TAÇA UA

Relatório Técnico

Autores:
Bernardo Borges 103592
Alexandre Regalado 124574
Mateus Rocha 122949
Dinis Sousa 119500
Diego Aguilar 117122
Gonçalo Simões 119412

Orientadores:
Osvaldo Pacheco
Carlos Costa
João Luís

Abstract

Taça UA represents one of the most significant academic sporting events at the University of Aveiro, involving extensive organizational coordination and community engagement. Ensuring efficient management is essential to sustaining its operational quality and institutional relevance.

Despite its scale, the competition is currently managed through fragmented tools and informal communication channels, resulting in inconsistent data handling, scheduling challenges, and limited transparency. No integrated digital platform exists that addresses the specific administrative and public needs of the event.

This project aims to conceptualize and specify a unified, web-based management system capable of centralizing operational workflows, improving information reliability, and supporting the diverse requirements of administrators, athletes and the general public.

The study employs requirement elicitation, comparative benchmarking of existing sports-management platforms, persona-based modelling and the development of functional specifications, non-functional requirements, user stories, and use-case diagrams to establish a rigid initial structure.

Analysis indicates a strong need for automation in specific areas, standardized data governance and real-time dissemination of the schedules, results and classifications. The system architecture derived from the study organizes responsibilities into three role-based profiles and modules covering tournaments, teams, scoring, scheduling and regulation management.

The proposed solution significantly reduces organizational overhead, enhances the accuracy and timeliness of competition data, and strengthens transparency for all users. It provides a scalable digital foundation for the modernization and long-term evolution of academic sports events of the University of Aveiro.

Lista de Abreviaturas

Sigla	Significado
AAUAv	Associação Académica da Universidade de Aveiro
API	Application Programming Interface
CD	Continuous Deployment
CI	Continuous Integration
CSS	Cascading Style Sheets
FADU	Federação Académica do Desporto Universitário
HTTP	Hypertext Transfer Protocol
JWT	JSON Web Token
MVC	Model–View–Controller
NGINX	Engine X (servidor web e reverse proxy)
OIDC	OpenID Connect
PDF	Portable Document Format
RBAC	OpenID Connect
REST	Representational State Transfer
SOLID	Conjunto de princípios de design orientado a objetos
TLS	Transport Layer Security
UA	Universidade de Aveiro
UML	Unified Modeling Language

1. Introdução

O presente relatório técnico documenta o trabalho realizado pelo grupo 15 no âmbito da unidade curricular de Projeto em Engenharia Informática (43675) da Licenciatura em Engenharia Informática, durante o ano letivo de 2025/2026. O projeto foi realizado sobre a orientação dos Prof. Dr. Carlos Costa, Prof. Dr. Osvaldo Pacheco e João Luís.

1.1. Contexto

A gestão da taça UA constitui atualmente um processo complexo, fragmentado e fortemente dependente de comunicação informal. A ausência de um sistema unificado leva à utilização simultânea de múltiplas ferramentas, originando dificuldades significativas na coordenação entre organizadores, núcleos e participantes.

No que diz a respeito à organização das modalidades e dos torneios, identificam-se várias limitações práticas. A calendarização dos jogos, frequentemente construída manualmente, é propensa a erros e exige constantes atualizações distribuídas por diferentes canais. A gestão de equipas e jogadores revela-se igualmente morosa, com o registo e validação de atletas, a associação de equipas por núcleo e a verificação do estatuto de sócio dependentes de processos repetitivos e não integrados. A publicação de regulamentos carece de um repositório central, resultando a inconsistências e atualizações demoradas.

A experiência dos administradores de núcleo enfrenta desafios paralelos. A submissão de jogadores elegíveis para cada jogo e o preenchimento da ficha de jogo envolvem tarefas redundantes e frequentemente duplicadas. A inexistência de mecanismos automáticos para validação de informação, como o estatuto de sócio, contribui ainda para ambiguidades e atrasos operacionais.

De ponto de vista do publico, a consulta de resultados, calendários e classificações encontra-se distribuída por várias páginas e publicações informais, dificultando uma perceção clara do estado da competição. A falta de transparência e centralização afeta a experiência global da comunidade académica, reduzindo o impacto e visibilidade do evento.

É neste cenário de ineficiências acumuladas que se insere o presente projeto. A proposta visa no desenvolvimento de uma plataforma web integrada que centralize a gestão de modalidades, torneios, equipas e resultados, garantindo atualizações em tempo real, uniformização da informação e uma experiência mais fluida para administradores, estudantes e público.

1.2. Objetivos

O desenvolvimento da plataforma digital para a Taça UA tem como propósito principal modernizar, unificar e otimizar as operações essenciais à gestão da competição, garantindo melhor eficiência, transparência e coerência organizacional. Assim, os objetivos do sistema são:

- Centralizar a gestão da Taça UA numa única plataforma web, eliminando a necessidade de ferramentas dispersas e comunicação manual.
- Apoiar a organização desportiva através de módulos dedicados, incluindo gestão de modalidades, torneios, equipas, jogos e classificações.
- Automatizar o cálculo de pontuações e classificações, reduzindo erros manuais e assegurando atualizações consistentes em tempo real.
- Disponibilizar informação publica atualizada, como calendários, resultados, classificações gerais e regulamentos, promovendo uma maior transparência para participantes e jogadores
- Facilitar a operação dos Administradores de Núcleo através de funcionalidades como a criação de equipas, gestão de jogadores e geração de fichas de jogo.

Estes objetivos refletem a necessidade de transformar a gestão da Taça UA num processo mais estruturado, eficiente e sustentável, alinhando com os requisitos modernos de informação, coordenação e fiabilidade num evento desportivo académico.

1.3. Estrutura do documento

Este relatório encontra-se organizado de forma a apresentar, de maneira clara e progressiva, o desenvolvimento do sistema de gestão da Taça UA, desde o enquadramento teórico e metodológico até à implementação e análise dos resultados obtidos. A estrutura do documento é a seguinte:

1. Estado da Arte: Apresenta o enquadramento do problema e uma análise comparativa de soluções existentes no domínio da gestão de competições e eventos desportivos académicos, identificando limitações e oportunidades que fundamentam a proposta do sistema desenvolvido.

2. Metodologia: Descreve a abordagem metodológica adotada ao longo do projeto, incluindo as ferramentas de comunicação, organização da equipa e gestão de tarefas, bem como as práticas utilizadas para garantir coordenação e acompanhamento do desenvolvimento.

3. Requisitos do Sistema: Define os requisitos do sistema com base no levantamento efetuado junto dos intervenientes, incluindo a identificação de atores, personas e casos de uso, bem como a especificação dos requisitos funcionais e não funcionais que orientaram o desenvolvimento da solução.

4. Arquitetura: Apresenta a arquitetura geral da solução, descrevendo a separação entre frontend, backend e serviços de suporte, bem como as principais decisões arquiteturais adotadas para garantir escalabilidade, segurança e manutenibilidade.

5. Implementação: Detalha o processo de implementação do sistema, abordando a gestão de repositórios, tecnologias utilizadas no frontend e backend, práticas de DevOps e CI/CD, bem como o modelo de dados adotado.

6. Documentação da API: Descreve a interface de programação disponibilizada pelo sistema, incluindo os principais endpoints, formatos de dados e mecanismos de autenticação, servindo de suporte à integração e manutenção futura.

7. Conclusão: Sintetiza os principais contributos do projeto, avaliando o cumprimento dos objetivos propostos e apontando direções para trabalho futuro.

8. Referências: Lista das fontes bibliográficas, documentação técnica e outros recursos utilizados ao longo do projeto.

2. Estado de Arte

A organização e gestão de competições desportivas académicas têm vindo a beneficiar de soluções digitais que procuram centralizar informação, automatizar processos e melhorar a comunicação entre entidades organizadoras, participantes e público. Em contexto nacional, a Federação Académica do Desporto Universitário (FADU) disponibiliza plataformas que suportam a gestão de competições universitárias a nível interinstitucional, servindo como referência funcional para eventos desportivos académicos de grande escala.

Contudo, a Taça UA apresenta características organizacionais específicas, nomeadamente a gestão por cursos, a existência de múltiplas modalidades em simultâneo e a forte dependência de administradores locais (Núcleos), que não são totalmente contempladas pelas soluções existentes. Atualmente, a gestão da Taça UA assenta num conjunto disperso de ferramentas, como folhas de cálculo, formulários online e comunicação informal, resultando em processos manuais, redundância de informação e maior probabilidade de erro.

