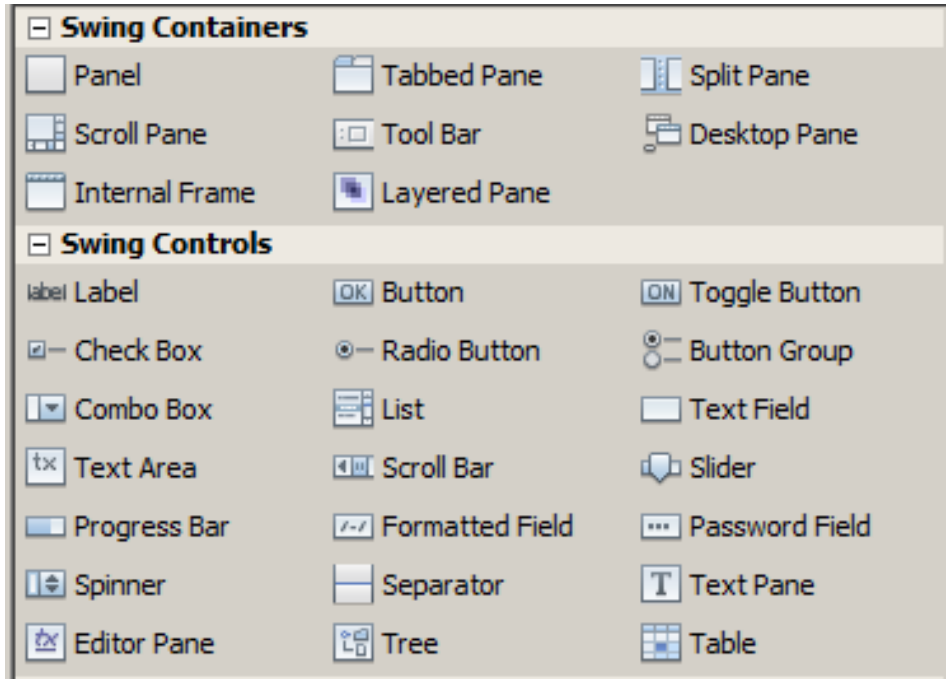
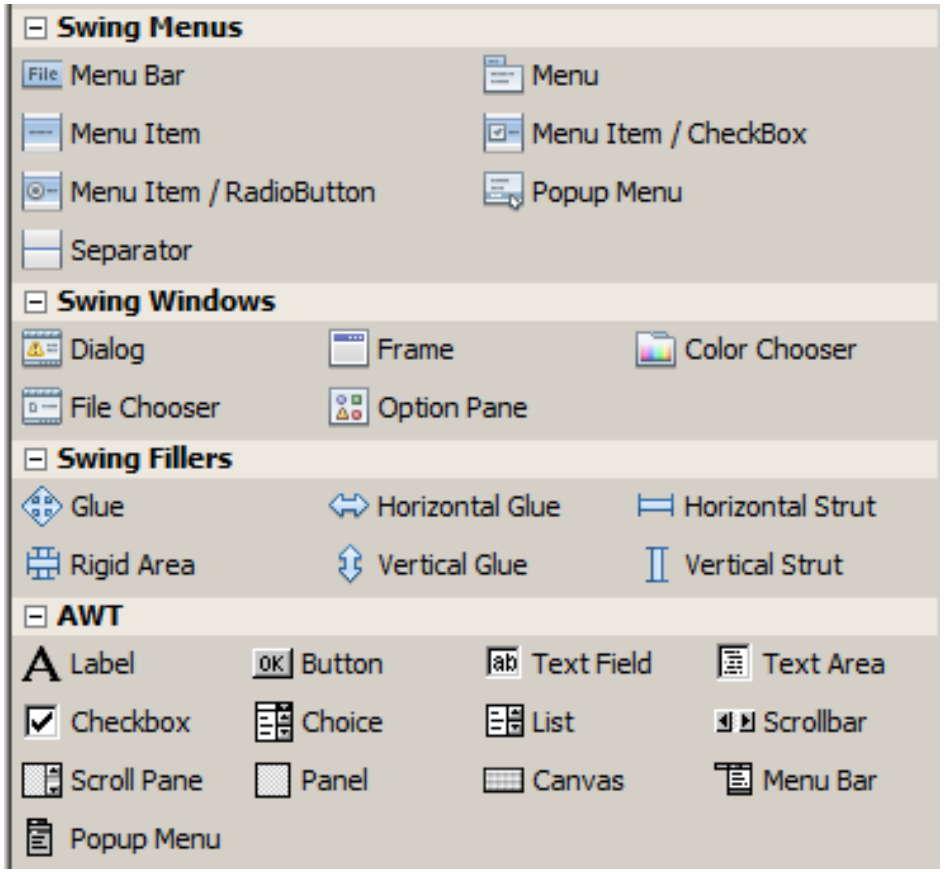
Interface Gráfica

