ▶ **ContainerListener:** Escucha un `ContainerEvent`, que puede ocurrir si el usuario agrega o elimina un objeto de la interfaz.
  - ▶ **KeyListener:** Escucha un `KeyEvent` en el que el usuario presiona, escribe o suelta una tecla.
  - ▶ **WindowListener:** Escucha un `WindowEvent`, por ejemplo, cuando una ventana se cierra, se activa o se desactiva.
  - ▶ **MouseListener:** Escucha un `MouseEvent`, como cuando se hace clic o se presiona el mouse.

# Examen Rápido 5

## (Programación basada en eventos)