Neste contexto, torna-se pertinente analisar o estado da arte das soluções existentes, identificando as suas funcionalidades, limitações e o posicionamento do projeto proposto face às práticas atuais.

2.1. Análise Comparativa

A tabela abaixo resume as funcionalidades disponíveis em cada aplicação:

Critério	FADU	Taça UA (Atual)	Taça UA (Projeto)
Gestão de torneios / modalidades	SIM	PARCIAL (Excel + Google Forms)	SIM
Gestão por Núcleos	NÃO	MANUAL (Via Responsável)	SIM
Hierarquia Administrativa	NÃO	GESTAO INFORMAL	SIM
Agendamento de Jogos	SIM	MANUAL	SIM
Registo de resultados	SIM	MANUAL (Redes Sociais)	SIM
Pontuação automática por modalidade	NÃO	EXCEL	SIM
Classificação geral intermodalidades	NÃO	EXCEL	SIM
Inscrição de atletas	SIM	FORMS	SIM
Validação de sócios	NÃO	MANUAL	MANUAL
Fichas de Jogo (PDF)	NÃO	CRIADAS MANUALMENTE	FORA DA APLICAÇÃO
Histórico multi-época	SIM	EXCEL	SIM
Calendário público com filtros	SIM	NÃO	SIM

Tabela 2.1: Comparação de funcionalidades entre FADU, a solução atual da Taça UA e do projeto da Taça UA

2.2. Análise das Soluções Existentes

A plataforma da FADU constitui uma referência no contexto do desporto universitário nacional, oferecendo suporte à gestão de modalidades, torneios, inscrições de atletas e resultados. Destaca-se pela disponibilização de calendários públicos e históricos multi-época. No entanto, trata-se de uma solução orientada para competições interuniversitárias, não contemplando a gestão interna por núcleos, nem uma hierarquia administrativa adaptada à realidade organizacional da Taça UA.

A versão atual da Taça UA recorre a um conjunto de ferramentas heterogéneas, incluindo folhas de cálculo, formulários online e redes sociais. Embora estas permitam assegurar o funcionamento básico da competição, grande parte dos processos, como o agendamento de jogos, o registo de resultados, o cálculo de pontuações e a gestão de equipas, é realizada manualmente. Esta abordagem dificulta a consistência dos dados, aumenta a carga administrativa e compromete a transparência e atualização em tempo real da informação disponibilizada ao público.

O projeto da Taça UA propõe uma solução integrada e centralizada, concebida especificamente para responder às necessidades organizacionais da competição. A plataforma suporta uma hierarquia administrativa clara, com distinção entre Administrador Geral e Administradores de Núcleo, automatiza o cálculo de pontuações e classificações, centraliza a gestão de equipas e jogos, e disponibiliza informação pública filtrável e atualizada em tempo real. Embora algumas funcionalidades, como a validação de sócios e a geração final das fichas de jogo em PDF, ainda dependam de processos externos ou complementares, a solução representa um avanço significativo face às práticas atuais.

2.3. Síntese Crítica

A análise comparativa evidencia que, apesar da existência de plataformas consolidadas como a da FADU, estas não respondem de forma plena às especificidades da Taça UA. Por outro lado, a abordagem atualmente utilizada pela organização carece de integração, automatização e escalabilidade. O projeto proposto posiciona-se, assim, como uma solução intermédia e especializada, combinando boas práticas existentes com funcionalidades adaptadas ao contexto académico local, contribuindo para uma gestão mais eficiente, transparente e digitalmente sustentável da Taça UA.

3. Metodologia

Ao longo do desenvolvimento da plataforma de gestão da Taça UA, optou-se por uma abordagem incremental e colaborativa. O trabalho foi estruturado em diferentes fases, desde a definição e análise dos requisitos até à sua correta implementação, garantindo a coerência entre os objetivos definidos e as soluções propostas.

Em relação à coordenação e comunicação envolvidas no projeto, foram utilizadas principalmente duas ferramentas digitais: o Discord, como meio de comunicação, e o Jira para a gestão de tarefas. Os aspetos referidos serão aprofundados ao longo das secções seguintes.

3.1. Ferramentas de Comunicação

A comunicação entre os membros da equipa, o cliente e os orientadores revelou-se fundamental ao longo das diversas etapas de desenvolvimento do projeto. Esta ocorreu tanto de forma presencial como virtual, assegurando um acompanhamento contínuo e eficaz.

Conforme referido anteriormente, a ferramenta mais utilizada foi o Discord, que possibilitou a troca de informação e de recursos relevantes, a discussão de decisões e o acompanhamento constante do progresso do projeto. Foram criados dois canais distintos: um que incluía o cliente, destinado à recolha de requisitos, partilha de progressos e esclarecimento de dúvidas; e outro restrito aos membros da equipa, mais focado na organização e planeamento das tarefas. Adicionalmente, através desta plataforma, foram realizadas reuniões remotas entre os membros da equipa, recorrendo a chamadas de voz e à partilha de ecrã, o que facilitou a análise conjunta de documentos e código.

No que respeita à comunicação com os orientadores, recorreu-se à plataforma Zoom para a realização de reuniões virtuais, complementadas por diversos encontros presenciais. Cada uma destas reuniões foi fundamental para o progresso do projeto, permitindo esclarecer dúvidas relacionadas com a estrutura do trabalho, a definição de prioridades e os aspetos a melhorar.

Tudo o que foi descrito foi transcendental no processo de validação das decisões e no alinhamento do projeto com os objetivos e requisitos previamente definidos.

3.2. Gestão de tarefas em Jira

As diversas tarefas em cada uma das etapas do projeto foram geridas com a ajuda do Jira, permitindo organizar o projeto de forma ordenada e acompanhar de forma clara a sua evolução.

Cada um dos diferentes aspetos a desenvolver foi registado na plataforma, originando um quadro do tipo *Kanban*, organizado em colunas que representavam os diferentes estados das tarefas (to do, in progress, in review, in testing, done). Esta estrutura facilitou a visualização do estado global do projeto, bem como a identificação atempada de obstáculos ou atrasos que pudessem surgir ao longo do seu desenvolvimento. Cada uma das tarefas criadas foi atribuída a um membro da equipa, de acordo com a sua função no projeto, o que permitiu uma distribuição clara das responsabilidades e facilitou a identificação do progresso individual e coletivo.

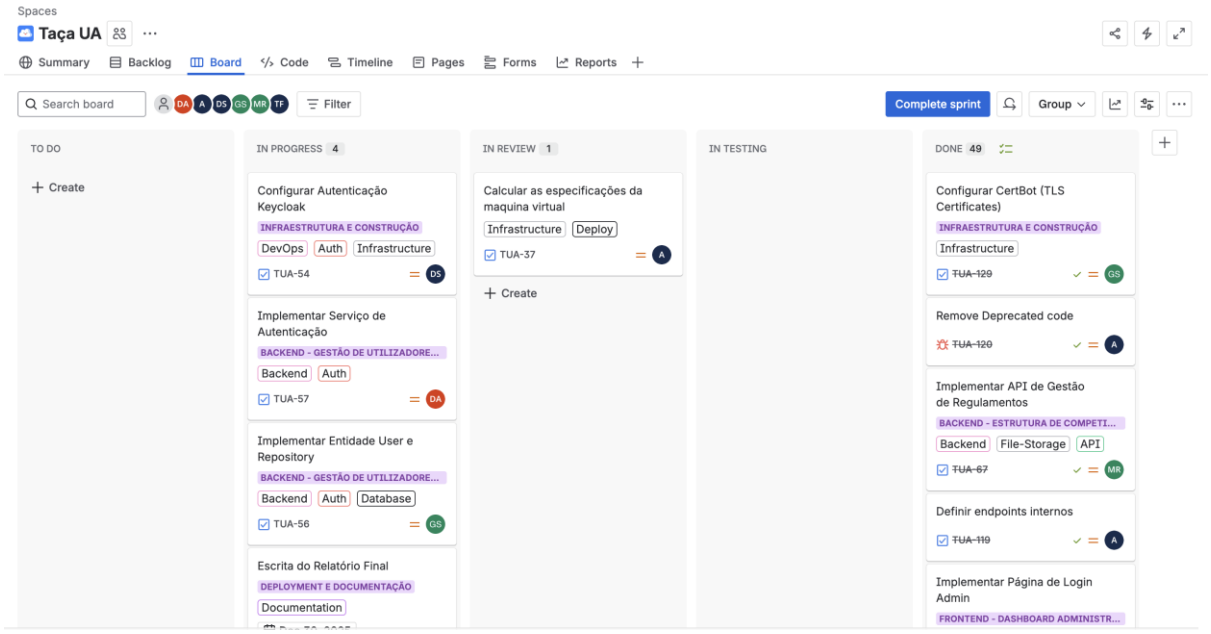

Figura 3.1: Board do Jira utilizada na gestão das tarefas do projeto.

