



Unidade 1

Seção 1

Programação Orientada a Objetos II

Webaula 1

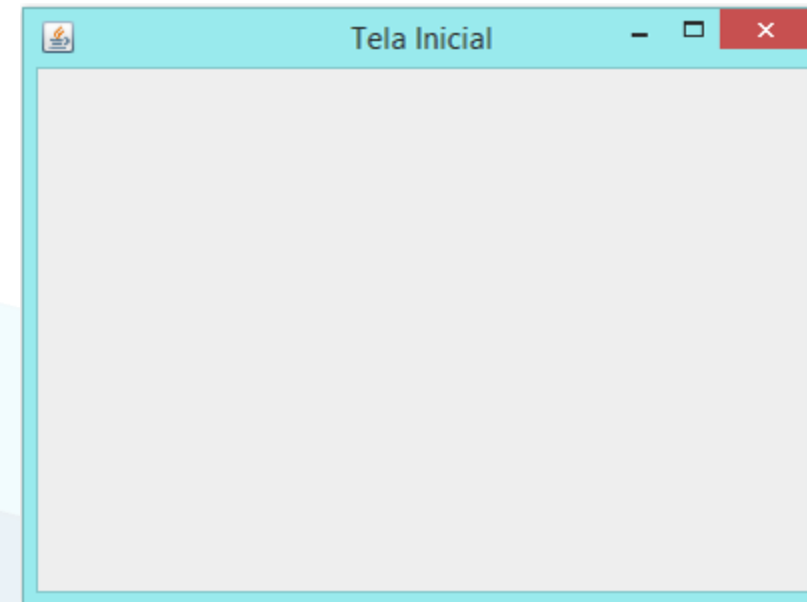
Desenvolvimento de interfaces gráficas na
linguagem Java

A criação de interfaces gráficas, também chamada de *Graphical User Interface* (GUI), é um processo que necessita de grande dedicação, pois por meio delas o usuário percebe e interage com o sistema.

A construção de interfaces gráficas é feita a partir da inserção de componentes em uma tela, como por exemplo, botões, caixas de seleção e a própria representação da tela, dentre outros. Todos esses componentes estão disponíveis dentro de bibliotecas específicas.

No caso do Java utilizaremos o **Swing**. O Swing é uma biblioteca que vem incorporada no **Java Development Kit**, dispondo de diversos elementos para a produção dessas telas. O primeiro elemento a ser utilizado em uma interface gráfica é a representação de uma área que apresenta uma barra de título e um espaço reservado para se adicionar componentes. No caso do Java com Swing, a classe que faz essa representação é o ***Jframe***. A figura a seguir exhibe um exemplo da interface criada por essa classe.

Exemplo de frame ou tela utilizando o Swing do Java



Fonte: captura de tela do software Java.





Mesmo com essas ferramentas é necessário entender como os componentes são alocados na tela e como são feitos os processos de posicionamento, entre outros. Em caso de manutenção ou outros problemas você será capaz de fazer as correções sem depender de uma ferramenta específica. Além disso, muitos componentes precisam ser criados em tempo de execução e isso só é possível através da programação.

Para se criar uma interface gráfica são necessários cinco passos (HORSTMANN, 2016, DEITEL; DEITEL, 2016):

Criar uma relação de especialização com a classe que representa sua tela.

Declarar como atributos os elementos que serão adicionados à tela.

Definir a forma de alocação dos elementos gráficos na tela.

No construtor, instanciar, configurar e posicionar os itens na tela.

Tratar os eventos dos componentes para tratar as ações do usuário com a interface gráfica.

A melhor forma de aprender a programar é programando!

Portanto, vamos agora implementar o código necessário para criar o layout da figura a seguir.

Clique na imagem para ver o código completo.

A imagem mostra uma captura de tela de uma janela de software Java. A janela tem um título "Tela Inicial" e uma barra de controle com botões de minimizar, maximizar e fechar. O layout da janela é o seguinte: no topo, um campo de texto rotulado "Nome"; abaixo dele, um campo de texto rotulado "CPF"; abaixo disso, uma lista suspensa rotulada "Tipo de usuário" com o valor "Administrador" selecionado; e no centro da janela, um botão azul com o texto "Enviar".

Fonte: captura de tela do software Java elaborada pelo autor.

Explore a galeria para ver a implementação do código para criar o layout anterior.

Etapas de 1 a 3.

