

Interfaces gráficas I

- Introducción -

75.07 - 95.02 Algoritmos y programación III

Pablo Rodríguez Massuh



FACULTAD
DE INGENIERIA
Universidad de Buenos Aires

Temario

- ★ ¿A qué llamamos vista?
- ★ Distintos tipos de vistas
- ★ ¿Qué es JavaFx?
- ★ Conceptos de JavaFX
- ★ Controles básicos en JavaFX
- ★ Eventos y Clases anónimas
- ★ Eventos y Expresión Lambda
- ★ Eventos y Clases custom
- ★ Ejemplo Integrador



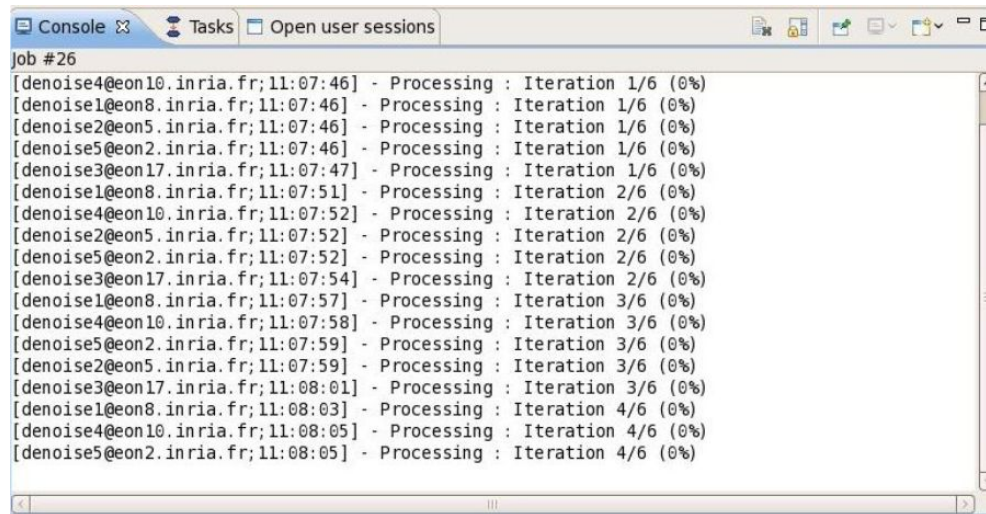
¿Qué es una vista?

Se conoce por vista a la colección de clases que representan los elementos de la interfaz de usuario. Es decir todas las “*cosas*” que el usuario puede ver y/o responder en la pantalla, como pueden ser botones, cajas de texto, etiquetas, etc.

