
Relatório Final do Projeto de Desenvolvimento de Aplicações Móveis

Sumário:	Este relatório documenta o desenvolvimento da aplicação móvel <i>AthleTrack</i> , criada no âmbito da unidade curricular de Desenvolvimento de Aplicações para Dispositivos Móveis. A aplicação visa apoiar a gestão de treinos, eventos e presenças em contexto académico-desportivo, recorrendo a tecnologias como Jetpack Compose, Kotlin e Spring Boot.
Grupo:	Grupo composto por: Gabriel Oliveira - 120364 João Gabriel – 117589 Luis Assis – 112763 Luís Nantes - 120401
Professor da Disciplina:	Prof. José Martins
Circulação:	Exclusiva para a Equipa de Desenvolvimento e Professor da Disciplina
Ano/Semestre	2º Ano/2º Semestre

Agradecimentos

Gostaríamos de expressar a nossa gratidão ao nosso professor da disciplina. pelo suporte, pelo conhecimento partilhado e pelas orientações fundamentais ao longo do desenvolvimento. Um especial agradecimento às nossas famílias e amigos pelo incentivo constante e compreensão nos momentos desafiantes. Também reconhecemos o empenho e colaboração de cada membro da equipa, cujo esforço conjunto permitiu a concretização deste trabalho. Por fim, agradecemos a todos os que, direta ou indiretamente, contribuíram para o sucesso deste projeto.

Índice

1	Introdução	4
1.1	Enquadramento e Objetivos do Projeto	4
1.2	Metodologia de Trabalho	4
1.2.1	Método de desenvolvimento	4
1.2.2	Tecnologias associadas	4
2	Planeamento do Trabalho	5
2.1	Calendarização	5
2.2	Público-Alvo	6
2.3	Levantamento de Requisitos	6
2.3.1	Requisitos Funcionais	6
2.3.2	Requisitos não Funcionais	7
2.4	Diagrama de Casos de Utilização	8
2.4.1	Visão Geral	8
2.4.2	Atores do Sistema	8
2.5	Protótipo da Aplicação	9
2.6	Modelo de Dados Persistente	9
2.7	Modelo de Estados da Aplicação	10
3	Testes e Validações	10
3.1	Testes Unitários	10
3.2	Testes Estruturais	11
4	Análise dos resultados	11
5	Anexos	12

Índice de Figuras

Figura 1- Diagrama de Gantt	5
Figura 2-Diagrama de Casos de Uso	8
Figura 3-Protótipo da Aplicação no Figma.....	9
Figura 4-Diagrama Conceitual da Base de Dados.....	9
Figura 5-Representação da Navegação na Aplicação.....	10

Índice de Tabelas

Tabela 1- Requisitos Funcionais.....	7
Tabela 2- Requisitos não Funcionais	7

Lista de Siglas, Abreviaturas, Estrangeirismos e Termos Técnicos

API – Interface de comunicação entre sistemas.

Backend – Lado do servidor da aplicação.

Framework – Estrutura base para desenvolvimento.

Frontend – Interface visível ao utilizador.

QR Code – Código visual para leitura rápida.

UI – Interface do utilizador.

ViewModel – Camada que gere dados para o UI.

REST – Padrão para APIs web.

.jar – Ficheiro executável Java.

1 Introdução

1.1 Enquadramento e Objetivos do Projeto

O **AthleTrack** tem como finalidade facilitar a gestão e acompanhamento de atividades desportivas em **contexto institucional**, promovendo a organização de **treinos, eventos e registos de presenças**. Dirige-se tanto a alunos, que participam nas modalidades e registam a sua presença, como a professores, que têm a responsabilidade de gerir as modalidades e planejar treinos e eventos. O principal objetivo do projeto é proporcionar uma **solução digital funcional, intuitiva e segura**, que **automatize processos** anteriormente manuais e contribua para a melhoria da comunicação e controlo das atividades desenvolvidas.

