

UT6-7 JAVAFX Parte 2: Ventanas

José Jarones

Cargar una nueva ventana

Aquí creamos el **Scene** con el **root** antes creado por el **loader**. Después, creo el **Stage**, lo inicializo en modo modal (hasta que no se cierre esa ventana, no me deja ir a cualquier otra). Por último, lo muestro y espero a que cierre, en el código se queda esperando de verdad.

```
// Cargo la vista
```

```
FXMLLoader loader = new FXMLLoader(getClass().getResource("/vista/VentanaHijaVista.fxml"));
```

```
// Cargo la ventana
```

```
Parent root = loader.load();
```

```
// Creo el Scene
```

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

```
Stage stage = new Stage();
```

```
stage.initModality(Modality.APPLICATION_MODAL);
```

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

```
stage.showAndWait();
```



Cerrar ventana

@FXML

```
private void cerrarVentana(ActionEvent event) {  
    Node source = (Node) event.getSource();  
    Stage stage = (Stage) source.getScene().getWindow();  
    stage.close();  
}
```



Alertas

Los tipos de alertas que hay son:

- ERROR
- INFORMATION
- WARNING
- CONFIRMATION



Ejemplo

@FXML

```
private void mostrarAlertError(ActionEvent event) {  
    Alert alert = new Alert(Alert.AlertType.ERROR);  
    alert.setHeaderText(null);  
    alert.setTitle("Error");  
    alert.setContentText("Error en la aplicacion");  
    alert.showAndWait();  
}
```

@FXML

```
private void mostrarAlertInfo(ActionEvent event) {  
    Alert alert = new Alert(Alert.AlertType.INFORMATION);  
    alert.setHeaderText(null);  
    alert.setTitle("Info");  
    alert.setContentText("Informacion sobre la aplicación");  
    alert.showAndWait();  
}
```

Ejemplo

@FXML

```
private void mostrarAlertWarning(ActionEvent event) {  
    Alert alert = new Alert(Alert.AlertType.WARNING);  
    alert.setHeaderText(null);  
    alert.setTitle("Info");  
    alert.setContentText("Warning en la aplica");  
    alert.showAndWait();  
}
```

@FXML

```
private void mostrarAlertConfirmation(ActionEvent event) {  
    Alert alert = new Alert(Alert.AlertType.CONFIRMATION);  
    alert.setHeaderText(null);  
    alert.setTitle("Confirmacion");  
    alert.setContentText("¿Deseas realmente confirmar?");  
    alert.showAndWait();  
}
```

Ejemplo

@FXML

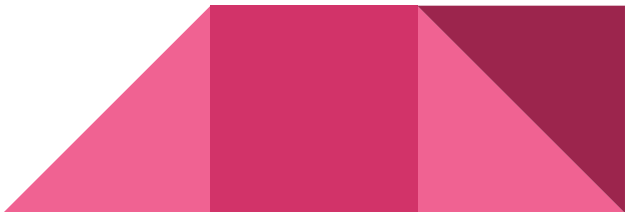
```
private void mostrarAlertCabecera(ActionEvent event) {  
    Alert alert = new Alert(Alert.AlertType.INFORMATION);  
    alert.setHeaderText("Cabecera");  
    alert.setTitle("Info");  
    alert.setContentText("Informacion sobre la aplicación");  
    alert.showAndWait();  
}
```

Ejemplo Alert de confirmación

```
Alert alert = new Alert(Alert.AlertType.CONFIRMATION);  
alert.setHeaderText(null);  
alert.setTitle("Confirmación");  
alert.setContentText("¿Estas seguro de confirmar la acción?");  
Optional<ButtonType> action = alert.showAndWait();
```




```
// Si hemos pulsado en aceptar
if (action.get() == ButtonType.OK) {
    this.txtPedirConfir.setText("Has pulsado en aceptar");
} else {
    this.txtPedirConfir.setText("Has pulsado en cancelar");
}
```