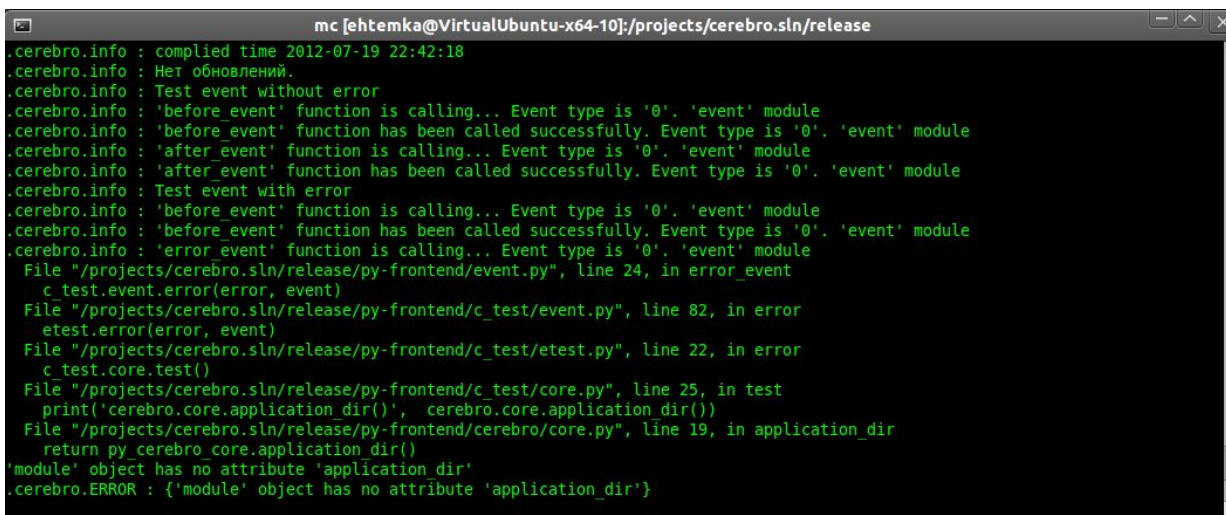


Tipo de vista I

Por consola



```
Console Tasks Open user sessions
Job #26
[denoise4@eon10.inria.fr;11:07:46] - Processing : Iteration 1/6 (0%)
[denoise1@eon8.inria.fr;11:07:46] - Processing : Iteration 1/6 (0%)
[denoise2@eon5.inria.fr;11:07:46] - Processing : Iteration 1/6 (0%)
[denoise5@eon2.inria.fr;11:07:46] - Processing : Iteration 1/6 (0%)
[denoise3@eon17.inria.fr;11:07:47] - Processing : Iteration 1/6 (0%)
[denoise1@eon8.inria.fr;11:07:51] - Processing : Iteration 2/6 (0%)
[denoise4@eon10.inria.fr;11:07:52] - Processing : Iteration 2/6 (0%)
[denoise2@eon5.inria.fr;11:07:52] - Processing : Iteration 2/6 (0%)
[denoise5@eon2.inria.fr;11:07:52] - Processing : Iteration 2/6 (0%)
[denoise3@eon17.inria.fr;11:07:54] - Processing : Iteration 2/6 (0%)
[denoise1@eon8.inria.fr;11:07:57] - Processing : Iteration 3/6 (0%)
[denoise4@eon10.inria.fr;11:07:58] - Processing : Iteration 3/6 (0%)
[denoise5@eon2.inria.fr;11:07:59] - Processing : Iteration 3/6 (0%)
[denoise2@eon5.inria.fr;11:07:59] - Processing : Iteration 3/6 (0%)
[denoise3@eon17.inria.fr;11:08:01] - Processing : Iteration 3/6 (0%)
[denoise1@eon8.inria.fr;11:08:03] - Processing : Iteration 4/6 (0%)
[denoise4@eon10.inria.fr;11:08:05] - Processing : Iteration 4/6 (0%)
[denoise5@eon2.inria.fr;11:08:05] - Processing : Iteration 4/6 (0%)
```



```
mc [ehtemka@VirtualUbuntu-x64-10]:/projects/cerebro.sln/release
.cerebro.info : compiled time 2012-07-19 22:42:18
.cerebro.info : Нет обновлений.
.cerebro.info : Test event without error
.cerebro.info : 'before_event' function is calling... Event type is '0'. 'event' module
.cerebro.info : 'before_event' function has been called successfully. Event type is '0'. 'event' module
.cerebro.info : 'after_event' function is calling... Event type is '0'. 'event' module
.cerebro.info : 'after_event' function has been called successfully. Event type is '0'. 'event' module
.cerebro.info : Test event with error
.cerebro.info : 'before_event' function is calling... Event type is '0'. 'event' module
.cerebro.info : 'before_event' function has been called successfully. Event type is '0'. 'event' module
.cerebro.info : 'error_event' function is calling... Event type is '0'. 'event' module
File "/projects/cerebro.sln/release/py-frontend/event.py", line 24, in error_event
  c test.event.error(error, event)
File "/projects/cerebro.sln/release/py-frontend/c_test/event.py", line 82, in error
  etest.error(error, event)
File "/projects/cerebro.sln/release/py-frontend/c_test/etest.py", line 22, in error
  c test.core.test()
File "/projects/cerebro.sln/release/py-frontend/c_test/core.py", line 25, in test
  print('cerebro.core.application dir()', cerebro.core.application_dir())
File "/projects/cerebro.sln/release/py-frontend/cerebro/core.py", line 19, in application_dir
  return py_cerebro_core.application_dir()
'module' object has no attribute 'application_dir'
.cerebro.ERROR : {'module' object has no attribute 'application_dir'}
```

