



**INSTITUTO  
FEDERAL**

Santa Catarina

---

Câmpus  
São José

## **Implementação de dados de 'Cafeteira' em CSV**

Banco de Dados

**Arthur Cadore Matuella Barcella**

24 de Setembro de 2024

Engenharia de Telecomunicações - IFSC-SJ

## Sumário

1. Objetivo .....	3
2. Desenvolvimento: .....	3

# 1. Objetivo

O objetivo desta tarefa é apresentar um sistema de gerenciamento de dados de uma cafeteria, onde o sistema deve ser capaz de armazenar informações de usuários e histórico de cafés em arquivos CSV.

## 2. Desenvolvimento:

O sistema foi desenvolvido em Java, e é composto por duas funções principais: escreve e le. A função escreve é responsável por escrever uma lista de listas de strings em um arquivo CSV, enquanto a função le é responsável por ler um arquivo CSV e retornar uma lista de listas de strings.

Abaixo está o código .java modificado durante a aula de BDC:

```
1 // BDC - Homework 1
2 // Aluno: Arthur Cadore M. Barcella
3 // Importando as bibliotecas necessárias
4 import java.io.File;
5 import java.io.FileNotFoundException;
6 import java.io.FileWriter;
7 import java.io.IOException;
8 import java.util.ArrayList;
9 import java.util.Arrays;
10 import java.util.List;
11 import java.util.Scanner;
12
13 // Classe Cafeteira
14 public class Cafeteira {
15
16     /* Método escreve
17     * Escreve uma lista de listas de strings em um arquivo CSV
18     * @param linhas: lista de listas de strings
19     * @param filename: nome do arquivo
20     */
21
22     public static void escreve(List<List<String>> linhas, String filename) {
23         try {
24             FileWriter arquivo = new FileWriter(filename, true);
25             for (List<String> elem : linhas) {
26                 arquivo.append(String.join(",", elem));
27                 arquivo.append("\n");
28             }
29             // imprimir o conteúdo da lista no terminal:
30             for (List<String> elem : linhas) {
31                 System.out.println(String.join(",", elem));
32             }
33             arquivo.flush();
34             arquivo.close();
35
36         } catch (IOException e) {
37             e.printStackTrace();
38         }
39     }
40 }
```

```

39     }
40
41     /* Método le
42     * Lê um arquivo CSV e retorna uma lista de listas de strings
43     * @param pathname: caminho do arquivo
44     * @return linhas: lista de listas de strings
45     */
46
47     public static ArrayList<ArrayList<String>> le(String pathname) {
48         ArrayList<ArrayList<String>> linhas = new
49 ArrayList<ArrayList<String>>(0);
50         try {
51             File entrada = new File(pathname);
52             Scanner linha = new Scanner(entrada);
53
54             while (linha.hasNext()) {
55                 String[] registro = linha.nextLine().split(",");
56
57                 ArrayList<String> list = new
58 ArrayList<>(Arrays.asList(registro));
59                 linhas.add(list);
60             }
61         } catch (FileNotFoundException e) {
62             e.printStackTrace();
63         }
64
65         return linhas;
66     }
67
68     /* Método main
69     * Função principal do programa
70     * @param args: argumentos da linha de comando
71     */
72     public static void main(String[] args) {
73
74         System.out.println("Cafeteira System");
75
76         // Inicializando variáveis
77         boolean continua = true;
78         int opcao = 0;
79         int id = 0;
80
81         // Inicializando o scanner
82         Scanner in = new Scanner(System.in);
83
84         while(continua) {
85
86             // Menu de opções
87             System.out.println("=====");
88             System.out.println("Digite 1: Para informações de usuário");
89             System.out.println("Digite 2: Para histórico de cafés");
90             System.out.println("Digite 3: Para informações da cafeteria");
91             System.out.println("Digite 4: Para sair");
92             System.out.print("Sua opção: ");
93             opcao = in.nextInt();
94
95             // Laço de verificação de opções
96             if(opcao == 1) {
97                 System.out.print("Entre com o id do usuário: ");

```

```

96         id = in.nextInt();
97         System.out.println("\tId " + id + " selecionado para
informações de usuário");
98
99         ArrayList<ArrayList<String>> linhas = le("usuarios.csv");
100         // imprime todas as linhas do arquivo:
101         for (ArrayList<String> elem : linhas) {
102             System.out.println(String.join(",", elem));
103         }
104
105     }else if(opcao == 2) {
106         System.out.print("Entre com o id do usuário: ");
107         id = in.nextInt();
108         System.out.println("\tId " + id + " selecionado para
histórico de cafés");
109
110         ArrayList<ArrayList<String>> linhas = le("usuarios.csv");
111         // imprime todas as linhas do arquivo:
112         for (ArrayList<String> elem : linhas) {
113             System.out.println(String.join(",", elem));
114         }
115
116     }else if(opcao == 3) {
117         System.out.println("Informações da cafeteira:");
118         System.out.println("\tÓtima cafeteira");
119     }else if(opcao == 4) {
120         continua = false;
121     }
122 }
123 }
124 }

```

O código acima pode consultar e imprimir os dados de usuários e histórico de cafés armazenados em arquivos CSV, conforme apresentado abaixo:

```

1  123,juca,arthur@email
2  124,maria,maria@email
3  125,joao,joao@email

```