



# Java Foundations

9-1

## Introducción a JavaFX

**ORACLE**  
Academy



Copyright © 2022, Oracle y/o sus filiales. Oracle, Java y MySQL son marcas comerciales registradas de Oracle y/o sus filiales. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios.

# Objetivos

- En esta lección se abordan los siguientes objetivos:
  - Crear un proyecto JavaFX
  - Explicar los componentes del proyecto JavaFX por defecto
  - Describir los diferentes tipos de nodos y paneles
  - Explicar el gráfico de escenas, el nodo raíz, las escenas y las etapas



# ¡Están a punto de llegar los exámenes finales!

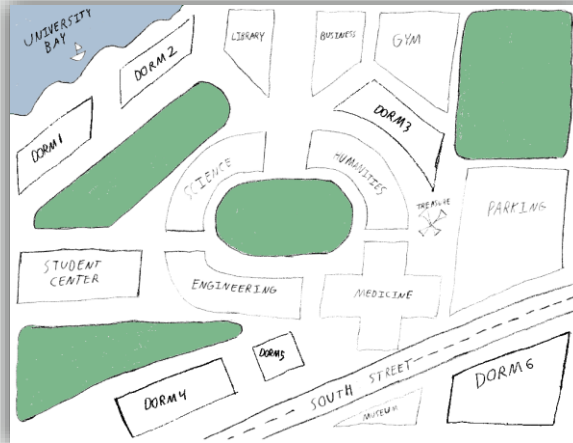
- Estudiar es importante
- ¿Quiere estudiar con un amigo?
  - ¿Pero sus amigos viven en otras residencias?
  - ¿Cuál es el mejor lugar para encontrarse?
  - ¿Cuál es el punto más céntrico del campus?

Gracias por recordármelo...



# JavaFX puede ayudarle

- JavaFX se utiliza para crear aplicaciones de GUI
- GUI: Interfaz gráfica de usuario
- Una aplicación de GUI nos permite ver la respuesta en un mapa



**ORACLE**  
Academy

JFo 9-1  
Introducción a JavaFX

Copyright © 2022, Oracle y/o sus filiales. Oracle, Java y MySQL son marcas comerciales registradas de Oracle y/o sus filiales. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios.

5

# Ejercicio 1



- En el material de referencia de esta diapositiva, `CampusMap.jar`, se muestra una aplicación completa escrita con JavaFX
- Reproduzca `CampusMap.jar`
- Cada cuadrado está alineado con la residencia correcta del mapa
- La población de cada residencia se ajusta haciendo clic y arrastrando el texto que aparece debajo de cada cuadrado
- Observe los cambios en los siguientes puntos centrales:
  - Todos los estudiantes en todas las residencias
  - Un grupo de estudio de tres amigos que viven en las residencias 1, 2 y 4

## Pero, ¿este no es mi campus!

- Tienes razón
- Sería mejor si la escuela utilizara el programa...
  - Mapa del campus
  - Nombre de las residencias
  - Población de las residencias
  - Y su grupo de amigos
- Este es el juego de problemas de la sección En la
- Sección 9 se explica todo lo que necesitará para volver a crear el programa

Java Puzzle Ball es también una aplicación JavaFX, pero se tardaría demasiado en volverla a crear.

## Ejercicio 2

- Cree un proyecto JavaFX
  - En el material de referencia de esta lección se han proporcionado instrucciones para crear un proyecto JavaFX en NetBeans o Eclipse
  - Si está utilizando otro IDE, consulte el documentación para obtener información sobre los pasos necesario para realizar este proceso
- Practique con el programa.
- ¿Puede hacer estos cambios?
  - Cambiar la etiqueta del botón
  - Cambiar lo que se imprime cuando se hace clic en el botón
  - Crear otro botón y mostrar ambos botones
  - Cambiar el tamaño por defecto de la ventana de la aplicación



# Proyecto JavaFX por defecto

```
public class JavaFXMain extends Application {

    @Override
    public void start(Stage primaryStage) {
        Button btn = new Button();
        btn.setText("Say 'Hello World'");
        btn.setOnAction(new EventHandler<ActionEvent>() {

            @Override
            public void handle(ActionEvent event) {
                System.out.println("Hello World!");
            }
        });

        StackPane root = new StackPane();
        root.getChildren().add(btn);
    }
}
```

Continúa en la siguiente diapositiva...

