



Relatório Trabalho Prático de EDA

Nome: Leonardo Kusnierz da Costa

Nº 31528 – Regime diurno

Professor: Luís Ferreira

Disciplina : Estrutura de dados avançados

Ano Letivo: 2024/2025

Licenciatura em Engenharia de Sistemas informáticos

Escola Superior de tecnologia

Instituto Politécnico do Cávado e Ave



**INSTITUTO POLITÉCNICO
DO CÁVADO E DO AVE
ESCOLA SUPERIOR
DE TECNOLOGIA**

Conteúdo

Conteúdo.....	2
1. Resumo	4
2. Introdução	5
2.1. Contextualização e Motivação	5
2.2. Enquadramento Académico	5
2.3. Objetivos do Projeto.....	6
2.4. Metodologia de Trabalho	6
3. Enquadramento Teórico e Prático	7
3.1. Conceitos Teóricos e Tecnologias a Explorar	7
3.2. Soluções Existentes e Comparação	8
4. Definição do Problema	9
4.1. Contexto do Problema	9
4.2. Requisitos Funcionais.....	9
4.3. Ferramentas Utilizadas	10
4.3.1 Linguagem C.....	10
4.3.2 Visual Studio Code	10
4.3.3 Windows Subsystem for Linux (WSL).....	10
4.3.4 Inteligência Artificial – ChatGPT.....	10
4.3.5 Doxygen.....	10
4.3.6 WinRAR	10
4.3.7 Microsoft Word.....	10
4.3.8 Inteligência Artificial – MyMap.....	10
5. Modelação da Solução	11
5.1. Representação das Antenas com Listas Ligadas	11
5.2. Estruturação em Grafo.....	12
5.3. Estratégias de Exploração	13
6. Implementação e Testes.....	14
6.1. Organização Modular.....	14

6.2. Resultados Obtidos.....	15
6.3. Desafios Encontrados.....	15
7. Conclusão	16
8. Bibliografia.....	17
9. Índice de Figuras.....	18

1. Resumo

Este projeto visa o desenvolvimento de uma aplicação em C para a gestão de uma rede de antenas de ressonância distribuídas numa cidade, focando-se na deteção de zonas com possível efeito nefasto. A partir de um ficheiro de texto representando uma matriz urbana, construiu-se uma estrutura de dados dinâmica baseada em listas ligadas e grafos, capaz de armazenar vértices (antenas), as suas ligações (arestas) e frequências de funcionamento.

Foram implementadas funcionalidades como inserção e remoção de antenas e ligações, ligação automática entre antenas com a mesma frequência, cálculo de efeitos nefastos, geração de ficheiros de saída (texto e binário), e percursos no grafo via BFS e DFS, permitindo detetar caminhos e analisar conectividade.

O projeto conta com uma estrutura modular separando cabeçalhos, implementação e função principal, e foi completamente documentado com recurso à ferramenta Doxygen, assegurando clareza, manutenção e reaproveitamento do código. A solução reflete de forma prática os conceitos aprendidos na unidade curricular de Estruturas de Dados Avançadas, com uma aplicação funcional, bem estruturada e orientada à resolução de problemas reais.

2. Introdução

Neste capítulo será apresentado o contexto do problema que motivou a realização deste projeto, bem como a sua importância atual no domínio das redes de comunicação. Serão abordados os principais objetivos traçados, a metodologia aplicada durante o desenvolvimento, e ainda o plano de trabalho que guiou as diferentes etapas do projeto.

2.1. Contextualização e Motivação

As redes de comunicação são hoje essenciais para o funcionamento das cidades inteligentes e para a ligação entre diversos dispositivos e serviços. À medida que o número de antenas aumenta, torna-se muito importante gerir bem as frequências usadas para evitar interferências que possam afetar o desempenho dessas redes. Essas interferências surgem principalmente quando antenas próximas operam na mesma frequência, causando o chamado efeito nefasto. Este fenómeno pode prejudicar a qualidade de vários serviços, desde comunicações móveis até redes de sensores, tornando fundamental a criação de métodos que permitam identificar e analisar estes problemas. Apesar de já existirem soluções comerciais para este tipo de situação, desenvolver uma abordagem simples, flexível e fácil de implementar é uma vantagem para melhor compreender e estudar este fenómeno.

