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

Descrição gerada automaticamente

**Instituto Politécnico do Cávado e do Ave**

**Escola Superior de Tecnologia**

**Licenciatura em**

**Engenharia de Sistemas Informáticos**

**Trabalho Prático I**

Fábio Rafael Gomes Costa

**Barcelos, outubro de 2025**

****

**Instituto Politécnico do Cávado e do Ave**

**Escola Superior de Tecnologia**

**Licenciatura em**

**Engenharia de Sistemas Informáticos**

**Trabalho Prático I**

Fábio Rafael Gomes Costa – a22997

**Unidade Curricular:**

Integração de Sistemas de Informação (ISI)

**Docente:**

Dr. Óscar Rafael da Silva Ferreira Ribeiro

**Barcelos, outubro de 2025**

# Ficha de Identificação

|  |  |
| --- | --- |
| **Elaborado por** | Fábio Rafael Gomes Costa |
| **Contato** | a22997@alunos.ipca.pt |
|  |  |
| **Unidade Curricular** | Integração de Sistemas de Informação |
| **Curso** | Licenciatura em Engenharia de Sistemas Informáticos |
| **Instituição** | Escola Superior de Tecnologia do  Instituto Politécnico do Cávado e do Ave |
| **Professor** | Doutor Óscar Rafael da Silva Ferreira Ribeiro |
| **Gabinete** | 3 |
| **Contato** | oribeiro@ipca.pt |
|  |  |
| **Data de início** | 5 de maio de 2025 |
| **Data de conclusão** | 18 de maio de 2025 |

# Resumo

O presente trabalho insere-se no âmbito da Unidade Curricular de Estruturas de Dados Avançadas (EDA) e tem como objetivo principal a aplicação prática dos conhecimentos adquiridos, através do desenvolvimento de uma solução que permita representar e manipular dados referentes a antenas e suas localizações numa cidade, utilizando estruturas de dados dinâmicas baseadas em grafos, implementadas em linguagem C.

Na Fase 2 do projeto, a estrutura de dados foi reformulada, substituindo as listas ligadas simples da fase anterior por um grafo representado por listas de adjacência, onde cada vértice corresponde a uma antena e cada aresta a uma ligação entre antenas de mesma frequência. Além disso, foi incorporada a lógica para armazenar as zonas nefastas diretamente nas arestas, sempre que duas antenas criassem interferência com base na sua posição espacial.

Foram implementados algoritmos clássicos de travessia em profundidade (DFS) e largura (BFS), assim como a funcionalidade para determinar todos os caminhos possíveis entre duas antenas. A construção e manipulação do grafo permitiram aprofundar o conhecimento sobre estruturas ligadas, algoritmos de grafos e análise de padrões espaciais.

O desenvolvimento da fase decorreu num ambiente controlado, com recurso ao Visual Studio, e todo o projeto foi registado através da plataforma GitHub. A documentação técnica foi produzida com o auxílio da ferramenta Doxygen, assegurando a clareza e organização do código-fonte.

# Índice

[Ficha de Identificação I](#_Toc198302057)

[Resumo II](#_Toc198302058)

[Índice III](#_Toc198302059)

[Índice de Fragmentos de Código IV](#_Toc198302060)

[Índice de Figuras V](#_Toc198302061)

[Lista de Siglas e Acrônimo V](#_Toc198302062)

[1. Introdução 1](#_Toc198302063)

[1.1 Contextualização 1](#_Toc198302064)

[1.2 Motivação/Pretensões e Objetivos 1](#_Toc198302065)

[1.3 Metodologia de Trabalho 2](#_Toc198302066)

[1.4 Ferramentas e Tecnologias Utilizadas 2](#_Toc198302067)

[1.5 Estrutura do Documento 2](#_Toc198302068)

[2. Fundamentos Teóricos 4](#_Toc198302071)

[2.1 Listas Ligadas 4](#_Toc198302072)