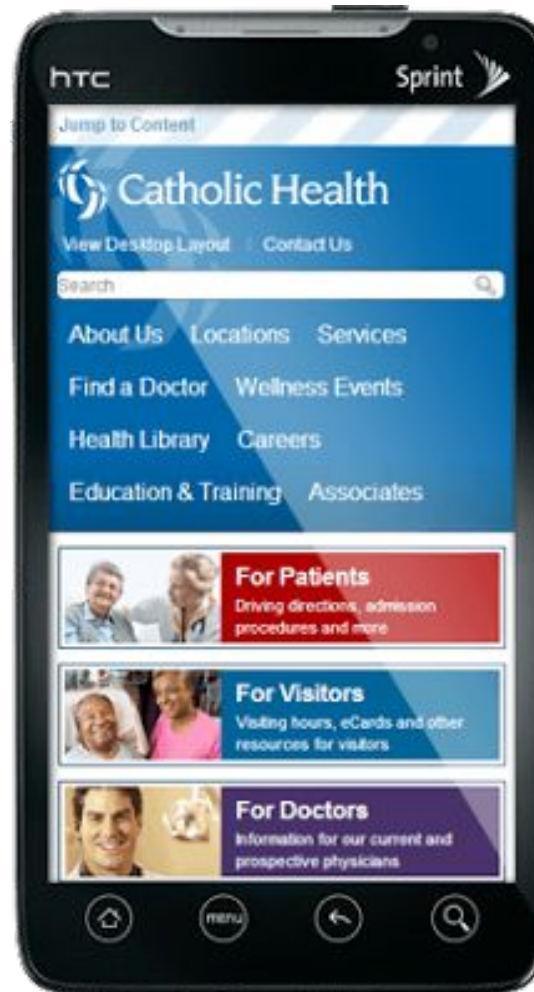
Tipo de vista II

Por ventanas



Tipo de vista III

Por dispositivo móvil



FACULTAD
DE INGENIERIA

Universidad de Buenos Aires

¿Qué es Java FX ?

JavaFX es un conjunto de gráficos y librerías multimedia que permite a los desarrolladores diseñar, crear, probar, depurar y desplegar aplicaciones que operan constantemente a través de diversas plataformas.

Es el reemplazo que Oracle ofrece a la ya deprecada librería gráfica swing.



JavaFx conceptos I

Application

```
import javafx.application.Application;

public class Main extends Application {

    public static void main(String[] args) {
        launch(args);
    }
}
```

JavaFX nos impone como condición heredar de la clase `Application`. En nuestro ***main*** necesitamos también llamar al método `launch` (ubicado en `Application`) en el cual se realizan las inicializaciones necesarias para una aplicación JavaFX .