2.2. Enquadramento Académico

Este trabalho foi desenvolvido no âmbito da Unidade Curricular de Estruturas de Dados Avançadas, integrada no curso de Engenharia Informática da Escola Superior de Tecnologia do Instituto Politécnico do Cávado e do Ave (EST-IPCA).

Realizado no segundo semestre do primeiro ano, o projeto baseia-se em conceitos fundamentais como estruturas dinâmicas, gestão de memória, modularização de código e manipulação de ficheiros, com o objetivo de aplicar estes conhecimentos na construção de uma solução funcional, eficiente e bem documentada.

2.3. Objetivos do Projeto

Os objetivos que guiaram este projeto incluem:

- Construir uma estrutura de dados dinâmica, utilizando listas ligadas, para representar a rede de antenas;
- Desenvolver algoritmos capazes de identificar zonas afetadas por interferências devido à sobreposição de frequências entre antenas;
- Expandir a modelação para a utilização de grafos, permitindo uma análise mais abrangente das relações e ligações entre antenas;
- Assegurar a modularidade do código e a geração automática da documentação através da ferramenta Doxygen;
- Criar uma interface por linha de comandos que permita a importação de dados, manipulação das estruturas e visualização clara dos resultados obtidos.

2.4. Metodologia de Trabalho

O processo de desenvolvimento seguiu uma abordagem faseada e iterativa, composta por duas etapas principais:

Implementação inicial das antenas e deteção dos efeitos nefastos com recurso a listas ligadas;

Evolução do modelo para grafos, facilitando a análise da conectividade entre as antenas e a identificação de caminhos possíveis.

Cada fase envolveu a implementação modular das funcionalidades, testes para validação do código e documentação contínua, que foi assegurada através da integração da ferramenta Doxygen desde as fases iniciais do desenvolvimento.

3. Enquadramento Teórico e Prático

Este capítulo pretende apresentar os conceitos básicos e as técnicas usadas no desenvolvimento do projeto, explicando as ideias por trás das estruturas de dados e das representações que suportam a análise das antenas e das suas interações. Também se faz uma comparação entre as abordagens existentes e a metodologia adotada no projeto.

3.1. Conceitos Teóricos e Tecnologias a Explorar

O trabalho baseia-se essencialmente em três áreas principais: uso de estruturas dinâmicas para armazenar dados, representação das ligações entre antenas por meio de grafos, e análise geométrica para identificar interferências.

As listas ligadas foram escolhidas para guardar as antenas e suas posições porque permitem inserir, remover ou procurar elementos facilmente, mesmo quando o número de antenas varia ao longo do tempo. Esta flexibilidade torna-as ideais para o nosso modelo.

Para representar as conexões entre antenas que funcionam na mesma frequência, foi utilizado o conceito de grafos, com listas de adjacência. Este método facilita a execução de algoritmos que exploram caminhos e relações entre antenas, ajudando a identificar grupos ou conexões importantes.

A deteção do chamado efeito nefasto baseia-se na análise da posição das antenas no espaço. Calcula-se, por exemplo, se o ponto entre duas antenas está numa posição que possa causar interferência, usando cálculos simples com as suas coordenadas.

Para garantir que o código está bem documentado, usamos a ferramenta Doxygen, que permite gerar documentação técnica diretamente a partir do código fonte. Isto ajuda a manter o projeto organizado e facilita a sua compreensão por outros programadores ou avaliadores.

3.2. Soluções Existentes e Comparação

A análise de interferências entre antenas é um tema comum em áreas como telecomunicações, planejamento urbano e redes de sensores. Muitas soluções existentes utilizam linguagens de programação de alto nível, como Python ou Java, que simplificam o desenvolvimento, mas escondem muitos detalhes sobre o funcionamento interno das estruturas de dados.