3.3. Equipa

A equipa de desenvolvimento foi constituída por seis elementos, todos estudantes da Licenciatura em Engenharia Informática, que colaboraram ativamente em todas as etapas do desenvolvimento do projeto.

Cada um dos membros assumiu funções e responsabilidades específicas, de acordo com as suas competências e interesses individuais.

Membro	**Função**
Bernardo Borges	Project Owner, Frontend
Alexandre Regalado	DevOps, Backend
Mateus Rocha	Arquiteto, Backend
Gonçalo Simões	Frontend
Diego Aguilar	Frontend
Dinis Sousa	Backend

Tabela 3.1: Tabela das funções de cada membro da equipa

4. Requisitos do Sistema

4.1. Levantamento de requisitos

O levantamento de requisitos do sistema foi realizado com base na análise do contexto atual da gestão da Taça UA, em reuniões informais com os responsáveis pela organização, bem como na observação dos processos atualmente utilizados, maioritariamente suportados por folhas de cálculo, formulários externos e comunicação manual. Este processo permitiu identificar necessidades funcionais e não funcionais associadas à gestão administrativa da competição, à organização das equipas e jogos, e ao acesso público à informação. De forma a garantir que o sistema responde de forma adequada às necessidades reais dos utilizadores, foi adotada uma abordagem centrada no utilizador, recorrendo à definição de atores e personas representativas dos principais perfis de utilização da plataforma.

4.2. Requisitos funcionais

Os requisitos funcionais definem as funcionalidades que o sistema deve disponibilizar de forma a suportar a gestão integral da Taça UA, abrangendo tanto as necessidades administrativas como o acesso público à informação. Estes requisitos encontram-se organizados por áreas funcionais, facilitando a compreensão das responsabilidades associadas a cada tipo de utilizador e garantindo uma cobertura completa dos processos identificados durante o levantamento de requisitos.

4.2.1. Gestão de Utilizadores e Autenticação

- O sistema deve permitir a autenticação de Administradores Gerais e Administradores de Núcleo através de credenciais válidas.
- O sistema deve atribuir permissões de acesso de acordo com o perfil do utilizador, assegurando uma separação clara de responsabilidades.
- O Administrador Geral deve ser capaz de criar, gerir e remover contas de Administradores de Núcleo.
- O público em geral deve conseguir aceder à informação pública da plataforma sem necessidade de autenticação.

4.2.2. Gestão da Competição

- O Administrador Geral deve ser capaz de criar, editar e remover cursos participantes na Taça UA.
- O Administrador Geral deve poder associar Administradores de Núcleo a cada curso.
- O sistema deve permitir a gestão e publicação de regulamentos gerais da competição, bem como regulamentos específicos por modalidade.
- O Administrador Geral deve ser capaz de iniciar e terminar épocas desportivas, definindo o ciclo competitivo de cada edição da Taça UA.

4.2.3. Gestão de Modalidades, Torneios e Jogos

- O Administrador Geral deve ser capaz de criar e configurar torneios para as diferentes modalidades.
- O sistema deve permitir o agendamento e edição de jogos, incluindo a definição de equipas, data, hora e local.
- O Administrador Geral deve poder gerir o calendário completo de jogos de cada modalidade.
- O sistema deve permitir o registo e atualização de resultados dos jogos.
- O Administrador Geral deve poder definir o sistema de pontuação de cada modalidade e tipo de torneio, em conformidade com os regulamentos aplicáveis.

4.2.4. Gestão de Equipas e Atletas por Curso

- O Administrador de Núcleo deve ser capaz de criar, editar e remover jogadores associados ao seu curso.
- O Administrador de Núcleo deve poder criar e gerir equipas por modalidade.
- O sistema deve permitir a inscrição de atletas nas equipas, incluindo o respetivo número mecanográfico.
- O sistema deve verificar automaticamente se o atleta é sócio, com base numa lista fornecida pelo Administrador Geral, sinalizando visualmente situações de incumprimento.

4.2.5. Gestão de Jogos ao Nível do Curso

- O Administrador de Núcleo deve ser capaz de atribuir aos jogos apenas os jogadores elegíveis pertencentes às equipas relevantes.

- O sistema deve permitir a atribuição de números de equipamento aos jogadores convocados.
- O Administrador de Núcleo deve poder adicionar comentários associados a cada jogo.
- O sistema deve permitir a geração de fichas de jogo em formato PDF, contendo toda a informação relevante do encontro.

4.2.6. Transparência, Consulta Pública e Comunicação

- O sistema deve disponibilizar um calendário completo da competição, organizado por dias.
- O público deve poder filtrar o calendário por modalidade e/ou curso.
- Os resultados dos jogos devem ser públicos e atualizados automaticamente.
- O público deve poder consultar classificações por modalidade e a classificação geral da Taça UA.
- O sistema deve permitir o acesso a regulamentos gerais e específicos.
- Deve ser possível consultar resultados e classificações de edições anteriores da competição.

4.2.7. Sistema de Pontuação e Classificação Geral

- O sistema deve calcular automaticamente a pontuação dos cursos por modalidade.
- O sistema deve atualizar de forma automática e consistente a Classificação Geral da Taça UA.

4.3. Requisitos não funcionais

Os requisitos não funcionais definem os critérios de qualidade do sistema, assegurando níveis adequados de desempenho, usabilidade, segurança, fiabilidade e capacidade de evolução. Estes requisitos são essenciais para garantir a aceitação da plataforma pelos utilizadores e a sua sustentabilidade técnica a longo prazo.

4.3.1. Usabilidade

- A plataforma deve apresentar um design responsivo, adaptado a dispositivos desktop, tablet e smartphone.
- A interface deve ser intuitiva e utilizar terminologia reconhecida pela comunidade académica.
- As atualizações de dados devem ocorrer de forma automática, sem necessidade de atualização manual da página.
- A navegação deve permitir o acesso a qualquer funcionalidade principal a partir da página principal em, no máximo, quatro cliques.
- Devem existir mecanismos de pesquisa e filtros eficientes para calendários, equipas e resultados.

4.3.2. Desempenho

- O tempo de resposta do sistema deve ser inferior a dois segundos em 95% das requisições.
- Devem ser implementadas técnicas de cache para reduzir a carga associada a consultas frequentes, como calendários e resultados.
- A arquitetura do sistema deve suportar escalabilidade horizontal, permitindo acomodar um aumento futuro de utilizadores e dados.

4.3.3. Compatibilidade

- A plataforma deve ser compatível com os navegadores mais utilizados, nomeadamente Chrome, Firefox, Edge e Safari.

4.3.4. Manutenibilidade e Evolução

- O código deve seguir boas práticas de desenvolvimento, como os princípios SOLID e arquiteturas MVC e RESTful.
- O sistema deve permitir atualizações contínuas com tempo de indisponibilidade mínimo.
- A documentação técnica deve ser atualizada a cada nova versão do sistema.
- A arquitetura deve permitir a inclusão futura de novas modalidades e categorias sem necessidade de reestruturação profunda.
- Devem existir manuais de utilizador dirigidos a administradores sem conhecimentos técnicos.

4.4. Atores e Personas

De forma a compreender melhor a interação entre os utilizadores e o sistema, foram identificados os principais atores e definidas personas representativas, permitindo alinhar os requisitos técnicos com necessidades reais de utilização.

4.4.1. Atores

Os atores do sistema correspondem aos diferentes tipos de utilizadores que interagem com a plataforma, cada um com permissões e responsabilidades distintas:

- Administrador Geral: responsável pela gestão global da competição, incluindo modalidades, torneios, jogos, regulamentos e classificação geral.
- Administrador de Núcleo: responsável pela gestão das equipas e atletas de um curso específico.
- Utilizador Público: qualquer utilizador sem autenticação que acede à plataforma para consultar informação pública.
- Sistema: ator lógico responsável por processos automáticos, como o cálculo de pontuações e atualização de classificações.

