# Relatório de Algoritmos e Estrutura de dados

Realizado por: Rúben Alves | Tomás Ferreira

Professor: Fernando Bento

Disciplina: Algoritmo e Estruturas de Dados

Índice

[Relatório de Algoritmos e Estrutura de dados 1](#_Toc187084952)

[Introdução 4](#_Toc187084953)

[ Implementar uma árvore e um grafo de acordo com a componente teórica dada em aula 4](#_Toc187084954)

[ Escolher três estruturas de dados de todas as estruturas de dados que implementamos nas aulas. Desenvolver uma aplicação (software) à nossa escolha que incorpore as três estruturas de dados que escolheu 4](#_Toc187084955)

[ Inventar uma estrutura de dados e referir a sua utilidade face a todas as outras que implementamos em aula 4](#_Toc187084956)

[Árvores e Grafos 5](#_Toc187084957)

[Árvores: 5](#_Toc187084958)

[Grafos 8](#_Toc187084959)

[Estruturas de dados e software desenvolvido 11](#_Toc187084960)

[Objetivos 11](#_Toc187084961)

[Descrição do Projeto 11](#_Toc187084962)

[Funcionalidades Implementadas 12](#_Toc187084963)

[Funcionalidades por Implementar 12](#_Toc187084964)

[Resultados e Conclusões 13](#_Toc187084965)

[Sugestões de Melhoria 13](#_Toc187084966)

[Nova estrutura de dados e utilidade da mesma 14](#_Toc187084967)

[Dificuldades 15](#_Toc187084968)

[Anexos da árvore de tarefas 16](#_Toc187084969)

[16](#_Toc187084970)

[19](#_Toc187084971)

[Conclusão 20](#_Toc187084972)

# Introdução

**Neste documento, estará presente toda a explicação e documentação dos seguintes pontos:**

## Implementar uma árvore e um grafo de acordo com a componente teórica dada em aula

## Escolher três estruturas de dados de todas as estruturas de dados que implementamos nas aulas. Desenvolver uma aplicação (software) à nossa escolha que incorpore as três estruturas de dados que escolheu

## Inventar uma estrutura de dados e referir a sua utilidade face a todas as outras que implementamos em aula

# Árvores e Grafos

Implementamos uma árvore e um grafo, tal e qual como na componente lecionada, fica então presente nos anexos abaixo o código mais importante das **Árvores** e o resultado do mesmo:

## Árvores:

Uma imagem com texto, eletrónica, captura de ecrã, software

Descrição gerada automaticamente

Figura 1 - Inserção dos valores/nós

Uma imagem com texto, captura de ecrã, software, Sistema operativo

Descrição gerada automaticamente

Figura 2- Remoção dos nós e validação do valor dos restantes

Uma imagem com texto, captura de ecrã, software, ecrã

Descrição gerada automaticamente

Figura 3 - Validação para encontrar o nó mínimo e ordenação dos mesmos

Uma imagem com texto, Tipo de letra, captura de ecrã

Descrição gerada automaticamente

Figura 4 - Amostra de resultados da árvore

## Grafos

Uma imagem com texto, eletrónica, captura de ecrã, ecrã

Descrição gerada automaticamente

Figura 5 - Adição de um nó e aresta

Uma imagem com texto, captura de ecrã, software, Sistema operativo

Descrição gerada automaticamente

Figura 6 - Remoção dos nós e arestas

Uma imagem com texto, captura de ecrã, multimédia

Descrição gerada automaticamente

Figura 7 - Inserção de valores

Uma imagem com captura de ecrã, Tipo de letra, texto, tipografia

Descrição gerada automaticamente

Figura 8 - Resultado

# Estruturas de dados e software desenvolvido

O ponto 2 deste trabalho foi dedicado à construção de um editor de texto usando tecnologias web, tendo sido usado o HTML, CSS e JavaScript. O objetivo principal era criar uma aplicação funcional que integrasse pelo menos três estruturas de dados distintas, demonstrando a sua aplicação prática e adequação ao problema proposto. Este relatório descreve o processo de desenvolvimento, as decisões tomadas e os resultados obtidos, com um foco especial nas estruturas de dados implementadas e nas funcionalidades do editor de texto.

