



5. Estilos CSS

En la medida de lo posible, JavaFX CSS sigue los estándares de W3C; sin embargo, los nombres de propiedad de JavaFX han sido prefijados con una **extensión <-fx->**.

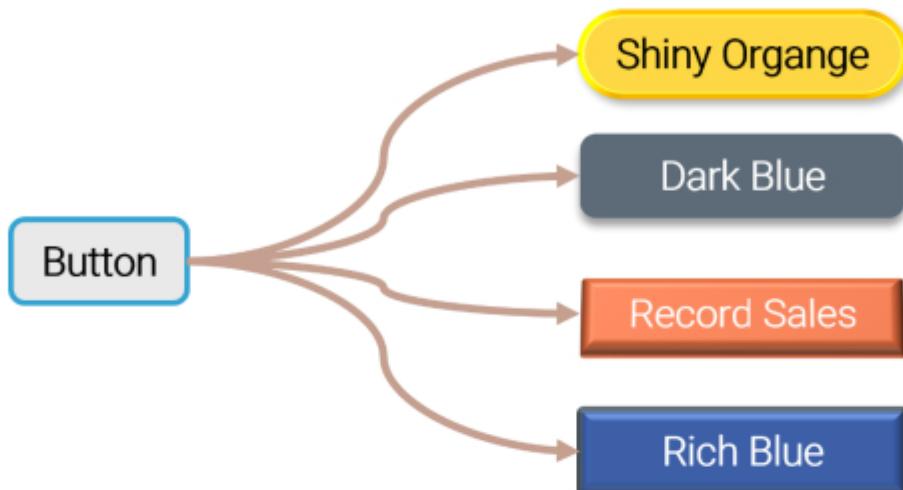


Fig. 13.16 Cómo transformar un botón con JavaFX CSS

5.1. Introducción a JavaFX CSS

Los estilos CSS se aplican a los nodos de la escena JavaFX de una manera similar a la forma en que se aplican los estilos CSS a los elementos del DOM HTML. En concreto, los estilos se aplican primero al padre y luego a sus hijos.

Igual que ocurre con los estilos CSS que se aplican en HTML, se existen tres **selectores** para modificar el estilo:

- El nombre de un componente gráfico; por ejemplo, un botón o un campo de texto.
- Una clase, cuyos componentes tendremos que marcar.
- Un identificador asociado a un único componente gráfico de nuestra escena.

Componentes

Únicamente debemos especificar las reglas CSS en la hoja de estilos. Esto tendrá efecto en todos los botones de nuestra aplicación donde esté enlazada la hoja de estilos.

```
.button {  
    -fx-background-color: #A0EAF5;  
    -fx-text-fill: #595C5D;  
}
```



Clases

Deberemos especificar, en aquellos componentes a los que queramos aplicar el estilo, a qué clase pertenecen mediante el atributo **styleClass**.

```
<Button layoutX="120.0" layoutY="180.0" mnemonicParsing="false"
prefHeight="40.0" prefWidth="160.0" styleClass="botonAceptar" text="ACEPTAR"/>
```

La clase en CSS la indicaremos precedida de un punto delante de su nombre. Por ejemplo:

```
.botonAceptar{
    -fx-background-color: blue;
    -fx-text-fill: #000000;
}
```

Identificadores

Deberemos especificar, en el componente al que queramos aplicar el estilo, qué identificador tiene mediante el atributo **id**.

```
<Button id="botonUnico" layoutX="120.0" layoutY="180.0" mnemonicParsing="false"
prefHeight="40.0" prefWidth="160.0" text="ACEPTAR"/>
```

El identificador en CSS lo indicaremos precedido del carácter «#».

```
#botonUnico {
    -fx-background-color: #A0EAF5;
    -fx-text-fill: #595C5D;
}
```

5.2. ¿Cómo incluir CSS en nuestro proyecto JavaFX?

Existen dos formas de asignar una hoja de estilos CSS a una escena de una aplicación JavaFX con IGU:

Opción A

La primera opción es añadir tantas hojas de estilos como deseemos, mediante código Java, en la creación de la escena. En nuestro caso, en la clase **Main.java**, en el método **start()**, añadimos las líneas de código que puedes ver a continuación:

```
@Override
public void start(Stage primaryStage) {
    ...
    // Asignar hoja de estilos
    scene.getStylesheets().add("/vista/css/estilos1.css");
    scene.getStylesheets().add("/vista/css/estilos2.css");
    ...
}
```



Opción B

La segunda opción es incluir la hoja de estilos desde la vista, es decir, desde el archivo .fxml que define nuestra IGU. Se añade el atributo **stylesheets** al contenedor principal de la escena. En el siguiente ejemplo se añade al *AnchorPane*.

```
<?xml version="1.0" encoding="UTF-8"?>
<?import javafx.scene.layout.AnchorPane?>

<AnchorPane prefWidth="600.0" prefHeight="400.0" stylesheets="@estilos.css"
xmlns="http://javafx.com/javafx/11.0.1" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="controlador.VentanaController">

    <!-- TODO Add Nodes -->

</AnchorPane>
```

Si queremos incluir más de una hoja de estilos, en lugar de añadir el atributo, añadiremos un elemento **stylesheet** con tantos nodos hijos como hojas de estilos queramos agregar:

```
<AnchorPane prefWidth="600.0" prefHeight="400.0"
xmlns="http://javafx.com/javafx/11.0.1" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="controlador.VentanaController">

    <stylesheets>
        <URL value="@estilos_1.css"/>
        <URL value="@estilos_2.css"/>
    </stylesheets>

</AnchorPane>
```



5.3. Ejemplo de JavaFX CSS

Considere la siguiente ventana simple de JavaFX:

miVentana.fxml

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.layout.AnchorPane?>

<AnchorPane prefHeight="400.0" prefWidth="400.0" stylesheets="@estilos.css"
xmlns="http://javafx.com/javafx/11.0.1" xmlns:fx="http://javafx.com/fxml/1">
    <children>
        <Button layoutX="120.0" layoutY="180.0" mnemonicParsing="false"
            prefHeight="40.0" prefWidth="160.0" text="ACEPTAR"/>
    </children>
</AnchorPane>
```

Esta ventana contiene únicamente un botón con el texto ACEPTAR. Si no aplicamos ningún estilo, se mostrará el botón de la siguiente manera:



Fig. 13.17 Botón sin ningún estilo

Si **estilos.css** contiene lo siguiente:

```
.button {
    -fx-background-color: #A0EAF5;
    -fx-text-fill: #595C5D;
}
```

entonces el botón se mostrará de color azul y con el texto en gris:



Fig. 13.18 Botón azul y texto gris

Si **estilos.css** contiene lo siguiente:

```
.button {
    -fx-background-color: red;
    -fx-text-fill: #FFFFFF;

    -fx-font-size: 20;
    -fx-background-radius: 50;

    -fx-border-radius: 50;
    -fx-border-width: 4;
    -fx-border-color: blue;
}
```

el resultado será un botón rojo, con el texto de tamaño 20 puntos, blanco, redondeado y con un borde azul:



Fig. 13.19 Botón rojo redondeado, con borde azul

