



**Politécnico
de Viseu**

Escola Superior
de Tecnologia
e Gestão de Viseu

Software para Registo de Estatísticas de Equipas de Andebol

Marcos Marcelo Assis Nunes – 33417

Pedro Miguel Gomes – 33628

Raquel Vaz Guerreiro - 33559

Bernardo Miguel Vieira - 33676

Relatório de Trabalho Prático

Curso Técnico Superior Profissional em

Redes e Sistemas Informáticos

Dezembro de 2025



Politécnico
de Viseu

Escola Superior
de Tecnologia
e Gestão de Viseu



departamento de
informática

Software para Resgisto de Estatísticas de Equipas de Andebol

Marcos Marcelo Assis Nunes – 33417

Pedro Miguel Gomes – 33628

Raquel Vaz Guerreiro - 33559

Bernardo Miguel Vieira - 33676

Relatório de Trabalho Prático

Curso Técnico Superior Profissional em
Redes e Sistemas Informáticos

Trabalho efetuado sob a orientação de
Manuel Lopes
Vitor Figueiredo
Raquel Sebastião

Dezembro de 2025

CENTRO 2030

Os Fundos Europeus mais próximos de si.

PORTUGAL
2030



Cofinanciado pela
União Europeia

Índice

1	Resumo.....	IV
2	Introdução	V
3	Descrição do Problema	V
4	Organização e Estrutura do Projeto	VI
5	Funcionalidades Implementadas.....	VIII
6	Compilação e Execução do Programa.....	IX
6.1	Instruções de Compilação:.....	X
6.2	Instruções de Execução	XI
7	Conclusão	XI
8	Referências	XI

1 Resumo

O presente relatório descreve o desenvolvimento de uma aplicação em linguagem C para o registo e gestão de dados estatísticos de equipas de andebol, no âmbito do projeto final da disciplina de Introdução à Programação, integrada no curso de Redes e Sistemas Informáticos da Escola Superior de Tecnologia e Gestão de Viseu, do Instituto Politécnico de Viseu. Realizado por um grupo de quatro alunos, o trabalho simula um pedido da Federação Portuguesa de Andebol, permitindo o armazenamento dinâmico de informações sobre campeonatos, equipas (com designação e até 15 atletas) e atletas (com atributos como nome, identificação única de sete dígitos, ano de nascimento, posição, médias estatísticas e minutos jogados). A valia dos atletas é calculada com base na posição (exceto guarda-redes), utilizando fórmulas específicas, e a aplicação suporta operações como registo, adição, atualização, remoção, listagens ordenadas, pesquisas por critérios (valia, ano de nascimento), cálculos de valia coletiva e identificação de extremos (equipas/atletas mais/menos valiosos), com carregamento e gravação automática em ficheiro.

A implementação prioriza a modularidade, com separação em ficheiros fonte (core.c para lógica central, io.c para entrada/saída e menus, stats.c para cálculos e main.c para execução), uso de alocação dinâmica de memória (malloc, realloc, free) para evitar limites fixos e libertação de recursos ao encerrar, e validações robustas para garantir integridade dos dados. O projeto foi organizado num repositório GitHub com suporte multiplataforma via Makefile, e testes foram efetuados com dados iniciais em equipas.txt. Apesar de desafios como gestão de memória e leitura de ficheiros, o resultado consolida os conceitos lecionados, promovendo uma solução simples, clara e funcional.

2 Introdução

No âmbito da disciplina de Introdução à Programação, no curso de Redes e Sistemas Informáticos da Escola Superior de Tecnologia e Gestão de Viseu, do Instituto Politécnico de Viseu, o presente projeto prático final visa aplicar os conceitos fundamentais lecionados ao longo do semestre.

O projeto consiste no desenvolvimento de uma aplicação de consola para o registo e gestão de dados estatísticos de equipas de andebol, simulando um pedido hipotético da Federação Portuguesa de Andebol. Esta aplicação permite o armazenamento dinâmico de informações sobre equipas e atletas, o cálculo de métricas de desempenho e a realização de operações como adições, remoções, pesquisas e listagens. Os objetivos gerais incluem promover o domínio de estruturas de dados, alocação dinâmica de memória (utilizando funções como malloc, calloc, realloc e free), leitura e escrita em ficheiros, validação de entradas e modularização do código para garantir clareza e reutilização.