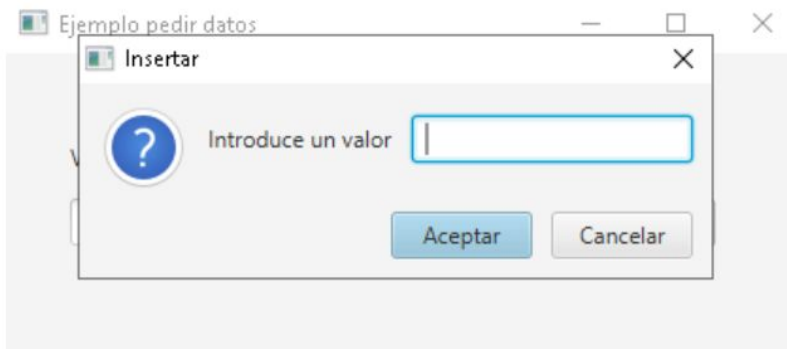


Pedir un dato en JavaFX

Es como **Alert** pero usando esa clase. Veamos un ejemplo:

```
1 | TextInputDialog tid = new TextInputDialog();  
2 | tid.setHeaderText(null);  
3 | tid.setTitle("Insertar");  
4 | tid.setContentText("Introduce un valor");  
5 | Optional<String> texto = tid.showAndWait();
```

Con esto hacemos que salga la ventana para introducir un valor.



Hojas de estilo

En JavaFX puedes dar estilo al interfaz de usuario utilizando hojas de estilo en cascada (CSS).

Estilo por defecto en JavaFX

Los estilos por defecto de JavaFX 8 se encuentran en un archivo denominado `modena.css`. Este archivo CSS se encuentra dentro del archivo jar `jfxrt.jar` que se encuentra en tu directorio de instalación de Java, en la ruta `/jdk1.8.x/jre/lib/ext/jfxrt.jar`.

Puedes descomprimir `jfxrt.jar` o abrirlo como si fuera un zip. Encontrarás el archivo `modena.css` en la ruta `com/sun/javafx/scene/control/skin/modena/`

Este estilo se aplica siempre a una aplicación JavaFX. Añadiendo un estilo personal podemos reescribir los estilos por defecto definidos en `modena.css`.



Vinculando hojas de estilo CSS

Añade un archivo CSS denominado `DarkTheme.css` al paquete *view*.

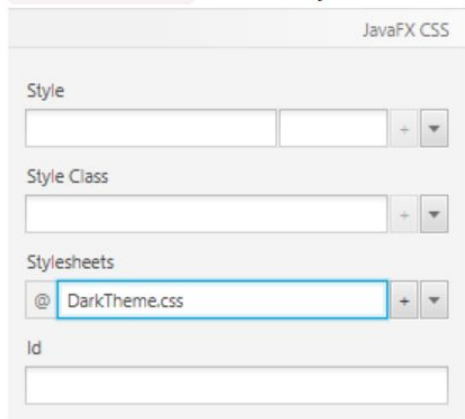
DarkTheme.css

```
.background {  
    -fx-background-color: #1d1d1d;  
}  
  
.label {  
    -fx-font-size: 11pt;  
    -fx-font-family: "Segoe UI Semibold";  
    -fx-text-fill: white;  
    -fx-opacity: 0.6;  
}  
  
.label-bright {  
    -fx-font-size: 11pt;  
    -fx-font-family: "Segoe UI Semibold";  
    -fx-text-fill: white;  
    -fx-opacity: 1;  
}
```

Vinculación con la scene

A continuación necesitamos vincular el CSS a nuestra escena. Podemos hacer esto programáticamente, mediante código Java, pero en esta ocasión vamos a utilizar Scene Builder para añadirlo a nuestros archivos FXML:

2. Selecciona el `BorderPane` raíz en la sección *Hierarchy*. En la vista *Properties* añade el archivo `DarkTheme.css` como hoja de estilo (campo denominado `Stylesheets`).



Establece el icono de la escena principal

Para establecer el icono de nuestra escena debemos añadir la línea de código siguiente al método

`start(...)` dentro de `MainApp.java`

MainApp.java

```
this.primaryStage.getIcons().add(new Image("file:resources/images/address_book_32.png"));
```

The whole `start(...)` method should look something like this now:

```
public void start(Stage primaryStage) {  
    this.primaryStage = primaryStage;  
    this.primaryStage.setTitle("AddressApp");  
  
    // Set the application icon.  
    this.primaryStage.getIcons().add(new Image("file:resources/images/address_book_32.png"));  
  
    initRootLayout();  
  
    showPersonOverview();  
}
```

Ejemplo código hoja de estilos css netbeans

```
public class Main extends Application {
    @Override
    public void start(Stage primaryStage) {
        try {
            AnchorPane root = (AnchorPane)FXMLLoader.load(getClass().getResource("Vista.fxml"));
            Scene scene = new Scene(root);
            scene.getStylesheets().add(getClass().getResource("application.css").toExternalForm());
            primaryStage.setScene(scene);
            primaryStage.show();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

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