

Layouts

FlowLayout

FlowLayout é o gerenciador mais simples do Java. Os componentes são organizados da esquerda para a direita e de cima para baixo, na mesma ordem que são adicionados ao painel. Podemos utilizar alguns métodos para mudarmos e configurarmos nosso programa para melhor visualização e entendimento.

Exemplo:

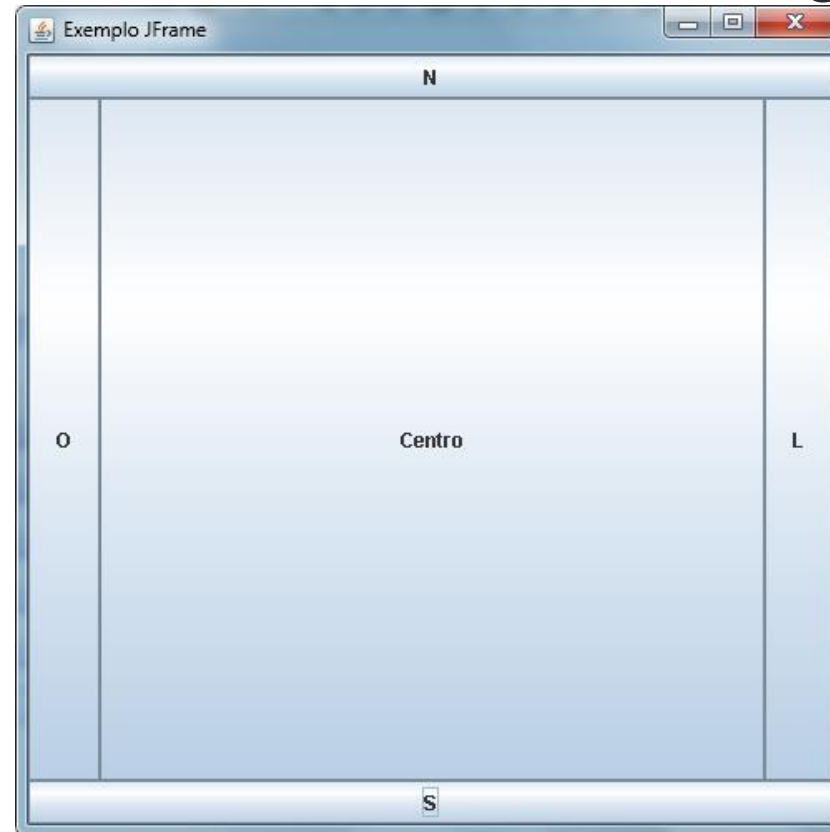
.setSize(L, A) – Define o tamanho do campo;

.setLocation(x, y) – Define a localização do componente no painel.

.setBounds()

BorderLayout

O gerenciador de Layout **BorderLayout** divide o painel em 5 posições definidas (North, South, East, West e Center). Ele limita o painel a conter no máximo 5 componentes, um em cada região. Esse componente pode ser outro painel e dentro dele conter outros componentes.



BorderLayout

Para utilizamos o BorderLayout devemos na hora de instanciar o objeto informar que usaremos o Layout Border.

Exemplo:

//Criando objeto informando o layout utilizado

JPanel painel = new JPanel(new BorderLayout());

//adicionando os botões nas posições do painel

painel.add(b1, BorderLayout.North);

painel.add(b2, BorderLayout.SOUTH);

painel.add(b3, BorderLayout.EAST);

painel.add(b4, BorderLayout.WEST);

painel.add(b5, BorderLayout.CENTER);