1.2 Metodologia de Trabalho

1.2.1 Método de desenvolvimento

O desenvolvimento da aplicação seguiu uma abordagem mista. Numa fase inicial, foi adotado um **modelo em cascata**, começando pelo levantamento e análise de requisitos, passando pela conceção e, posteriormente, pela implementação. Este método foi útil para estruturar a base do projeto e definir uma direção clara.

Contudo, à medida que o trabalho avançava, tornou-se evidente a necessidade de maior agilidade e capacidade de adaptação. Assim, migrámos para uma abordagem **iterativa baseada no trabalho em duplas**, onde os elementos da equipa colaboraram em ciclos curtos de desenvolvimento, focando-se em funcionalidades específicas. Esta estratégia permitiu uma resposta mais eficaz a imprevistos e facilitou a integração contínua de código, com apoio em ferramentas como o GitHub para controlo de versões e colaboração.

1.2.2 Tecnologias associadas

A aplicação foi inteiramente desenvolvida em **Kotlin**, tanto no lado do frontend (aplicação Android) como no backend (API REST), tirando partido da sua concisão, interoperabilidade com Java e suporte moderno para programação assíncrona.

As principais tecnologias adotadas foram:

- **Frontend:** Utilizámos **Jetpack Compose**, o toolkit moderno da Google para construção de interfaces em Android. Esta ferramenta permite uma abordagem **declarativa** ao design de interfaces, facilitando a criação de ecrãs responsivos e adaptáveis a diferentes tamanhos de dispositivo. A arquitetura escolhida foi **Model-View-ViewModel (MVVM)**, promovendo uma clara separação de responsabilidades, maior manutenibilidade do código, e facilidade de teste e reutilização de componentes.
- **Backend:** Foi implementado com **Spring Boot**, que possibilitou a criação de uma API REST modular e eficiente. Esta framework fornece um ecossistema robusto para desenvolvimento de serviços web, facilitando a integração com a base de dados, o tratamento de autenticação e a configuração de segurança.

- **Comunicação HTTP:** A comunicação entre o frontend e o backend é feita por meio da biblioteca **Retrofit**, que permite realizar chamadas de rede de forma segura, escalável e com tratamento eficiente de respostas e erros.
- **Base de Dados:** O sistema utiliza **PostgreSQL**, uma base de dados relacional de alto desempenho. Os dados são hospedados na plataforma **Azure**, assegurando acessibilidade remota e fiabilidade.
- **Deploy do Backend:** O serviço backend foi empacotado num ficheiro .jar e disponibilizado através da plataforma **Render**, permitindo a execução contínua da API na nuvem. Esta abordagem simplificou o processo de publicação do serviço e garantiu a sua disponibilidade pública para consumo da aplicação Android.
- **Controlo de Versões e Colaboração:** Foi utilizado o **GitHub** como ferramenta central de versionamento e colaboração em equipa, garantindo organização, histórico de alterações e integração contínua de código.

Estas tecnologias foram seleccionadas para garantir boa integração entre os componentes, facilitar o desenvolvimento e permitir um produto final fiável e escalável.