No setor profissional, são usados softwares especializados para mapear e analisar redes de comunicação, com interfaces gráficas avançadas. No entanto, o nosso projeto destaca-se por ser implementado em linguagem C, o que permite um maior controle sobre a gestão de memória e os detalhes da implementação, tornando possível uma compreensão mais profunda dos processos envolvidos.

4. Definição do Problema

Neste capítulo será apresentada uma descrição clara e detalhada do problema que o projeto pretende resolver. Iremos expor o contexto onde o problema surge, as limitações a considerar e os requisitos funcionais necessários para o desenvolvimento da solução. A explicação será feita de forma progressiva, acompanhando as etapas e desafios que foram ultrapassados durante a implementação. Este enquadramento ajuda a clarificar os objetivos do projeto e a validar as escolhas técnicas adotadas.

4.1. Contexto do Problema

Num ambiente urbano, existe uma rede de antenas distribuídas geograficamente, cada uma a operar numa frequência específica. Quando várias antenas próximas utilizam a mesma frequência, podem ocorrer interferências que prejudicam a qualidade do sinal, originando zonas problemáticas designadas por áreas com efeito nefasto. A identificação e gestão destas zonas é essencial em sistemas de comunicação críticos. Para tal, o modelo computacional deve representar a cidade através de uma estrutura dinâmica, onde cada antena possui atributos como a frequência, as coordenadas X e Y, e as ligações a outras antenas. A análise do comportamento coletivo destas antenas requer percorrer essa estrutura, verificar a relação espacial entre pares de antenas e interpretar o impacto da sua disposição.

4.2. Requisitos Funcionais

A solução desenvolvida deverá assegurar as seguintes funcionalidades:

Leitura de ficheiros contendo a matriz de antenas e transformação dessa informação numa estrutura dinâmica de dados;

Capacidade de adicionar e remover antenas, mantendo a organização e integridade dos dados;

Identificação automática de zonas com efeito nefasto, com base em critérios geométricos previamente definidos;

Representação das antenas como vértices num grafo, com ligações apenas entre antenas que partilham a mesma frequência;

Implementação de algoritmos de busca em profundidade (DFS) e em largura (BFS) para encontrar possíveis caminhos entre antenas;

Apresentação de todos os caminhos existentes entre dois vértices;

Deteção de pares de antenas com frequências distintas que ocupam a mesma área lógica (interseções).

4.3. Ferramentas Utilizadas

Para o desenvolvimento deste projeto, foram utilizadas várias ferramentas que facilitaram cada etapa do trabalho, contribuindo para a sua conclusão de forma eficiente.

4.3.1 Linguagem C

A implementação foi feita em linguagem C, escolhida pela sua capacidade de controlo direto da memória e eficiência na manipulação de estruturas dinâmicas.

4.3.2 Visual Studio Code

O Visual Studio Code foi o editor de código utilizado, graças à sua interface leve, extensibilidade e suporte para desenvolvimento em C/C++, bem como à possibilidade de executar comandos diretamente no terminal integrado.

4.3.3 Windows Subsystem for Linux (WSL)

O WSL permitiu a utilização de um ambiente Linux dentro do Windows, facilitando a compilação com GCC e o debugging com GDB, ferramentas essenciais para este projeto.

4.3.4 Inteligência Artificial – ChatGPT

A ferramenta ChatGPT foi usada como apoio para esclarecer dúvidas, ajudar no debug e melhorar a estrutura do código.

4.3.5 Doxygen

O Doxygen foi utilizado para criar automaticamente documentação técnica a partir dos comentários no código-fonte, garantindo uma documentação clara e organizada.

4.3.6 WinRAR

O WinRAR serviu para compactar ficheiros, permitindo-me transportar facilmente o projeto entre o meu computador principal e o portátil, devido à necessidade de deslocações frequentes.

4.3.7 Microsoft Word

O relatório final do projeto foi elaborado no Microsoft Word, aproveitando as suas ferramentas para uma formatação cuidada e apresentação profissional.

4.3.8 Inteligência Artificial – MyMap

A ferramenta MyMap ajudou a organizar e estruturar as ideias através de mapas mentais, facilitando a visualização do fluxo do código e a compreensão global do projeto.

