



INSTITUTO SUPERIOR POLITÉCNICO DE TECNOLOGIAS E CIÊNCIAS
DEPARTAMENTO DE ENGENHARIAS E TECNOLOGIAS
CURSO DE ENGENHARIA INFORMÁTICA

TRABALHO DE ESTRUTURA DE DADOS
GERENCIADOR DE TAREFAS (TASK MANAGER)

Integrantes do Grupo:

- Kélsio Mateus - 20221473
- Lukeny da Silva - 20220718
- Matateu André – 20212549

Luanda, 01.27.2024

ÍNDICE

| | |
|---|---|
| 1. INTRODUÇÃO | 1 |
| 1. METODOLOGIA | 2 |
| 1.1. TIPOS ABSTRACTOS DE DADOS | 2 |
| 1.2. ESTRUTURAS DE DADOS UTILIZADAS | 3 |
| 2. BIBLIOTECAS EXTRAS | 5 |
| 3. CONCLUSÃO | 6 |

1. INTRODUÇÃO

A eficaz administração de tarefas desempenha um papel crucial na organização e otimização do tempo em contextos profissionais e pessoais. No âmbito da disciplina de estrutura de dados, apresentamos a concepção de um gerenciador de Tarefas (Tasker) - um programa destinado a simplificar a gestão, monitoramento e execução de tarefas de maneira sistemática.

1. METODOLOGIA

1.1. TIPOS ABSTRACTOS DE DADOS

- **User:** representa o usuário, cujos campos são: 7 vectores de caracteres (id, name, username, password, email, phone, gender) que servem para armazenar o ID, nome verdadeiro do usuário, nome da conta, palavra-passe, e-mail, número de telemóvel e gênero do usuário.
- **Date:** representa a data de criação e expiração de uma tarefa, cujos campos são: 3 variáveis do tipo inteiro (day, month, year) que representam o dia, mês e ano.
- **Task:** representa a nossa tarefa, cujos campos são: 6 vectores de caracteres (name, id, description, createdAt, updatedAt, expiresAt) que servem para armazenar o nome, ID, descrição da tarefa e as datas de criação, alteração e expiração da tarefa, e duas variáveis do tipo inteiro (state e priority), uma que armazena o estado da tarefa (0 ou 1) e outra, o seu nível de prioridade, dois ponteiros do tipo struct Task que possibilitam a ligação entre as demais tarefas, e um ponteiro do tipo User.
- **TaskManager:** representa o nosso gerenciador de tarefas, cujos campos são: uma Lista Duplamente Ligada que representa o conjunto de usuários que utilizaram o gerenciador de tarefas (List* users), uma Fila que representa o conjunto de tarefas organizadas por prioridade, de ordem decrescente (Queue* tasks) e uma Pilhas de tarefas concluídas (Stack* concludedTasks).

1.2. ESTRUTURAS DE DADOS UTILIZADAS

- **Lista Duplamente Ligada:** A estrutura de lista duplamente ligada foi implementada para representar a lista de tarefas no Task Manager, onde cada elemento da lista é uma TAD Task. A flexibilidade oferecida por essa estrutura para operações de inserção e remoção em qualquer ponto da lista é essencial. Além disso, a capacidade de voltar a estados anteriores é estendida para as tarefas individuais, permitindo que o usuário reverta alterações específicas nas propriedades das tarefas, como nome, descrição, data e estado.

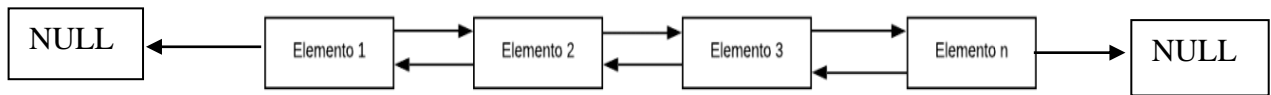


Fig.1: Listas Duplamente Ligadas

- **Fila:** A estrutura de fila (FIFO) concentra-se na gestão de prioridades e na execução sequencial de tarefas. Cada tarefa adicionada à fila representa um ponto na linha do tempo do gerenciador. A natureza da fila garante que a primeira tarefa adicionada seja a primeira a ser processada, proporcionando uma abordagem sequencial para a execução de tarefas. A combinação de fila e Task oferece uma organização eficiente das tarefas em termos de prioridade e execução.