2 Planeamento do Trabalho

2.1 Calendarização

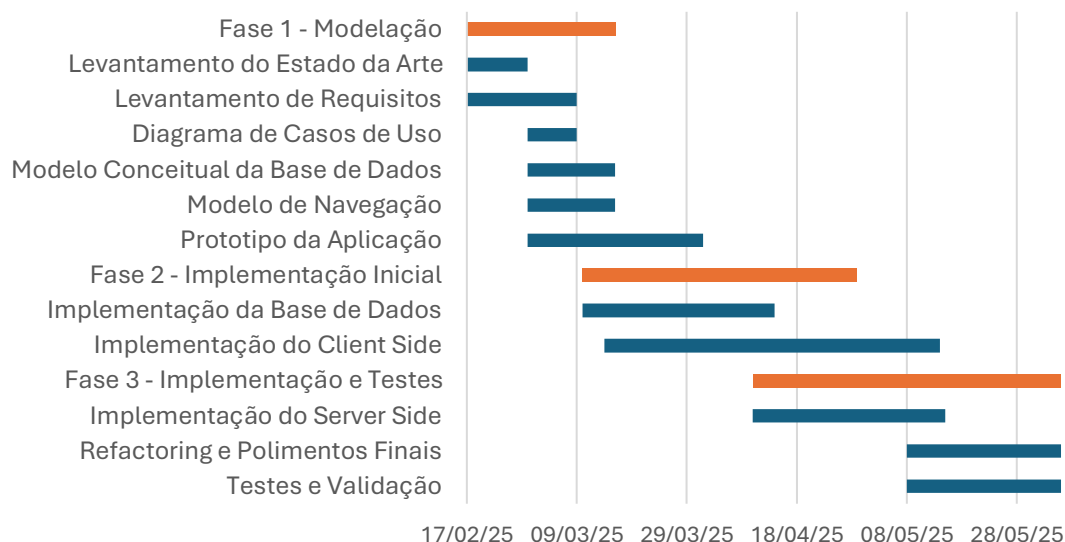


Figura 1- Diagrama de Gantt

2.2 Público-Alvo

A aplicação foi desenvolvida para responder às necessidades de **instituições multidesportivas**, como clubes, academias ou escolas com várias modalidades desportivas, que procuram **modernizar e automatizar a gestão das suas atividades internas**.

2.3 Levantamento de Requisitos

2.3.1 Requisitos Funcionais

Prioridade	Nº	Requisito
1	RF1	Consulta de Treinos: Os alunos podem visualizar a data e hora dos próximos treinos.
	RF2	Registo de Presença: Os alunos podem confirmar a presença nos treinos através da leitura de um código QR.
	RF3	Validação de QR Code: O sistema deve validar o código QR e associá-lo ao aluno que fez o scan.
	RF4	Visualização de Presenças em Tempo Real: O sistema deve permitir que o professor visualize, em tempo real, a lista de alunos presentes numa aula.
	RF5	Autenticação de Utilizadores: O sistema deve permitir que os alunos e professores façam login na aplicação.
	RF6	Restrição de Acesso para Alunos: O sistema deve garantir que os alunos apenas possam visualizar e marcar presença nos seus próprios treinos.

	RF7	Restrição de Acesso para Professores: O sistema deve garantir que os professores apenas possam visualizar as suas próprias aulas.
	RF10	Gestão de Treinos: O sistema deve permitir que os treinos sejam criados com data, hora e dia da semana.
	RF11	Gestão de Credenciais de Alunos: O sistema deve permitir que credenciais do tipo aluno sejam criadas

Tabela 1- Requisitos Funcionais

2.3.2 Requisitos não Funcionais

Prioridade	Nº	Requisito
1	RF1	Escalabilidade: O sistema deve ser capaz de processar múltiplos pedidos simultaneamente sem degradação do desempenho.
	RF2	Tempo de Resposta: O tempo de resposta para ações como login e registo de presença não deve ultrapassar 2 segundos.
	RF3	Proteção de Dados: Os dados dos utilizadores devem ser armazenados de forma segura, utilizando encriptação para senhas e informações sensíveis.
	RF4	Autenticação Segura: Apenas utilizadores autenticados podem aceder às funcionalidades do sistema.
	RF5	Gestão de Permissões: O acesso às funcionalidades deve ser controlado por níveis de permissão (alunos e professores).
	RF6	Leitura de QR Code Eficiente: A leitura do código QR deve ser rápida e precisa, garantindo uma boa experiência do utilizador.
2	RF7	Suporte ao Crescimento de Utilizadores: A aplicação deve ser capaz de suportar um aumento no número de utilizadores sem comprometer a performance.

