

TEMA 2: JAVA FX-CSS

1 APLICANDO ESTILOS CSS A JAVA FX

A la hora de diseñar una aplicación JavaFX hay que tener claro su estructura. Una aplicación desarrollada con JavaFX está compuesta por tres componentes esenciales los cuales son:

- **Stage** - Escenario.
- **Scene** – Escena.
- **Pane** → Nodo
- **Nodes** - Nodos.

La API de JavaFX nos da las siguientes funciones.

- **Scene:** `getStylesheets().add("dirección de archivo css")`
- **Node** - `getClass().add("nombre de clase css")`
- **Node** – `setStyle("-fx-estilo-css")`

Ejemplos:

```
button1.setStyle("-fx-background-color: darkslateblue; -fx-text-fill: white;");
```

```
button2.setStyle("-fx-background-color: darkslateblue; -fx-text-fill: white;");
```

```
text1.setStyle("-fx-font: normal bold 20px 'serif' ");
```

```
text2.setStyle("-fx-font: normal bold 20px 'serif' ");
```

```
gridPane.setStyle("-fx-background-color: BEIGE;");
```

JavaFX permite la personalización de los elementos gráficos de una aplicación Java usando CSS.

La **escena (Scene)** que se utilice en una aplicación JavaFX ofrece la posibilidad de asignarle una serie de hojas de estilos a través de una lista. A dicha lista se le podría añadir una única hoja de estilos, varias o dejarla vacía.

Para acceder a dicha lista de hojas de estilos se puede utilizar el método `getStylesheets()` de la clase Scene:

```
scene.getStylesheets()
```

La función **`getStylesheets()`** nos permite agregar una hoja de estilo CSS a nuestro programa, este archivo contendrá todas las clases con las propiedades que se le aplicaran al objeto que tenga ese nombre de clase asignada.

Para añadir una hoja de estilos a la lista se puede usar el método **`add()`** como es habitual en las listas de Java, indicando en este caso el nombre del archivo que contiene los estilos:

Por ejemplo podemos hacer lo siguiente:

```
Scene escena = new Scene();  
escena.getStylesheets().add("/resources/miArchivo.css");
```

De esta manera le estamos pasando como parámetro al objeto “escena” la dirección del archivo CSS que contiene la información de los estilos.

La escena de la aplicación puede tener varias hojas de estilos asociadas si se van añadiendo a la lista más de una hoja.

Si en un momento determinado deseas vaciar completamente la lista de hojas de estilos (para dejar de nuevo los estilos predefinidos) puedes utilizar el método **clear()** que se encuentra disponible como en otros tipos de listas en Java.

```
scene.getStylesheets().clear();
```

Con la función **getStyleClass()** le asignaremos al objeto nodo la clase CSS que contiene las propiedades que le hayamos configurado.

Por ejemplo:

```
Button boton = new Button("Click me");  
boton.getStyleClass().add("nombre-de-clase-css");
```

Para que este método funcione, se debe agregar primero una hoja de estilo a la escena y dentro de la hoja de estilo CSS agregar la clase con las propiedades que se aplicaran al objeto. Y la función **setStyle()** nos facilita agregar una propiedad de CSS al nodo sin necesidad de asignarle una clase o agregar una hoja de estilo a la escena.

Por ejemplo:

```
Button boton = new Button("Click me");  
boton.setStyle("-fx-background-color: red;");
```

En este caso el color de fondo del botón será rojo

2 CONTENIDO DE LA HOJA DE ESTILOS

- **Selectores**

Las clases de estilos que son habituales en CSS se corresponden en JavaFX con el nombre de las clases utilizadas en el código fuente Java. Se deben indicar el nombre de la clase, en minúsculas, precedido por un punto (.), y has de tener en cuenta que si el nombre de la clase está compuesto por varias palabras se deben separar con un guión (-).

Por ejemplo:

```
.button  
.check-box  
.scroll-bar
```

También se pueden definir estilos personalizados para elementos concretos asignándoles un identificador. Los identificadores deben indicarse precedidos del carácter almohadilla (#).

Por ejemplo:

#boton1

Para asignarle un identificador a un elemento (nodo) de JavaFX se debe usar el método `setId()`, especificando como parámetro de tipo `String` el nombre del identificador que se le quiere asignar. Por ejemplo:

```
Button btnEstilo1 = new Button();
```

```
btnEstilo1.setText("Estilo 1");
```

```
btnEstilo1.setId("boton1");
```

También permite el uso de pseudo-clases de la misma manera que se usan en otros lenguajes que soportan CSS, es decir, usando como separador dos puntos (:) y el nombre del estado. Por ejemplo, para cambiar el estilo del botón cuando se posiciona el ratón sobre él:

.button:hover

• Reglas y propiedades

Las reglas de una definición de estilo asignan valores a propiedades. Los identificadores de las propiedades en JavaFX están precedidos por `-fx-`. Debes usar el separador dos puntos (:) entre la propiedad y el valor que se le asigna y se finaliza cada regla con punto y coma (;).

Todo el conjunto de reglas que desees aplicara un selector se deben encerrar entre llaves { }.

En el siguiente ejemplo se asigna un tamaño de fuente de 14px a todos los botones, y sólo al botón que tenga como identificador "boton1" se le aplica color rojo al texto:

```
.button {  
    -fx-font-size: 14px;  
}  
  
#boton1 {  
    -fx-text-fill: red;  
}
```

En el artículo [JavaFX CSS Reference Guide](#) puedes encontrar todas las propiedades y los posibles valores que se le puede asignar a cada propiedad.