

***INSTITUTO SUPERIOR POLÍTECNICO DE
TECNOLOGIAS E CIÊNCIAS CURSO DE
ENGENHARIA INFORMÁTICA***

Estrutura de Dados I

2025/26 1º Semestre | Trabalho Final

Relatório de Desenvolvimento: Sistema de Voto Eletrónico

Elaborado por:

- António Nangikola - 20240694
- Josemar da Silva - 20240610
- Valdnir Albino - 20241757

Docente: Sílvia António

Fevereiro, 2026

Relatório de Desenvolvimento: Sistema de Voto Eletrónico
Fundamentação Teórica

1. Introdução e Metodologia de Trabalho

O desenvolvimento de sistemas de software fundamentados em Estruturas de Dados representa um pilar central na Engenharia de Software e na Ciência da Computação. Trabalhos desta natureza visam aplicar conceitos de organização lógica e manipulação eficiente de informações na memória, utilizando estruturas como listas ligadas, filas e pilhas. A escolha adequada de um Tipo Abstrato de Dados (TAD) permite que problemas complexos sejam decompostos em módulos independentes, garantindo que o software não seja apenas funcional, mas também otimizado em termos de processamento e uso de memória dinâmica. Em projetos acadêmicos, esta prática consolida o entendimento sobre ponteiros, gestão de memória e a modularização de sistemas robustos.

O presente trabalho foi desenvolvido em grupo, sendo as tarefas distribuídas de forma equilibrada entre os membros, de acordo com as responsabilidades e competências de cada um. O objetivo central foi garantir uma construção uniforme e organizada do sistema proposto, no âmbito da disciplina de Estruturas de Dados I.

2. Módulo de Gestão de Eleitores

Este módulo constitui uma parte essencial do sistema, pois garante que apenas eleitores devidamente registados possam participar no processo de votação, assegurando a integridade e a fiabilidade dos resultados.

- **Estrutura de Dados:** Os eleitores foram representados através de uma lista ligada simples, onde cada nó corresponde a um eleitor. A estrutura é composta pelo id (identificador único), nome, votou (0 para não votou, 1 para votou) e um ponteiro prox para o próximo nó. A escolha desta estrutura deve-se à flexibilidade para inserções dinâmicas sem número total predefinido.
- **Funcionalidades Implementadas:**
 - **Criação e Inserção:** A função criarEleitor aloca memória dinamicamente e inicializa os campos (estado inicial votou = 0). A inserção ocorre ao final da lista via inserirEleitor, utilizando um ponteiro duplo para permitir a alteração do início da lista.
 - **Procura e Validação:** A função procurarEleitor permite localizar um eleitor pelo ID, servindo de base para validar se operações de atualização e votação são realizadas sobre eleitores registados.
 - **Controlo e Atualização:** A função atualizarEleitor possibilita a alteração do nome mantendo a integridade do ID e do estado de votação. O estado de votação é controlado pela função marcarComoVotou, que atualiza o campo apenas após o voto, impedindo a duplicidade.
 - **Gestão de Ficheiros e Memória:** A função carregarEleitoresFicheiro permite o registo automático via ficheiro de texto, reiniciando o estado para "não votou" conforme as regras do sistema. A função libertarEleitores percorre a lista e liberta a memória alocada, prevenindo vazamentos (*memory leaks*).

3. Módulo de Candidatos e Filas de Votação

Este módulo tem como finalidade garantir a correta gestão dos candidatos concorrentes, bem como a organização eficiente das filas de votação associadas às urnas.

- **Estruturas de Dados Utilizadas:**
 - **Lista Ligada (Candidatos):** Os candidatos são armazenados numa lista ligada simples que permite a inserção e remoção dinâmica antes do início da votação. Cada nó contém o ID do candidato, nome e número de votos obtidos.
 - **Fila Encadeada (Urnas):** As urnas são representadas por três filas distintas. A utilização da fila garante que os eleitores sejam atendidos segundo o princípio FIFO (*First In, First Out*), respeitando a ordem de chegada.
- **Funcionalidades Implementadas:** Incluem a inserção e remoção de candidatos, a inicialização das três filas de votação e a distribuição dos eleitores por elas. O sistema permite ainda a desistência de eleitores das filas e a impressão do tamanho e conteúdo de cada fila para fins de controlo e visualização.

4. Processo de Votação, Auditoria e Resultados

Este módulo é responsável pelo processamento central dos votos, auditoria interna e geração da documentação final do pleito.

- **Lógica de Votação:** São implementadas as funções de validação e classificação dos votos (válidos, brancos e nulos), além do cálculo e apresentação dos resultados parciais e do resultado final da votação.
- **TAD Pilha (Auditoria):** Na pasta referente à pilha, foi implementado o Tipo Abstrato de Dados Pilha, utilizado para registar os votos segundo o princípio LIFO (*Last In, First Out*) para fins de auditoria, permitindo o acompanhamento da sequência do processo.
- **Encerramento e Saída:** Foi implementada uma funcionalidade para a geração de um ficheiro final com todos os resultados da votação. O módulo garante também a libertação adequada de toda a memória alocada dinamicamente durante a execução, assegurando a eficiência do sistema. Os ficheiros centrais desenvolvidos foram pilha.h/c e votacao.h/c.

5. Recursos de utilização e Conclusão

O cronograma de desenvolvimento e os recursos utilizados foram organizados da seguinte forma:

- **Início do Projeto:** As atividades tiveram início no dia 22 de janeiro do ano em curso.
- **Planeamento:** Foram elaborados esboços e planeamentos detalhados com o objetivo de definir a organização geral do código e a forma de integração de cada funcionalidade.
- **Ferramentas de Apoio:** A fase inicial de codificação utilizou um editor de texto simples (Bloco de Notas), evoluindo com o suporte de vídeos educativos no YouTube para aprofundar conceitos de organização e estruturas de dados.
- **Auxílio Tecnológico:** Utilizou-se o ChatGPT como ferramenta de apoio à lógica de programação, à sintaxe da linguagem C e à organização estrutural do projeto, incluindo a distribuição de ficheiros e pastas.
- **Estrutura Modular:** O projeto foi organizado de forma modular, dividindo-se em pastas com ficheiros de cabeçalho (.h) para declaração de funções e ficheiros de implementação (.c) para a definição do comportamento das mesmas.
- **Fase de Testes:** O ficheiro `main.c` foi utilizado de forma estratégica para testar o funcionamento individual das funções e facilitar a integração final entre os módulos desenvolvidos pelos diferentes membros.

O desenvolvimento deste sistema permitiu a aplicação prática de conceitos fundamentais como listas ligadas, filas, pilhas, ponteiros e alocação dinâmica de memória. A abordagem modular facilitou a integração dos componentes através do ficheiro `main.c`, utilizado para testar a funcionalidade de cada módulo individualmente. As principais dificuldades superadas envolveram a correta gestão de ponteiros e a ligação técnica entre os ficheiros de interface e implementação. O sistema cumpre o objetivo de ser uma solução eficiente para o processo de voto eletrónico.