5. Modelação da Solução

Neste capítulo apresenta-se o processo de desenvolvimento e estruturação da solução técnica adotada. Optou-se por uma abordagem modular, com uma clara divisão em ficheiros distintos, focada em garantir uma representação eficiente dos dados através de estruturas dinâmicas. Foram aplicados conceitos essenciais, tais como listas ligadas, grafos e algoritmos de busca em profundidade e em largura. A modelação progrediu de forma gradual, iniciando-se pela representação das antenas na cidade e avançando para a análise das suas interligações e possíveis interferências.

5.1. Representação das Antenas com Listas Ligadas

As antenas são os elementos básicos da solução. Cada antena é definida por uma frequência, representada por um carácter, e pela sua posição em coordenadas X e Y num plano bidimensional. Para armazenar e gerir estas informações, optou-se pela utilização de uma lista ligada simples, devido às suas vantagens, nomeadamente:

Permitir a inserção ordenada dos elementos segundo as suas coordenadas, facilitando a organização espacial;

Garantir a remoção eficiente de nós, assegurando a manutenção da integridade da estrutura;

Possibilitar a navegação sequencial para análise e deteção de padrões ou condições específicas entre as antenas.

Cada elemento da lista é uma estrutura do tipo Antena, que contém os campos para a frequência, as coordenadas e um apontador para o próximo elemento da lista.

A Figura 1 apresenta um diagrama esquemático do grafo que agrega todas as antenas da área urbana, elaborado com o suporte da ferramenta de inteligência artificial MyMap.

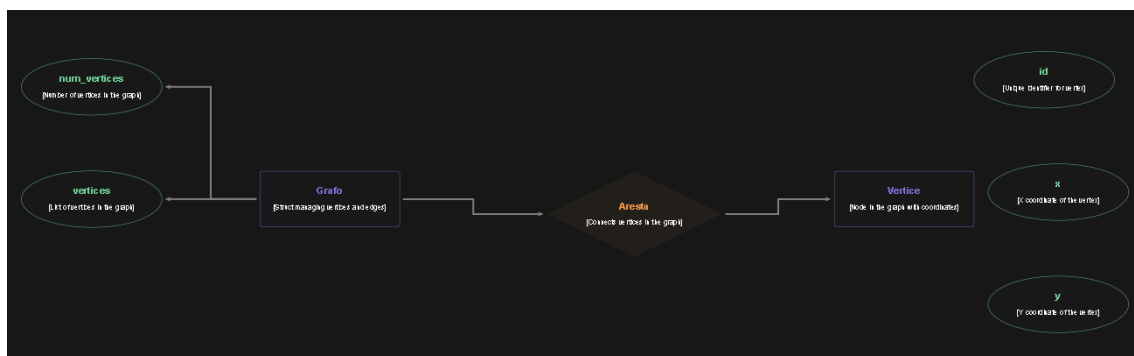


Figura 1 esquema da estrutura do grafo

5.2. Estruturação em Grafo

Para permitir a análise da interligação entre as antenas, desenvolveu-se uma representação baseada em grafos. Neste modelo, cada antena assume o papel de vértice (designado como Vértice), e cada possível ligação determinada pela coincidência de frequência entre antenas é modelada como uma aresta (Aresta).

A estrutura do grafo foi implementada através de uma lista ligada de vértices, sendo que cada vértice armazena:

As informações específicas da antena (frequência, coordenadas X e Y);

Uma lista ligada de arestas que aponta para os vértices com os quais a antena está diretamente conectada.

Esta abordagem permite a inserção dinâmica de novas antenas e a criação de ligações entre elas sem limites fixos de dimensão, oferecendo flexibilidade e escalabilidade.

A Figura 2 ilustra a interconexão entre antenas que operam na mesma frequência, demonstrando visualmente a estrutura do grafo resultante.