4.4.2. Personas

De forma a compreender melhor os objetivos, motivações e dificuldades dos diferentes tipos de utilizadores, foram definidas várias personas representativas, descritas de seguida.

Michael Hondt – Estudante / Participante

- Idade: 22 anos
- Curso: Química
- Papel: Participante na Taça UA e utilizador público da plataforma

Michael participa regularmente na Taça UA e acompanha ativamente o desempenho do seu curso. Atualmente, sente frustração devido à dispersão da informação e aos atrasos na divulgação de resultados e horários. Procura uma plataforma centralizada onde possa consultar rapidamente calendários, resultados e classificações atualizadas em tempo real, de forma simples e intuitiva.

Lahra Cough – Administradora Geral

- Idade: 26 anos
- Cargo: Membro da Direção da AAUAv
- Papel: Administradora Geral da plataforma

Lahra integra a comissão organizadora da Taça UA e é responsável pela coordenação global da competição. No modelo atual, depende fortemente de folhas de Excel, formulários e comunicação manual, o que compromete a eficiência e o controlo do processo. O seu objetivo principal é gerir modalidades, cursos e responsáveis, torneios, equipas e regulamentos num único sistema digital, reduzindo erros e tempo administrativo.

Zach Daniel – Administrador Geral

- Idade: 24 anos
- Cargo: Membro da Administração da AAUAv
- Papel: Administrador Geral, responsável pelas modalidades

Zach desempenha funções de gestão transversal das modalidades da Taça UA. É responsável pela criação de torneios, agendamento de jogos, atualização de resultados e validação das classificações. Enfrenta dificuldades associadas a atrasos na atualização de dados e correções manuais frequentes. Procura uma solução que lhe permita operar com rapidez, precisão e confiança, garantindo classificações automáticas e comunicação imediata com os participantes.

Teelor Shift – Administradora de Núcleo

- Idade: 24 anos
- Curso: Engenharia Informática
- Papel: Representante do Núcleo de Engenharia Informática

Teelor é responsável pela gestão das equipas do seu núcleo, incluindo a inscrição de atletas e a organização das equipas por modalidade. A comunicação fragmentada e alterações tardias nos horários afetam a logística das equipas. O seu objetivo é dispor de uma plataforma que lhe permita gerir jogadores, equipas e fichas de jogo de forma clara, centralizada e fiável.

Clara Almeida – Utilizadora Pública

- Idade: 19 anos
- Curso: Engenharia Mecânica
- Papel: Utilizadora pública da plataforma

Clara não participa diretamente na competição, mas acompanha os jogos para apoiar o seu curso. Sente dificuldade em encontrar horários e locais atualizados através dos meios atuais. Procura uma plataforma simples, acessível e visualmente clara, onde possa consultar rapidamente os jogos do dia, resultados e classificações.

4.4.3. Casos de Uso

Na Figura 4.1 apresenta-se o diagrama UML de casos de uso do sistema proposto para a gestão da Taça UA. Este diagrama ilustra as principais interações entre os diferentes atores e a plataforma, evidenciando as funcionalidades disponibilizadas a cada perfil de utilizador.

Os casos de uso foram definidos com base no levantamento de requisitos e nas responsabilidades associadas a cada ator, permitindo descrever de forma clara e estruturada as operações suportadas pelo sistema. De forma complementar ao diagrama, a Tabela 4.1 apresenta a descrição detalhada de cada caso de uso identificado.

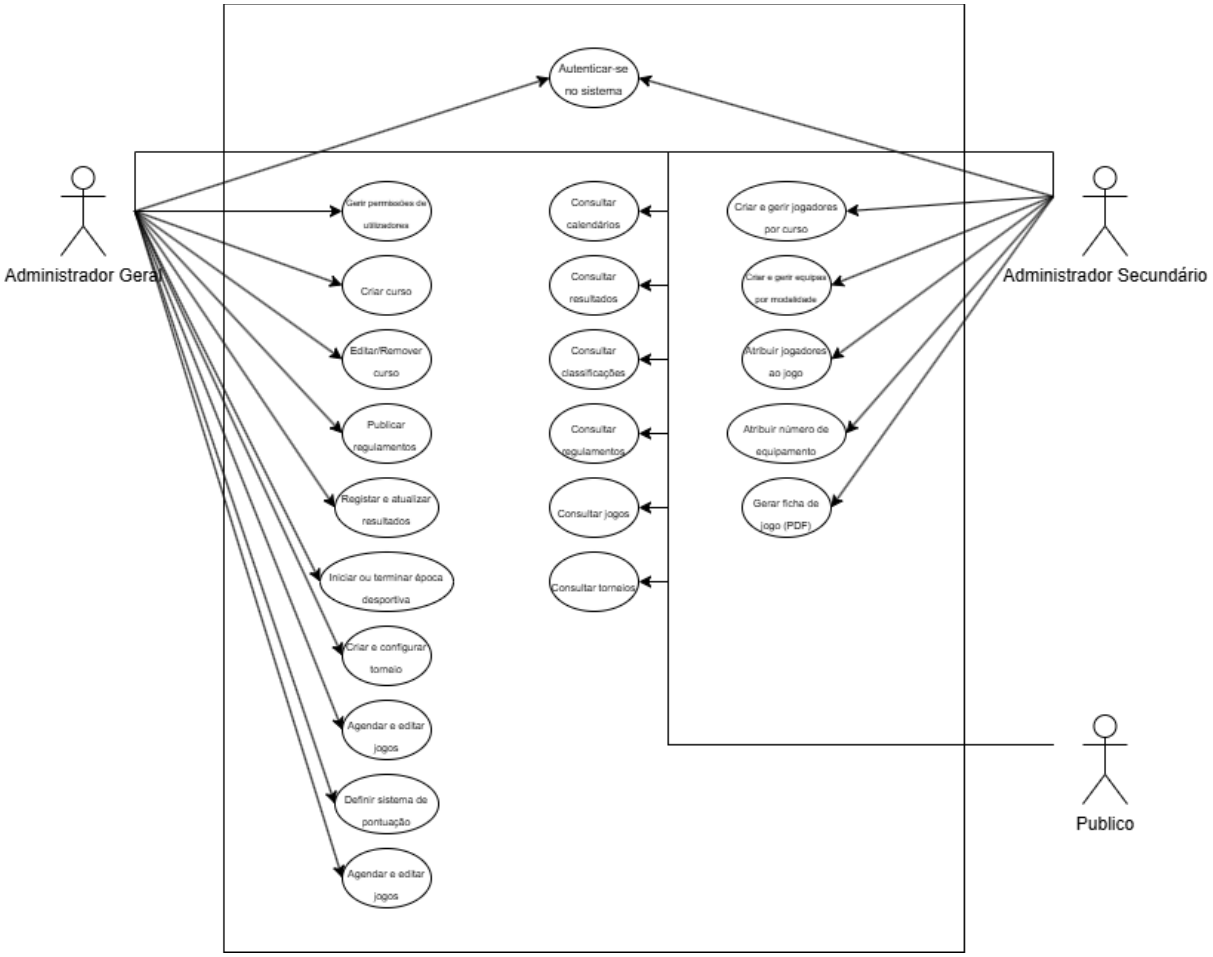

Figura 4.1: Diagrama de Casos de Uso do sistema de gestão da Taça UA.

ID	Ator	Caso de Uso	Descrição
UC1	Todos os Administradores	Autenticar-se no sistema	Permite ao Administrador Geral e ao Administrador de Núcleo autenticar-se na plataforma através de credenciais válidas, garantindo acesso seguro às funcionalidades restritas.
UC2	Administrador Geral	Gerir permissões de utilizadores	Permite atribuir e gerir permissões de acesso de acordo com o tipo de administrador, assegurando controlo hierárquico
UC3	Administrador Geral	Criar curso	Permite criar cursos participantes na Taça UA.
UC4	Administrador Geral	Editar ou remover curso	Permite alterar ou remover cursos previamente criados.
UC5	Administrador Geral	Publicar regulamentos	Permite gerir e publicar regulamentos gerais e específicos por modalidade.
UC6	Administrador Geral	Iniciar ou terminar época desportiva	Permite definir o ciclo competitivo de cada edição da Taça UA.
UC7	Administrador Geral	Criar e configurar torneios	Permite criar torneios por modalidade e definir a sua estrutura competitiva.
UC8	Administrador Geral	Agendar e editar jogos	Permite definir equipas, data, hora e local dos jogos, bem como efetuar alterações posteriores.
UC9	Administrador Geral	Definir sistema de pontuação	Permite configurar o sistema de pontuação por modalidade e tipo de torneio, conforme regulamentos.
UC10	Administrador Geral	Registar e atualizar resultados	Permite introduzir e corrigir resultados dos jogos, desencadeando atualizações automáticas das classificações.
UC11	Administrador Geral	Consultar torneios	Permite visualizar torneios existentes e respetiva informação associada.