Tabela 2- Requisitos não Funcionais

2.4 Diagrama de Casos de Utilização

2.4.1 Visão Geral

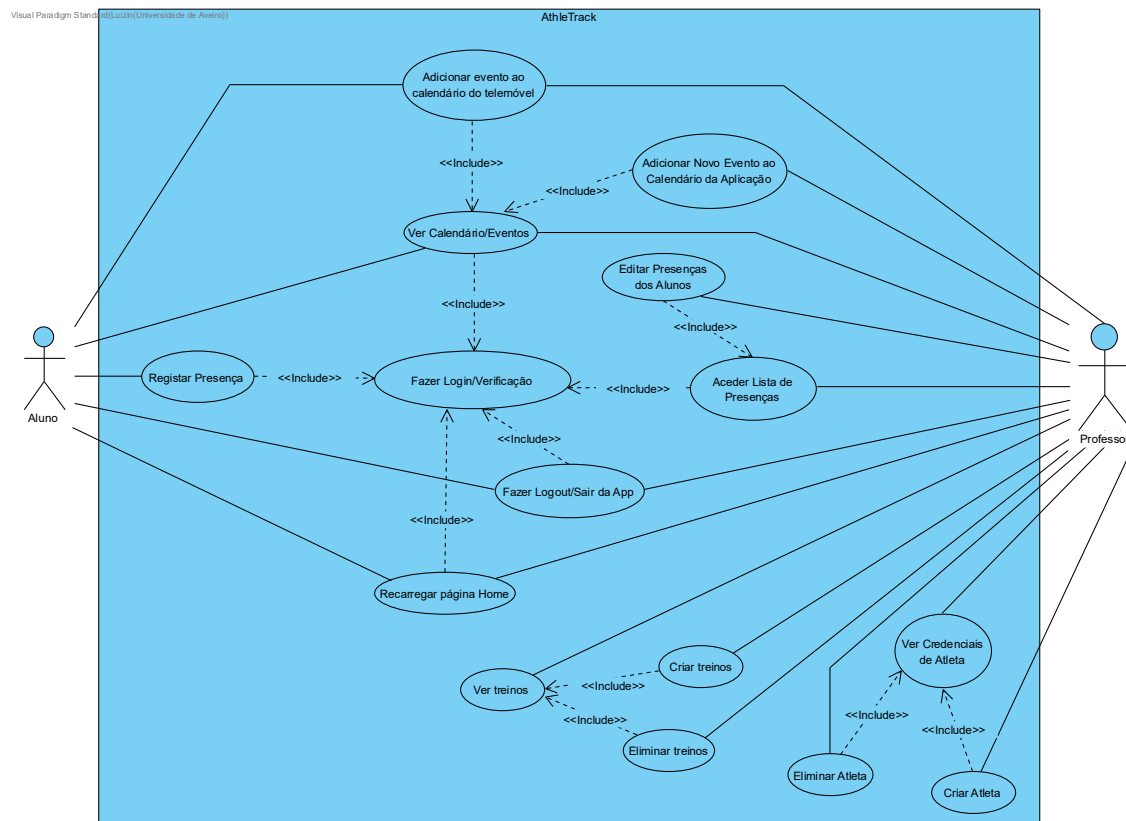


Figura 2-Diagrama de Casos de Uso

2.4.2 Atores do Sistema

Aluno - O utilizador final da aplicação que tem como principal objetivo aceder às informações das modalidades em que está inscrito. Através da aplicação, o aluno pode visualizar os **treinos regulares** e os **eventos especiais** agendados, bem como realizar o **registo de presença através de um QR Code** gerado pelo sistema.

A interface disponibiliza ainda a opção de **logout**, garantindo uma utilização segura da aplicação.

Professor - Possui permissões mais alargadas, atuando como gestor dos treinos da academia e das credenciais dos alunos. Cabe-lhe a criação e edição de **treinos regulares** e de **eventos**. O professor pode também **visualizar as listas de presença** dos treinos e **editar** presenças dos alunos.

Tal como o aluno, o professor pode também realizar o **logout** da aplicação.