GridLayout

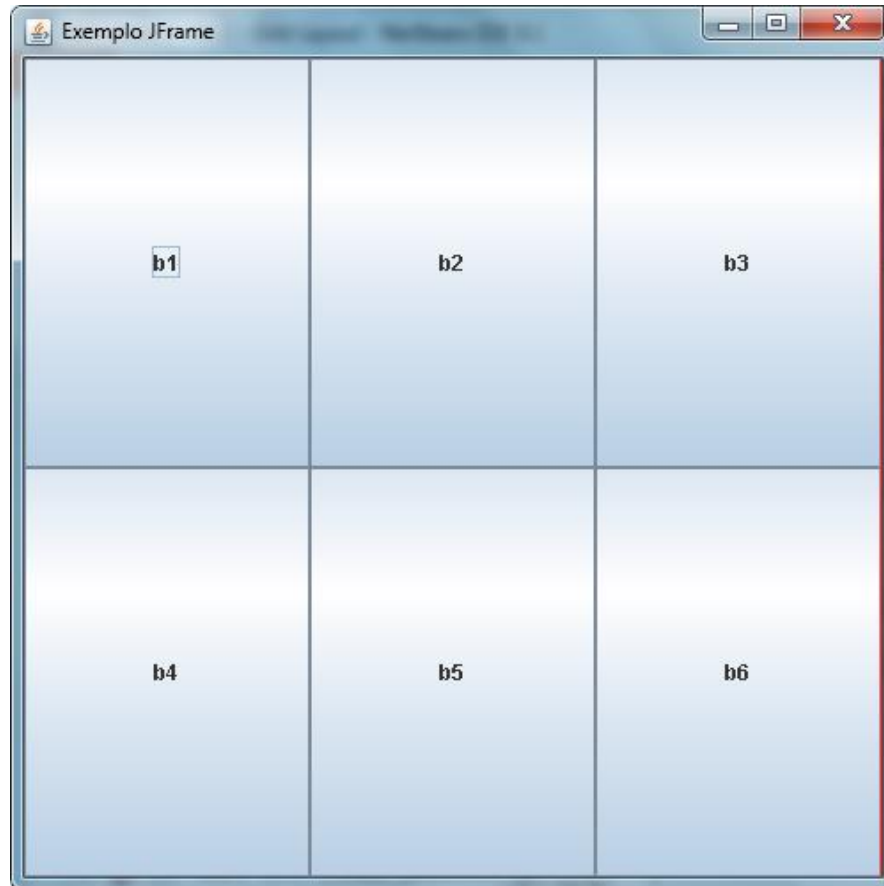
O gerenciador de Layout **GridLayout(L, C, EL, EI)** divide o painel em linhas e colunas, espaçamento lateral e espaçamento inferior, definidas na hora de instanciar o objeto e informado esse tipo de Layout.

L -> Quantidades de Linhas.

C -> Quantidade de Colunas.

EL -> Espaço entre as Colunas.

EI -> Espaço entre as Linhas.



Exemplo GridLayout

//Criando objeto informando o layout utilizado

JFrame janela = new JFrame();

JPanel painel = new JPanel(new GridLayout(2, 3, 5, 50));

//adicionando o painel na posição norte do painel

painel.add(b1);

painel.add(b2);

painel.add(b3);

painel.add(b4);

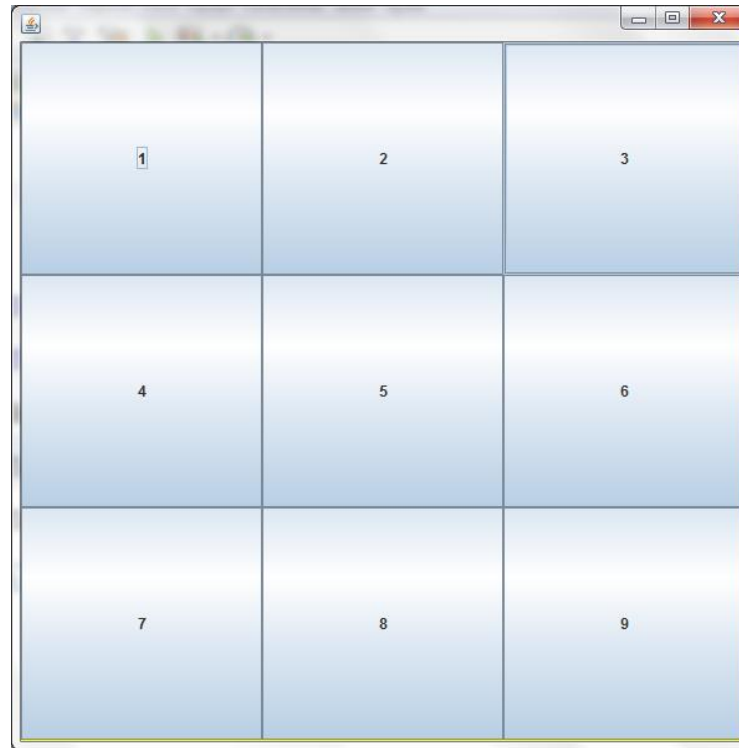
painel.add(b5);