ID	Ator	Caso de Uso	Descrição
UC12	Administrador Geral	Consultar jogos	Permite consultar jogos agendados ou já realizados.
UC13	Utilizador Público	Consultar calendários	Permite visualizar o calendário geral de jogos da competição.
UC14	Utilizador Público	Consultar calendário com filtros	Permite filtrar o calendário por modalidade e/ou curso, facilitando o acesso à informação relevante.
UC15	Utilizador Público	Consultar resultados	Permite consultar resultados atualizados dos jogos.
UC16	Utilizador Público	Consultar classificações	Permite visualizar classificações por modalidade e a classificação geral da Taça UA.
UC17	Utilizador Público	Consultar regulamentos	Permite aceder aos regulamentos gerais e específicos das competições.
UC18	Administrador de Núcleo	Criar e gerir jogadores por curso	Permite criar, editar e remover jogadores associados ao respetivo curso.
UC19	Administrador de Núcleo	Criar e gerir equipas por modalidade	Permite criar e organizar equipas do curso por modalidade.
UC20	Administrador de Núcleo	Atribuir jogadores ao jogo	Permite selecionar os jogadores elegíveis para participar num jogo específico
UC21	Administrador de Núcleo	Atribuir número de equipamento	Permite associar um número de equipamento a cada jogador convocado.
UC22	Administrador de Núcleo	Gerar ficha de jogo (PDF)	Permite gerar uma ficha de jogo em formato PDF contendo jogadores, números de equipamento e comentários associados.

Tabela 4.1: Tabela de Casos de Uso do sistema de gestão da Taça UA

5. Arquitetura

A arquitetura do sistema proposto para a gestão da Taça UA foi concebida com base numa abordagem modular, distribuída e orientada a serviços, tendo como principais objetivos a escalabilidade, a manutenibilidade e a separação clara de responsabilidades. A solução adota um modelo de sistema web multicamada, no qual as diferentes componentes comunicam através de interfaces bem definidas, recorrendo a APIs REST e a mecanismos de comunicação assíncrona baseados em eventos.

A Figura 5.1 apresenta a arquitetura global do sistema, ilustrando os principais módulos funcionais, os fluxos de comunicação estabelecidos entre eles e a interação com os diferentes tipos de utilizadores. A arquitetura encontra-se organizada em camadas distintas, nomeadamente: camada de apresentação, camada de autenticação e autorização, camada de orquestração e APIs, serviços de domínio, persistência de dados e observabilidade.

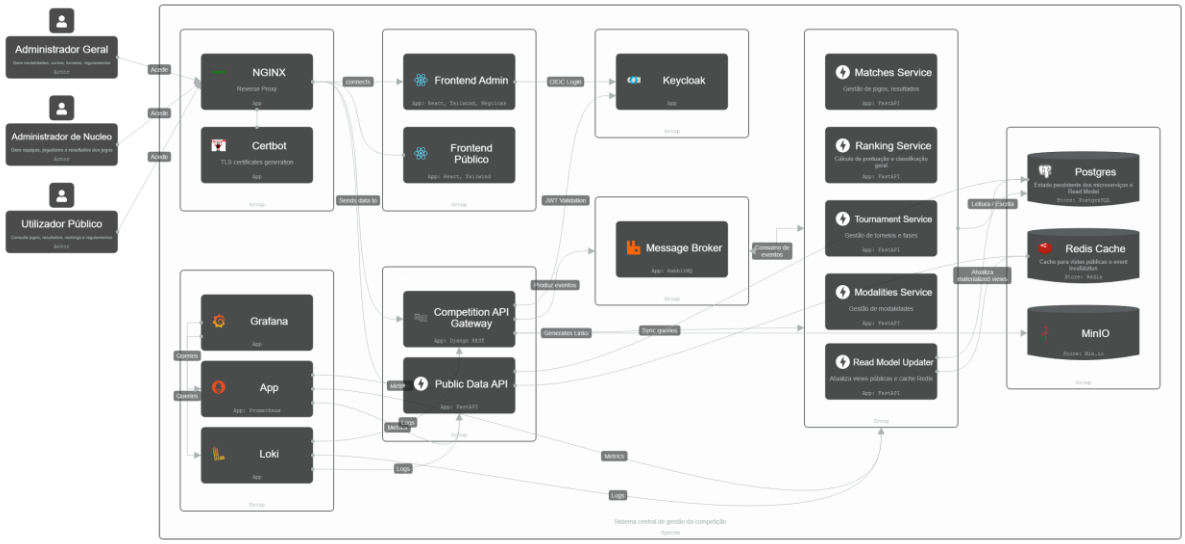

Figura 5.1: Visão geral da arquitetura do sistema

5.1. Camada de Apresentação

A camada de apresentação é responsável pela interação direta entre os utilizadores e o sistema, sendo composta por duas aplicações web distintas:

- Frontend Administrativo, destinado aos Administradores Gerais e Administradores de Núcleo;
- Frontend Público, acessível a utilizadores não autenticados.

Ambas as interfaces foram desenvolvidas com recurso à biblioteca React, em conjunto com Tailwind CSS, garantindo consistência visual, reutilização de componentes e adaptação a diferentes dispositivos. O acesso às aplicações é efetuado através de um NGINX Reverse Proxy, responsável pelo encaminhamento de pedidos HTTP, bem como pela integração com certificados TLS geridos automaticamente pelo Certbot, assegurando comunicações seguras.

5.2. Camada de Autenticação e Autorização

A autenticação e a autorização são elementos fundamentais do sistema, uma vez que a plataforma suporta diferentes perfis de utilizador com níveis distintos de acesso às suas funcionalidades. Para garantir a segurança e o controlo de acessos, o sistema recorre ao Keycloak como serviço centralizado de gestão de identidade.

A autenticação é realizada através do protocolo OpenID Connect (OIDC). Os utilizadores administrativos são redirecionados para o serviço de autenticação, onde validam as suas credenciais. Após autenticação bem-sucedida, é emitido um JSON Web Token (JWT), que identifica o utilizador e é utilizado nos pedidos subsequentes às APIs do sistema, permitindo uma arquitetura “stateless” e escalável.

A autorização baseia-se num modelo de controlo de acesso por papéis (RBAC). Cada utilizador possui papéis específicos, como Administrador Geral ou Administrador de Núcleo, que determinam as operações que pode executar. Os serviços validam o token recebido, verificando os papéis associados antes de permitir o acesso aos recursos solicitados.

5.3. Camada de APIs e Orquestração

A comunicação entre os frontends e a lógica de negócio do sistema é realizada através de um conjunto de APIs REST, organizadas de acordo com o tipo de acesso. O Competition API Gateway centraliza o acesso às funcionalidades administrativas, funcionando como ponto único de entrada para operações que requerem autenticação e autorização.

Em paralelo, a Public Data API é responsável por disponibilizar dados públicos, como calendários e resultados, otimizados para consultas frequentes e de elevada concorrência. Esta separação permite melhorar o desempenho global do sistema, ao mesmo tempo que reduz a exposição direta dos serviços internos.

O uso de um gateway de APIs facilita ainda a aplicação uniforme de políticas de segurança, validação de tokens, controlo de acessos e futura integração com outros sistemas ou serviços externos.

5.4. Serviços de Domínio

A lógica de negócio do sistema encontra-se distribuída por vários serviços independentes, cada um responsável por um conjunto específico de funcionalidades, como a gestão de torneios, jogos, equipas, atletas e regulamentos. Esta abordagem promove uma separação clara de responsabilidades e reduz o acoplamento entre componentes.