2.5 Protótipo da Aplicação

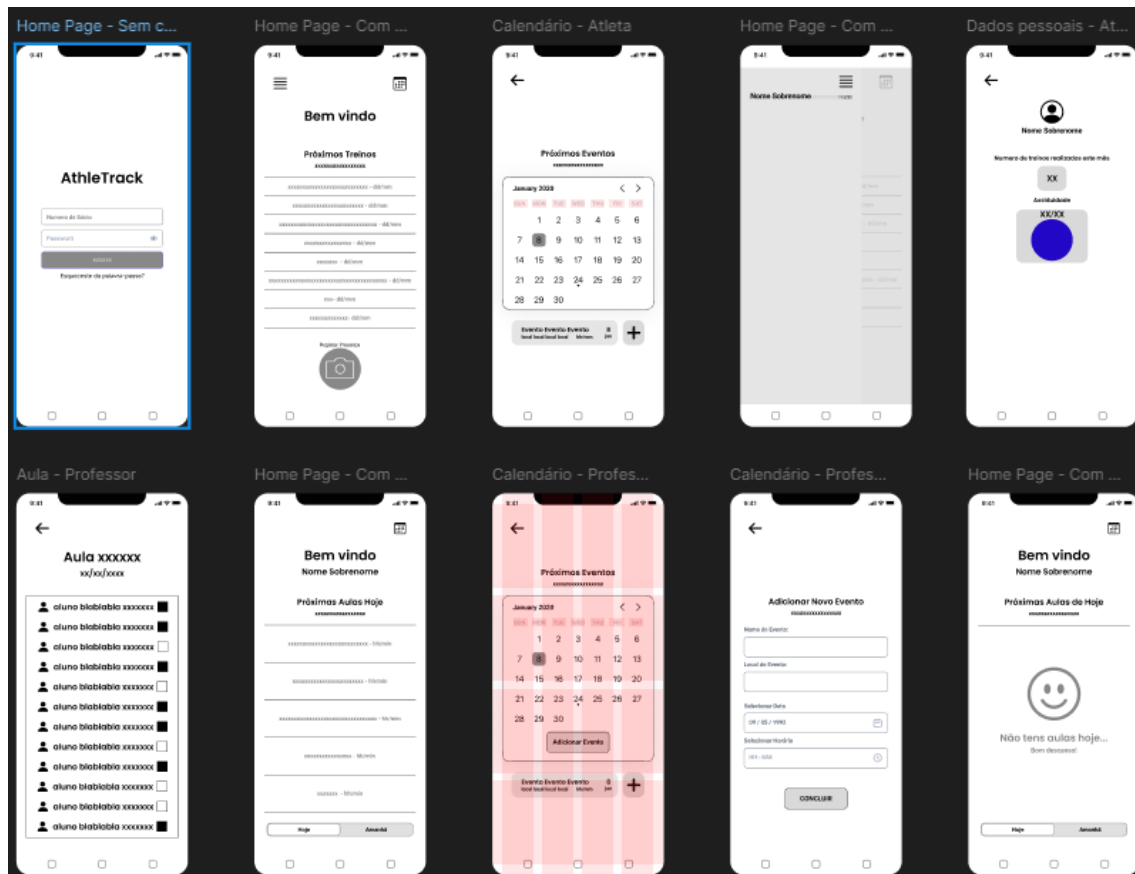


Figura 3-Protótipo da Aplicação no Figma

2.6 Modelo de Dados Persistente

Visual Paradigm Standard (Luís Norton/Universidade de Aveiro)