JavaFX debe haberle proporcionado un programa similar a este. Observaremos con atención los componentes de este código.

# Proyecto JavaFX por defecto

## Continuación

```
Scene scene = new Scene(root, 300, 250);

primaryStage.setTitle("Hello World!");
primaryStage.setScene(scene);
primaryStage.show();
} //end method start

public static void main(String[] args) {
    launch(args);
} //end method main
} //end class JavaFXMain
```

JavaFX debe haberle proporcionado un programa similar a este. Observaremos con atención los componentes de este código.

## Dos métodos: start() y main()

- start() es el punto de entrada para todas las aplicaciones JavaFX
  - Se puede considerar como el método main para JavaFX

```
public void start(Stage primaryStage) {  
    ...  
} //end method start
```

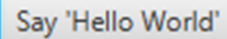
- main() también es necesario en sus programas
  - Inicia la aplicación JavaFX

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

# Los botones son objetos

- Los botones son como cualquier otro objeto

- Se pueden instanciar
- Contienen campos
- Contienen métodos



```
public void start(Stage primaryStage) {  
    Button btn = new Button();  
    btn.setText("Say 'Hello World'");  
    ...  
} //end method start
```

- A partir de este código podemos decir que...

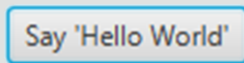
- Los botones contienen un campo de texto
- Los botones contienen un método para cambiar el campo de texto

## Los botones son nodos

- Algunos de estos campos y métodos están diseñados para almacenar y manipular propiedades visuales:
  - `btn.getText()`
  - `btn.setMinHeight()`
  - `btn.setLayoutX()`      **//set x position**
  - `btn.setLayoutY()`      **//set y position**
  - `btn.isPressed()`      **//is it pressed?**
- Los objetos como este se denominan nodos JavaFX

# Nodos

- Existen muchos tipos de nodos JavaFX:



Button



Rectangle



PieChart



ScrollBar



Text



ImageView

- La mayoría de objetos visuales que creará probablemente...
  - Serán un nodo o
  - Incluirán un nodo como campo

# Interacción de nodo

- Esto ayuda a manejar la interacción del botón:

```
public void start(Stage primaryStage) {  
    ...  
    btn.setOnAction(new EventHandler<ActionEvent>() {  
        @Override  
        public void handle(ActionEvent event) {  
            System.out.println("Hello World!");  
        } //end method handle  
    }); //end setOnAction  
    ...  
} //end method start
```

- Se denomina "**clase interna anónima**"
  - ¿La sintaxis no parece un lío?
  - Las expresiones Lambda Java SE 8 son una alternativa elegante
  - Trataremos las expresiones Lambda más adelante en esta sección

# Creación de nodos

- Los nodos se instancian como cualquier otro objeto Java:

```
public void start(Stage primaryStage) {  
    Button btn1 = new Button();  
    Button btn2 = new Button();  
    btn1.setText("Say 'Hello World'");  
    btn2.setText("222");  
    ...  
} //end method start
```

- Después de instanciar un nodo:
  - Hay una memoria asignada para almacenar el objeto
  - Los campos se pueden manipular y se puede llamar a los métodos
  - Pero puede que no se muestre... ←

*Al menos por ahora...*



# Muestra de nodos

- Hay algunos pasos para mostrar un nodo

```
public void start(Stage primaryStage) {  
    Button btn1 = new Button();  
    Button btn2 = new Button();  
    btn.setText("Say 'Hello World'");  
    btn.setText("222");  
    StackPane root = new StackPane();  
    root.getChildren().add(btn1);  
    root.getChildren().add(btn2);  
    ...  
} //end method start
```