Um dos serviços centrais é o responsável pelo cálculo automático das classificações, que aplica as regras de pontuação definidas para cada modalidade. Sempre que são registados ou atualizados resultados, este serviço assegura a atualização imediata das classificações, garantindo consistência e fiabilidade da informação disponibilizada.

5.5. Comunicação Assíncrona e Processamento de Eventos

Para suportar processos que não requerem resposta imediata e reduzir dependências diretas entre serviços, o sistema recorre a um mecanismo de comunicação assíncrona baseado num “message broker”. Este componente permite a publicação e consumo de eventos relevantes, como atualizações de resultados ou alterações na configuração de torneios.

A utilização de eventos possibilita que múltiplos serviços reajam automaticamente a alterações no sistema, sem necessidade de comunicação síncrona direta. Por exemplo, após o registo de um resultado, podem ser desencadeados processos de atualização de classificações e invalidação de cache.

5.6. Persistência e Gestão de Dados

A persistência da informação é assegurada por uma base de dados relacional (PostgreSQL), utilizada para armazenar dados estruturados, como utilizadores, cursos, equipas, jogos, resultados e regulamentos. A escolha de uma base de dados relacional garante integridade referencial e suporte a consultas complexas.

Para melhorar o desempenho e reduzir a carga sobre a base de dados principal, o sistema utiliza um mecanismo de cache em memória (Redis). Este componente é particularmente relevante para dados de leitura frequente, como calendários, resultados e classificações públicas.

O armazenamento de ficheiros gerados pelo sistema, nomeadamente fichas de jogo em formato PDF, é realizado através de um serviço de armazenamento de objetos (MinIO), garantindo persistência fiável e acesso controlado a conteúdos não estruturados.

5.7. Observabilidade e Monitorização

A monitorização do sistema é assegurada através da integração de ferramentas de observabilidade, como Prometheus, Grafana e Loki. Estas permitem a recolha de métricas, visualização do estado dos serviços e análise centralizada de logs.

A disponibilização de métricas e registos detalhados facilita a deteção precoce de falhas, a análise de desempenho e a identificação de gargalos no sistema. Esta abordagem é essencial para garantir a fiabilidade e a continuidade do serviço durante o período competitivo da Taça UA.

6. Implementação

O processo de implementação da plataforma de gestão da Taça UA foi orientado por princípios de engenharia de software modernos, privilegiando a modularidade, a escalabilidade, a manutenibilidade e a automação dos processos de desenvolvimento e integração. As decisões técnicas tomadas refletem a necessidade de suportar um sistema distribuído, com múltiplos perfis de utilizador, elevado volume de acessos públicos e requisitos de fiabilidade e segurança.

6.1. Gestão dos repositórios em GitHub

O desenvolvimento do sistema foi suportado por um repositório principal alojado no GitHub, organizado de forma modular e alinhado com a arquitetura definida. A estrutura do repositório reflete uma separação clara entre frontend, backend, micro-serviços, configurações de infraestrutura e documentação técnica, facilitando o desenvolvimento paralelo e a manutenção do código.

Foram adotadas boas práticas de versionamento e colaboração, nomeadamente:

- Utilização de branches dedicadas ao desenvolvimento de funcionalidades específicas (feature branches);
- Submissão de pull requests com revisão de código antes da integração na branch principal;
- Integração contínua automatizada através do GitHub Actions, garantindo validação sistemática do código.

A documentação técnica encontra-se centralizada na diretoria docs/, permitindo um acesso estruturado a informação relevante sobre autenticação, definição de endpoints, modelos de dados, migrações e configuração do ambiente de desenvolvimento. Esta abordagem contribui para a rastreabilidade das decisões técnicas e para a sustentabilidade do projeto a longo prazo.

6.2. Frontend

O frontend da plataforma foi desenvolvido com o objetivo de proporcionar uma experiência de utilização intuitiva, responsiva e adequada aos diferentes perfis de utilizador identificados. A separação entre interface pública e painel administrativo permite adaptar a complexidade e o nível de interação às necessidades específicas de cada tipo de utilizador.

6.2.1. Tecnologias Usadas

O desenvolvimento do frontend baseou-se num conjunto de tecnologias modernas amplamente utilizadas no ecossistema web:

- React: desenvolvimento de interfaces dinâmicas baseadas em componentes reutilizáveis;
- TypeScript: reforço da robustez do código através de tipagem estática;
- Vite: ambiente de desenvolvimento e *build* otimizado;
- Tailwind CSS: definição de estilos consistentes e responsivos;
- Docker: containerização das aplicações frontend.

No painel administrativo foi integrado o sistema de autenticação baseado em Keycloak, assegurando o controlo de acessos e a correta gestão de perfis e permissões.

6.2.2. Estrutura no Repositório

O frontend encontra-se organizado em duas aplicações distintas, localizadas na diretoria src/frontend:

- admin-panel: painel administrativo destinado ao Administrador Geral e aos Administradores de Núcleo;
- public-panel: interface pública para consulta de calendários, resultados, classificações e regulamentos.

Ambas as aplicações seguem uma estrutura interna consistente, organizada por responsabilidades:

- pages/: páginas principais da aplicação;
- components/: componentes reutilizáveis;
- api/: módulos de comunicação com as APIs backend;
- contexts/ e hooks/: gestão de estado global e autenticação;

- routes/: definição de rotas da aplicação.

Esta organização promove a separação de preocupações (separation of concerns) e facilita a evolução futura das interfaces.

6.2.3. Interfaces

As interfaces do sistema foram concebidas de acordo com uma separação clara de perfis de utilização, refletindo os diferentes níveis de responsabilidade e necessidades de acesso à informação identificados durante o levantamento de requisitos. Esta abordagem permite otimizar a experiência do utilizador, reduzir a complexidade das interações e reforçar os mecanismos de segurança, ao disponibilizar apenas as funcionalidades estritamente necessárias a cada perfil.

O frontend encontra-se, assim, organizado em três interfaces principais: interface pública, painel do Administrador Geral e painel do Administrador de Núcleo.

6.2.3.1. Interface Publica

A interface pública destina-se a utilizadores não autenticados e tem como principal objetivo assegurar a transparência e a divulgação eficiente da informação relativa à Taça UA junto da comunidade académica. Esta interface privilegia a simplicidade de navegação, a clareza visual e o acesso rápido à informação mais relevante, uma vez que representa o principal ponto de contacto entre o sistema e o público em geral.

Através desta interface, é possível:

- Consultar o calendário de jogos da competição, com suporte a filtros por curso e modalidade, facilitando a localização de eventos de interesse específico;
- Aceder aos resultados dos jogos, atualizados de forma automática após o seu registo no sistema;
- Visualizar classificações por modalidade, por torneio e a classificação geral da Taça UA, refletindo o estado competitivo atualizado;
- Consultar regulamentos gerais da competição e regulamentos específicos por modalidade;
- Aceder ao histórico de edições anteriores da Taça UA, permitindo uma análise evolutiva da competição ao longo do tempo.

Esta interface foi otimizada para consultas frequentes e elevada concorrência, recorrendo a modelos de leitura e mecanismos de cache descritos nas secções anteriores.

6.2.3.2. Administrador Geral

O painel do Administrador Geral constitui o núcleo central de gestão da plataforma, disponibilizando funcionalidades de carácter estratégico e transversal a toda a competição. Este perfil é responsável pela configuração global do sistema e pela definição das estruturas organizativas que suportam cada edição da Taça UA.

Através deste painel, o Administrador Geral pode:

- Gerir cursos participantes e respetivos núcleos, estabelecendo a hierarquia administrativa da competição;
- Criar, configurar e administrar modalidades e torneios, definindo a sua estrutura competitiva;
- Gerir e publicar regulamentos gerais e específicos, assegurando a consistência normativa da competição;
- Controlar o ciclo de vida das épocas desportivas, incluindo a abertura e o encerramento de cada edição da Taça UA.

O acesso a estas funcionalidades encontra-se protegido por mecanismos de autenticação e autorização baseados em RBAC, garantindo que apenas utilizadores devidamente autorizados podem executar operações críticas para o funcionamento do sistema.

6.2.3.3. Administrador de Núcleo

O painel do Administrador de Núcleo destina-se à gestão operacional da competição ao nível de cada curso, funcionando como interface intermédia entre a organização central e os participantes. Este perfil assume um papel essencial na preparação e acompanhamento das equipas ao longo da competição.

