



## Exercício 1: Galeria de Imagens

### Enunciado

Crie uma aplicação em Java Swing que simule uma galeria de imagens. A interface deve conter:

- Um rótulo (JLabel) que exibe uma imagem carregada de um arquivo `.png`.
- Dois botões: "Next" (próximo) e "Previous" (anterior) para navegar entre as imagens.
- As imagens devem estar armazenadas no mesmo diretório do programa e ter o mesmo nome (pode usar as imagens dos códigos de exemplo).

### Resultado Esperado:

- Inicialmente, a janela exibe a primeira imagem.
- Ao clicar em "Next", a próxima imagem é exibida (ex.: de `locate.png` para `analog.png`); ao atingir `stopwatch.png`, volta para `locate.png`.
- Ao clicar em "Previous", a imagem anterior é exibida (ex.: de `analog.png` para `locate.png`); ao atingir `locate.png`, volta para `stopwatch.png`.

## Exercício 2: Seletor de Estados e Cidades

### Enunciado

Crie uma aplicação em Java Swing que permita ao usuário selecionar um estado e visualizar suas cidades correspondentes. A interface deve conter:

- Uma JComboBox com os estados "MT", "GO" e "MS".
- Uma segunda JComboBox que exibe cidades conforme o estado selecionado:
  - "MT": Barra do Garças, Cuiabá, Rondonópolis.
  - "GO": Goiânia, Anápolis, Aparecida.
  - "MS": Campo Grande, Naviraí, Coxim.
- Sugestão: Use um HashMap para mapear os estados às suas cidades.

### Resultado Esperado:

- Ao iniciar, "MT" está selecionado e a segunda JComboBox mostra "Barra do Garças", "Cuiabá" e "Rondonópolis".
- Ao mudar para "GO", a segunda JComboBox atualiza para "Goiânia", "Anápolis" e "Aparecida".
- Ao selecionar "MS", exibe "Campo Grande", "Naviraí" e "Coxim".

## Exercício 3: Lista de Tarefas Simples

### Enunciado

Crie uma aplicação em Java Swing chamada "ToDoListApp" que funcione como uma lista de tarefas. A interface deve conter:

- Um campo de texto (`TextField`) para digitar uma nova tarefa.
- Um botão "Adicionar" para incluir a tarefa na lista.
- Uma lista (`JList`) que exibe todas as tarefas adicionadas.
- Um botão "Concluir" que marca a tarefa selecionada como concluída, adicionando "[Concluída]" ao texto.
- Um botão "Remover" que exclui a tarefa selecionada da lista.
- Use `DefaultListModel` para gerenciar os itens da lista.
- Valide que o campo de texto não esteja vazio ao adicionar e exiba mensagens de erro com `JOptionPane` se o usuário tentar concluir ou remover sem selecionar uma tarefa.

### Resultado Esperado:

- Ao digitar "Estudar Java" e clicar em "Adicionar", a lista exibe "Estudar Java".
- Ao selecionar "Estudar Java" e clicar em "Concluir", a lista atualiza para "Estudar Java [Concluída]".
- Ao selecionar "Estudar Java [Concluída]" e clicar em "Remover", a tarefa é removida da lista.
- Se o campo estiver vazio ao clicar em "Adicionar" ou nenhuma tarefa estiver selecionada ao clicar em "Concluir" ou "Remover", uma mensagem de erro é exibida.

### Código Base

Você deve complementar o código deste [link](#) nos locais indicados a fim de cumprir os requisitos do enunciado.