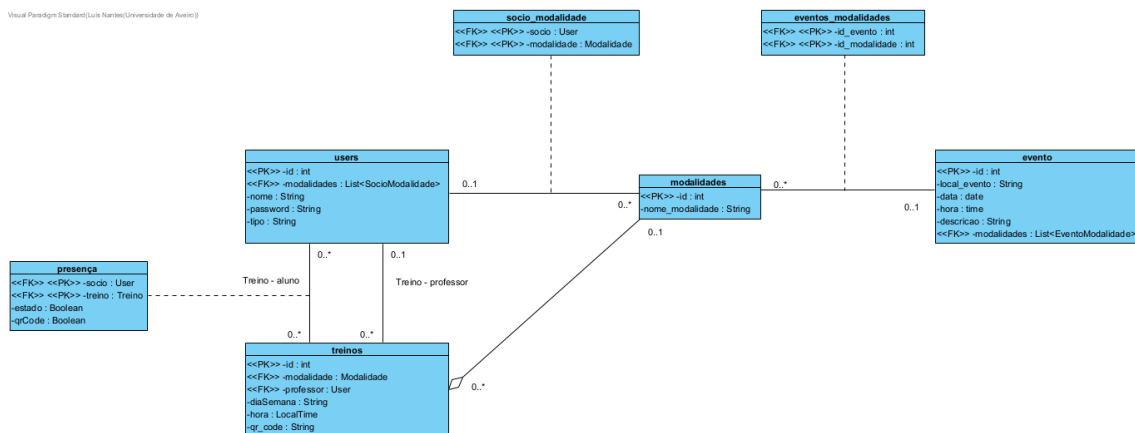


Figura 4-Diagrama Conceitual da Base de Dados

2.7 Modelo de Navegação da Aplicação

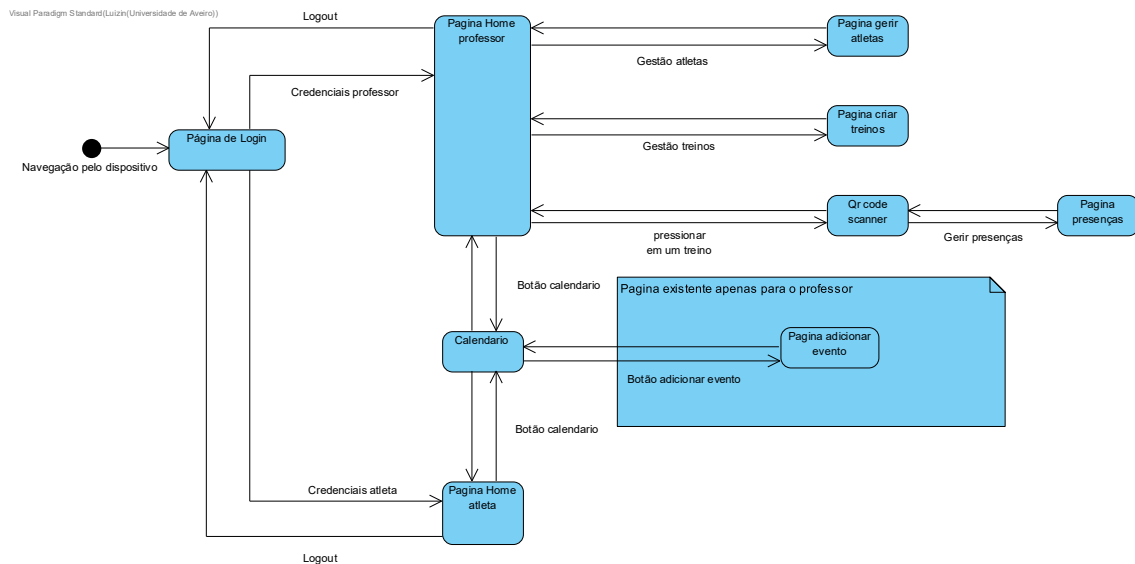


Figura 5-Representação da Navegação na Aplicação

3 Testes e Validações

3.1 Testes Unitários

Os **testes unitários** realizados abrangeram diversas funcionalidades do sistema, assegurando a **integridade** e o **comportamento esperado** das implementações. Foram testados tanto os **controladores** como os **view models**, incluindo métodos de **criação**, **listagem**, **atualização** e **eliminação** de entidades como **treinos**, **eventos**, **modalidades**, **presenças** e **utilizadores**.

Nos **controladores**, foi verificada a **interação correta com os repositórios simulados**, validando **cenários de sucesso e falha**, como:

- Criação de treinos e eventos com dados válidos,
- Eliminação de entidades inexistentes,
- Manipulação de associações entre entidades.

Nos **view models**, foram testados:

- Os **estados iniciais**,
- **Carregamento de dados simulados**,
- **Resposta a ações do utilizador**, como:
 - Detecção de dias da semana,
 - Carregamento de treinos e modalidades,
 - Adição de eventos.

Adicionalmente, foram simuladas **situações de erro**, como **falhas de conexão** e **dados duplicados**, para validar o **tratamento de exceções** e garantir a **robustez do sistema**.