Utilizando Paleta

Permite criar e organizar componentes no painel com um simples clicar e arrastar!





Exercício


CADASTRO DE VEÍCULOS
Código
1
Fabricante
Modelo
Volks
Gol
Ano
Chassi
2013
122-3405/05
KMs
35
SALVAR
DADOS CADASTRADOS
Código.....:
Fabricante...:
Modelo.....:
Ano.....:
Chassi.....:
Kms.....:


CADASTRO DE VEÍCULOS
Código
Fabricante
Modelo

Ano
Chassi
- /
KMs
0
SALVAR
DADOS CADASTRADOS
Código.....: 1
Fabricante...: Volks
Modelo.....: Gol
Ano.....: 2013
Chassi.....: 122-3405/05
Kms.....: 35

Exercício

Desafio Locadora de Filmes

Internacionalização (I18N)

Com as necessidades de criação de aplicações globais, sejam essas em ambiente Web ou não, fica cada vez mais comum o uso de aplicações "internacionalizadas", ou seja, aplicações que mudam o idioma de acordo com o "language" e "country" do usuário.

Internacionalização (I18N)

A classe **Locale** possui vários métodos que podem ser usados para informar a localização de um aplicativo internacionalizado.

- Padrão
  - https://pt.wikipedia.org/wiki/Internacionalização_(software)

Passo 1

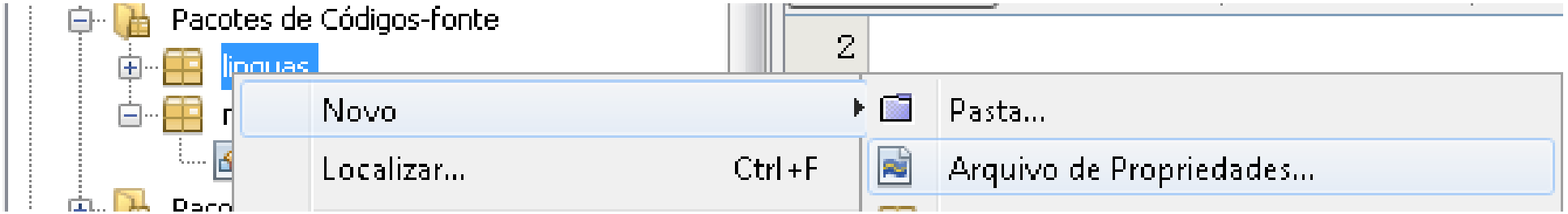
- Criar os arquivos com as traduções **MessagesBundle (Arquivos de Propriedades).** Nesses arquivos serão inseridas as palavras chaves de cada idioma.
- Cada arquivo segue o padrão **MessagesBundle_xx_YY**, no qual xx é o **idioma** e o YY o **país**.

