

Desenvolvimento Backend

2º Semestre de 2024 – Prof. Ms Gustavo Molina

SWING

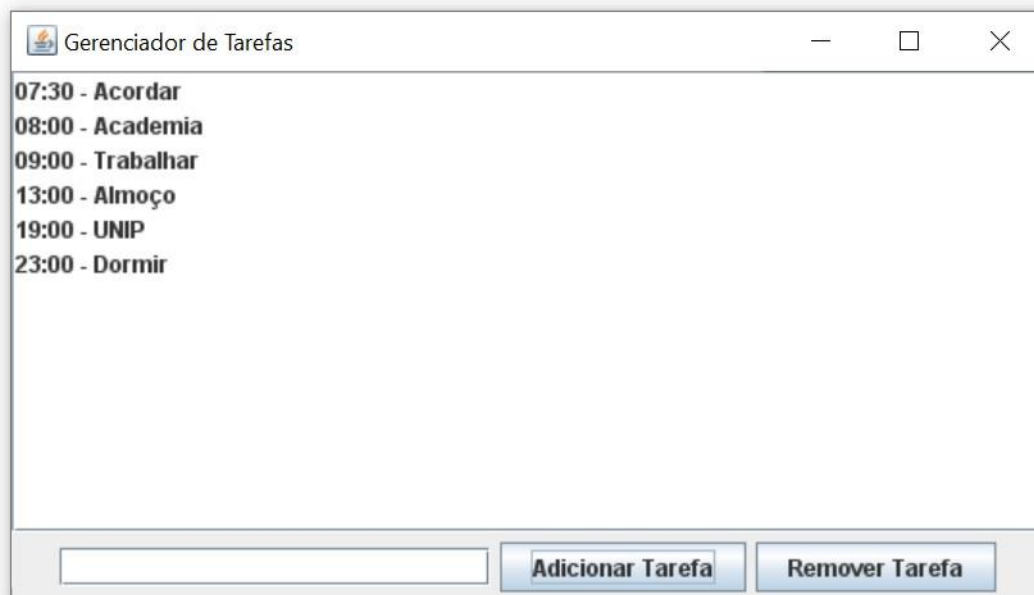
1. **Gerenciador de tarefas:** O código abaixo implementa um exemplo do gerenciador de tarefas, com o uso de AWT + Swing + Eventos.

```
1  import javax.swing.*;
2  import java.awt.*;
3
4
5  public class TaskManagerApp {
6      Run | Debug
7      public static void main(String[] args) {
8          JFrame frame = new JFrame("Gerenciador de Tarefas");
9
10         /*Criação de um modelo de Lista para armazenar as tarefas.
11          * Dessa forma é possível que a lista de tarefas seja atualizada
12          * de forma dinâmica
13          */
14         DefaultListModel<String> taskListModel = new DefaultListModel<>();
15         JList<String> taskList = new JList<>(taskListModel);
16
17         JTextField taskTextField = new JTextField(20);
18         JButton addButton = new JButton("Adicionar Tarefa");
19         JButton removeButton = new JButton("Remover Tarefa");
20
21         addButton.addActionListener(e -> {
22             String task = taskTextField.getText();
23             taskListModel.addElement(task);
24             taskTextField.setText("");
25         });
26     }
```

```

27         removeButton.addActionListener(e -> {
28             int selectedIndex = taskList.getSelectedIndex();
29             if (selectedIndex != -1) {
30                 taskListModel.remove(selectedIndex);
31             }
32         });
33
34         JPanel controlPanel = new JPanel();
35         controlPanel.add(taskTextField);
36         controlPanel.add(addButton);
37         controlPanel.add(removeButton);
38
39         frame.setLayout(new BorderLayout());
40         frame.add(new JScrollPane(taskList), BorderLayout.CENTER);
41         frame.add(controlPanel, BorderLayout.SOUTH);
42
43         frame.setSize(400, 300);
44         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
45         frame.setVisible(true);
46     }
47 }

