

UNIVERSIDADE FEDERAL DO PARANÁ
SEPT – SETOR DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

MARIA FERNANDA ZANDONA CASAGRANDE

PEDRO EDUARDO DALL’ AGNOL

GABRIELA HARRES RODRIGUES

JOÃO VÍTOR KRETZSCHMAR

TRABALHO PRÁTICO II-
SIMULADOR DE ATENDIMENTO MULTICAIXA BANCÁRIO

CURITIBA

2025

MARIA FERNANDA ZANDONA CASAGRANDE - GRR20241727

PEDRO EDUARDO DALL' AGNOL - GRR20240844

GABRIELA HARRES RODRIGUES - GRR20246215

JOÃO VÍTOR KRETZSCHMAR - 20240632

SIMULADOR DE ATENDIMENTO MULTICAIXA BANCÁRIO

Trabalho apresentado à disciplina DS130- Estrutura de Dados I, Setor de Educação Profissional e Tecnológica (SEPT), Universidade Federal do Paraná (UFPR), como requisito parcial à conclusão da disciplina de Tecnologia em Análise e Desenvolvimento de Sistemas, apresentado no Semestre 01/2025

Professor Dr. Helcio Soares Padilha Junior

CURITIBA

2025

SUMÁRIO

1. INTRODUÇÃO
2. DIVISÃO E RESPONSABILIDADES INDIVIDUAIS
3. DESENVOLVIMENTO DO PROJETO
 - 3.1. SEMANA I
 - 3.2. SEMANA II
 - 3.3. SEMANA III
 - 3.4. COMENTÁRIOS
4. LINKS EXTERNOS

1. INTRODUÇÃO

Este trabalho tem como objetivo desenvolver, em linguagem C, um sistema simulador de atendimento bancário com múltiplas filas, integrando os conceitos de pilhas e filas lineares estudados na disciplina de Estrutura de Dados I.

A proposta consiste em representar, de forma prática e funcional, o atendimento simultâneo em dois tipos de caixas: um comum, destinado a qualquer cliente, e outro rápido, voltado a clientes com até dois procedimentos.

A aplicação deve permitir o cadastro de clientes, o encaminhamento automático para a fila correta, o atendimento alternado entre as filas, além do registro de cada atendimento em uma pilha de histórico. Também estão previstas funcionalidades de exibição da situação atual das filas, cálculo do tempo médio de atendimento por tipo de caixa e listagem dos atendimentos realizados.

O desenvolvimento do projeto busca aplicar na prática os conceitos de estruturação modular em C, além de fortalecer a compreensão da dinâmica de inserção, remoção e armazenamento de dados em estruturas como filas e pilhas, utilizando um cenário-problema realista e próximo do cotidiano.

2. DIVISÃO E RESPONSABILIDADES INDIVIDUAIS

Cada integrante ficou responsável por um ou mais módulos do sistema, conforme descrito abaixo:

- Maria Fernanda Zandona Casagrande ficou encarregada do módulo clientes, com os arquivos *clientes.c* e *clientes.h*, responsáveis pela estrutura e criação de novos clientes, e pelo módulo fila, composto por *fila.c* e *fila.h*, que contém a lógica de enfileiramento e desenfileiramento dos clientes nas filas comum e rápida;
- Pedro Eduardo Dall' Agnol foi responsável pelo arquivo *main.c*, que implementa a interface de menu, captura entradas do usuário, encaminhamento dos clientes para as filas e integração dos demais módulos do sistema;
- Gabriela Harres Rodrigues desenvolveu o módulo histórico, com os arquivos *historico.c* e *historico.h*, responsáveis por armazenar os atendimentos realizados na pilha e fornecer acesso aos registros para exibição e cálculo de médias;
- João Vítor Kretschmar foi responsável pela implementação do módulo atendimento, composto pelos arquivos *atendimento.c* e *atendimento.h*, responsáveis pela lógica de atendimento dos clientes, cálculo de tempo médio, registro no histórico e exibição das informações coletadas.