  Exemplo para o idioma **Inglês** e País **Estados Unidos**

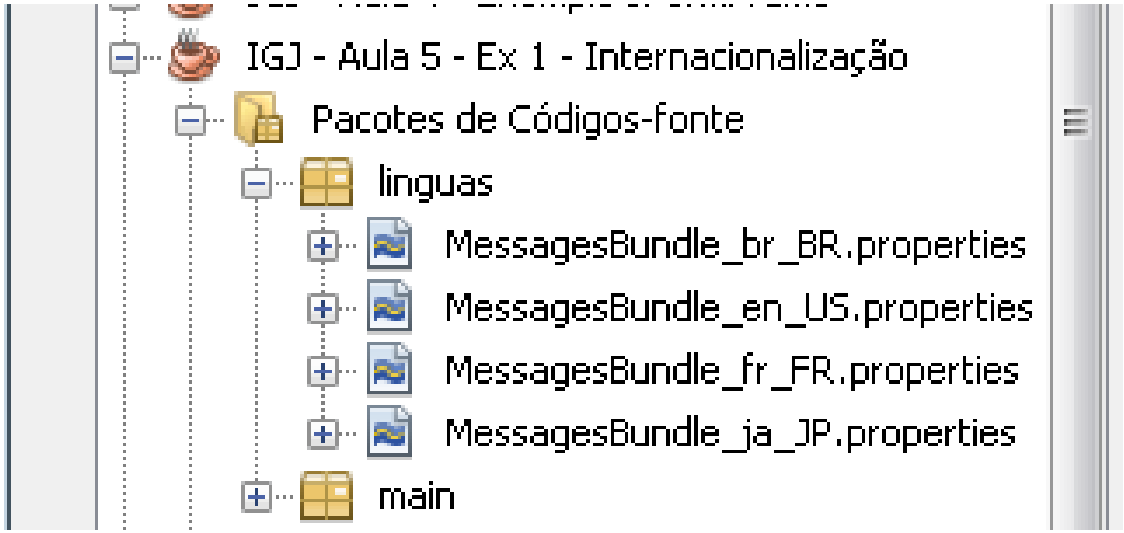
  MessagesBundle_en_US

Passo 1

Criando os arquivos -> Botão direito -> Novo -> Arquivo de Propriedades



Nome Padrão dos arquivos



Palavras Chaves (Idioma Japonês)

Main.java | MessagesBundle_ja_JP.properties

Código-Fonte | Histórico

```
1  texto_dia = ahayougosaimasu
2  texto_tarde = konnitiwa
3  texto_noite = konbawa
4
5  BR = Brasil
6  FR = Francês
7  JP = Japonês
8  US = EUA
```

Passo 2

- Capturar o idioma e o país do usuário através da classe **Locale**
- Locale **brasil** = new Locale(“br", “BR");
- E pegar o conteúdo do idioma do usuário, criado em nosso pacote **línguas**, através da classe **ResourceBundle.**
- **ResourceBundle tradBR = ResourceBundle.getBundle("linguas.MessagesBundle", brasil);**

Passo 3

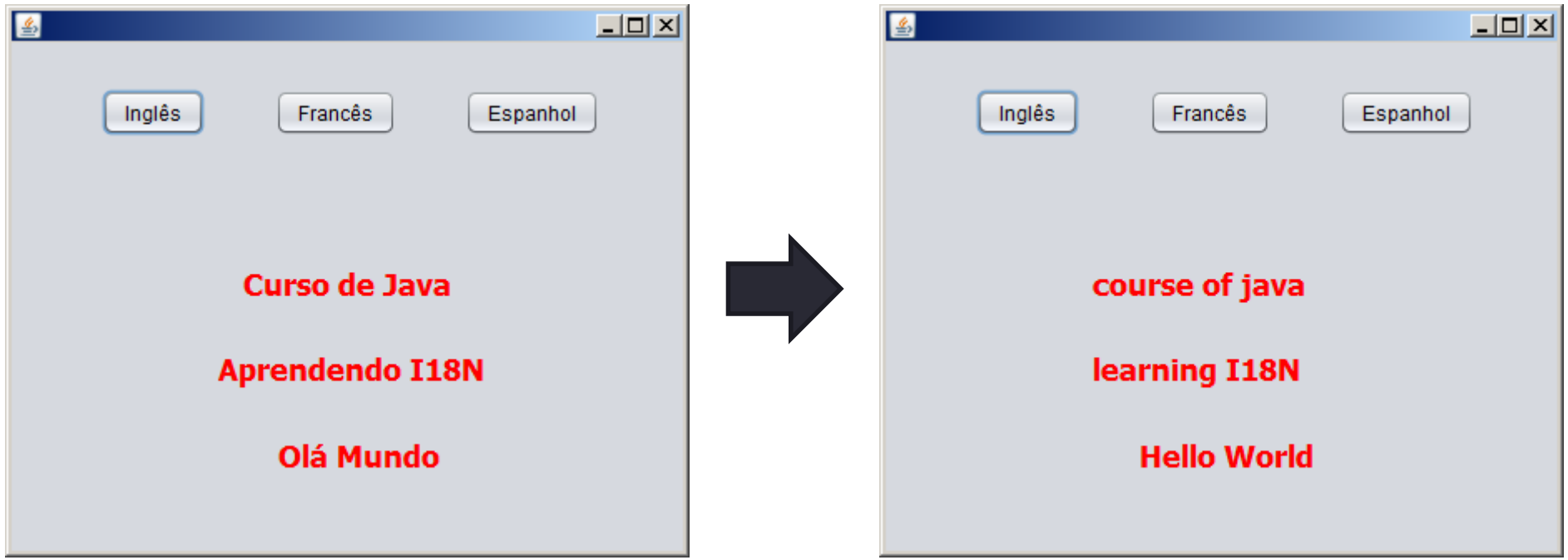
- Preparar nosso programa para receber o conteúdo traduzido dos outros idiomas que criamos em nosso pacote **idiomas** com a classe **ResourceBundle** e a **classe Locale.**