Todos os testes foram realizados com **dados hardcoded** e **mocks**, permitindo isolar as funcionalidades testadas e assegurar que cada **componente funcione corretamente de forma independente**.

3.2 Testes Estruturais

*Os testes estruturais realizados podem ser consultados no **Anexo A: Testes Estruturais***

4 Análise dos resultados

O desenvolvimento da aplicação móvel **AthleTrack**, destinada à gestão de presenças em academias multidesportivas, pode ser considerado **bem-sucedido**. A aplicação atingiu os seus **principais objetivos** funcionais dentro do prazo estipulado, oferecendo uma interface **clara e funcional** tanto para atletas como para professores.

Durante o desenvolvimento da aplicação, foi dada especial atenção à **experiência do utilizador** e à **integração eficiente com o backend**, assegurada através da implementação de funcionalidades essenciais como **autenticação de utilizadores**, **visualização de treinos futuros** e **registo de presenças por QR Code**. A escolha de tecnologias modernas, nomeadamente **Jetpack Compose** no frontend e **Spring Boot com Kotlin** e **PostgreSQL** no backend, permitiu a construção de uma base sólida, modular e escalável.

Apesar do sucesso alcançado, **algumas funcionalidades inicialmente previstas não foram implementadas na aplicação móvel**, nomeadamente:

- A **análise de assiduidade** dos atletas por parte dos professores;
- As **estatísticas de presença e desempenho** para os alunos;
- A implementação dos **perfis individuais** de utilizador (aluno e professor);
- O **ecrã de definições da aplicação**, que permitiria ajustar preferências ou gerir sessões.

Após uma análise prática da utilização real da aplicação numa academia desportiva, concluiu-se que certas **operações administrativas e de análise de dados**, como a gestão de credenciais, treinos ou estatísticas detalhadas, seriam mais adequadas a uma **plataforma web de administração** ou a uma **aplicação de secretariado para computador**. Esta decisão reflete uma adaptação sensata às necessidades do utilizador final e permite uma **separação lógica entre a interface móvel**, orientada ao utilizador comum, e as **ferramentas administrativas**, destinadas ao corpo técnico da academia.

Importa ainda destacar que todas as atividades do projeto foram **realizadas dentro do cronograma estabelecido**, demonstrando um bom planeamento, organização e cumprimento dos objetivos definidos desde o início.

O projeto revelou-se, assim, uma **base funcional robusta**, com clara **margem para futuras extensões**, como o desenvolvimento da **plataforma administrativa**, a complementação das funcionalidades em falta e a **melhoria contínua da experiência do utilizador**.

*O Manual de Utilização da aplicação pode ser consultados no **Anexo B: Manual de Utilização***

5 Anexos

Anexo A - Testes Estruturais:

Funcionalidade/Teste	Status	Resultado
Criação de atleta com apenas a modalidade	Êxito	O atleta não é criado e é exibido uma mensagem de erro
Criação de atleta com apenas a password e modalidade	Êxito	O atleta não é criado e é exibido uma mensagem de erro
Criação de atleta com apenas a password	Êxito	O atleta não é criado e é exibido uma mensagem de erro
Criação de atleta com apenas o nome e modalidade	Êxito	O atleta não é criado e é exibido uma mensagem de erro
Criação de atleta com apenas o nome e password	Êxito	O atleta não é criado e é exibido uma mensagem de erro
Criação de atleta com apenas o nome	Êxito	O atleta não é criado e é exibido uma mensagem de erro
Criação de atleta com todos os parâmetros preenchidos	Êxito	O atleta é criado, adicionado à BD e é exibido uma mensagem indicando seu número de socio
Criação de atleta sem nenhum parâmetro	Êxito	O atleta não é criado e é exibido uma mensagem de erro
Criação de treino com todos os parâmetros preenchidos	Êxito	O treino é criado e adicionado à BD
Criação de treino sem definir dia e modalidade	Êxito	O treino não é criado e é exibido uma mensagem de erro
Criação de treino sem definir dia	Êxito	O treino não é criado e é exibido uma mensagem de erro
Exibição de aulas do dia de hoje na tela do professor	Êxito	Caso tenha aulas, elas são exibidas. Caso contrário a mensagem "Sem aulas para hoje." é exibida
Leitura de um QR code valido	Êxito	A aplicação marca a presença do aluno no treino