3. DESENVOLVIMENTO DO PROJETO

O desenvolvimento do projeto foi realizado ao longo de três semanas. As decisões técnicas envolveram a escolha de estruturas de dados adequadas ao problema, como filas para gerenciar o fluxo de atendimento e arrays para armazenar registros e clientes. Abaixo, apresentamos o progresso semanal, os desafios enfrentados e as justificativas para as escolhas de implementação:

3.1. SEMANA I

Durante a primeira semana, foi definido o escopo do projeto, a divisão de funções e estrutura inicial do projeto. Também foram criados os arquivos *.c* e *.h* de cada módulo. As filas foram definidas como estáticas usando *arrays*, por sua simplicidade e fácil acesso por índice, o que se mostrou adequado ao porte da aplicação. A pilha de histórico também foi estruturada com *array* fixo, permitindo controle direto do topo e facilitando a iteração no momento de exibir os atendimentos realizados.

3.2. SEMANA II

Durante a segunda semana, os módulos começaram a ser desenvolvidos individualmente de forma mais aprofundada. A estrutura de fila foi finalizada com as funções de enfileiramento e desenfileiramento. A lógica de criação de clientes também foi implementada, assim como o módulo atendimento, com o cálculo do tempo médio por atendimento, o registro no histórico e a exibição dos dados coletados. Por fim, os módulos foram integrados ao *main.c*. O grupo realizou testes de uso com diferentes clientes e cenários, validando o comportamento das filas, da pilha de histórico e da exibição de dados.

3.3. SEMANA III

Durante a última semana, foi gravado o vídeo de demonstração do sistema. A documentação do projeto, com a elaboração do diário de bordo e do *readme*, também foram realizados nessa semana.

3.4. COMENTÁRIOS

Abaixo, comentamos as principais funções implementadas por cada membro do grupo, com foco na lógica adotada, na escolha das estruturas de dados e nos motivos que levaram à sua utilização, conforme solicitado:

Maria Fernanda - módulos clientes e fila:

-Função *criar_cliente*:

Objetivo: encapsular a criação do cliente com nome, número e quantidade de procedimentos.

-Funções *enfileirar* e *desenfileirar*

Objetivo: inserir e remover clientes das filas comum e rápida.

Por que usar filas com array fixo?

- Simplicidade na implementação e uso direto com índices;
- Boa performance em contextos com número limitado de clientes;
- Fácil depuração e visualização para fins didáticos.

Pedro Eduardo - arquivo *main.c*:

-Funções *menu* e *main*:

Objetivo: encapsular a criação do cliente com nome, número e quantidade de procedimentos.

Motivos de implementação:

- A divisão clara das opções torna o programa acessível;
- A estrutura modular permite isolar as responsabilidades, mantendo o *main.c* como controlador.

Gabriela - Módulo histórico:

-Função *empilhar*:

Objetivo: armazenar um registro de atendimento no topo da pilha.

Por que usar *array* na pilha?

- Estrutura leve e direta para simular o armazenamento em ordem de chegada inversa;
- Não exige alocação dinâmica, o que reduz erros e simplifica o controle.

-Funções auxiliares:

Implementadas com base na ideia de acesso constante ao topo e percorrimento simples por índice.

João Vítor - Módulo atendimento:

-Função *atender_cliente*:

Objetivo: simular o atendimento de um cliente da fila e registrar esse atendimento na pilha de histórico. Calcula também o tempo estimado.

Motivos de implementação:

- Permite registrar os atendimentos de forma ordenada;
- Facilita análises estatísticas e rastreabilidade do processo de atendimento;
- Separação lógica entre atendimento e histórico.

4. LINKS EXTERNOS

Link para o vídeo: <https://youtu.be/yEdCJgo-4HM>

Repositório GitHub:

https://github.com/NeroPRDO/EDI_Pratico_Multicaixa_Bancario.git