[2.2 Grafos e Listas de Adjacência 4](#_Toc198302073)

[2.3 Algoritmos de Travessia (DFS e BFS) 5](#_Toc198302074)

[2.4 Backtracking para todos os caminhos 5](#_Toc198302075)

[3. Problemas 6](#_Toc198302076)

[4. Resolução dos Problemas 10](#_Toc198302077)

[4.1 Estrutura do grafo (Problema 1) 10](#_Toc198302078)

[4.1.1 Antenas/Vértices 10](#_Toc198302079)

[4.1.2 Arestas e Zonas Nefastas 10](#_Toc198302080)

[4.1.3 Grafo 11](#_Toc198302081)

[4.2 Carregamento de dados (Problema 2) 12](#_Toc198302082)

[4.2.1 Função LerLista 12](#_Toc198302083)

[4.2.2 Função InserirAntena 13](#_Toc198302084)

[4.2.3 Função CriarListaArestas 14](#_Toc198302085)

[4.3 Procura em profundidade DFS (Problema 3.a.) 17](#_Toc198302086)

[4.3.1 Função DFS 17](#_Toc198302087)

[4.3.2 Função IniciarDFS 18](#_Toc198302089)

[4.4 Procura em largura BFS (Problema 3.b.) 19](#_Toc198302090)

[4.4.1 Função BFS 19](#_Toc198302091)

[4.5 Caminhos possíveis entre antena (Problemas 3.c.) 21](#_Toc198302092)

[4.5.1 Função EncontrarCaminhos 21](#_Toc198302093)

[4.5.2 Função TodosCaminhos 22](#_Toc198302094)

[5. Repositório GitHub 23](#_Toc198302135)

[6. Considerações Finais 24](#_Toc198302136)

[7. Referencias bibliográfica 25](#_Toc198302137)

# Índice de Fragmentos de Código

[Código 1 - struct Ant 10](#_Toc198302046)

[Código 2 - struct Ars 11](#_Toc198302047)

[Código 3 - struct Grafo 11](#_Toc198302048)

[Código 4 - Função LerLista 13](#_Toc198302049)

[Código 5 - Função InserirAntena 14](#_Toc198302050)

[Código 6 - Função CriarListaArestas 16](#_Toc198302051)

[Código 7 - Função DFS 18](#_Toc198302052)

[Código 8 - Função IniciarDFS 18](#_Toc198302053)

[Código 9 - Função BFS 20](#_Toc198302054)

[Código 10 - Função EncontrarCaminhos 21](#_Toc198302055)

[Código 11 - Função TodosCaminhos 22](#_Toc198302056)

# Índice de Figuras

[Figura 1 - Matriz Problema I 6](#_Toc198302038)

[Figura 2 - Matriz Problema II 7](#_Toc198302039)

[Figura 3 - Matriz Problema III 7](#_Toc198302040)

[Figura 4 - Matriz Problema IV 8](#_Toc198302041)

[Figura 5 - Matriz Problema V 9](#_Toc198302042)

[Figura 6 - Arestas e Nefastos 11](#_Toc198302043)

[Figura 7 - DFS – Depth-First Search 17](#_Toc198302044)

[Figura 8 - BFS – Breadth-First Search 19](#_Toc198302045)

# Lista de Siglas e Acrônimo

IPCA – Instituto Politécnico do Cavado e do Ave

UC – Unidade Curricular

VS – Visual Studio 2022

# Introdução

## Contextualização

No âmbito da Unidade Curricular (UC) de Estruturas de Dados Avançadas (EDA), inserida no 2º semestre do 1º ano do curso, foi-me atribuída a realização de um projeto prático como instrumento de avaliação, sob a orientação do docente Dr. Luís G. Ferreira. Este projeto tem como principal objetivo a aplicação e consolidação dos conhecimentos adquiridos ao longo do semestre, através da implementação e manipulação de estruturas de dados dinâmicas na linguagem de programação C.