## Objetivos

O projeto tinha como objetivo principal o desenvolvimento de um editor de texto funcional, incorporando pelo menos três estruturas de dados diferentes. Os objetivos específicos incluíam:

1. Implementar as estruturas de dados necessárias, construindo-as manualmente.

2. Integrar as estruturas de dados no funcionamento do editor.

3. Desenvolver um conjunto de funcionalidades básicas de edição de texto, como escrita, manipulação de linhas, e gestão do cursor.

4. Garantir que a arquitetura do software fosse modular e escalável para futuras melhorias.

## Descrição do Projeto

O editor de texto foi desenvolvido utilizando quatro estruturas de dados principais:

• Arrays: Implementados para usar em vez da solução padrão oferecida pelo JS.

• DoublyLinkedLists: Implementadas para gerir as operações de criação, remoção e atualização de linhas, bem como para construir o clipboard.

• Stacks: Planeadas para funcionalidades futuras como undo/redo (não implementadas).

• Queues: Utilizadas para capturar e armazenar as teclas inseridas pelo utilizador num buffer.

A aplicação utilizou uma abordagem baseada num Singleton para gerir algumas das estruturas de dados de forma global, permitindo a partilha de dados entre diferentes componentes do sistema.

## Funcionalidades Implementadas

O editor de texto desenvolvido apresenta várias funcionalidades que garantem o seu funcionamento básico. O utilizador pode escrever e apagar texto em qualquer linha, sendo esta a funcionalidade mais elementar do sistema. Para além disso, foi implementado um sistema de gestão de linhas, permitindo criar e remover linhas dinamicamente. Sempre que uma nova linha é adicionada ou eliminada, o número das restantes é atualizado automaticamente, assegurando uma numeração consistente e clara.

A gestão das linhas foi construída sobre uma DoublyLinkedList, que se revelou uma escolha eficiente para este propósito, graças à sua flexibilidade na manipulação de elementos. Outra funcionalidade integrada foi a capacidade de copiar e colar texto, onde o conteúdo copiado é armazenado numa DoublyLinkedList que funciona como clipboard. Esta abordagem permitiu operações rápidas e intuitivas.

Relativamente à interação do utilizador com o editor, o sistema também suporta a captura de teclas, armazenando-as temporariamente num buffer, que foi implementado através de uma Queue. Adicionalmente, o cursor pode ser manipulado tanto pelo utilizador como pelo sistema, ajustando-se automaticamente em resposta às operações realizadas.

## Funcionalidades por Implementar

Embora o editor inclua as funcionalidades essenciais mencionadas, algumas características planeadas não foram concretizadas devido ao tempo limitado. Não foi possível criar um sistema para gerir ficheiros, que permitiria abrir, guardar e editar documentos diretamente. Da mesma forma, o sistema de undo e redo, que teria sido implementado com recurso a pilhas, acabou por não ser incluído.

Outras funcionalidades como a divisão da interface em múltiplas secções editáveis (split views) e a gestão de comandos contextuais (por exemplo, diferenciar as ações de atalhos como CTRL+C no editor e no sistema de ficheiros) também ficaram por implementar. Além disso, o projeto necessitaria de uma última fase de aperfeiçoamento, incluindo polimento visual e ajustes de usabilidade, para atingir o nível de maturidade desejado.

Apesar destas limitações, o editor conseguiu demonstrar, de forma sólida, a integração das estruturas de dados utilizadas e o potencial para evolução futura.

## Resultados e Conclusões

Apesar das limitações impostas pelo tempo e pela necessidade de desenvolver projetos noutras disciplinas, o editor de texto desenvolvido demonstrou a integração prática de várias estruturas de dados e a sua aplicabilidade a problemas reais.

O projeto foi uma experiência enriquecedora, permitindo a aplicação de conceitos teóricos aprendidos na disciplina de AED, bem como o desenvolvimento de competências práticas em programação e design de software. Ficou evidente o potencial para melhorias e expansão do projeto, caso fosse disponibilizado mais tempo.