- En primer lugar, agregue cada nodo al nodo raíz
  - Se suele denominar root
  - Es muy similar a una clase ArrayList de todos los nodos

## Adición de nodos al nodo raíz

- Puede agregar cada nodo por separado:



```
root.getChildren().add(btn1);  
root.getChildren().add(btn2);  
root.getChildren().add(btn3);
```

- O bien, puede agregar muchos nodos a la vez:



```
root.getChildren().addAll(btn1, btn2, btn3);
```

## Adición de nodos al nodo raíz

- Pero no agregue el mismo nodo más de una vez
  - Causa un error del compilador:



```
root.getChildren().add(btn1);  
root.getChildren().add(btn1);
```

# Nodo raíz StackPane

- El nodo raíz de este ejemplo es un StackPane

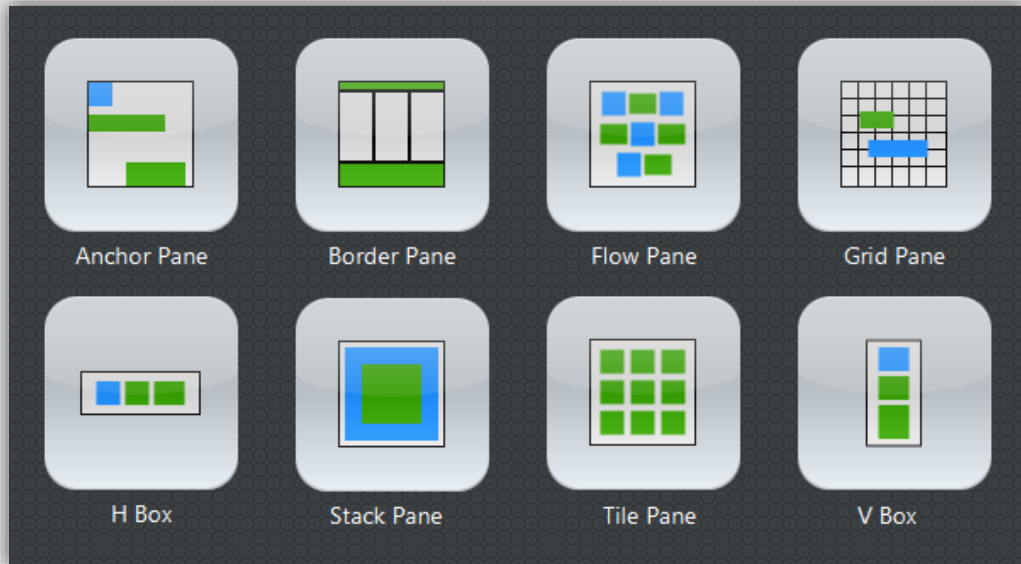
```
StackPane root = new StackPane();  
root.getChildren().addAll(btn1, btn2);
```

- El StackPane apila los nodos uno por encima de otro
- Pero los botones pequeños podrían volverse ocultos e inaccesibles



# Los paneles como nodos Raíz

- Cada panel determina el diseño de los nodos



## Programación de distintos paneles como nodos raíz

- Diseñar el nodo raíz como un panel diferente es fácil
- Solo tiene que especificar un tipo de referencia y un tipo de objeto distintos

Cambie este

Y este

```
StackPane root = new StackPane();  
root.getChildren().addAll(btn1, btn2);
```

```
TilePane root = new TilePane();  
root.getChildren().addAll(btn1, btn2);
```

```
VBox root = new VBox();  
root.getChildren().addAll(btn1, btn2);
```

## Ejercicio 3

- Edite el proyecto `JavaFX` actual
  - Vamos a experimentar un poco
- Después de agregar un botón al nodo raíz, pruebe a cambiar su posición
  - `btn1.setLayoutY(100);`
- ¿La posición de un botón cambiará si el nodo raíz no era un `StackPane`? Pruebe estas alternativas:
  - `TilePane`
  - `VBox`
  - `Group`