painel.add(b6);

OBS: Os componentes são organizados da esquerda para direita e de cima para baixo, conforme estão sendo adicionados

Exercício GridLayout

Faça um interface gráfica 3x3 com 9 botões, mas que não seja possível apertar o botão localizado no meio (botão 5).



Painel dentro de painel

É possível colocar vários componentes dentro de um painel, inclusive painel dentro de painel.

Painel dentro de painel

```
JPanel painelmestre = new JPanel(new GridLayout(1, 2));
```

```
JPanel painel1 = new JPanel();
```

```
JPanel painel2 = new JPanel();
```

```
painel1.setBackground(Color.red);
```

```
painel2.setBackground(Color.blue);
```

```
painelmestre.add(painel1);
```

```
painelmestre.add(painel2);
```

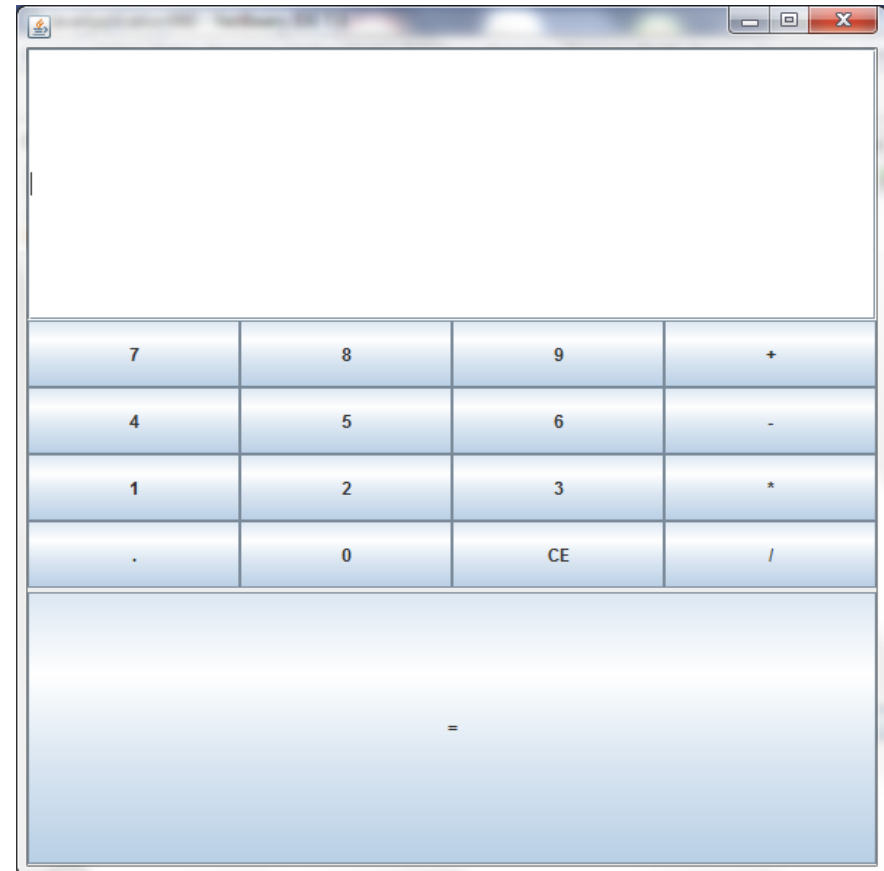
```
janela.add(painelmestre);
```

Exercício

Criar apenas a interface de uma calculadora.

