



## Relatório DAW

**(31/12/2024)**

Docente: Professora Noélia Correia

Hugo Conceição (Nº79722), João Ventura (Nº79882), José Lima(Nº79738) e  
João Guerreiro(Nº81430)

Grupo 1

Desenvolvimento de Aplicações Web

Licenciatura em Engenharia Informática

Faculdade De Ciências e Tecnologias - Universidade Do Algarve

### **i. Introdução**

O nosso projeto consiste no desenvolvimento de uma aplicação inspirada no Flashscore, projetada para fornecer informações desportivas em tempo real, com foco no acompanhamento de futebol. O principal objetivo é permitir aos utilizadores acompanhar as suas equipas e competições favoritas de forma prática e eficiente.

A aplicação disponibilizará resultados ao vivo, classificações atualizadas e estatísticas detalhadas. Além disso, o projeto visa integrar e aplicar os conteúdos lecionados ao longo do ano, promovendo a consolidação do conhecimento adquirido de forma a construir uma aplicação própria adicionando alguns requisitos extras.

### **ii. Descrição da Rest API criada**

A REST API é responsável pela gestão de autenticação de utilizadores, dados desportivos, atualizações em tempo real e também as equipas/ligas/jogos favoritos do usuário definindo as rotas para realizar as operações CRUD (Create, Read, Update, Delete). Ela foi criada com recurso a base de dados para a gestão dos utilizadores (adicionar, obter e remover) e também à FreeSports API que nos permitiu obter uma variedade de dados das ligas, equipas e jogos tornando possível a criação desta RestAPI. Falando um bocado mais sobre cada endpoint da nossa Rest API.

Rota dos usuários (/user):

- POST /register: Registrar novos utilizadores
- GET /verify/token: Verificação de email por token JWT
- POST /login: Login de utilizadores
- PUT /:username/change-username: Alteração de username
- POST /logout: Logout do utilizador

Rota da Liga (/portugal, /espanha, /inglaterra, /alemanha, /italia, /franca):

- GET /classificacoes: A lista de classificação das equipas da liga
- GET /lista: A lista de próximos jogos da liga
- GET /resultados: A lista de resultados dos jogos da liga

Rota de Favoritos (/favorites)

- POST / : Adicionar ou remover uma liga/equipa aos favoritos do utilizador
- GET / : Obter a lista de favoritos do utilizador

Rota do Jogo (/jogo)

- GET /estatisticas/:id: A lista de estatísticas de um jogo (ex: Remates à baliza, Faltas, etc.)
- GET /formacao/:id: A lista de jogadores que presentes na equipa inicial do jogo
- GET /timeline/:id: A lista de eventos que ocorreram durante o jogo

Rota da Equipa (/equipa)

- GET /:id/lista: A lista dos próximos jogos da equipa
- GET /:id/resultados: A lista de resultados da equipa

### iii. Funcionalidades e Requerimentos do lado do Client e do Server

#### Client

##### Funcionalidades:

- **Autenticação de Utilizadores:** Componentes de registo, login, logout, atualização do username do utilizador, e também de eliminação da conta do utilizador.
- **Favoritos:** Componentes para gerir ligas e equipas favoritas, gerindo o id, logo, e nome de ambas. Exibição de eventos das ligas favoritas.
- **Eventos:** Exibição dos jogos das equipas organizado pelas ligas correspondentes. Capacidade para filtrar os eventos das ligas por eventos ao vivo, terminados ou agendados
- **Estatísticas:** Componentes para gerir e mostrar as estatísticas dos jogos ao vivo e terminados
- **Contextos:** Forma de variável global para gerir os dados dos users, das ligas, das equipas e dos eventos
- **Interface do Utilizador:** Uso de componentes do Bootstrap, como Button, Card, Container, Nav.bar e Image. Uso de componentes personalizados como ButtonLeague, ButtonTeam, NextEventButton para melhor navegação e exibição de informação.

