

LUIZA DA COSTA

AVA3:
ALGORITMOS E ESTRUTURAS DE DADOS

1. INTRODUÇÃO E PROBLEMATIZAÇÃO

A fim de integrar os conhecimentos adquiridos nas disciplinas do módulo, o projeto de extensão de criação de uma startup continuará sendo base para o desenvolvimento desta atividade de extensão.

A minha startup, idealizada em grupo com outros colegas, se chama Melhor Preço e é um aplicativo que visa facilitar a vida dos consumidores que desejam comprar produtos de supermercado com o melhor preço. Por meio do app, o usuário pode comparar os preços dos produtos em diferentes mercados da cidade, permitindo assim que ele escolha a melhor opção de compra para seu orçamento. O aplicativo funciona da seguinte forma: o usuário pesquisa pelo produto que deseja comprar, e o aplicativo mostra os preços em diferentes mercados parceiros. O usuário pode adicionar os produtos que deseja em um carrinho virtual, que sempre será composto pelos produtos de melhor preço, mesmo que sejam de mercados diferentes. Quando a compra do usuário envolve produtos de dois ou mais mercados diferentes, ele terá duas opções: retirar a compra pessoalmente nos mercados ou pagar uma taxa para ter a compra entregue em casa. Se todos os produtos do carrinho forem de um só mercado, o frete é grátis. Com essa solução, a startup visa tornar o processo de compra de produtos de supermercado mais prático, rápido e econômico para os consumidores, além de ajudar a fomentar o comércio local dos mercados participantes do aplicativo.

2. ESTRUTURAS DE DADOS PRESENTES NA IMPLEMENTAÇÃO DE UM PROBLEMA REAL

Considerando as estruturas de dados que foram apresentadas durante a disciplina de algoritmos e estruturas de dados, foi simulada algumas das funções principais do aplicativo Melhor Preço (processo de compras e finalização do pedido).

No processo de compras, o aplicativo mostra ao usuário quais são os cinco produtos mais populares do dia. Assim, ele pode escolher o que deseja comprar e vê quais os preços para o mesmo produto em diferentes mercados. O aplicativo indica qual é o melhor preço e o cliente escolhe se deseja adicionar o produto ao carrinho e

continuar comprando. **É importante notar que nessa parte do código, há uma estrutura de pilha para armazenar os preços dos produtos.** A pilha é, portanto, percorrida para se encontrar o melhor preço. Por ora, a estrutura de pilha é a adequada para armazenar esses dados. No futuro, com o desenvolvimento de mais partes do aplicativo, no entanto, outras estruturas poderão ser adicionadas.

Em seguida, na finalização do pedido, o usuário indica ao aplicativo que não gostaria de continuar comprando e, assim, é mostrado o preço final da compra.

É possível observar estas partes mencionadas anteriormente no código abaixo:

```
1 import javax.xml.transform.Source;
2 import java.sql.SQLOutput;
3 import java.util.*;
4
5 public class ShoppingCart {
6     public static void main(String[] args) {
7         Scanner input = new Scanner(System.in);
8         boolean continuarComprando = true;
9         double precoFinal = 0;
10        Map<String, Double> productsCartPrice = new HashMap<>();
11
12        System.out.println("Bem vindo à seção de promoções do dia do App Melhor Preço!");
13        System.out.println("Aqui, você acha os melhores preços para os 5 produtos mais procurados no App.");
14        System.out.println("-----");
15        System.out.println("Confira os produtos do dia: chips, carne, refrigerante, chiclete, pão.");
16        System.out.println("-----");
17        while (continuarComprando) {
18            System.out.println("O que você gostaria de comprar hoje?");
19            String product = input.nextLine();
20
21            Stack<Double> prices = new Stack<Double>();
22            switch (product) {
23                case "chips":
24                    prices.push( item: 3.99);
25                    prices.push( item: 4.00);
26                    prices.push( item: 3.75);
27                    prices.push( item: 4.60);
28                    prices.push( item: 2.19);
29                    break;
30                case "refrigerante":
31                    prices.push( item: 4.99);
32                    prices.push( item: 3.50);
33                    prices.push( item: 4.80);
34                    prices.push( item: 5.75);
35                    prices.push( item: 6.50);
36                    break;
37                case "carne":
38                    prices.push( item: 22.99);
39                    prices.push( item: 24.50);
40                    prices.push( item: 23.89);
41                    prices.push( item: 35.10);
42                    prices.push( item: 31.25);
43                    break;
```

```

44     case "pão":
45         prices.push( item: 5.99);
46         prices.push( item: 4.25);
47         prices.push( item: 3.89);
48         prices.push( item: 5.75);
49         prices.push( item: 4.50);
50         break;
51     case "chiclete":
52         prices.push( item: 0.99);
53         prices.push( item: 1.00);
54         prices.push( item: 1.99);
55         prices.push( item: 0.55);
56         prices.push( item: 0.50);
57         break;
58     default:
59         System.out.println("Opa! Esse produto é inválido. Tente escrever de novo do mesmo modo que é apresentado pela mensagem principal: ");
60         continue;
61 }
62
63 System.out.println("Esse são os preços, em reais, em 5 mercados diferentes " + product + ": " + prices);
64
65 double cheaperPrice = prices.pop();
66 while (!prices.empty()) {
67     double currentPrice = prices.pop();
68     if (currentPrice < cheaperPrice) {
69         cheaperPrice = currentPrice;
70     }
71 }
72
73 System.out.println("O melhor preço para " + product + " é " + cheaperPrice);
74
75 System.out.println("Gostaria de adicionar " + product + " ao seu carrinho? (sim ou não)");
76 String addToCart = input.nextLine();
77
78 if (addToCart.equals("sim")) {
79     preçoFinal += cheaperPrice;
80     productsCartPrice.put(product, cheaperPrice);
81     System.out.println(product + " foi adicionado ao carrinho!");
82 }
83
84 System.out.println("Vai aproveitar para comprar mais? (sim ou não)");
85 String continueShopping = input.nextLine();
86
87 if (!continueShopping.equals("sim")) {
88     continuarComprando = false;
89     System.out.println("Obrigada por comprar com o Melhor Preço!\n");
90
91     // mostrar o preço de cada produto adicionado no carrinho
92     System.out.println("Preços de cada produto");
93     for (Map.Entry<String, Double> entry : productsCartPrice.entrySet()) {
94         System.out.println(entry.getKey() + ": " + entry.getValue());
95     }
96     // mostrar o preço final de toda a compra
97     System.out.printf("O preço total é %.2f\n", preçoFinal);
98 }
99 }
100 }
101
102

```