- **ResourceBundle tradUS = ResourceBundle.getBundle("linguas.MessagesBundle", Locale.US);**

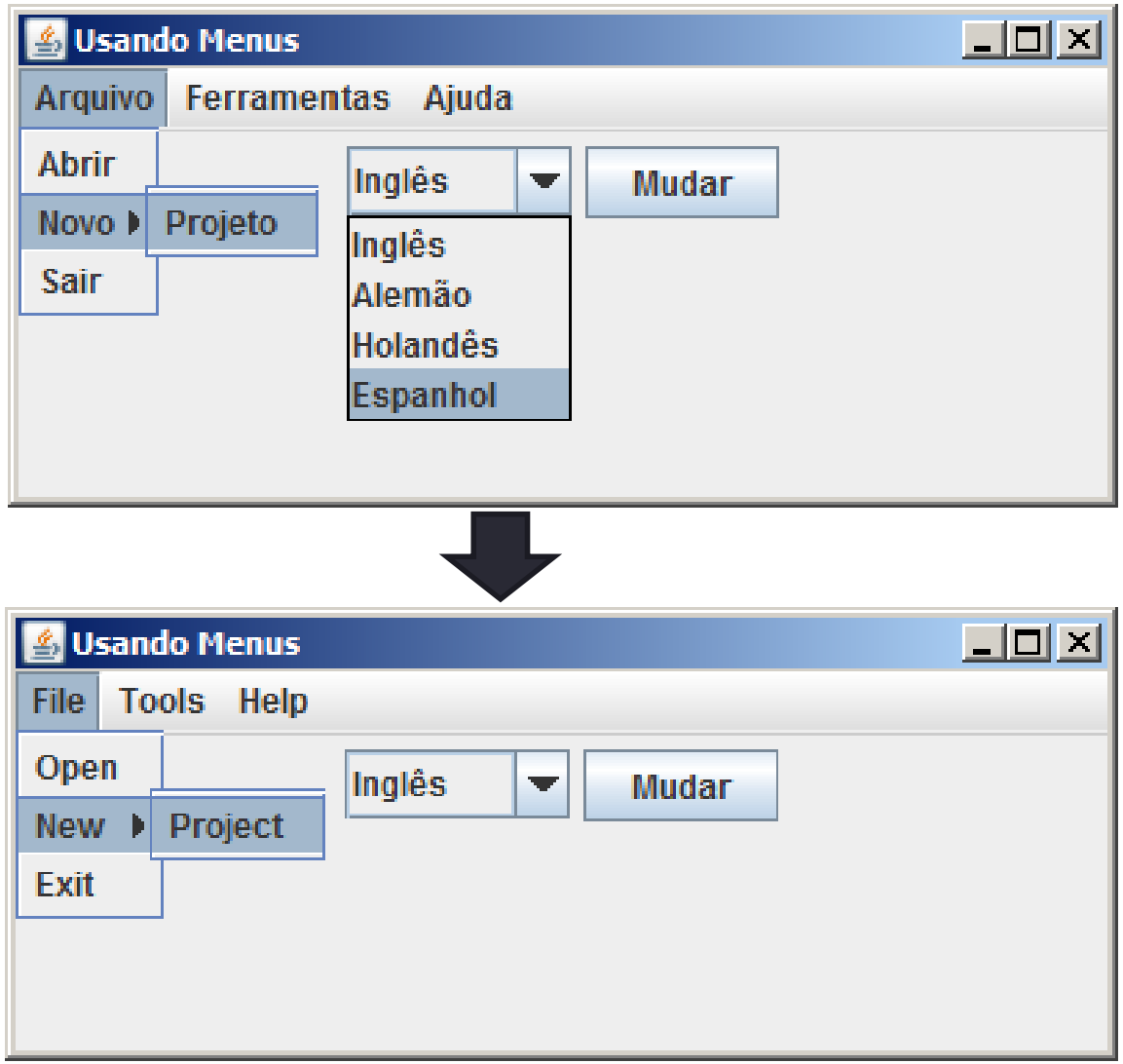
Exemplo

```
//Pegando o idioma do usuário (passo 2)
Locale brasil = new Locale("br", "BR");
ResourceBundle tradBR = ResourceBundle.getBundle("linguas.MessagesBundle", brasil);

//Preparar para receber o idioma escolhido pelo usuário.
     ResourceBundle tradFR = ResourceBundle.getBundle("linguas.MessagesBundle",
Locale.FRANCE);
       ResourceBundle tradUS = ResourceBundle.getBundle("linguas.MessagesBundle",
Locale.US);
       ResourceBundle tradJP = ResourceBundle.getBundle("linguas.MessagesBundle",
Locale.JAPAN);

       //Pedindo pra mostrar o texto_dia em francês
       String palavra = tradFR.getString("texto_dia");
       JOptionPane.showMessageDialog(null, palavra);
```