```



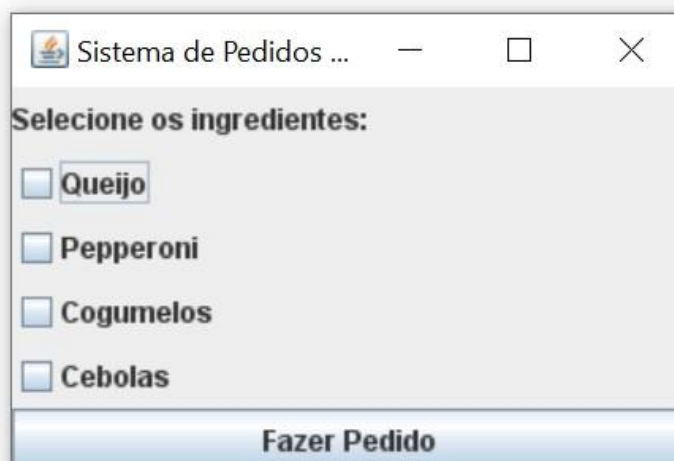
2. **Sistemas de Pedido de Pizzaria:** Implemente o Código abaixo para ver o funcionamento do Programa.

```
1  import javax.swing.*;
2  import java.awt.*;
3  import java.awt.event.ActionEvent;
4  import java.awt.event.ActionListener;
5
6  public class PizzaOrderSystem {
7      Run | Debug
8      public static void main(String[] args) {
9
10         JFrame frame = new JFrame("Sistema de Pedidos de Pizzaria");
11
12         JPanel toppingsPanel = new JPanel();
13         JLabel toppingsLabel = new JLabel("Selecione os ingredientes:");
14
15         JCheckBox cheeseCheckBox = new JCheckBox("Queijo");
16         JCheckBox pepperoniCheckBox = new JCheckBox("Pepperoni");
17         JCheckBox mushroomsCheckBox = new JCheckBox("Cogumelos");
18         JCheckBox onionsCheckBox = new JCheckBox("Cebolas");
19
20         toppingsPanel.setLayout(new GridLayout(0, 1));
21         toppingsPanel.add(toppingsLabel);
22         toppingsPanel.add(cheeseCheckBox);
23         toppingsPanel.add(pepperoniCheckBox);
24         toppingsPanel.add(mushroomsCheckBox);
25         toppingsPanel.add(onionsCheckBox);
26
27         JButton orderButton = new JButton("Fazer Pedido");
```

```

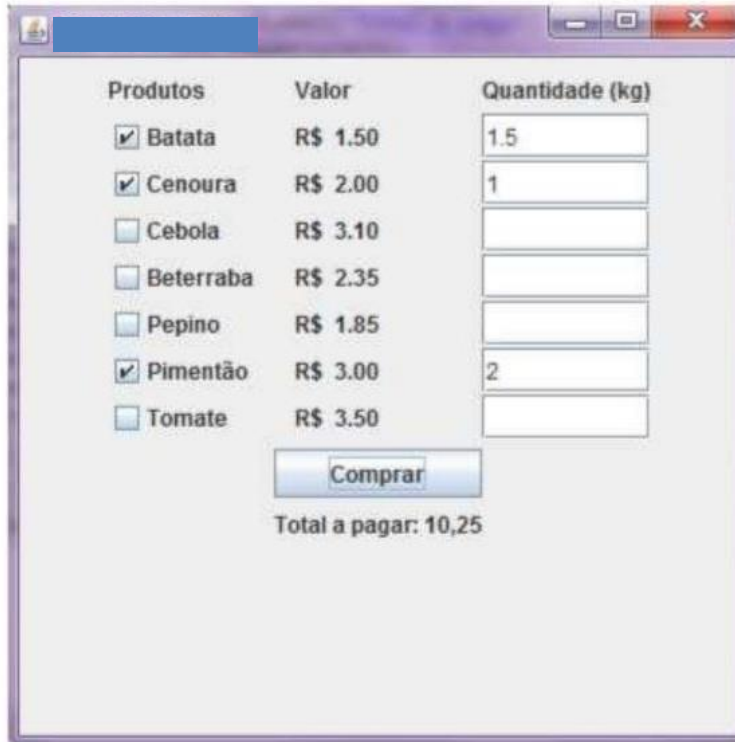
27     orderButton.addActionListener(new ActionListener() {
28         public void actionPerformed(ActionEvent e) {
29             StringBuilder orderDetails = new StringBuilder("Pedido: ");
30             if (cheeseCheckBox.isSelected()) {
31                 orderDetails.append("Queijo ");
32             }
33             if (pepperoniCheckBox.isSelected()) {
34                 orderDetails.append("Pepperoni ");
35             }
36             if (mushroomsCheckBox.isSelected()) {
37                 orderDetails.append("Cogumelos ");
38             }
39             if (onionsCheckBox.isSelected()) {
40                 orderDetails.append("Cebolas ");
41             }
42             if (orderDetails.toString().equals("Pedido: ")) {
43                 orderDetails.append("Nenhum ingrediente selecionado");
44             }
45             JOptionPane.showMessageDialog(frame, orderDetails.toString());
46         }
47     });
48
49     frame.setLayout(new BorderLayout());
50     frame.add(toppingsPanel, BorderLayout.CENTER);
51     frame.add(orderButton, BorderLayout.SOUTH);
52
53     frame.setSize(300, 200);
54     frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
55     frame.setVisible(true);
56 }
57 }

```



Exercícios

1 - Criar uma interface gráfica em Java (utilizando AWT/ SWING) para simular os pedidos de um mercadinho. Pode-se usar como exemplo a interface apresentada ou criar sua própria interface.



The screenshot shows a Java Swing window titled "Mercadinho" with a standard Mac OS X-style title bar (blue, yellow, and red buttons). The window contains a table with three columns: "Produtos", "Valor", and "Quantidade (kg)". There are seven rows of products, each with a checkbox, a price, and a quantity input field. The products are Batata, Cenoura, Cebola, Beterraba, Pepino, Pimentão, and Tomate. The prices are R\$ 1.50, R\$ 2.00, R\$ 3.10, R\$ 2.35, R\$ 1.85, R\$ 3.00, and R\$ 3.50 respectively. The quantities entered are 1.5, 1, and 2 for Batata, Cenoura, and Pimentão. Below the table is a "Comprar" button and a label "Total a pagar: 10,25".

Produtos	Valor	Quantidade (kg)
<input checked="" type="checkbox"/> Batata	R\$ 1.50	1.5
<input checked="" type="checkbox"/> Cenoura	R\$ 2.00	1
<input type="checkbox"/> Cebola	R\$ 3.10	
<input type="checkbox"/> Beterraba	R\$ 2.35	
<input type="checkbox"/> Pepino	R\$ 1.85	
<input checked="" type="checkbox"/> Pimentão	R\$ 3.00	2
<input type="checkbox"/> Tomate	R\$ 3.50	

Total a pagar: 10,25

2 – Criar uma interface gráfica em Java (utilizando AWT / SWING) para criar um programa que faça o cálculo de IMC de uma pessoa. Não esqueça de fazer a sua aplicação o mais completa possível, incluindo eventos para exibição de mensagens de acordo com a faixa de IMC.