3 Descrição do Problema

O problema proposto envolve a criação de uma aplicação que simule a gestão de um campeonato de andebol, com foco no registo de dados estatísticos de equipas e atletas. Cada equipa é caracterizada pela sua designação e por um grupo de até 15 atletas, enquanto cada atleta possui atributos como nome, número de identificação único e positivo de sete dígitos, ano de nascimento, posição de jogo (ponta, lateral, central, pivô ou guarda-redes), médias de pontos, remates, perdas de bola, assistências e fintas por jogo, e total de minutos jogados. Com exceção dos guarda-redes, cuja valia é zero, o valor de cada atleta é calculado com base na sua posição, utilizando fórmulas específicas:

- Ponta: valia = $5 \times mPontos + 4 \times mRemates + 2 \times mAssist - 3 \times mPerdas$
- Lateral: valia = $3 \times mPontos + 2 \times mRemates + 4 \times mAssist - 3 \times mPerdas$
- Central: valia = $2 \times mPontos + 5 \times mAssist + 4 \times mFintas - 2 \times mPerdas$
- Pivô: valia = $2 \times mPontos + 1 \times mRemates + 5 \times mFintas - 2 \times mPerdas$

A aplicação carrega automaticamente dados de um ficheiro no arranque (com pelo menos duas equipas e atletas em todas as posições para testes) e garante a gravação automática de alterações durante a execução e ao sair.

A gestão de memória é dinâmica, permitindo a criação e remoção de equipas e atletas sem limites pré-definidos, com libertação correta de recursos para evitar fugas de memória. A validação de dados é priorizada, incluindo verificações de unicidade do número de identificação, validade das posições e limites numéricos. A estrutura do programa é modular, com variáveis e estruturas de dados adequadas, promovendo a simplicidade, indentação apropriada e inclusão de comentários para facilitar a interpretabilidade.

4 Organização e Estrutura do Projeto

O projeto foi organizado de forma modular e estruturada, com o objetivo de promover a clareza, a manutenção e a reutilização do código, em conformidade com as boas práticas lecionadas na disciplina. O repositório no GitHub, denominado "[protoclcoders/andebol-statistics-register-software](#)", segue uma hierarquia lógica que separa responsabilidades e facilita a colaboração entre os membros do grupo.

A estrutura do repositório é a seguinte:

- **Raiz do repositório:** Contém o ficheiro LICENSE, sob a licença MIT, que define as permissões de uso e distribuição do software. Inclui também o Makefile, responsável pela compilação e execução multiplataforma (compatível com Windows, Linux e macOS), utilizando comandos adaptados ao sistema operativo detetado. O Makefile define regras para compilar os ficheiros fonte, criar diretórios “obj/” e “bin/”, executar o programa e limpar artefactos gerados, com flags como “-Wall”, “-Wextra” e “-g” para depuração e segurança.
- **Pasta data/:** Armazena os ficheiros de dados, incluindo “equipas.txt”, que contém os dados iniciais do campeonato (nome, ano, equipas como FC Porto e Sporting CP, e atletas com atributos completos, delimitados por “|”, para testes). Inclui ainda o ficheiro “Estrutura-dados-equipa.txt”, que descreve o formato dos dados no ficheiro “equipas.txt”, facilitando a compreensão da estrutura de armazenamento.