As funcionalidades disponibilizadas incluem:

- Gestão de atletas e membros associados ao curso, incluindo criação, edição e remoção de registos;
- Criação e organização de equipas por modalidade, de acordo com as regras definidas;
- Consulta e acompanhamento dos jogos em que as equipas do curso participam;
- Atribuição de jogadores a jogos específicos, assegurando que apenas atletas elegíveis são convocados;
- Consulta detalhada de partidas, resultados e informação associada.

Esta interface foi concebida para minimizar a carga administrativa e reduzir erros operacionais, centralizando num único sistema tarefas que anteriormente dependiam de comunicação informal e ferramentas externas.

6.3. Backend

O backend da plataforma da Taça UA foi concebido segundo uma arquitetura distribuída baseada em micro-serviços, com o objetivo de garantir escalabilidade, modularidade e facilidade de manutenção. Esta abordagem permite que diferentes componentes do sistema evoluam de forma independente, reduzindo o acoplamento entre domínios funcionais e facilitando a introdução de novas funcionalidades ou serviços no futuro.

6.3.1. Tecnologias Usadas

A seleção das tecnologias do backend teve como critério principal a robustez, o desempenho e o suporte a arquiteturas orientadas a serviços. As principais tecnologias utilizadas são:

- FastAPI, utilizada no desenvolvimento de micro-serviços de alto desempenho, beneficiando de validação automática de dados, tipagem estática e geração de documentação OpenAPI;
- Django REST Framework, adotado na implementação da API administrativa, funcionando também como ponto de entrada centralizado (API Gateway) para operações autenticadas;
- PostgreSQL, como sistema de gestão de base de dados relacional, garantindo integridade referencial e suporte a consultas complexas;
- Redis, utilizado como mecanismo de cache para otimização de consultas públicas de elevada frequência;
- RabbitMQ, responsável pela comunicação assíncrona baseada em eventos entre serviços;
- Keycloak, como serviço centralizado de autenticação, autorização e gestão de controlo de acessos baseado em papéis (RBAC);
- Docker, utilizado para a containerização de todos os serviços, assegurando portabilidade e consistência entre ambientes.

Esta combinação tecnológica permite responder de forma eficaz aos requisitos funcionais e não funcionais identificados, nomeadamente em termos de desempenho, segurança e escalabilidade.

6.3.2. Serviços implementados

O backend encontra-se organizado em vários serviços especializados, cada um responsável por um domínio funcional específico do sistema. Esta decomposição facilita o desenvolvimento independente e a manutenção de cada componente. Os principais serviços implementados são:

- Competition API (Django): atua como API Gateway, concentrando os endpoints administrativos, assegurando a validação de tokens JWT e o controlo de permissões de acordo com os papéis do utilizador;
- Public API (FastAPI): disponibiliza endpoints otimizados para consumo público, baseados em modelos de leitura e preparados para elevada concorrência;
- Matches Service: responsável pela gestão de jogos e respetivos resultados;
- Modalities Service: gere modalidades, cursos, membros e equipas;
- Tournaments Service: responsável pela definição da estrutura dos torneios e respetivas fases;
- Ranking Service: encarregado do cálculo automático das pontuações e classificações;
- Read Model Updater: responsável pela atualização dos modelos de leitura e invalidação de cache com base em eventos.

Apesar de partilharem a mesma base de dados física, cada serviço opera sobre domínios bem definidos, com migrações independentes, reduzindo o acoplamento lógico e facilitando a evolução incremental do sistema.

6.3.3. Estrutura no Repositório

A organização do código-fonte do backend reflete diretamente a arquitetura baseada em micro-serviços adotada pelo sistema. A estrutura do repositório foi concebida com o objetivo de promover uma separação clara de responsabilidades, facilitar a manutenção do código e permitir o desenvolvimento, teste e evolução independentes de cada componente.

O código do backend encontra-se maioritariamente concentrado no diretório src/, o qual está organizado em quatro grandes áreas funcionais: APIs centrais, micro-serviços, componentes partilhados e ferramentas auxiliares. Esta organização contribui para uma visão clara da arquitetura do sistema e reduz o acoplamento entre diferentes domínios funcionais.

6.3.3.1. APIs Centrais

O diretório src/apis/ agrega as aplicações responsáveis pela interação direta com os frontends e pela orquestração dos restantes serviços do sistema. Estas APIs assumem um papel central na validação de acessos, exposição de endpoints e encaminhamento de pedidos.

As principais APIs centrais são:

- competition-api: aplicação desenvolvida em Django que atua como API Gateway do sistema. É responsável pela exposição dos endpoints administrativos, pela validação de tokens JWT, pela aplicação das regras de autenticação e autorização, bem como pelo controlo de permissões baseado em RBAC.
- public-api: aplicação desenvolvida em FastAPI, dedicada à disponibilização de informação pública da competição. Esta API baseia-se em modelos de leitura otimizados, permitindo consultas eficientes a calendários, resultados, classificações e regulamentos, sem expor diretamente a lógica interna dos serviços de escrita.

Esta separação entre APIs administrativas e públicas contribui para uma maior segurança, desempenho e clareza na definição dos contratos de acesso ao sistema.

6.3.3.2. Micro-serviços

Os micro-serviços do sistema encontram-se organizados no diretório src/microservices/, sendo cada serviço responsável por um domínio funcional específico da plataforma. Esta abordagem permite isolar a lógica de negócio, reduzir dependências diretas entre componentes e facilitar a evolução independente de cada serviço.

Os micro-serviços atualmente implementados incluem:

- matches-service: responsável pela gestão de jogos e resultados;
- modalities-service: responsável pela gestão de modalidades, cursos, membros e equipas;
- tournaments-service: responsável pela definição e gestão da estrutura dos torneios e respetivas fases;
- ranking-service: responsável pelo cálculo automático das pontuações e pela atualização das classificações;
- read-model-updater: responsável pela atualização dos modelos de leitura utilizados pela API pública, bem como pela invalidação de cache sempre que ocorrem alterações relevantes.

Cada micro-serviço segue uma estrutura interna consistente, promovendo uniformidade e facilidade de manutenção:

- app/: contém a lógica principal da aplicação, incluindo modelos, rotas, esquemas de validação e gestão de eventos;
- alembic/: responsável pela gestão de migrações da base de dados associadas ao serviço;
- Dockerfile e Dockerfile.env: definem os ambientes de execução do serviço;
- requirements.txt: lista as dependências específicas de cada micro-serviço.

6.3.3.3. Componentes Partilhados

O diretório src/shared/ concentra bibliotecas e módulos comuns utilizados por múltiplos serviços, promovendo a reutilização de código e evitando duplicação de lógica transversal ao sistema.

Entre os principais componentes partilhados destacam-se:

- read-model-shared: define os modelos partilhados utilizados nos modelos de leitura, sendo consumido tanto pela public-api como pelo serviço read-model-updater;

- taca-messaging: fornece uma abstração para a comunicação assíncrona baseada em RabbitMQ, encapsulando a lógica de publicação e consumo de eventos e uniformizando a interação entre serviços.

A centralização destes componentes facilita a manutenção, melhora a consistência do sistema e reduz o esforço de desenvolvimento.

6.3.3.4. Ferramentas Auxiliares

Para além do código aplicacional, o repositório inclui um conjunto de diretórios dedicados à configuração da infraestrutura e a ferramentas de suporte ao desenvolvimento e operação do sistema.

Estes encontram-se organizados da seguinte forma:

- src/configs/: contém as configurações associadas a componentes de infraestrutura, como NGINX, Prometheus, Grafana, Loki e Certbot;
- scripts/: inclui scripts auxiliares para automatização de tarefas, nomeadamente a configuração de TLS e a gestão de certificados;
- tools/: reúne ferramentas auxiliares destinadas a testes, povoamento de dados e apoio ao desenvolvimento.

Esta organização contribui para uma separação clara entre código aplicacional e configurações operacionais, facilitando o deployment e a gestão do sistema em diferentes ambientes.

6.4. DevOps & CI/CD

A estratégia de DevOps adotada visa garantir consistência entre ambientes, automatização de processos e redução de erros humanos durante o ciclo de desenvolvimento e entrega do software.

6.4.1. Docker

Todos os componentes do sistema encontram-se containerizados, incluindo frontend, backend, micro-serviços e ferramentas de observabilidade. Foram definidos ficheiros Dockerfile e docker-compose distintos para ambientes de desenvolvimento e produção, permitindo reproduzir localmente condições próximas do ambiente final de execução e facilitando o processo de deploy.