## Nodos raíz Group

- Un Group permite colocar los nodos en cualquier lugar

```
Group root = new Group();  
root.getChildren().addAll(btn1, btn2);  
btn1.setLayoutY(100);
```

- Un panel puede restringir dónde se colocan los nodos
  - No se pueden mover, aunque lo desee
  - No se puede hacer clic ni arrastrar un nodo que se ha bloqueado en un panel

```
StackPane root = new StackPane();  
root.getChildren().addAll(btn1, btn2);  
btn1.setLayoutY(100); //Has no effect
```



## Un grupo puede contener un panel

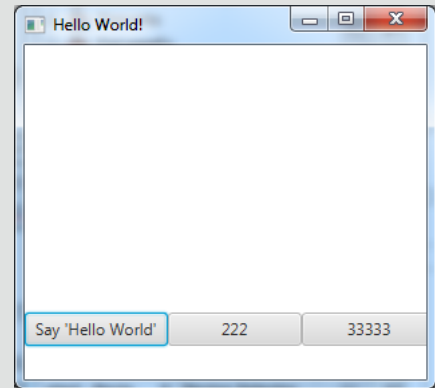
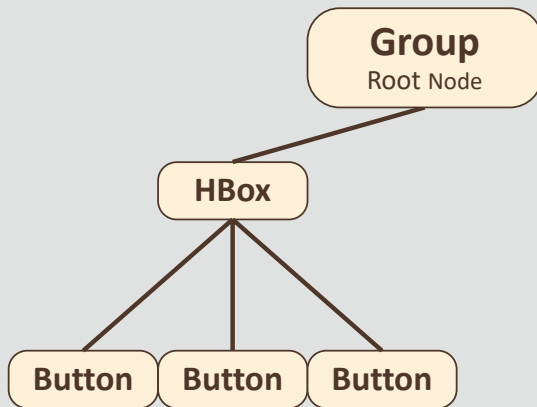
- Los paneles también son nodos
  - Cualquier nodo se puede agregar al nodo raíz
- Un panel puede ser una buena opción para almacenar los botones, los cuadros de diálogo de entrada de texto y otros elementos de GUI
  - No puede mover los nodos individuales en un panel
  - Pero puede mover todo el panel a un Group
  - Mueva el panel como lo haría con cualquier otro nodo

## Ejercicio 4

- Edite el proyecto `JavaFX` actual
  - Es el momento de experimentar más
- ¿Puede averiguar cómo hacer lo siguiente?
  - Cree un panel `HBox` y agregue varios botones para él
  - Agregue el panel `HBox` a un nodo raíz `Group`
  - Coloque el `HBox` cerca del botón de la ventana

# El gráfico de escenas JavaFX

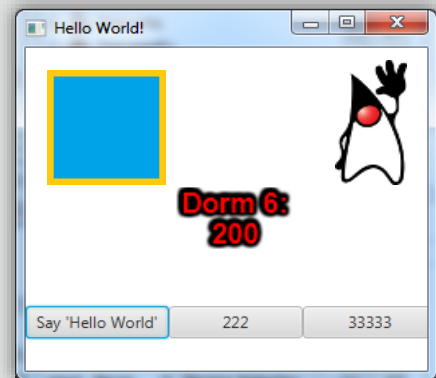
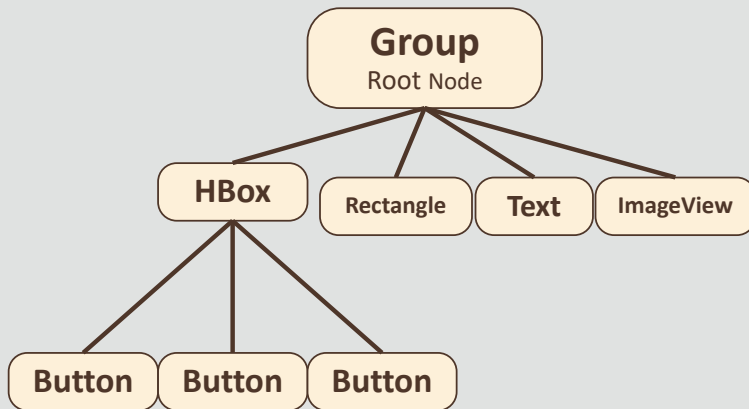
- Cómo decidir que los nodos agregados se pueden dibujar como un gráfico de escenas
  - El nodo raíz contiene un Hbox
  - El HBox actúa como un contenedor para los botones



Nota para los instructores: Los gráficos en estas dos diapositivas deben estar en la misma posición.