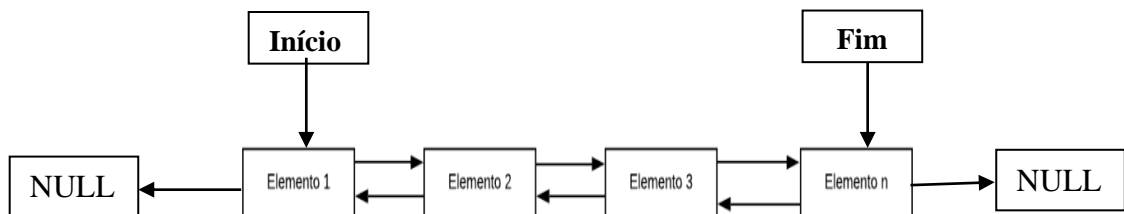
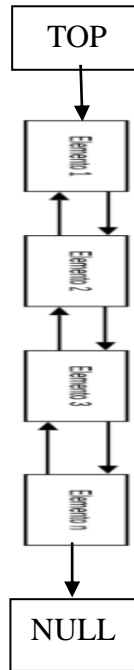


Fig.2: Fila (FIFO)

- **Pilha:** A pilha é essencial para armazenar tarefas concluídas. Cada vez que uma tarefa é concluída, é armazenada no topo pilha, com o objectivo de manter um histórico das tarefas concluídas da nossa fila.



2. BIBLIOTECAS EXTRAS

Para além das bibliotecas padrão da Linguagem C que são a biblioteca `stdio.h` e `stdlib.h`, o nosso projeto teve a inclusão das seguintes bibliotecas extras:

- `string.h`: para manipulação de ponteiros do tipo `char`.
- `locale.h`: para representação de caracteres especiais (acentos).
- `time.h`: para obter o tempo actual do computador.

3. CONCLUSÃO

Durante a implementação, pudemos explorar diversas estruturas de dados, desde listas encadeadas para representar a lista de tarefas até árvores para hierarquizar as prioridades. A escolha criteriosa de algoritmos também se mostrou essencial para garantir a eficiência na busca, ordenação e manipulação das tarefas registradas.

Ao longo do desenvolvimento, enfrentamos desafios que nos obrigaram a repensar abordagens e otimizar o código, consolidando assim o aprendizado prático. A interação entre a teoria estudada em sala de aula e a implementação real foi fundamental para a consolidação do conhecimento.

As listas duplamente ligadas foram adotadas para representar a lista de tarefas, proporcionando uma flexibilidade crucial na inserção, remoção e navegação pelos elementos. A escolha desta estrutura revelou-se vital para manter a dinâmica da lista de tarefas, permitindo operações eficientes de adição e remoção em qualquer ponto da lista.

A utilização de filas, por sua vez, foi estratégica na gestão da prioridade das tarefas. A implementação de uma fila permitiu uma abordagem justa e sequencial para o processamento das tarefas, garantindo que aquelas com prioridade mais alta fossem atendidas primeiro. Essa abordagem de filas agregou uma camada de organização e eficiência ao sistema.

Por fim, a adoção de pilhas de filas proporcionou uma maneira eficaz conseguirmos organizar as tarefas concluídas de forma reversa, possibilitando uma visualização do histórico de atividades e facilitando eventuais revisões.

A combinação harmoniosa de listas duplamente ligadas, filas e pilhas de filas proporcionou um Task Manager eficiente, capaz de lidar com as complexidades do gerenciamento de tarefas de maneira coerente e organizada. Este trabalho reforça a premissa de que a seleção cuidadosa de estruturas de dados desempenha um papel fundamental na eficácia e desempenho de aplicações do mundo real.