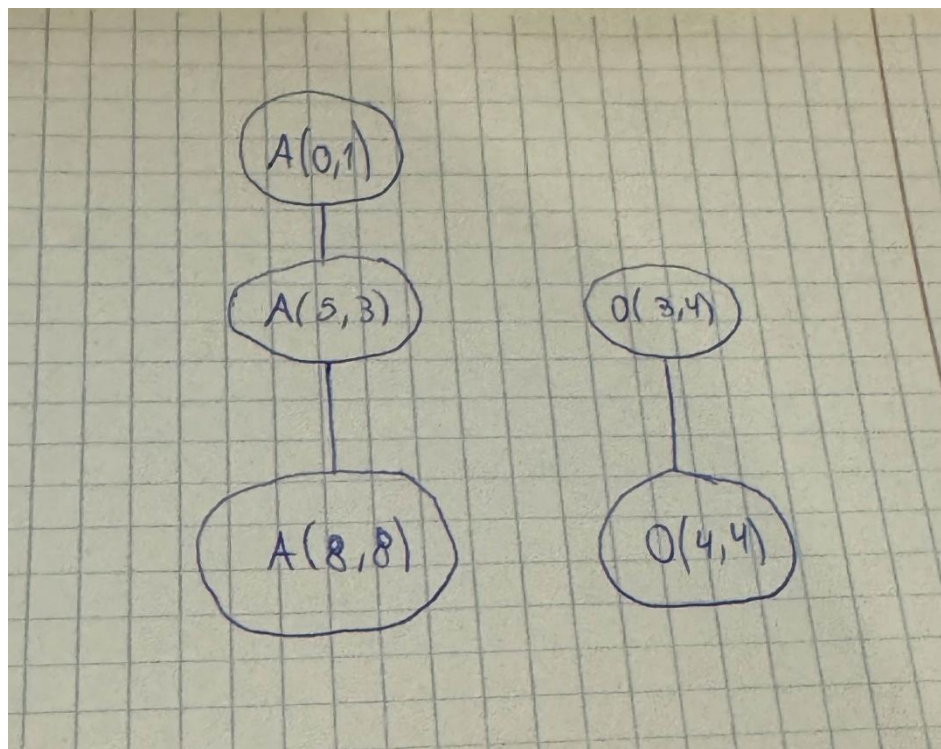


Figura 2 conexão de antenas

5.3. Estratégias de Exploração

Com o intuito de percorrer e analisar o grafo criado, foram implementadas duas abordagens clássicas de exploração:

Pesquisa em profundidade (DFS – Depth-First Search): aplicada para investigar percursos mais extensos entre antenas, sendo particularmente eficaz na verificação da conectividade total entre nós.

Pesquisa em largura (BFS – Breadth-First Search): utilizada para encontrar caminhos mais curtos em termos de número de ligações entre antenas, permitindo identificar rotas de comunicação mais eficientes.

Ambos os algoritmos operam diretamente sobre listas ligadas e foram estruturados de acordo com os princípios estabelecidos por Wayne (2011).

Para garantir que os vértices não sejam visitados repetidamente durante a execução dos algoritmos, foram utilizadas listas auxiliares baseadas nas coordenadas das antenas, assegurando uma marcação eficaz dos elementos já percorridos.

6. Implementação e Testes

Este capítulo apresenta o percurso seguido na implementação da solução desenvolvida, desde a estruturação modular do código-fonte até à verificação funcional de cada componente. A abordagem incluiu a realização de testes específicos para assegurar o comportamento esperado das funcionalidades essenciais, nomeadamente: leitura de ficheiros, manipulação de estruturas dinâmicas (listas e grafos), procura de caminhos entre antenas e identificação de zonas com interferência.

6.1. Organização Modular

A implementação da solução foi planeada com uma arquitetura modular, de forma a garantir clareza, manutenibilidade e separação de responsabilidades. A estrutura do projeto baseia-se em três ficheiros principais, cada um com uma função bem definida no desenvolvimento e teste da aplicação:

Main.c - Contém a função principal (main) e é responsável pela orquestração geral da execução do programa.

Functest.c - Implementa as funcionalidades principais do sistema, incluindo manipulação de listas ligadas, construção do grafo, e algoritmos de procura e análise.

Functest.h - Declara as estruturas de dados, constantes e protótipos das funções implementadas em **functest.c**, permitindo a sua reutilização e inclusão no **main.c**.