## El gráfico de escenas

- El HBox mantiene la GUI organizada y colocada en el lugar apropiado
- El resto de la ventana se puede utilizar para otros nodos



**ORACLE**  
Academy

JFo 9-1  
Introducción a JavaFX

Copyright © 2022, Oracle y/o sus filiales. Oracle, Java y MySQL son marcas comerciales registradas de Oracle y/o sus filiales. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios.

28

Nota para los instructores: Los gráficos en estas dos diapositivas deben estar en la misma posición.

## La escena y la etapa

- Si observamos el resto del programa JavaFX por defecto, vemos dos cosas más:
  - Una escena (que contiene el nodo raíz)
  - Una etapa (que contiene la escena)

```
public void start(Stage primaryStage) {  
    ...  
    Scene scene = new Scene(root, 300, 250);  
  
    primaryStage.setTitle("Hello World!");  
    primaryStage.setScene(scene);  
    primaryStage.show();  
} //end method start
```

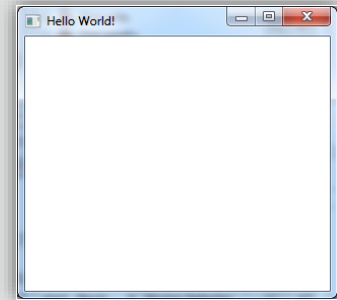
## ¿Qué es la escena?

- Hay algunas propiedades importantes que describen una Scene:
- Gráfico de escenas
  - La Scene es el contenedor para todo el contenido en el gráfico de escenas JavaFX
- Size
  - La anchura y la altura de la Scene se pueden definir
- antecedentes
  - El fondo se puede establecer como un Color o una imagen de fondo
- Información del cursor
  - La Scene puede detectar los eventos del mouse y manejar las propiedades del cursor.

```
Scene scene = new Scene(Nodo raíz root, ancho 300, altura 250, fondo Color.BLACK);
```

## ¿Qué es la etapa?

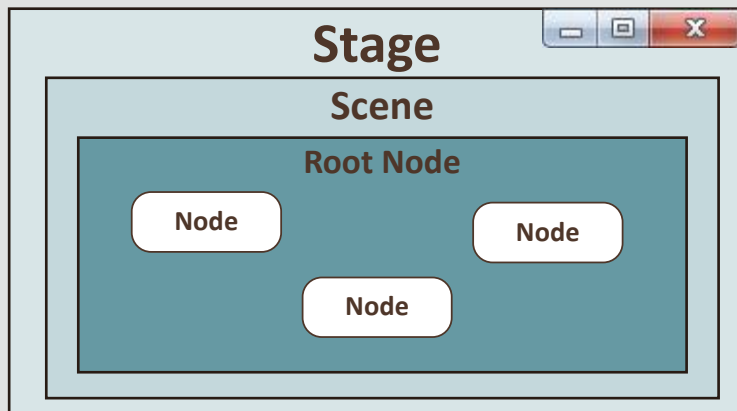
- Considere la etapa como la ventana de la aplicación
- Estas son dos de las propiedades importantes de la etapa:
  - Title
    - El título de la etapa se puede definir
  - Escena
    - La etapa contiene una Scene



```
primaryStage.setTitle("Hello World!");  
primaryStage.setScene(scene);  
primaryStage.show();
```