JavaFx conceptos II

Stage

```
import javafx.application.Application;
import javafx.stage.Stage;

public class Main extends Application {

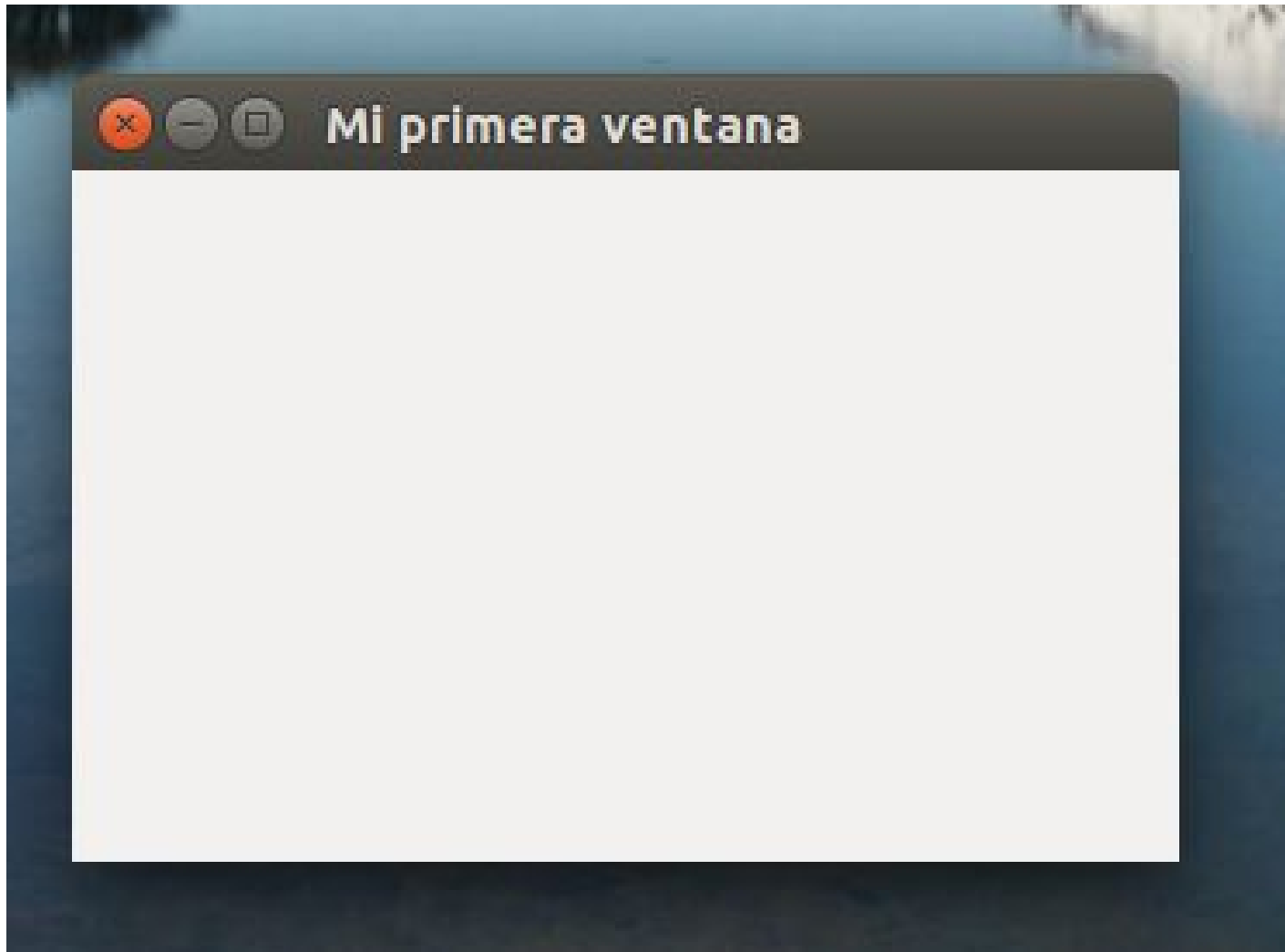
    public static void main(String[] args) {
        launch(args);
    }

    @Override
    public void start(Stage stage) throws Exception {
        stage.setTitle("Mi primera ventana");
        stage.show();
    }
}
```



JavaFx conceptos II

Stage