Esta organização modular permitiu uma melhor gestão do código-fonte, facilitando a identificação de erros, a realização de testes individuais por componente e a posterior integração de todas as funcionalidades num sistema coeso.

6.2. Resultados Obtidos

A implementação do sistema revelou-se bem-sucedida em todos os módulos desenvolvidos. Verificou-se que as funcionalidades previstas operaram conforme o esperado, respeitando os princípios de modularidade, desempenho e legibilidade definidos na fase de modelação.

Os algoritmos de procura em profundidade (DFS) e em largura (BFS) apresentaram uma execução eficiente para o volume de dados utilizado nos testes, conseguindo explorar as estruturas de grafo com precisão e rapidez.

Adicionalmente, a funcionalidade de deteção de zonas com efeito nefasto demonstrou-se fiável, identificando corretamente as áreas onde ocorriam interferências, de acordo com os critérios geométricos estabelecidos. Estes resultados confirmam a robustez das estruturas de dados dinâmicas e a adequação da lógica implementada à resolução do problema proposto.

6.3. Desafios Encontrados

Um dos principais obstáculos enfrentados durante o desenvolvimento do projeto prendeu-se com a interpretação detalhada do enunciado, em especial no que respeita à definição e deteção de zonas com efeito nefasto. Esta componente exigiu um estudo rigoroso das condições geométricas envolvidas, bem como a formulação precisa de critérios que pudessem ser traduzidos eficazmente para código.

Para além disso, a gestão do tempo e do planeamento das tarefas constituiu outro desafio significativo. A complexidade progressiva do projeto exigiu uma reestruturação contínua da abordagem, de forma a assegurar que cada fase do desenvolvimento fosse concluída com sucesso.

7. Conclusão

Este projeto representou uma oportunidade valiosa para consolidar os conhecimentos adquiridos na unidade curricular de Estruturas de Dados Avançadas, permitindo aplicar, de forma concreta, estruturas dinâmicas em linguagem C para resolver um problema inspirado num contexto realista de análise de redes de antenas.

Ao longo do desenvolvimento, foi possível explorar profundamente o uso de listas ligadas simples, que serviram de base para representar as antenas, as ligações entre elas, os caminhos de procura e os locais com efeito nefasto. A escolha desta estrutura permitiu garantir flexibilidade na gestão dos dados, sem limitação de tamanho ou número fixo de elementos.

A construção de um grafo dinâmico onde cada antena corresponde a um vértice e as ligações são condicionadas à igualdade de frequência revelou-se eficaz para modelar a conectividade entre dispositivos. A implementação dos algoritmos de procura em profundidade (DFS) e em largura (BFS) demonstrou ser adequada para explorar os caminhos possíveis entre antenas, detetar componentes ligados e verificar acessibilidade.

A componente de deteção de locais com efeito nefasto introduziu uma camada adicional de complexidade, obrigando à análise geométrica da propagação do sinal e à criação de regras precisas para identificar essas zonas. A correta integração desta funcionalidade reforçou a lógica modular do projeto, onde cada parte desempenha um papel específico, mas interligado com o todo.

Em suma, os objetivos propostos foram plenamente alcançados. O projeto resultou numa solução funcional, clara e facilmente extensível, que alia a estruturação eficiente dos dados à capacidade de análise espacial e de conectividade.

Para além dos resultados técnicos, este trabalho permitiu desenvolver competências importantes, como a abstração algorítmica, a gestão de memória dinâmica em C, a divisão de tarefas por módulos e a organização de código com boa legibilidade e documentação (através do Doxygen). Esta experiência reforça a preparação para desafios futuros que envolvam análise de dados, estruturas não-lineares e problemas de redes em sistemas computacionais.

8. Bibliografia

Programação Descomplicada | Linguagem C Estrutura de Dados em C | Aula 56
- Grafos – Definição. YouTube.

OpenAI (2025). ChatGPT – modelo de linguagem.

9. Índice de Figuras

<u>Figura 1 esquema da estrutura do grafo</u>	11
<u>Figura 2 conexão de antenas</u>	12