Usando:

- **painelmestre**
- **TextField**
- **painel1 com 16 botões**
- **Button igual**



Alinhamento usando **SetBounds()**

Método utilizado para alinhar os componentes no painel, definindo o seu posicionamento e dimensões.

- **setBounds**(coluna, linha, comprimento, altura);
- Utiliza os **Pixels** para o posicionamento do componente.
- Configurar o painel (JPanel) para não possuir gerenciador de Layout através do método **.setLayout(null);**

Sintaxe

objeto.setBounds(linha,coluna,largura,altura)

objeto.setBounds(10,10,50,50)

Exemplo usando SetBounds()

```
JFrame janela = new JFrame("Formulário");  
JPanel painel = new JPanel();  
painel.setLayout(null);  
JTextField c1 = new JTextField();  
JButton b1 = new JButton("Enviar");  
c1.setBounds(10,10,50,20);  
b1.setBounds(10,70,50,50);  
painel.add(c1);  
painel.add(b1);
```

Exercício

Crie um programa com uma interface gráfica em Java que permita calcular o peso de uma pessoa em vários planetas. O programa deve receber o peso do usuário na Terra, o planeta desejado (de acordo com a tabela abaixo) e exibir na tela o peso do usuário no respectivo planeta.

#	Gravidade relativa	Planeta
1	0,37	Mercúrio
2	0,88	Vênus
3	0,38	Marte
4	2,64	Júpiter
5	1,15	Saturno
6	1,17	Urano

Fórmula

PesoTerra * gravidade relativa

Conversão de Pesos

Peso na Terra (Kg):

Calcular Peso

#Resultado#

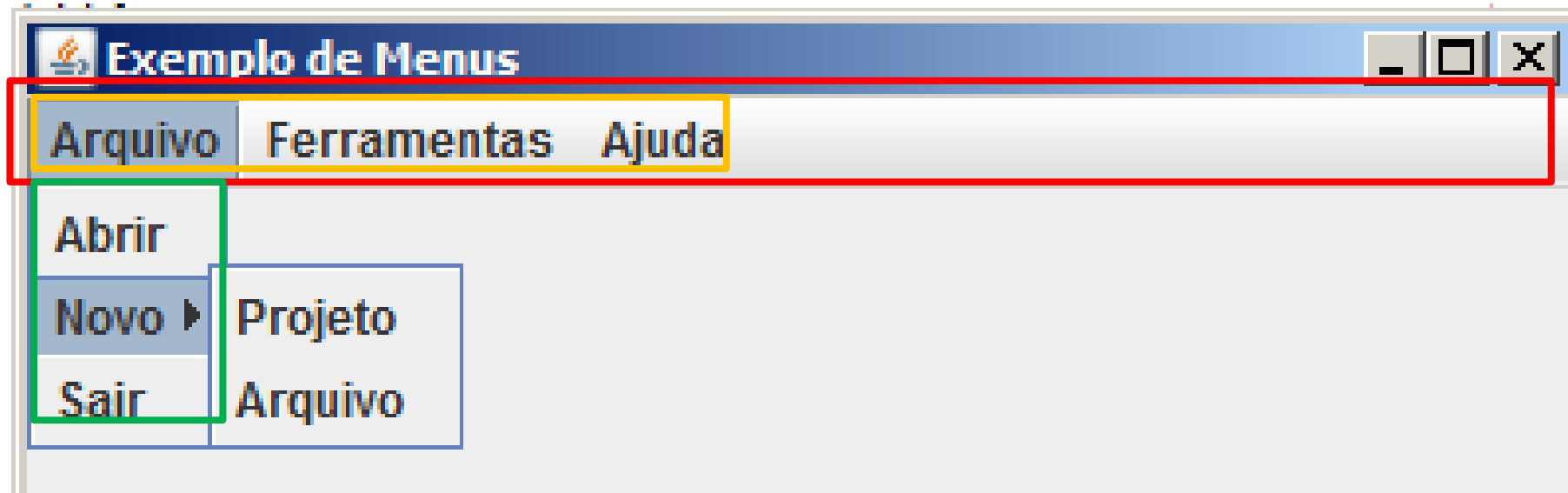
Mercúrio
Mercúrio
Marte
Saturno
Urano

JMenu

É um componente que fica na parte superior da tela, permitindo ações sem poluir a tela com componentes extras.

