Memorando

**De:** nome do estudante

**Nº de Matrícula:** ….

**Disciplina:** ….

**Assunto:** ….

**Data:** …

# 1. Introdução

Este relatório apresenta o desenvolvimento de um sistema de gestão de tarefas utilizando a linguagem C. O sistema permite criar, listar e completar tarefas, organizadas por prioridade e data de criação. Foram utilizados conceitos de estruturas de dados (listas ligadas) e manipulação de strings. A estrutura de dados principal é baseada em listas ligadas simples, garantindo eficiência na inserção e remoção de tarefas. O projeto foi implementado com o objetivo de praticar habilidades em manipulação de ponteiros e gestão de memória.

# 2. Experiências Realizadas

A implementação do projeto seguiu uma abordagem modular, com arquivos separados para as funcionalidades principais: gestão de sequências de caracteres (sequencia.c e sequencia.h) e gestão de tarefas (tarefa.c e tarefa.h). O arquivo principal (main.c) integra os módulos e realiza o controle da interface de comandos.

Para compilar e executar o programa, utilizou-se um Makefile com os seguintes alvos:

* **main**: Gera o executável principal.
* **sequencia.o**: Compila o módulo de sequências.
* **tarefa.o**: Compila o módulo de tarefas.
* **main.o**: Compila o arquivo principal.
* **clean**: Remove arquivos objeto e o executável.

**Comandos Principais**

* new <prioridade> <id>: Cria uma nova tarefa com prioridade e ID especificados.
* list <prioridade>: Lista as tarefas da prioridade especificada.
* complete <id>: Marca a tarefa com o ID fornecido como concluída.
* exit: Encerra o programa.