## Motivação/Pretensões e Objetivos

O desenvolvimento da Fase 2 deste projeto tem como principal motivação o aprofundamento do estudo e aplicação de estruturas de dados do tipo grafo, uma das mais relevantes na modelação de sistemas complexos. Pretende-se consolidar o conhecimento sobre a representação de grafos por listas de adjacência, integrando esse modelo com conceitos previamente abordados, como listas ligadas e modularização de código.

Os principais objetivos desta fase são:

* Implementar algoritmos clássicos de travessia, como DFS (Depth-First Search) e BFS (Breadth-First Search);
* Desenvolver mecanismos para identificar todos os caminhos possíveis entre duas antenas, recorrendo a backtracking;
* Reforçar a capacidade de resolução de problemas de conectividade, acessibilidade e propagação em redes.

Adicionalmente, esta fase promove boas práticas de desenvolvimento, como:

* Organização modular do código;
* Documentação técnica com Doxygen;
* Utilização eficiente da memória através de vetores e estruturas dinâmicas;
* Aplicação de estruturas e algoritmos a dados espaciais em formato matricial.

## Metodologia de Trabalho

Nesta segunda fase, foi mantida uma abordagem faseada e incremental, dando continuidade à estrutura desenvolvida anteriormente. Inicialmente, procedeu-se à análise do enunciado da fase 2, com o objetivo de identificar as novas exigências técnicas, nomeadamente a representação do grafo, o cálculo das zonas nefastas, e a implementação de algoritmos de percursos.

O código foi organizado de forma modular e extensível, garantindo compatibilidade com as estruturas já existentes. A implementação do grafo por lista de adjacência foi integrada diretamente à estrutura das antenas, permitindo uma representação eficiente das conexões entre vértices.

Os algoritmos de DFS, BFS e backtracking foram desenvolvidos individualmente, testados de forma isolada, e depois integrados no sistema global. Durante o processo, foram realizados testes frequentes para garantir o correto funcionamento das travessias e a coerência dos caminhos identificados.

A documentação foi atualizada com o auxílio do Doxygen, mantendo a clareza da arquitetura do código e facilitando a compreensão das funções envolvidas. Todo o desenvolvimento foi acompanhado com controlo de versões através do GitHub.

## Ferramentas e Tecnologias Utilizadas

No desenvolvimento do projeto, utilizei a linguagem de programação C, utilizada no contexto de sala de aula. O *Visual Studio* foi o ambiente escolhido pela sua interface intuitiva e funcionalidades de depuração. Para o controlo de versões, recorreu-se ao *GitHub*, garantindo o acompanhamento e preservação do código. A documentação foi gerada automaticamente com o *Doxygen*, a partir de comentários no código. Finalmente, o *Microsoft Word* foi utilizado para a redação do relatório.

## Estrutura do Documento

Este relatório foi organizado de forma a apresentar o trabalho desenvolvido de maneira clara e sequencial. Inicia-se com a capa, subcapa e Ficha de Identificação, seguida do Resumo e do Índice, que inclui tanto os tópicos principais como o Índice de Figuras, o Índice de Fragmentos de Código e a Lista de Siglas e Acrónimos.

A Introdução apresenta o enquadramento geral do projeto, incluindo a contextualização, os objetivos, a metodologia e as ferramentas utilizadas.

Seguindo a introdução, encontra-se o Enquadramento Teórico e Prático, onde são discutidos os fundamentos teóricos.

Na secção seguinte, Problemas, são identificados e descritos os problemas enfrentados durante o desenvolvimento do projeto.

O Desenvolvimento do projeto está dividido em duas fases: a resolução dos problemas antes da defesa, detalhada na Resolução dos Problemas (Antes da Defesa), e as modificações feitas após a defesa, descritas em Resolução dos Problemas (Pós-Defesa). Cada problema é analisado individualmente, com as respetivas soluções apresentadas nas subsecções de Resolução do Problema 1, Resolução do Problema 2, e assim por diante, até o Problema 3.d.