6.4.2. Continuous Integration

A integração contínua é suportada por GitHub Actions, automatizando tarefas como a validação de builds, execução de testes e verificação da integridade das imagens Docker. Complementarmente, são utilizados pre-commit hooks para garantir conformidade com normas de estilo e qualidade de código antes da submissão de alterações para o repositório.

Esta abordagem contribui para a deteção precoce de erros e para a manutenção de um elevado padrão de qualidade ao longo do desenvolvimento.

6.4.3. Continuous Deployment

O processo de deployment foi concebido para suportar atualizações contínuas do sistema. O acesso externo é mediado por um reverse proxy NGINX, responsável pelo encaminhamento de pedidos e pela integração com certificados TLS geridos automaticamente através do Certbot, assegurando comunicações seguras.

Embora o pipeline de Continuous Deployment ainda não se encontre totalmente operacional, a infraestrutura foi preparada para a sua ativação futura.

6.5. Modelo de Dados

O modelo de dados do sistema foi concebido de acordo com os princípios da arquitetura de micro-serviços, atribuindo a cada serviço a responsabilidade pelas suas próprias entidades e regras de integridade. Esta abordagem reduz dependências diretas entre serviços e facilita a manutenção e evolução do sistema.

As principais entidades do sistema incluem:

- Curso
- Tipo de Modalidade
- Modalidade
- Membro
- Torneio
- Jogo
- Equipa
- Classificação

De forma a otimizar o acesso público à informação, foi adotada uma separação entre modelos de escrita e modelos de leitura, sincronizados através de eventos assíncronos.

Esta estratégia melhora significativamente o desempenho das consultas públicas, garante consistência eventual dos dados e prepara o sistema para uma escalabilidade futura adequada ao crescimento da competição.

7. Documentação da API

A documentação da API da plataforma de gestão da Taça UA foi concebida de forma a garantir clareza, consistência e alinhamento contínuo com a implementação do sistema. Para esse efeito, adotou-se o padrão OpenAPI, amplamente utilizado na especificação formal de interfaces de serviços web REST, permitindo a geração automática de documentação técnica estruturada e interativa.

A documentação é disponibilizada através da ferramenta ReDoc, que possibilita uma navegação clara e organizada pelos diferentes recursos expostos pela API. Esta abordagem assegura que a documentação reflete fielmente o estado atual do sistema, reduzindo o risco de divergências entre a especificação e a implementação efetiva dos serviços.

Importa salientar que a documentação é gerada automaticamente a partir das definições dos endpoints, modelos de dados e esquemas de validação presentes no código-fonte dos serviços backend, não sendo necessária manutenção manual adicional.

7.1. Ferramentas e Abordagem

Para a especificação e geração da documentação da API foram utilizadas as seguintes tecnologias e frameworks:

- OpenAPI, para a definição formal dos contratos das APIs, incluindo endpoints, métodos, parâmetros e esquemas de dados;
- ReDoc, para a visualização estruturada e interativa da documentação;
- FastAPI e Django Rest Framework, utilizados no backend, que permitem a geração automática dos esquemas OpenAPI a partir do código da aplicação.

A adoção desta abordagem integrada garante que a documentação acompanha automaticamente a evolução do backend, constituindo um recurso fiável tanto para a compreensão da arquitetura dos serviços como para o desenvolvimento, manutenção e futura extensão da plataforma.

7.2. Conteúdo da Documentação

A documentação da API encontra-se organizada por recursos e serviços, refletindo a estrutura lógica do sistema. Para cada endpoint são disponibilizadas informações essenciais, nomeadamente:

- Método HTTP e respetivo caminho do endpoint;
- Descrição funcional da operação;
- Parâmetros de entrada, incluindo parâmetros de caminho, query e corpo do pedido;
- Estrutura dos pedidos e das respostas, com esquemas de dados associados;
- Códigos de resposta HTTP e respetivos significados;
- Requisitos de autenticação e autorização, quando aplicável.

Estão devidamente documentadas tanto a API administrativa, responsável pela gestão interna da competição (modalidades, torneios, jogos, equipas e classificações), como a API pública, destinada à consulta de informação aberta, incluindo calendários, resultados, classificações e regulamentos.

Esta documentação constitui, assim, um elemento fundamental de suporte à utilização correta da API, à integração com outros sistemas e à evolução futura da plataforma, promovendo boas práticas de engenharia de software e facilitando a continuidade do projeto.

8. Conclusão

O desenvolvimento do sistema de gestão da Taça UA permitiu concretizar uma solução técnica coerente com os requisitos inicialmente levantados, validando as decisões de engenharia adotadas ao longo do projeto. A plataforma resultante apresenta uma arquitetura modular, orientada a serviços e suportada por princípios consolidados aprendidos no curso de Engenharia Informática, assegurando separação de responsabilidades, escalabilidade e facilidade de manutenção.

8.1. Resultados Obtidos

O desenvolvimento do sistema de gestão da Taça UA permitiu concretizar uma solução técnica coerente com os requisitos inicialmente levantados, validando as decisões de engenharia adotadas ao longo do projeto. A plataforma resultante apresenta uma arquitetura modular, orientada a serviços e suportada por princípios consolidados da Engenharia Informática, assegurando separação de responsabilidades, escalabilidade e facilidade de manutenção.

A nível arquitetural, a adoção de uma arquitetura multicamada, baseada em APIs REST e comunicação assíncrona por eventos, permitiu validar uma abordagem moderna e alinhada com boas práticas de sistemas distribuídos. A integração de serviços como Keycloak para autenticação e autorização, Redis para cache, PostgreSQL para persistência relacional e ferramentas de observabilidade demonstra a aplicação consistente de conceitos fundamentais de Engenharia Informática, nomeadamente segurança, desempenho e monitorização.

Adicionalmente, a definição clara de atores, personas e casos de uso revelou-se essencial para alinhar o desenvolvimento técnico com as necessidades reais dos utilizadores, contribuindo para uma solução funcionalmente adequada e conceptualmente bem estruturada.

8.2. Trabalho Futuro

Apesar dos resultados alcançados com o desenvolvimento da plataforma de gestão da Taça UA, foram identificadas várias linhas de evolução que permitirão consolidar e maturar a solução proposta, tanto do ponto de vista técnico como funcional.

Em primeiro lugar, torna-se necessária a conclusão do processo de deploy completo do sistema, nomeadamente através da configuração da infraestrutura numa máquina virtual dedicada e da ativação integral do pipeline de CD. A implementação deste mecanismo permitirá a disponibilização automatizada, controlada e consistente de

novas versões da aplicação, reforçando boas práticas de engenharia de software e garantindo maior fiabilidade operacional em ambiente de produção.

Ao nível funcional, prevê-se a implementação de funcionalidades atualmente pendentes, com particular destaque para a validação automática do estatuto de sócio dos atletas, de forma integrada com fontes de dados oficiais, reduzindo dependências de processos externos e mitigando erros administrativos. Adicionalmente, a conclusão da geração automática de fichas de jogo em formato PDF, totalmente integrada no sistema, permitirá encerrar o ciclo administrativo dos jogos de forma digital, assegurando maior coerência, rastreabilidade e eficiência nos processos organizativos.

No que respeita à qualidade do software, a definição e implementação de uma estratégia abrangente de testes automatizados, incluindo testes unitários, de integração e end-to-end, constitui um passo fundamental para aumentar a robustez do sistema. Esta abordagem permitirá detetar regressões de forma precoce, facilitar a manutenção evolutiva e assegurar a estabilidade do sistema à medida que novas funcionalidades forem introduzidas.

Por fim, identificam-se oportunidades de melhoria ao nível da segurança da aplicação, nomeadamente através do reforço dos mecanismos de controlo de acessos, da monitorização de atividades e da proteção dos serviços expostos. Estas melhorias são particularmente relevantes num sistema que gere informação sensível e suporta múltiplos perfis de utilizador com diferentes níveis de permissão.

A concretização destas linhas de trabalho futuro irá permitir consolidar a plataforma como uma solução tecnicamente madura, alinhada com boas práticas de Engenharia Informática, e preparada para suportar de forma sustentável futuras edições da Taça UA, bem como potenciais adaptações a outros contextos de gestão de competições académicas.

9. Referencias

[1] Federação Académica do Desporto Universitário. *Plataforma de Gestão Desportiva Universitária*