Login com apenas número de socio	Êxito	É exibido uma mensagem de erro
Login com apenas senha	Êxito	É exibido uma mensagem de erro
Login com credenciais existentes	Êxito	O login é concluído
Login com credenciais inválidas	Êxito	É exibido uma mensagem de erro
Login sem credenciais	Êxito	É exibido uma mensagem de erro
Pressionar em alguma aula na lista de aulas de hoje ou amanhã	Êxito	É exibido o QR Code da aula
Pressionar o botão "Adicionar Evento" com um dia selecionado previamente	Êxito	A aplicação exibe a página de adicionar eventos com a data definida para o dia selecionado
Pressionar o botão "Adicionar Evento" sem selecionar um dia	Êxito	A aplicação exibe a página de adicionar eventos com a data definida para o dia de hoje
Pressionar o botão "Gerir presenças"	Êxito	A aplicação exibe a tela de gestão de presenças
Pressionar o botão "Gestão de Atletas" na tela do professor	Êxito	A aplicação exibe a tela de gestão de atletas
Pressionar o botão "Gestão de Treinos" na tela do professor	Êxito	A aplicação exibe a tela de criação de treinos
Pressionar o botão cancelar no pop-up de eliminação de atletas	Êxito	O Atleta não é eliminado e a aplicação retorna a página de gestão de atletas
Pressionar o botão de calendário na página do Atleta	Êxito	A aplicação exibe a página do calendário e exibe os eventos existentes
Pressionar o botão de calendário na página do Professor	Êxito	A aplicação exibe a página do calendário
Pressionar o botão de criar evento com apenas a descrição preenchida	Êxito	A aplicação não cria o evento e exibe uma página de erro
Pressionar o botão de criar evento com apenas a modalidade preenchida	Êxito	A aplicação não cria o evento e exibe uma página de erro

Pressionar o botão de criar evento com apenas o local preenchido	Êxito	A aplicação não cria o evento e exibe uma página de erro
Pressionar o botão de criar evento com todos os parâmetros preenchidos	Êxito	A aplicação cria um evento no calendário
Pressionar o botão de criar evento sem nenhum parâmetro preenchido	Êxito	A aplicação não cria o evento e exibe uma página de erro
Pressionar o botão de eliminar atletas	Êxito	É exibido o pop-up de confirmar eliminação
Pressionar o botão de leitura de QR code na página de atleta	Êxito	A aplicação faz uso da câmara do telefone que irá ler o qr code
Pressionar o botão de logout	Êxito	A aplicação retorna para a página de login
Pressionar o botão de voltar	Êxito	A aplicação retorna para a página principal
Pressionar o botão eliminar no pop-up de eliminação de atletas ao inserir a password do professor	Êxito	O Atleta é eliminado e a aplicação retorna a página de gestão de atletas
Pressionar o botão eliminar no pop-up de eliminação de atletas ao inserir uma password invalida	Êxito	O Atleta não é eliminado e a aplicação retorna a página de gestão de atletas
Pressionar o botão eliminar no pop-up de eliminação de atletas sem inserir a password do professor	Êxito	O Atleta não é eliminado e a aplicação retorna a página de gestão de atletas
Pressionar o ícone de utilizador ao lado do nome do utilizador	Êxito	É exibido a opção de logout
Pressionar o botão que alterna entre os treinos de hoje e amanhã	Êxito	As aulas exibidas alternam entre os dias selecionados

Anexo B – Manual de Utilização

O Manual de Utilização da aplicação **AthleTrack** encontra-se disponível em ficheiro separado, incluído na mesma pasta que este relatório.

Este manual complementa a presente documentação, apresentando passo a passo o funcionamento da aplicação do ponto de vista do utilizador (aluno e professor), com descrições de ecrãs, funcionalidades e instruções de uso.