A seguir, encontra-se o Repositório GitHub, onde é apresentado o endereço do repositório do projeto.

O relatório é concluído com as Considerações Finais, onde são apresentadas as reflexões sobre o trabalho realizado, as aprendizagens obtidas e sugestões de melhoria.

Por fim, são apresentadas as Referências Bibliográficas, onde se encontram listados todos os links consultados durante o desenvolvimento do projeto.

Enquadramento — identificação da equipa, Unidade Curricular e Curso.​

Problema — descrição do que se pretende demonstrar e/ou resolver.​

Estratégia utilizada — operadores e processos envolvidos.​

Transformações — diagramas e explicação.​

Jobs — diagramas e explicação.​

Vídeo com demonstração (acesso via QR Code).​

Conclusão e trabalhos futuros.​

Referências bibliográficas, incluindo referências web relevantes.

De acordo com o documento, o relatório final do trabalho deve documentar todo o trabalho desenvolvido e incluir os seguintes tópicos:

1. **Enquadramento**
   * Equipa, Unidade Curricular, Curso
2. **Problema**
   * Descrição do que se pretende demonstrar e/ou resolver
3. **Estratégia utilizada**
   * Que operadores e processos estarão envolvidos
4. **Transformações**
   * Diagramas
   * Explicação
5. **Jobs**
   * Diagramas
   * Explicação
6. **Vídeo com demonstração (QR Code)**
7. **Conclusão e Trabalhos futuros**
8. **Referências bibliográficas utilizadas**, incluindo referências web que tenham sido relevantes.

# Repositório GitHub

O repositório *GitHub* com o projeto de Estruturas de Dados e Algoritmos (EDA), a sua documentação, o enunciado do projeto, e o seu relatório pode ser acedido através do seguinte link:

<https://github.com/fabiocosta191/Projeto_EDA>

Neste repositório, poderão ser consultados todos os detalhes e documentação relacionados com a primeira (antes e pós correções sugeridas na defesa) e segunda fase do projeto, incluindo as análises, implementações e conclusões até ao momento.

# Considerações Finais

Ao longo da realização desta segunda fase do projeto prático, aprofundei significativamente a minha compreensão sobre a implementação e manipulação de grafos dinâmicos em linguagem C. A transição de estruturas simples, como listas ligadas (exploradas na Fase 1), para uma estrutura mais complexa baseada em listas de adjacência, representou um desafio importante, sobretudo ao nível da organização dos dados, da gestão eficiente da memória e da integração modular do código.

A implementação de operações como a criação de arestas entre antenas, o armazenamento de zonas nefastas, e a execução de algoritmos de travessia como DFS, BFS e backtracking, permitiu-me consolidar conhecimentos fundamentais sobre grafos e perceber como a estruturação correta dos dados influencia diretamente a eficiência e legibilidade da solução.

A necessidade de determinar todos os caminhos possíveis entre duas antenas destacou ainda mais a importância de raciocínios lógicos e estruturados, especialmente no uso de algoritmos recursivos com backtracking, reforçando a minha capacidade de resolver problemas com múltiplas ramificações e estados.

Este projeto contribuiu também para o reforço de boas práticas de desenvolvimento, tais como:

* A modularização do código;
* A documentação clara através do Doxygen;
* O controlo de versões com GitHub, essencial para o acompanhamento da evolução do trabalho.

De forma geral, considero que este projeto foi uma excelente oportunidade para aplicar e consolidar os conteúdos abordados na unidade curricular de Estruturas de Dados Avançadas, não só cumprindo os objetivos propostos, mas também me preparando para problemas mais complexos no âmbito da programação, da análise de algoritmos e da modelação de sistemas.

A experiência adquirida nesta fase será uma mais-valia para o meu percurso académico e profissional, dotando-me de ferramentas sólidas para abordar desafios relacionados com redes, estruturas interligadas e otimização de caminhos.

# Referencias bibliográfica