- **Pasta docs/**: Destinada a documentação adicional, a conter o ficheiro referente ao enunciado do projeto (nomeadamente “RSI_IP_Projeto_2025_26_Normal.pdf”) e o presente relatório em formato PDF.
- **Pasta include/**: Contém o cabeçalho andebol.h, onde são definidas as estruturas de dados principais (Atleta, Equipa e Campeonato), constantes (como MAX_ATLETAS = 15, MAX_NOME_ATLETA = 100 e MAX_DESIGNACAO = 100) e protótipos de funções. Esta separação permite a inclusão centralizada de definições comuns, promovendo a consistência.
- **Pasta src/**: Agrupa os ficheiros fonte em C, divididos por módulos funcionais:
 - **core.c**: Implementa funções centrais de gestão de dados, como alocação dinâmica de equipas e atletas (alocaEquipa, alocaAtleta, alocaCampeonato), adição e remoção de equipas/atletas (adicionarEquipa, removerEquipa, adicionarAtleta, removerAtleta), validações (validarNumId), limpeza de buffer (limparBuffer), atualização de atletas (atualizarAtleta), libertação de memória (libertarMemoria) e apagar dados (ApagarDados). Inclui tratamentos de erros para alocação falhada e uso de realloc para expansão dinâmica.
 - **io.c**: Responsável pelas operações de entrada/saída, incluindo leitura de ficheiros (ler_ficheiro, com tratamento de erros como criação automática se inexistente), impressão de dados de atletas (imprimirDadosAtleta) e interfaces de menu (menuMI, menuA, menuB, etc.), com suporte a limpeza de ecrã multiplataforma (system("cls") ou system("clear")).
 - **main.c**: Ponto de entrada da aplicação, onde é inicializado o campeonato e chamado o menu principal, garantindo o fluxo de execução.
 - **stats.c**: Dedicado a cálculos estatísticos, como valia de atletas (calcularValiaAtleta, atualizarTodasValias) e equipas (calcularValiaEquipa), ordenações (ordenarAtletasPorValia, ordenarAtletasPorNome), listagens (listarAtletasEquipaOrdenado), pesquisas (pesquisarAtletasPorValia, pesquisarAtletasPorAno, pesquisarAtletasPorAnoExato) e identificações de extremos (encontrarEquipaMaisValiosa, encontrarEquipaMenosValiosa, encontrarMelhorAtletaPorPosicao).

Esta modularização reflete uma divisão clara de responsabilidades: core.c para lógica principal, io.c para interação com o utilizador e ficheiros, stats.c para análises e cálculos e main.c para organização e orquestração. A gestão de ficheiros é

centralizada, com carregamento automático de equipas.txt no arranque e gravação automática de alterações, utilizando delimitadores "|" para persistência. As validações são implementadas em vários níveis, como verificação de unicidade e formato do número de identificação, posições válidas (com strcasecmp para insensibilidade a maiúsculas/minúsculas), limites de capacidade e verificações da existência de equipas ou atletas, contribuindo para a robustez da aplicação.

5 Funcionalidades Implementadas

Todas as funcionalidades requeridas pelo enunciado foram implementadas e estão alinhadas aos conhecimentos lecionados. O código prioriza a simplicidade e a ausência de repetições, com uso de funções reutilizáveis e tratamentos de erros para evitar falhas em tempo de execução. A aplicação é intuitiva, com menus claros na consola, e suporta operações automáticas como carregamento e gravação de dados em ficheiro.

A descrição detalhada das funcionalidades, numeradas conforme o enunciado, consiste no seguinte:

1. **Registo de uma equipa e respetivos atletas:** Implementado através de adicionarEquipa e adicionarAtleta, que solicitam dados ao utilizador (designação, capacidade, número de atletas, atributos individuais) e validam entradas. A função imprimirDadosAtleta permite visualizar informações de um atleta específico pelo número de identificação, com pesquisa em todas as equipas.
2. **Adicionar atletas a equipas:** A função adicionarAtleta seleciona uma equipa existente, verifica capacidade disponível e adiciona o atleta com cálculo automático de valia, atualizando o ficheiro.
3. **Atualizar informação de um atleta de modo eficaz:** A função atualizarAtleta localiza o atleta pelo ID, exibe os dados atuais e permite edição seletiva ou total de campos, recalculando a valia e gravando alterações automaticamente.
4. **Listar os atletas de cada equipa, ordenados:** Suportado por ordenarAtletasPorValia (crescente/decrescente) e ordenarAtletasPorNome (alfabética pelo primeiro nome), com listagem via listarAtletasEquipaOrdenado, exibindo todos os atributos em formato tabular na consola.

5. **Pesquisar atletas com base em critérios definidos pelo utilizador:**
Implementadas pesquisas por valia (pesquisarAtletasPorValia, superior/inferior) e ano de nascimento (pesquisarAtletasPorAno, anterior/posterior), iterando sobre todas as equipas e exibindo resultados relevantes.
6. **Calcular a valia de uma equipa:** A função calcularValiaEquipa soma as valias individuais dos atletas, com atualização prévia via atualizarTodasValias.
7. **Indicar qual a equipa mais valiosa e qual a menos valiosa:** Utiliza encontrarEquipaMaisValiosa e encontrarEquipaMenosValiosa, comparando valias totais e exibindo as equipas correspondentes.
8. **Indicar qual o atleta mais valioso em cada posição:** A função encontrarMelhorAtletaPorPosicao filtra atletas com mais de 100 minutos, calcula valias e seleciona o melhor por posição (pon, lat, cen, piv, gua), exibindo detalhes.
9. **Permitir pesquisar quais os atletas nascidos em determinado ano:**
Implementado em pesquisarAtletasPorAnoExato, com entrada do ano e listagem de atletas correspondentes.
10. **Remover atletas de equipas:** Suporta remoção específica (removerAtleta, libertando memória) ou total (removerTodosAtletas), com atualização do ficheiro e realloc para otimização de espaço.
- 11-13. **Gravação de informação:** A gravação manual é acionada por menu, enquanto alterações são gravadas automaticamente durante a execução (após adições/remoções/atualizações) e ao sair (via libertarMemoria e ApagarDados opcionais), utilizando fprintf para persistir no formato delimitado por "|".