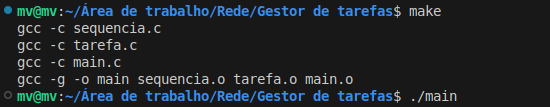


Figura 1 - Compilando o programa usando o make

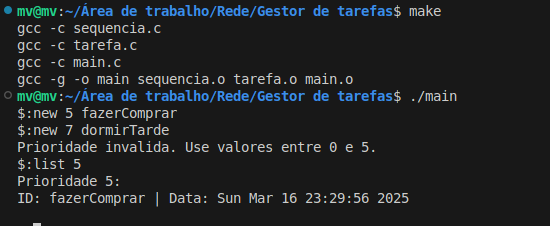


Figura 2 - Executando o arquivo main gerado

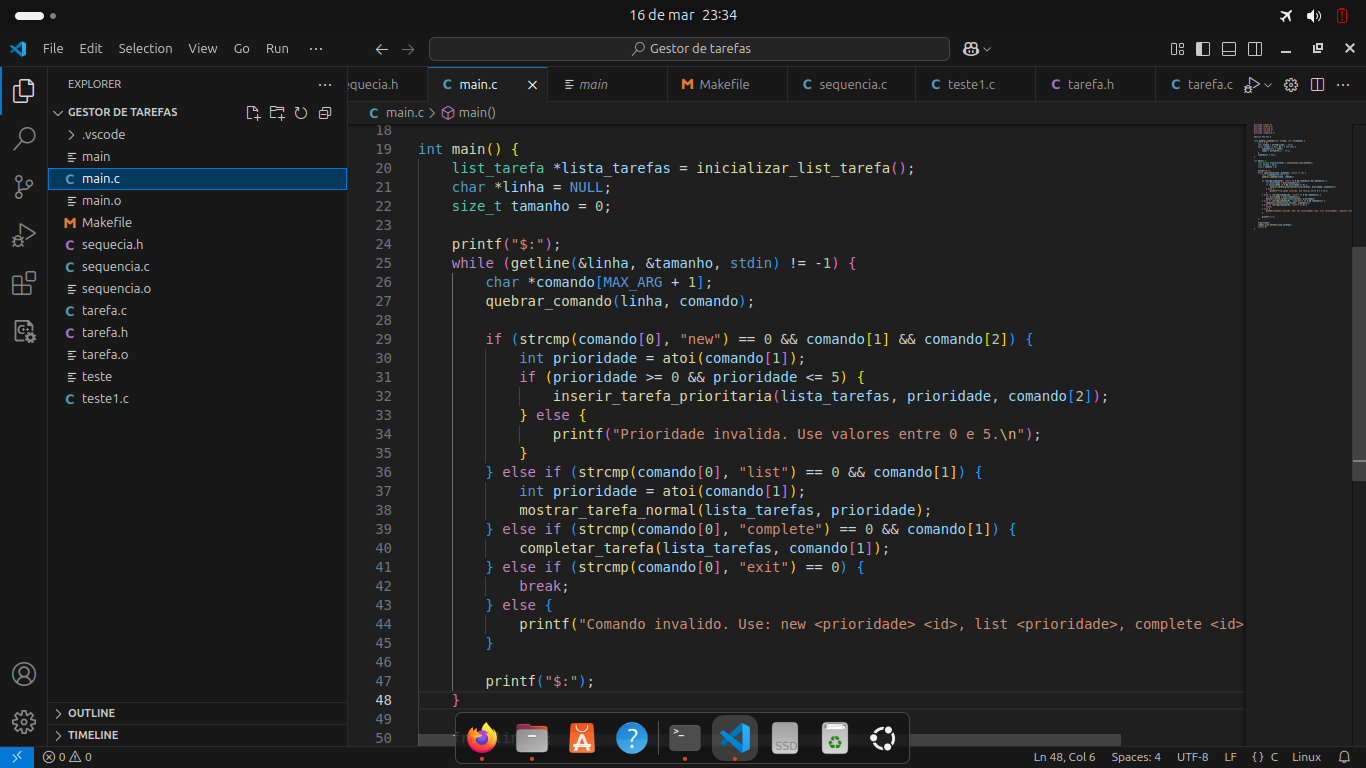


Figura 3 - Arquivo main.c

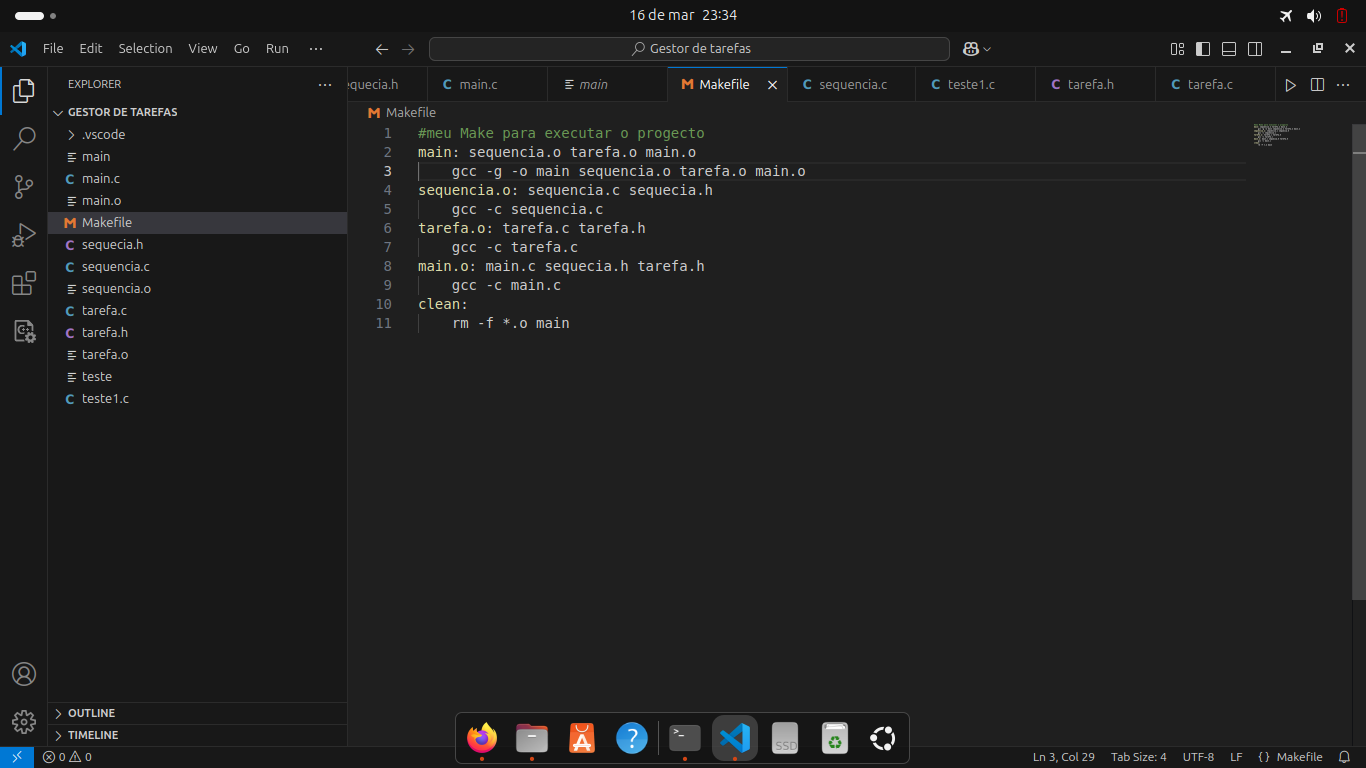


Figura 4 - Makefile

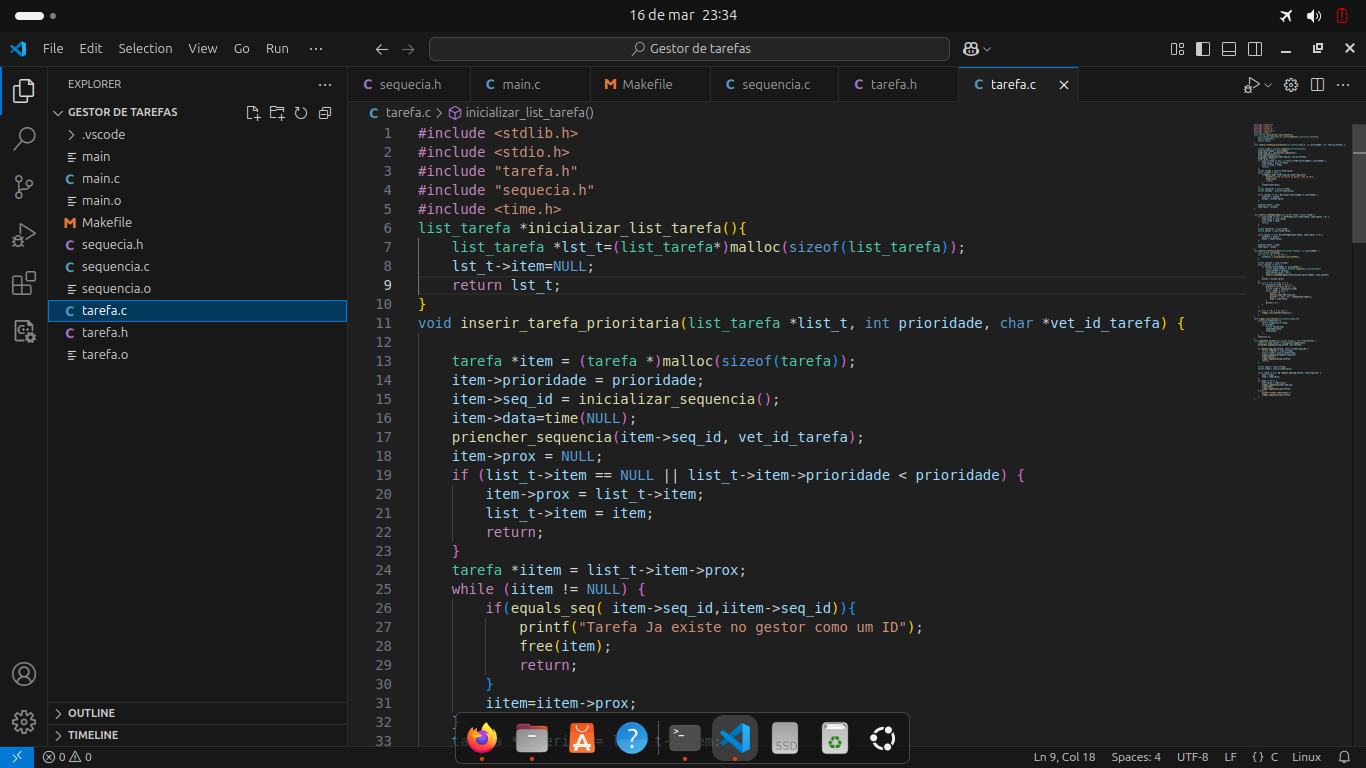


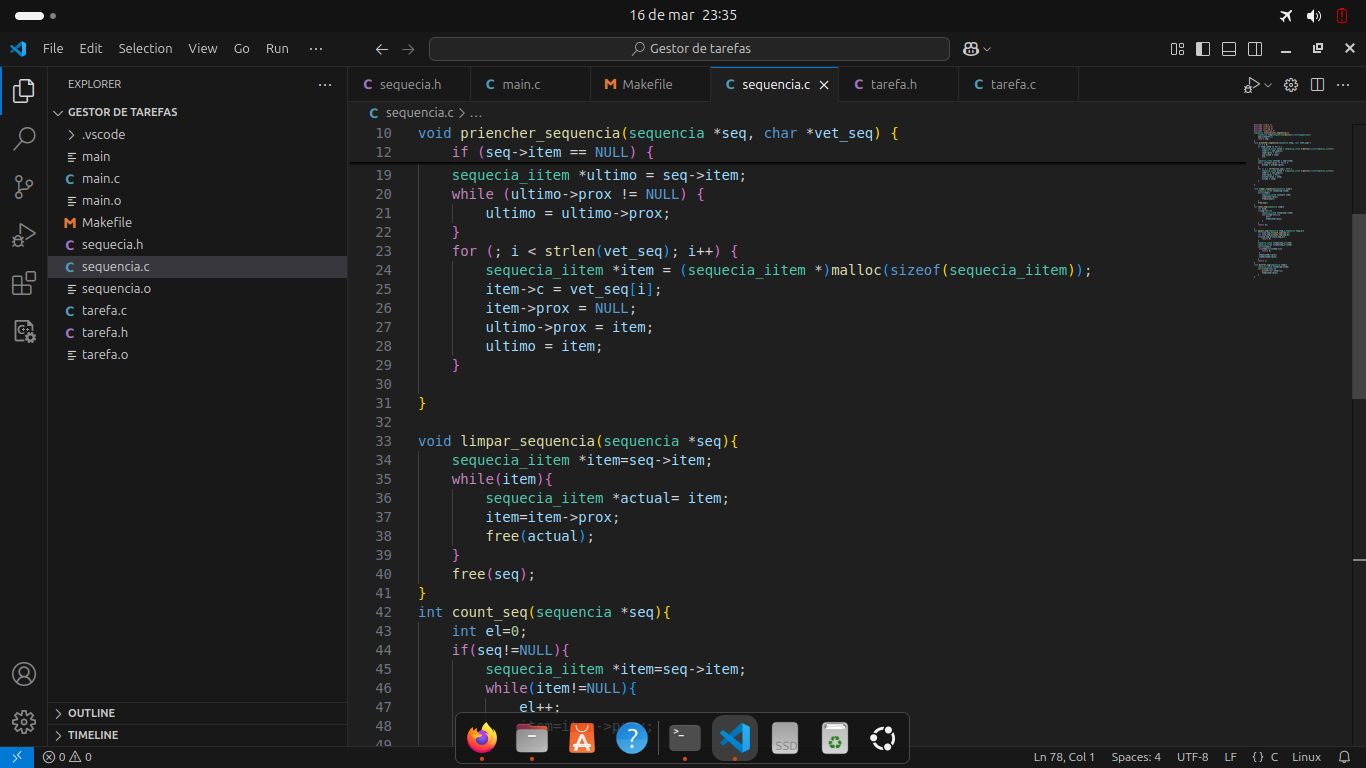
Figura 5 - Arquivo tarefa.c

