

## APUNTES T3

### 1. Función lambda:

Con la llegada de Java 8, la manera de manejar los eventos evolucionó para poder hacerlo mucho más simple.

Anteriormente, los eventos se controlaban de la siguiente manera:

```
Button boton = new Button("Haz clic");
```

```
boton.setOnAction(new EventHandler<ActionEvent>() {  
    @Override  
    public void handle(ActionEvent e) {  
        System.out.println("Botón pulsado");  
    }  
});
```

Es un método de varias líneas y poco intuitivo, pero con la llegada de Java 8 los eventos ahora se manejan de la siguiente manera:

```
Button boton = new Button("Haz clic");  
boton.setOnAction(e -> System.out.println("Botón pulsado"));
```

Una manera más simple y rápida de gestionar los eventos para JavaFX.

### 2. Tipos de eventos

#### A) setOnAction:

Este evento sirve para manejar acciones de controles interactivos, ya sean: botones, menu item, check box (al marcarlo), text filed(al darle al enter).

Cuando se quiere interactuar con alguno de esos controles, se hace uso del setOnAction.

#### B) setOnMouse

- setOnMouseEntered

Este evento gestiona la entrada del ratón sobre el control deseado, este se puede usar en todos los controles para que reaccionen cuando sobrepongas el ratón sobre ellos.

- setOnMouseExited

Este evento gestiona la salida del ratón sobre el control deseado, este se puede usar en todos los controles para que reaccionen cuando el ratón no está sobrepuerto en aquellos controles que deseas.

- setOnMouseClicked

Este evento gestiona el clickeo del ratón sobre el control deseado, este se puede usar en todos los controles para que reaccionen cuando cliquees sobre ellos.

### C) setOnKey

Este evento gestiona los eventos de entrada por teclado, y hay varios:

- setOnKeyPressed: Cuando una tecla se presiona
- setOnKeyReleased: Cuando una tecla se suelta
- setOnKeyTyped: Cuando se genera un carácter

Controles como button o la misma scene pueden hacer uso de este evento