Exercício


Inglês
Francês
Espanhol
Curso de Java
Aprendendo I18N
Olá Mundo
Inglês
Francês
Espanhol
course of java
learning I18N
Hello World

Exercício


Usando Menus
Arquivo
Ferramentas
Ajuda
Abrir
Novo
Projeto
Sair
Inglês
Mudar
Inglês
Alemão
Holandês
Espanhol
Usando Menus
File
Tools
Help
Open
New
Project
Exit
Inglês
Mudar

Java Mail

O JavaMail oferece uma API (Application Programming Interface) independente da plataforma para criar mensagens de correio e aplicações.

Embora o JavaMail seja uma das APIs da Sun, ela não é incluída nos downloads do JDK.

Requisitos

- Baixar os arquivos da API no site da Oracle

http://www.oracle.com/technetwork/java/index-138643.html

- Adicionar os arquivos *.jar no projeto

Enviando e-mail

- Por padrão, já temos a classe com as configurações necessárias para o envio de e-mail (Gmail). Essa classe é responsável por efetuar a autenticação do usuário no servidor de e-mail.

Enviando e-mail

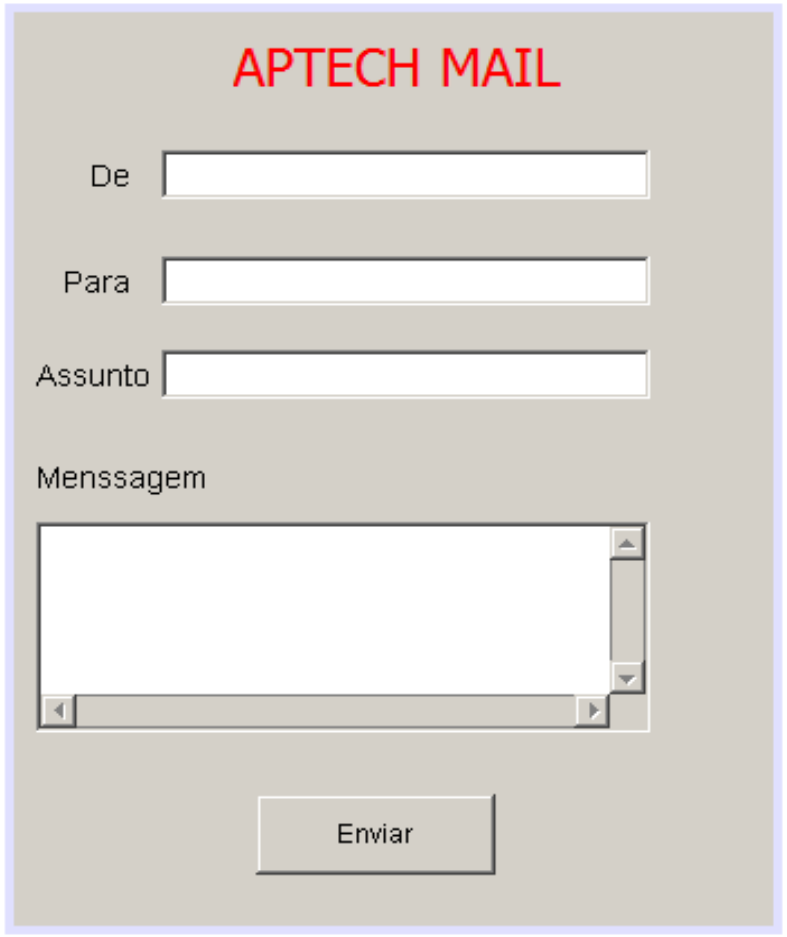
```
String assunto="Teste1";