- O **JMenuBar** contém métodos necessários para gerenciar uma barra de menus, também chamado de barra principal.
- O **JMenu** gerencia os menus, contém os itens de menu e é adicionado na barra principal. O JMenu abre itens.
- **JMenuItem** é um componente dentro do JMenu e executa uma ação quando selecionado. Também pode ser chamado de subitens. O **JMenuItem** é o ultimo componente, dele não abre outro e possui o mesmo evento do **JButton**.

Criando Menus



JMenuBar

JMenu

JMenuItem

**DICA : Possuem o mesmo
evento do JButton
(addActionListener())**

Exemplo – Criando Menus

```
JMenuBar menu = new JMenuBar(); // Criando a Barra Principal  
JMenu menu1 = new JMenu("Arquivo"); // Criando o Menu1  
JMenu menu2 = new JMenu("Editar"); // Criando o Menu2  
JMenu menu3 = new JMenu("Ajuda"); // Criando o Menu3  
JMenu menu1a = new JMenu("Novo"); // Criando o Menu1a  
JMenuItem menu1a1 = new JMenuItem("Projeto"); // Criando o MenuItem1  
JMenuItem menu2a = new JMenuItem("Fonte"); // Criando o Menu2a  
menu1a.add(menu1a1); // Adicionando o MenuItem1a1  
menu1.add(menu1a); // Adicionando o Menu1a  
janela.setJMenuBar(menu); // Adicionando o JMenuBar em nossa janela
```


Exemplo – Criando Menus

Criar Menu conforme a imagem de 2 slides anterior

Criando Abas - JTabbedPane

O **JTabbedPane** organiza em camadas os componentes na tela, deixando apenas uma aba de cada vez visível. Quando clicar em sua guia, a aba correspondente é aberta, exibindo assim seus componentes junto.

Assim que instanciar o objeto, temos que adicionar a aba através do método **.addTab(n, c)** no qual **n** é o nome que queremos na aba e **c** o componente que estamos adicionando a ela, geralmente um painel.

Criando Abas

//Criando o Objeto das Abas

JTabbedPane t1 = new JTabbedPane();

//Adicionando as Abas com o painel em meu JTabbedPane

t1.addTab("Aba1", painel);

t1.addTab("Aba2", painel2);

//Inserindo o JTabbedPane em minha Janela

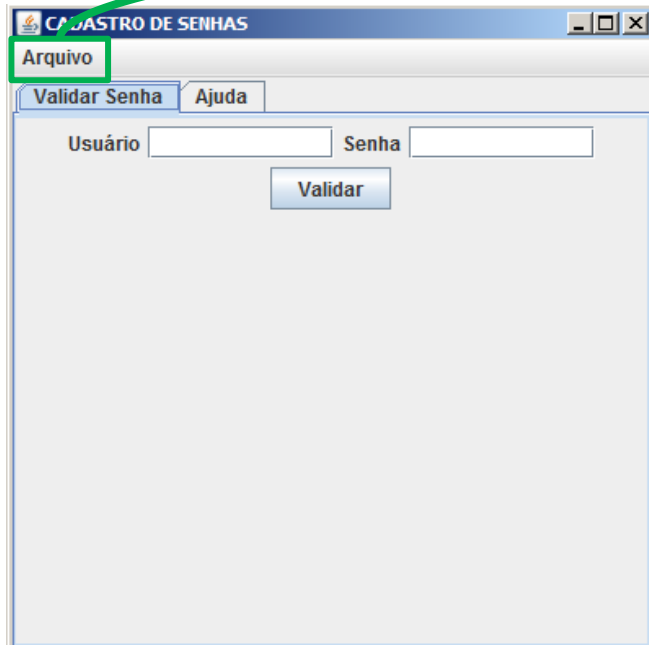
janela.add(t1);

Exemplo - Criando Abas

- Usar a classe **JTabbedPane**.
- Cada aba terá um **Jpanel**.