As linhas 1 a 4 descrevem quais classes necessitam ser agregadas ao código para que seja possível construir a interface.

```
1. import java.awt.Container; //biblioteca para containers
2. import java.text.ParseException;
3. import javax.swing.*; //simplificando a inclusão de bibliotecas
4. import javax.swing.text.MaskFormatter;
```

Fonte: elaborado pelo autor.

Na **etapa 4** de criação de interfaces gráficas, o primeiro passo é instanciar os objetos e configurar os componentes. Explore a galeria para saber mais.

O primeiro componente a ser instanciado é *JLabel*, o qual possui um construtor que recebe o texto a ser apresentado na tela, na linha 19. Esse processo é repetido nas linhas 21 e 26. Os outros componentes também são inicializados como *JTextField*, que é um componente para receber entrada de texto.

```
19. lblNome = new JLabel("Nome");  
20. txtNome = new JTextField();  
21. lblCPF = new JLabel("CPF");  
22. try {  
23.     txtCPF = new JFormattedTextField(new MaskFormatter("###.###.###-##"));  
24. } catch (ParseException e) {  
25.     e.printStackTrace();  
26. }  
27. lblTipo = new JLabel("Tipo de usuário");  
28. cmbTipo = new JComboBox(tiposUsuarios);  
29. btnOK = new JButton("Enviar");  
    ctn.setLayout(null);
```

Fonte: elaborado pelo autor.

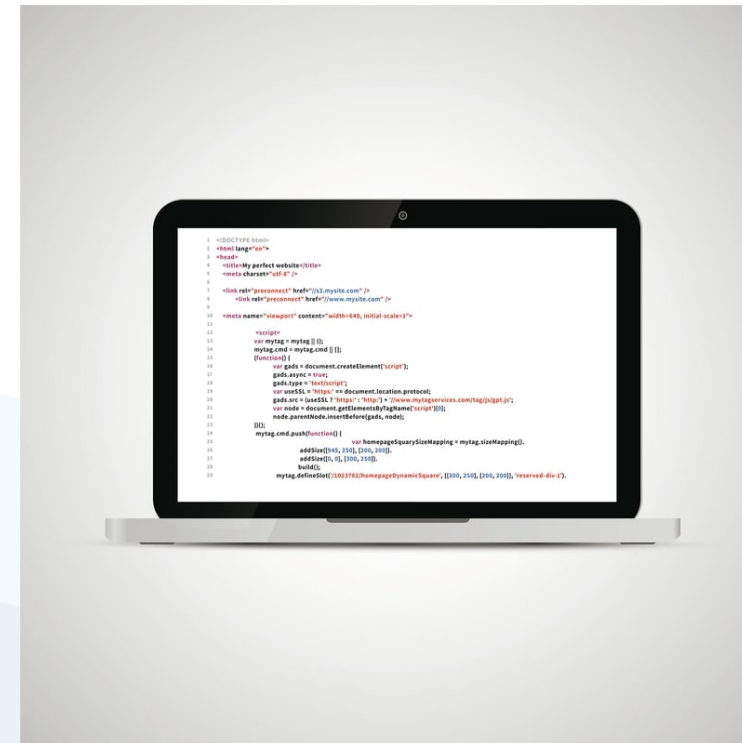
Com todos os elementos já configurados, é necessário posicioná-los na janela, e com isso é possível **concluir a etapa 4** da criação de uma interface gráfica. Explore a galeria para saber mais.

Entre a linha 30 a 43 são feitos o posicionamento e a definição de tamanho de cada componente com o método *setBounds()*, e são inseridos no *Container* que pela estrutura do *JFrame* é adicionado na tela.

```
30.         lblNome.setBounds(0, 0, 100, 25);
31.         txtNome.setBounds(150, 0, 200, 25);
32.         lblCPF.setBounds(0, 50, 100, 25);
33.         txtCPF.setBounds(150, 50, 200, 25);
34.         lblTipo.setBounds(0, 100, 100, 25);
35.         cmbTipo.setBounds(150, 100, 200, 25);
36.         btnOK.setBounds(150, 150, 100, 100);
37.         ctn.add(lblNome);
38.         ctn.add(txtNome);
39.         ctn.add(lblCPF);
40.         ctn.add(txtCPF);
41.         ctn.add(lblTipo);
42.         ctn.add(cmbTipo);
43.         ctn.add(btnOK);
```

Fonte: elaborado pelo autor.

Com esse exemplo de interface gráfica utilizando o Java Swing é possível adicionar outros elementos e criar a tela que você está precisando para concluir sua tarefa. Lembre-se de que uma interface gráfica apresenta muitas linhas de código, sendo necessário manter a organização do código.





Bons estudos!