

HERÓIS DO CÓDIGO



JAVAFX E CSS EM PODEROSA AÇÃO



MICHEL ALVES

JavaFX com Estilo

Dominando CSS na Interface dos Seus Apps

JavaFX é uma das ferramentas mais poderosas para criar interfaces gráficas ricas em Java. E um dos seus maiores trunfos é a capacidade de estilizar os elementos da interface usando CSS — algo que torna seu aplicativo muito mais profissional e organizado.

Neste eBook, vamos direto ao ponto: mostrar como usar CSS para estilizar seus componentes JavaFX com exemplos práticos, explicações simples e foco no essencial.

Começando com CSS no JavaFX

Para aplicar um arquivo CSS no seu app JavaFX, basta fazer assim:

Conectando o CSS à Aplicação

```
• • •
```

```
michel_app.java
```

```
1 scene.getStylesheets().add(getClass()  
2 .getResource("/michel_estilo.css").toExternalForm());  
3
```

Isso adiciona o CSS à Scene. Você também pode aplicar diretamente a um componente, mas na maioria dos casos, a Scene é o lugar ideal.

Principais Seletores CSS em JavaFX

JavaFX reconhece uma boa parte dos seletores CSS tradicionais, mas com algumas particularidades. Veja os principais com exemplos reais.

01

Seleccionando por Tipo de Componente

Alterando o estilo de todos os botões

CSS:

```
● ● ● michel_estilo.css  
1 .button {  
2     -fx-background-color: #3498db;  
3     -fx-text-fill: white;  
4     -fx-font-weight: bold;  
5 }  
6
```

Todos os componentes Button terão fundo azul, texto branco e fonte em negrito.

02

Seleccionando por ID

Estilizando um botão específico

JavaFX:

```
● ● ●               michel_app.java  
1 botaoSalvar.setId("botao-salvar");  
2
```

CSS:

```
● ● ●               michel_estilo.css  
1 #botao-salvar {  
2     -fx-background-color: green;  
3     -fx-text-fill: white;  
4 }  
5
```

Ideal quando você precisa destacar um botão ou elemento específico.

03

Selecionando por Classe

Aplicando estilos reutilizáveis

JavaFX:

```
● ● ●               michel_app.java  
1 labelTitulo.getStyleClass().add("titulo");  
2
```

CSS:

```
● ● ●               michel_estilo.css  
1 .titulo {  
2     -fx-font-size: 20px;  
3     -fx-text-fill: #2c3e50;  
4 }  
5
```

Permite aplicar o mesmo estilo em vários componentes, como títulos, destaque, etc.

04

Pseudo-classes

Mudando a aparência ao passar o mouse

CSS:

```
● ● ● michel_estilo.css  
1 .button:hover {  
2     -fx-background-color: #2980b9;  
3 }  
4
```

Isso deixa o botão mais escuro quando o cursor passa por cima — uma ótima forma de indicar interatividade.

Estilizando componentes desativados

CSS:

```
michel_estilo.css  
1 .text-field:disabled {  
2     -fx-opacity: 0.5;  
3     -fx-background-color: #cccccc;  
4 }  
5
```

Campo de texto desativado com aparência visual condizente.

05

Hierarquia de Componentes

Estilizando elementos dentro de containers

CSS:

```
● ● ● michel_estilo.css  
1 .vbox > .label {  
2     -fx-text-fill: #e67e22;  
3 }  
4
```

Todos os Label dentro de uma VBox terão texto laranja. Isso é útil para aplicar estilos contextuais.

06

Exemplo Real: Tela de Login Estilizada

Estilizando um formulário com JavaFX e CSS

JavaFX:

```
● ● ● michel_app.java

1 TextField usuario = new TextField();
2 usuario.setPromptText("Usuário");
3 usuario.getStyleClass().add("campo");
4
5 PasswordField senha = new PasswordField();
6 senha.setPromptText("Senha");
7 senha.getStyleClass().add("campo");
8
9 Button entrar = new Button("Entrar");
10 entrar.setId("btn-login");
11
```

Continuação...

CSS (login.css):

```
michel_estilo.css

1 .campo {
2     -fx-background-color: white;
3     -fx-border-color: #bdc3c7;
4     -fx-border-radius: 4;
5     -fx-padding: 8;
6 }
7
8 #btn-login {
9     -fx-background-color: #27ae60;
10    -fx-text-fill: white;
11    -fx-font-weight: bold;
12 }
13
14 #btn-login:hover {
15     -fx-background-color: #1e8449;
16 }
17
```

Esse estilo cria uma interface limpa, responsiva e moderna para login.

Agradecimentos

Obrigado por ler até aqui!

Esse e-book foi gerado por IA e diagramado por humano.

O passo a passo se encontra no meu github.

Esse e-book foi gerado com fins didáticos de construção, não foi realizado uma validação cuidadosa humana no conteúdo e pode conter erros gerados por uma IA.