Exercício



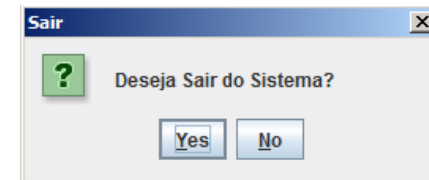
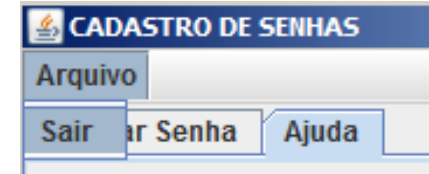
VALIDAR SENHA

Usuário: JTextField
Senha: JPasswordField
Validar: JButton. **Função: verificar se o usuário digitou as credenciais corretas**
Usuário: Aptech e Senha: Java



AJUDA

Exibir uma imagem qualquer para informar ao usuário as credenciais corretas.
Imagem = JLabel



No menu “**Arquivo**” terá uma opção “**Sair**”, ao clicar nessa opção será exibido o painel acima, o sistema somente será fechado se o usuário clicar em **YES**

Método Externo

São métodos em outras classes que podem ser chamado de dentro de um projeto mesmo com a Interface Gráfica.

Para Utilizar o Método externo temos que instanciar o objeto da classe.

Exercício Método Externo

- 1-** Criar um programa com uma caixa que receba 1 valor, criar 2 botões, 1 para converter quilos para gramas e outro de gramas para quilos. Utilizar os métodos em outra classe.
- 2-** Criar um programa que dado o peso e a altura, utilize um método para calcular o IMC a partir dos dados fornecidos.

JSlider

O **JSlider** é um componente que consiste em uma barra horizontal ou vertical, que permite ao usuário selecionar um intervalo de números inteiros através do mouse. Ao deslizar o mouse para esquerda ou para baixo o valor decrementa e ao deslizar para direita ou para cima esse valor é incrementado.

Na hora de Instanciar o objeto do JSlider é necessário passar por 4 argumentos. Exemplo: **Jslider bar1 = new JSlider(o, min, max, c);**

Onde: **o** -> pode ser 0 para horizontal ou 1 para vertical.

mim -> Valor mínimo do objeto.

Max -> Valor Máximo do objeto.

C -> Onde começa nosso marcador.

JSlider



Para pegar o valor do JSlider utilizamos o método **.getValue()**;

Para atribuir um valor utilizamos o método **.setValue()**;

Obs: Os valores utilizados são do tipo inteiro.

Exemplo JSlider

*// Instacia objeto Jslider horizontal com valores de 0 à 200 e inicia
// na posição 40*

JSlider bar1 = new JSlider(0, 0, 200, 40);

//Pega o Valor da posição da barra e salva na variável valor
int valor = bar1.getValue();

//adiciona o JSlider no painel
painel.add(bar1);

Exercício JSlider

Criar um programa com um Jslider que vai de 0 à 250, iniciando na posição 45. Colocar um botão para verificar a posição do marcador cada vez que é pressionado. Mostar o valor em um texto(Jlabel).

JProgressBar

JProgressBar é uma barra de progresso, também conhecida como barra de Download.

Assim como o JSlider, o JProgressBar pode ser colocado na vertical ou na horizontal e definimos o mínimo e o máximo através dos seus parâmetros.