# Animación de jerarquía

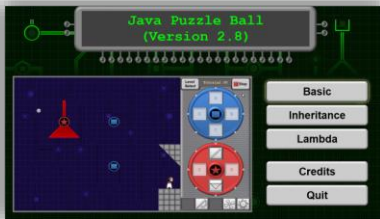
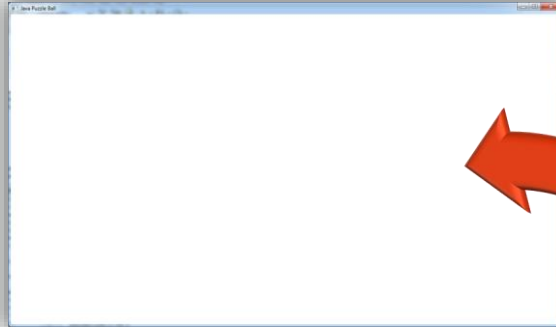
- Una etapa es un contenedor de nivel superior
- Una etapa contiene una escena
- Una escena contiene un nodo raíz
- El nodo raíz contiene otros nodos





## Varias escenas, una etapa

- Es posible intercambiar cualquier escena en una sola etapa



**ORACLE**  
Academy

JFo 9-1  
Introducción a JavaFX

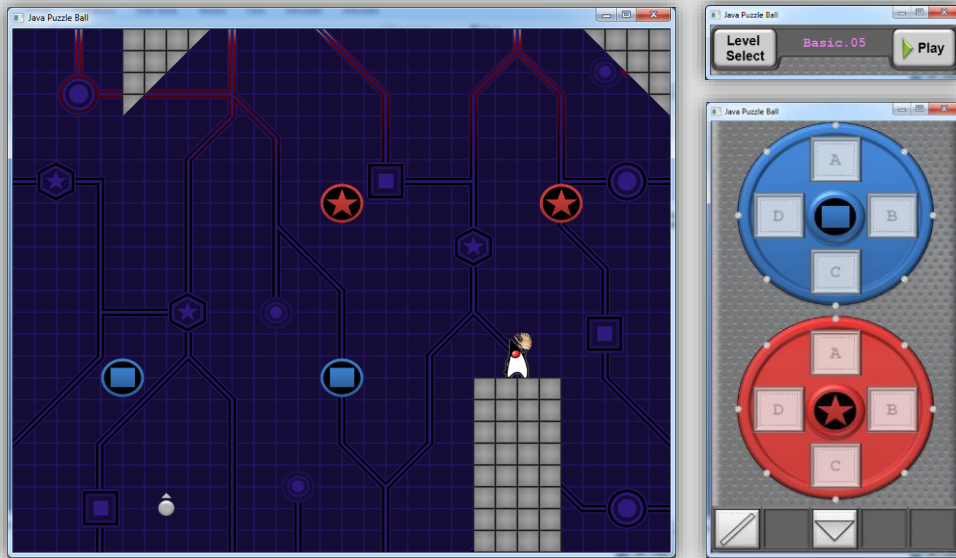
Copyright © 2022, Oracle y/o sus filiales. Oracle, Java y MySQL son marcas comerciales registradas de Oracle y/o sus filiales. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios.

33

En Java Puzzle Ball utilizamos una sola etapa. Si hubiésemos utilizado varias etapas, sería un lío ver como las ventanas se abren y se cierran al desplazarse por los menús.

## Varias escenas, varias etapas

- También se pueden crear varias etapas



**ORACLE**  
Academy

JFo 9-1  
Introducción a JavaFX

Copyright © 2022, Oracle y/o sus filiales. Oracle, Java y MySQL son marcas comerciales registradas de Oracle y/o sus filiales. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios.

34

Normalmente se realiza con las herramientas. Pero no tanto con los juegos.

# Resumen

- En esta lección se abordan los siguientes objetivos:
  - Crear un proyecto JavaFX
  - Explicar los componentes del proyecto JavaFX por defecto
  - Describir los diferentes tipos de nodos y paneles
  - Explicar el gráfico de escenas, el nodo raíz, las escenas y las etapas