Figura 6 - Arquivo sequencia.c

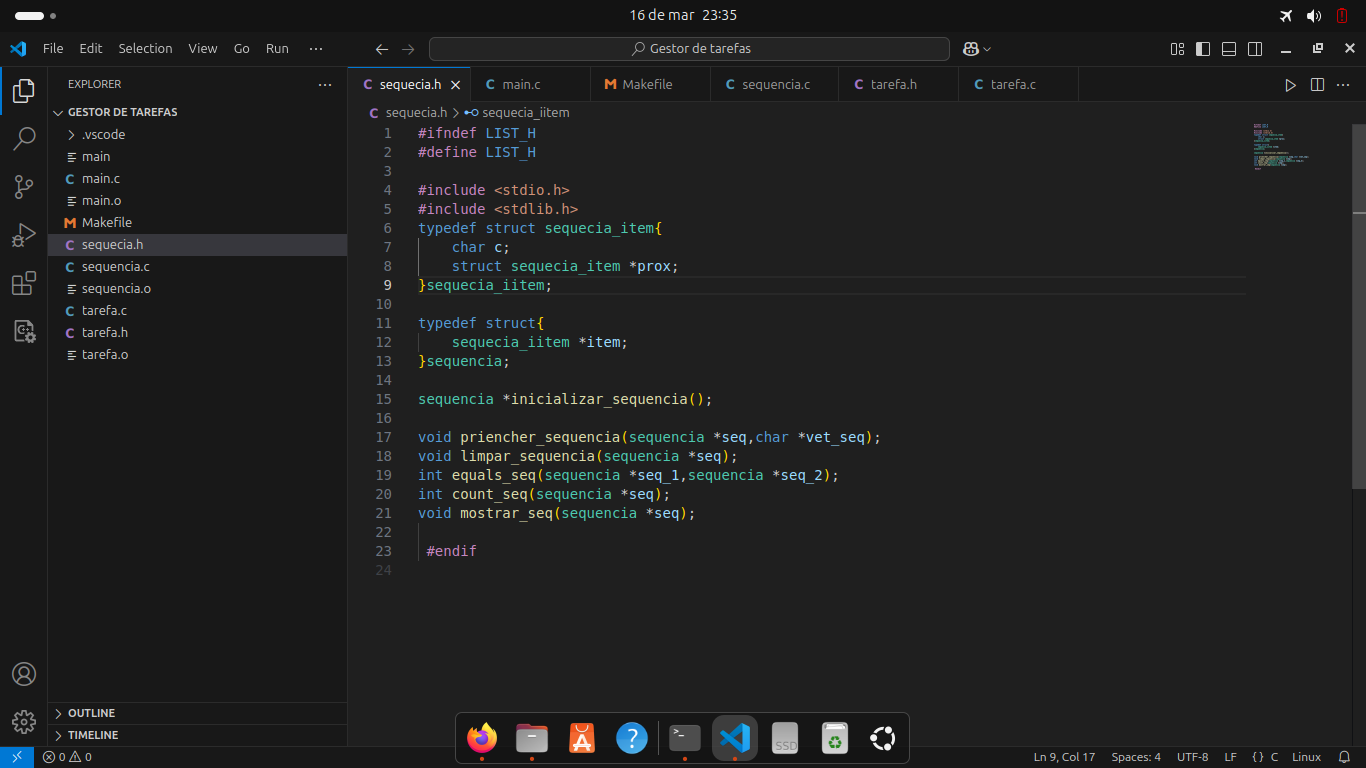


Figura 7 - Arquivo sequencia.h

# 3. Desafios

Um dos principais desafios foi a gestão de memória dinâmica, especialmente para manipular corretamente as listas ligadas e garantir que não houvesse vazamentos de memória. A função de completar tarefas exigiu um tratamento cuidadoso para a remoção eficiente e segura dos nós da lista. Também houve a necessidade de garantir que os IDs fossem únicos para evitar duplicidade de tarefas.

Outro grande desafio foi a utilização de uma sequência (lista ligada de caracteres) no lugar de um vetor de caracteres para armazenar os identificadores das tarefas. Isso permitiu que o número de caracteres em cada ID fosse ilimitado, garantindo maior flexibilidade ao gestor de tarefas. A implementação da sequência está detalhada nos arquivos sequencia.c e sequencia.h.

# 4. Referências Bibliográficas

* Documentação oficial da linguagem C.
* Artigos sobre listas ligadas e gestão de memória em C.
* Exemplos de Makefile para projetos em C.
* Arquivo **cpd-lab1** do professor João Costa

# 5. Repositório GitHub

Coloque aqui o link do repositório e envie o convite de colaborador para o utilizador GitHub **joaojdacosta**.