Adicionalmente, foram introduzidas melhorias como suporte multiplataforma no Makefile, delays visuais para feedback (Sleep/sleep), tratamento de ficheiros inexistentes (criação automática) e menus intuitivos com opções de confirmação para ações irreversíveis, todas enquadradas nos conceitos da unidade curricular.

6 Compilação e Execução do Programa

A compilação e execução da aplicação foram projetadas para serem simples e multiplataforma, utilizando um Makefile que automatiza o processo e adapta-se ao

sistema operativo do utilizador (Windows, Linux ou macOS). Esta abordagem garante uma experiência intuitiva, com tratamento de erros e depuração facilitada, em linha com os princípios de robustez lecionados na disciplina. Os requisitos incluem um compilador GCC instalado (disponível nativamente em Linux, ou via MinGW em Windows).

6.1 Instruções de Compilação:

Via Makefile: Na raiz do repositório, execute o comando make. Este cria as pastas “obj/” (para ficheiros objeto) e “bin/” (para o executável), compila os ficheiros fonte da pasta “src/” e gera o executável “bin/andebol” (ou bin\andebol.exe em Windows). As flags de compilação incluem “-Wall -Wextra -g -fstack-protector-strong -Wformat-security”, que ativam avisos detalhados, depuração e proteções de segurança, promovendo a identificação precoce de problemas.

Manual: Caso o Makefile não seja utilizado, compile individualmente com o comando “gcc -Wall -Wextra -g -fstack-protector-strong -Wformat-security src/core.c src/io.c src/main.c src/stats.c -o bin/andebol”.

O Makefile deteta automaticamente o sistema operativo, adaptando comandos como “mkdir”, “rm” e execução de terminal, para garantir compatibilidade.

6.2 Instruções de Execução

Via Makefile: Execute “make run”. Este abre um terminal e inicia o executável “bin/andebol”, utilizando o comando “cmd /c start” em Windows, “x-terminal-emulator -e” em Linux ou “open -a Terminal” em macOS.

Manual: Navegue para a pasta “bin/” e execute “./handebol” (Linux/macOS) ou “andebol.exe” (Windows). A aplicação carrega automaticamente os dados de “data/equipas.txt”, apresenta menus na consola e grava alterações no mesmo ficheiro.

Limpeza: Para remover os arquivos gerados, execute “make clean”, que elimina as pastas “obj/” e “bin/”.

7 Conclusão

Em síntese, o projeto alcançou os objetivos propostos, implementando uma aplicação completa e funcional para a gestão de dados estatísticos de andebol, com ênfase na modularidade, na gestão dinâmica de memória e na validação de dados. Todas as funcionalidades requeridas foram desenvolvidas, demonstrando o domínio dos conceitos lecionados, como estruturas de dados, alocação dinâmica e operações com ficheiros.

Entre as dificuldades enfrentadas, destacam-se a gestão precisa de memória dinâmica para evitar fugas e acessos inválidos, a implementação de leitura e escrita em ficheiros delimitados por "|", e a adaptação multiplataforma do Makefile e das funções de interface. Estas foram superadas através de testes iterativos e consulta aos materiais das aulas, reforçando a importância da depuração e da estruturação do código.

8 Referências

- Enunciado do projeto: RSI_IP_Projeto_2025_26_Normal.pdf. Disponível no Moodle da unidade curricular “Introdução à Programação”, Escola Superior de Tecnologia e Gestão de Viseu, Instituto Politécnico de Viseu, 2024/2025.
- Repositório GitHub: [protocolcoders/andebol-statistics-register-software](https://github.com/protocolcoders/andebol-statistics-register-software). Acedido em 28 de dezembro de 2025.