### Sugestões de Melhoria

Para projetos futuros, seria importante considerar:

1. Planeamento mais detalhado das funcionalidades a implementar.

2. Uso de testes unitários para garantir a robustez das estruturas de dados.

3. Investigação e domínio prévio das bibliotecas externas a utilizar.

# Nova estrutura de dados e utilidade da mesma

**Descrição**:  
Uma árvore binária onde os nós representam tarefas organizadas por data ou prazo. As tarefas mais próximas de expirar ficam mais à esquerda, e as de maior data, à direita.

Ideal para gerir projetos ou sistemas de controlo de prazos, onde é necessário priorizar tarefas com prazos mais próximos

A Árvore Dinâmica de Prazos oferece vantagens quando comparada a outras estruturas de dados, especialmente em cenários que exigem a ordenação e pesquisas rápidas.

**Estrutura**

1. **Nó da Árvore**:
   * Cada nó contém uma tarefa (task) e um prazo (deadline).
   * Cada nó possui dois filhos: o filho esquerdo e o filho direito.
2. **Propriedade de Ordenação**:
   * Os prazos na sub-árvore esquerda de um nó são menores do que o prazo do próprio nó.
   * Os prazos na sub-árvore da direita vão ser maiores.
3. **Operações Fundamentais**:
   * **Inserção (add\_task)**: Adiciona uma tarefa em sua posição correta com base no prazo.
   * **Remoção (remove\_task)**: Remove uma tarefa específica, mantendo a propriedade de ordenação.
   * **Procura pela próxima tarefa (get\_next\_task)**: Encontra a tarefa com o prazo mais próximo.
   * **Exibição (display\_in\_order)**: Mostra as tarefas por ordem crescente de prazo

# Dificuldades

Ao longo deste projeto encontramos diversas dificuldades, das principais obter ideias para o desenvolvimento de alguns pontos, tais como:

* O software a desenvolver com 3 dos métodos usados nas atividades letivas
* E a criação de uma “nova” estrutura de dados, visto que já existem milhares estruturas.

Tivemos de pensar em equipa e fazer várias pesquisas para verificar o que se adequava melhor ao desenvolvimento desses pontos.

No desenvolvimento do software (ponto 2), a maior dificuldade encontrada foi a inexperiência com as bibliotecas do JavaScript utilizadas para manipulação de texto e cursores, o que atrasou o desenvolvimento de algumas funcionalidades. Em contrapartida, funcionalidades independentes, como o sistema de gestão de linhas, foram desenvolvidas de forma rápida e eficiente.

Na criação da nova estrutura de dados tivemos de averiguar todas as estruturas já estudadas, fazer pesquisas e discutir em grupo para saber o que realmente mais se adequava.

Acabámos para fazer a árvore dinâmica para gerir tarefas e as datas das mesmas, achamos um assunto pertinente e bastante interessante, e que utiliza tambem técnicas usadas nas atividades letivas

# Anexos da árvore de tarefas

# Uma imagem com texto, captura de ecrã, multimédia, software Descrição gerada automaticamente

Figura 9 - Inserção na árvore

Uma imagem com texto, eletrónica, captura de ecrã, software

Descrição gerada automaticamente

Figura 10 - Remover da árvore

Uma imagem com texto, captura de ecrã, ecrã, multimédia

Descrição gerada automaticamente

Figura 11 - Verificar a próxima tarefa

Uma imagem com texto, captura de ecrã, ecrã, multimédia

Descrição gerada automaticamente

Figura 12 - Mostrar por ordem as tarefas

Uma imagem com texto, eletrónica, captura de ecrã, software

Descrição gerada automaticamente

Figura 13 - Menu

# Uma imagem com texto, captura de ecrã, Tipo de letra Descrição gerada automaticamente

Figura 14 - Exemplo de funcionamento

# Conclusão

Podemos concluir que o desenvolvimento deste projeto foi desafiador, desde a conceção de ideias até a sua implementação. No entanto, graças à boa comunicação e ao trabalho em equipa, conseguimos atingir os objetivos propostos. Este projeto não apenas nos proporcionou uma oportunidade de aprendizagem significativa, como também contribuiu para enriquecer o nosso conhecimento e aprimorar a nossas competências.

Estamos confiantes de que as experiências adquiridas ao longo deste processo serão fundamentais para o nosso desenvolvimento profissional e nos permitirão enfrentar futuros desafios com ainda mais competência e confiança.