Exememplo:

JProgressBar progresso1 = new JProgressBar(o , min, max);

o -> orientação 0 horizontal e 1 vertical

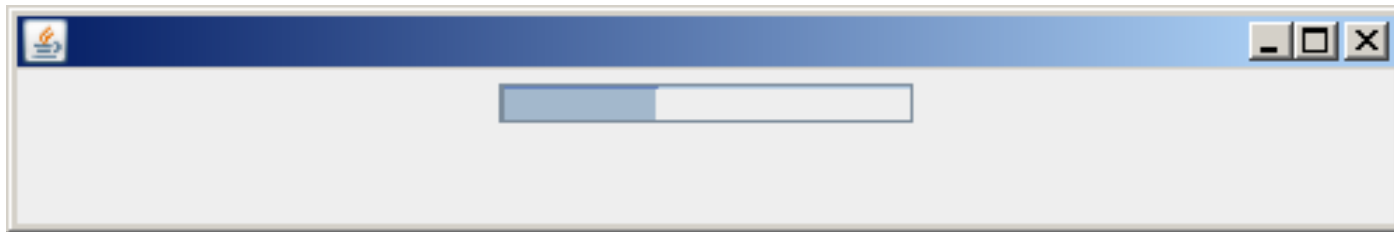
Min -> valor mínimo (início).

Max -> valor máximo do progresso.

JProgressBar

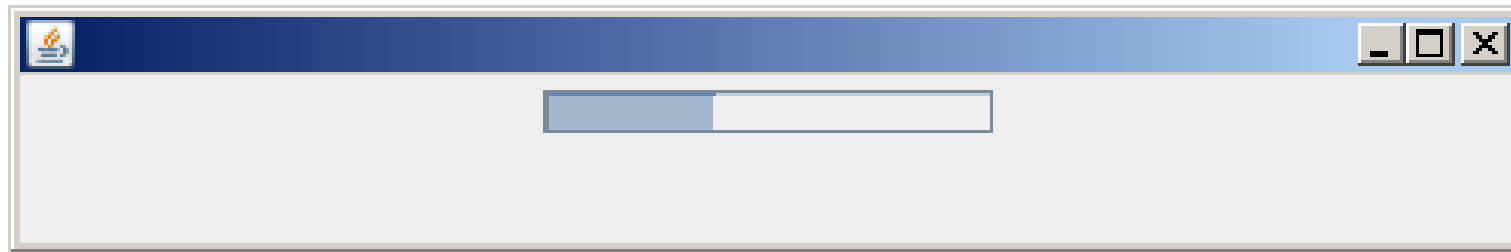
No **JProgressBar**, assim como no **JSlider**, Para pegar os valores utilizamos o método **.getValue();**

E para atribuir o valor utilizamos o método **.setValue(valor);**



JProgressBar

Criar uma GUI utilizando JProgressBar, mostrar o valor do progresso em um JLabel.



TIMER

Timer é um componente que executa de X em X segundos determinados comandos que estão dentro de seu looping. Mesmo que outros comandos estão executando fora de seu looping.

Para utilizar o timer temos que instanciar o objeto **Timer** e já criar sua ação, looping.

Exemplo:

```
Timer relógio = new Timer(2000, new ActionListener() {
```

```
Para iniciar o Timer utilizamos o método .start();
```

```
Para parar o método .stop();
```

```
Para parar o método dentro de seu looping
```

```
((Timer)ae.getSource()).stop();
```

Exemplo TIMER

*//Iniciando o Timer. O **Ac + Ctrl + Espaço** também funciona*

// O 2000 é o tempo em milissegundos.

```
Timer relógio = new Timer(2000, new ActionListener() {
```

```
    @Override
```

```
    public void actionPerformed(ActionEvent ae) {
```

```
        System.out.println("Aptech");
```

```
    }
```

```
});
```

relógio.start(); // inicia o Timer, geralmente dentro de um botão

relógio.stop(); //Finaliza o Timer também pode ser em outro botão

Exercício TIMER

- 1- Utilizar um botão para iniciar o progresso de uma barra de downloads, utilizar o timer para ler e incrementar o valor.
- 2- Fazer o progresso inverso do ex anterior
- 3- Criar um programa que verifica em tempo real a progressão do JSlider.