JavaFx conceptos III

Layout & Scene

@Override

```
public void start(Stage stage) throws Exception {
```

```
    stage.setTitle("Mi primera ventana");
```

```
    StackPane layout = new StackPane();
```

```
    Scene scene = new Scene(layout);
```

```
    stage.setScene(scene);
```

```
    stage.show();
```

```
}
```



JavaFx conceptos IV

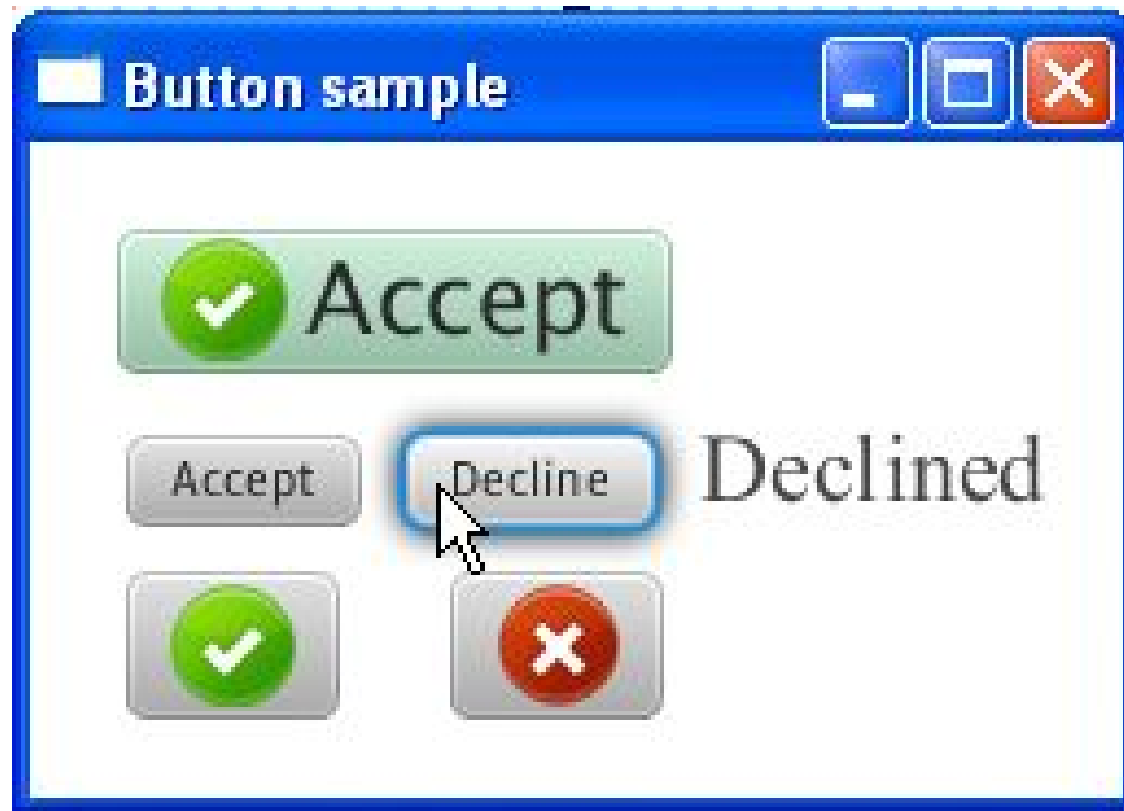
Diferencias entra Stage , Layout & Scene

