

PRG22105 — Programação de Computadores II

Projeto

22 de maio de 2025

Professor: João Cláudio Elsen Barcellos, joao.barcellos@ifsc.edu.br

Estudante:

Exercício 1: Máquina de refrigerantes

Objetivo e visão geral: Neste projeto propõe-se o desenvolvimento um sistema, executado via terminal, para simular uma máquina de refrigerantes. O sistema fará uso de uma máquina de estados, estruturas de dados (e.g., pilhas, filas, listas simplesmente e duplamente encadeadas) e manipulação de arquivos.

A máquina disponibiliza, inicialmente, dois refrigerantes: **MEET** e **ETIRPS**, ambos custando R\$ 1,50. O usuário pode inserir moedas de R\$ 1,00, R\$ 0,50 ou R\$ 0,25. Moedas inválidas são devolvidas automaticamente. O usuário também pode cancelar a operação ou desfazer a última ação, por meio de uma pilha de *undo*. O sistema fornecerá troco automaticamente quando necessário.

Requisitos mínimos:

1. Máquina de Estados:

- Implementação via `switch-case`;
- Estados obrigatórios: menu inicial, seleção de produtos, pagamento e configuração pelo dono da máquina de refrigerantes.

2. *user interface* (UI):

- Entrada e saída via terminal;
- Atualização “dinâmica” da interface (e.g., por meio do comando *clean*).

3. Gerenciamento de propagandas:

- Propagandas gerenciadas em estrutura de fila;
- Apresentadas ciclicamente no terminal.

4. Estruturas de Dados:

- Pilhas para desfazer (*undo*) últimas ações;
- Filas para gerenciar propagandas;
- Listas encadeadas para configuração e *logs* de vendas.

5. Menu de Configuração:

- Arquivo de configuração protegido por senha;
- Atualização de nome, preço e detalhes dos produtos acessível apenas pelo dono.

6. Persistência e *Logs*:

- *Logs* das vendas armazenados em arquivo;
- Recuperação dos estados anteriores por meio dos arquivos de log.

7. Documentação e Repositório:

- Uso obrigatório de GitHub;
- Documentação do código e do projeto utilizando Doxygen.

8. Funcionalidade adicional:

- Implemente uma funcionalidade extra, como alarme, relatórios gerenciais, ou outras sugestões utilizando estruturas adicionais.

Sugestão de entregas parciais:

Entrega 1: Levantamento de requisitos, diagramas da FSM e especificação das estruturas;

Entrega 2: Implementação da FSM básica, estruturas principais (pilha, fila, lista encadeada), arquivo de configuração inicial e *logs*;

Entrega 3: Integração completa, configuração segura, recuperação de estado via log, documentação e demonstração.

Critérios de avaliação:

- Eficiência das estruturas implementadas;
- Clareza e organização do código;
- Uso correto de máquinas de estados e manipulação de arquivos;
- Qualidade da documentação;
- Uso efetivo do GitHub.

Observações:

- Implementação obrigatória em C;
- O sistema deve rodar integralmente no terminal;
- Estruturas devem ser implementadas sem uso de bibliotecas externas (apenas `stdio.h`, `stdlib.h`, etc.);
- É esperado um comportamento coerente e estável em todas as operações.