String remetente = "Aptech";

String para = “emailquequerenviar@gmail.com";

String mensagem = "Olá, teste Java";


Session mailSession = null;

Properties props = null;

Message message = null;
```

Enviando e-mail

```
String smtp = "smtp.gmail.com";


props = new Properties();
props.put("mail.smtp.host", smtp);
props.put("mail.transport.protocol", "smtp");
props.put("mail.smtp.port", "465");
props.put("mail.smtp.ssl.enable", "true");
props.put("mail.smtp.auth", "true");
props.put("mail.debug", "true");
```

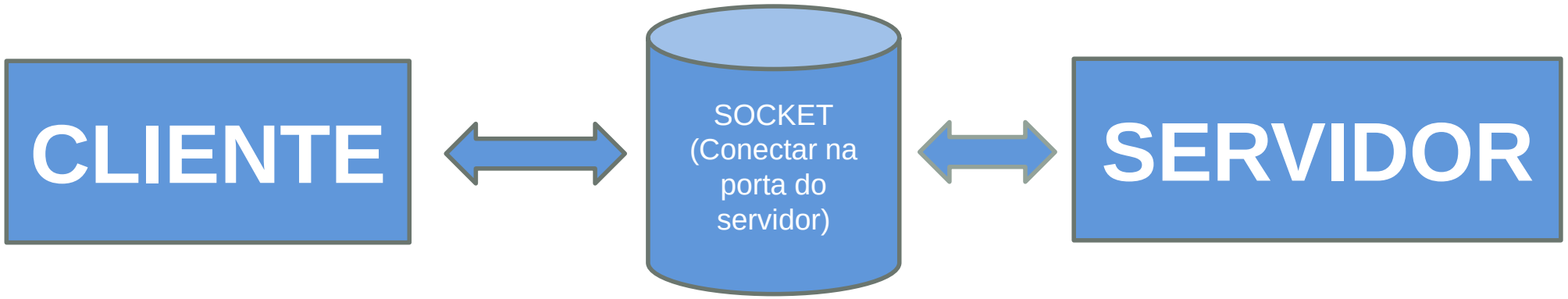
Exercício

Criar uma GUI para o envio de e-mail conforme exemplo.



Sockets

Uma aplicação que utiliza sockets normalmente é composta por uma parte servidora e por diversos clientes. Um cliente solicita determinado serviço ao servidor, o servidor processa a solicitação e devolve a informação ao cliente.



Exercício

Criar um aplicativo (Cliente/Servidor) em que:

- O cliente envia o peso e a altura
- O servidor retorna o IMC