- Stage: *Ventana con el marco para arrastrarla , su título y botones de minimizar, maximizar y cerrar.*
- Layout: *Indica la disposición (forma, manera) en que serán ubicados los elementos.*
- Scene: *Anteriormente conocido como “canvas” es el contenedor en el cual se colocan los elementos visuales.*



JavaFx controles I

Button



JavaFx controles I

Button

```
import javafx.scene.control.Button;

...

public void start(Stage stage) throws Exception {
    ...
    StackPane layout = new StackPane();

    Button button = new Button();

    button.setText("Texto del botón");

    layout.getChildren().add(button);

    ...
}
```



JavaFx controles II

Label



JavaFx controles II

Label

```
import javafx.scene.control.Label;

...

public void start(Stage stage) throws Exception {
    ...
    StackPane layout = new StackPane();

    Label label = new Label();

    label.setText("Texto de la etiqueta");

    layout.getChildren().add(label);

    ...
}
```


JavaFx controles III

TextField



JavaFx controles III

TextField

```
import javafx.scene.control.TextField;

...

public void start(Stage stage) throws Exception {
    ...
    StackPane layout = new StackPane();

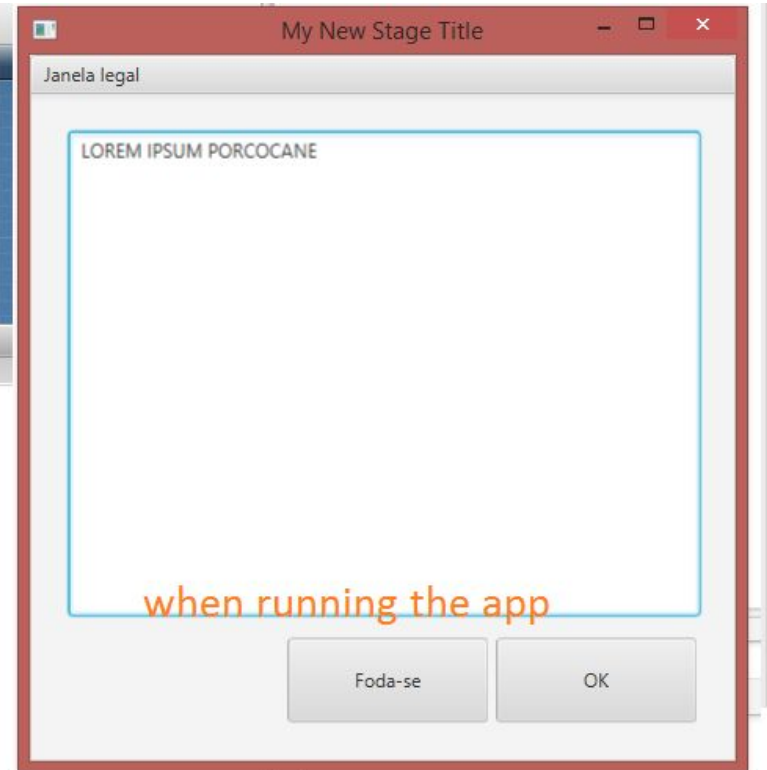
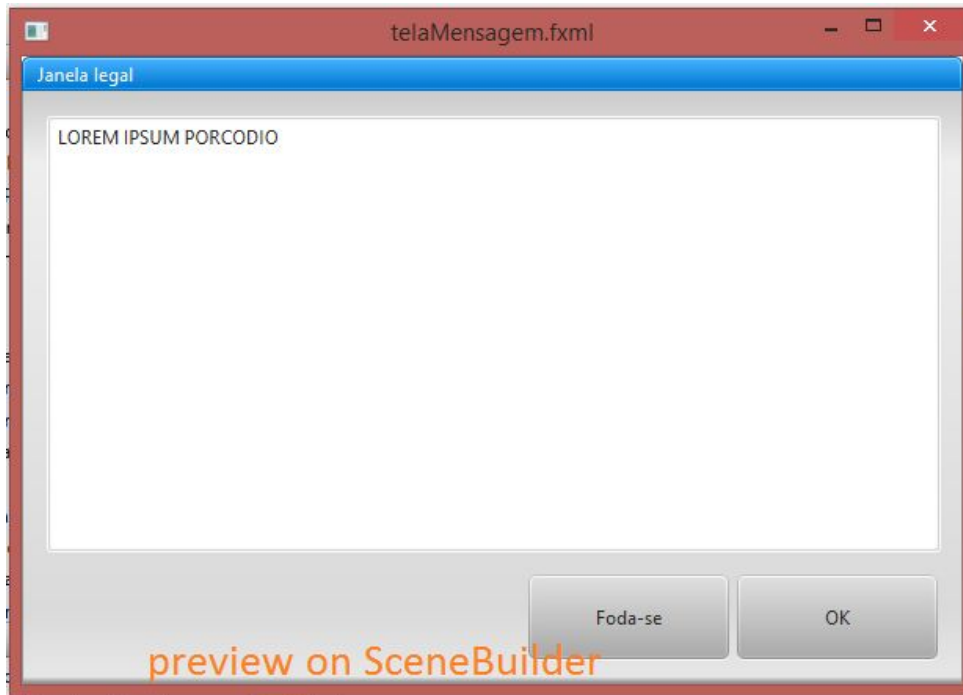
    TextField textField = new TextField();

    layout.getChildren().add(textField);

    ...
}
```

JavaFx controles IV

TextArea



JavaFx controles IV

TextArea

```
import javafx.scene.control.TextArea;

...

public void start(Stage stage) throws Exception {
    ...

    StackPane layout = new StackPane();

    TextArea textArea = new TextArea();
    textArea.setPrefRowCount(10);
    textArea.setPrefWidth(200);

    layout.getChildren().add(textArea);

    ...
}
```



Eventos en JavaFX I

Clases anónimas

```
Button button = new Button();  
button.setText("Clickeame");  
  
button.setOnAction(new EventHandler<ActionEvent>() {  
  
    public void handle(ActionEvent actionEvent) {  
  
        System.out.println("; Me han clickeado !");  
    }  
  
});
```



Eventos en JavaFX II

Expresión Lambda

```
Button button = new Button();  
button.setText("Clickeame");  
  
button.setOnAction(e-> {  
    System.out.println("Expresión");  
    System.out.println("Lambda");  
});
```

Las expresiones lambdas fueron incorporadas en Java 8



Eventos en JavaFX III

Usando clases propias para manejar eventos

```
public class MiBotonEventHandler implements EventHandler<ActionEvent> {  
  
    private Button miBoton;  
  
    public MiBotonEventHandler(Button miBoton) {  
        this.miBoton = miBoton;  
    }  
  
    @Override  
    public void handle(ActionEvent actionEvent) {  
  
        String textoClickeado = "; Me han clickeado !";  
  
        System.out.println(textoClickeado);  
  
        this.miBoton.setText(textoClickeado);  
    }  
}
```



Eventos en JavaFX III

Usando clases propias para manejar eventos

...

```
Button button = new Button();  
button.setText("Clickeame");
```

```
MiBotonEventHandler miBotonEventHandler = new MiBotonEventHandler(button);  
button.setOnAction(miBotonEventHandler);
```

...

Con esta forma más prolija evitamos duplicar código en caso de querer replicar el comportamiento con otro botón, al mismo tiempo que nos permite encapsular comportamiento mejorando la claridad.

Ejemplo Integrador