### Requisitos:

- **Dependências:** React, React-DOM, Bootstrap, Axios, WebSocket e ferramentas de desenvolvimento como Webpack e Babel.
- **Configuração:** Arquivo tsconfig.json para a transpilação e package.json para scripts de build e desenvolvimento
- **Estilo:** Uso de estilos personalizados nos componentes em main.css

### Servidor

#### Funcionalidades

- **Autenticação e sessão do utilizador:** Configuração de sessões com express-session. Autenticação de usuário com Passport.js e estratégia local (LocalStrategy)
- **Conexão com a Base de Dados:** Conexão com MongoDB com a utilização do Mongoose. Modelos de dados para utilizadores e favoritos
- **Rest API:** Rotas para gerir utilizadores, favoritos, dados de ligas e equipas, já referido acima.
- **WebSocket:** Configuração de servidor WebSocket para fornecer atualizações em tempo real de eventos, para permitir aos clientes receber atualizações instantâneas.
- **Middleware:** Passport utilizado para gerir sessões e a autenticação dos utilizadores. Utilização do NodeCache para gerir o cache na memória.
- **Tratamento de Dados:** Funções para receber e transformar dados de eventos, ligas, estatísticas de eventos em transformData.ts.
- **Envio de Emails:** Configuração de envio de emails de confirmação da conta do utilizador utilizando nodemailer em SMTP.ts.

#### Requisitos

- **Dependências:** Express, Mongoose, Passport, Passport-Local, Express-Session, WebSocket, NodeCache, NodeMailer, JWT
- **Configuração:** Arquivo tsconfig.json configurado para compilar o TypeScript. Utilização de um ficheiro com a configuração para a conexão com MongoDB, e também a chave da API

## iv. Layout

O layout foi estruturado para ser consistente e intuitivo, utilizando as mesmas cores e mantendo os vários elementos em posições fixas em diferentes páginas. Esta abordagem garante que os utilizadores consigam navegar facilmente pela aplicação sem a necessidade de reaprender a interface de cada página.

**Cabeçalho:** Todas as páginas possuem um cabeçalho uniforme que inclui:

- Um botão para redirecionar à página principal;
- Um botão para visualizar os resultados dos jogos;
- Um botão de favoritos para gerir as equipas ou ligas marcadas como favoritas;
- Um botão de login para autenticação do utilizador.

**Página Principal:** Na página inicial, são exibidos os resultados dos jogos e os favoritos, com filtros para facilitar a navegação, como:

- "Todos", "Ao Vivo", "Agendados", "Terminados" e seleção por data;
- É possível também aceder à página dos jogos nos resultados ao clicar nos mesmos.
- Além disso, foram adicionados botões que permitem o acesso às páginas das ligas.

**Página favoritos:** A página “favoritos”, possui o mesmo layout que a página principal, porém apenas apresenta os jogos e ligas que o utilizador marcou como favorito.

**Página do Jogo:** Na página dedicada aos jogos é possível observar detalhes dos jogos como:

- Informações do jogo;
- Estatísticas;
- Formações;
- Classificação.

**Página da Liga:** A página dedicada a cada liga inclui:

- Um cabeçalho com o nome e o emblema da liga, acompanhado de um botão para marcar a liga como favorita;
- Três botões que permitem alternar entre classificações, resultados, e lista de próximos jogos;
- Funcionalidade para acessar a página de uma equipa ao clicar no emblema correspondente.

**Página da Equipa:** O layout da página da equipa foi projetado para manter a consistência visual e estrutural com a página da liga. Isso inclui:

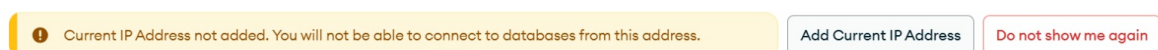
- Um cabeçalho com as informações principais da equipa, como nome e emblema;
- Botões que permitem aos utilizadores explorar classificações, resultados, e próximos jogos relacionados à equipa.

O foco no design consistente e reutilização de elementos garante uma experiência de utilização intuitiva, reduzindo o tempo de aprendizagem do utilizador e proporcionando uma navegação fluida.

## v. Project set up

Para iniciar o projeto será necessário entrar no mongo db atlas

(<https://account.mongodb.com>) com a conta: email: [joaotesteconta8@gmail.com](mailto:joaotesteconta8@gmail.com) pass: DawProjetoFinal e de seguida clicar no projeto com o nome: [Hugo's Org - 2024-11-16](#) e por fim clicar na mensagem para adicionar o IP. (Como a seguinte:)



**Aviso:** Não funciona com a rede da ualg.

De seguida abrir o código e instalar as dependências fazendo “npm i” tanto na pasta server como na pasta cliente e por fim na pasta server fazer “npm run dev” e aceder no browser a <http://localhost:8080>.

## **vi. Três extras/ alterações**

### **MongoDB:**

Como complemento, optámos por utilizar uma base de dados diferente da aplicada nos laboratórios práticos. Para este projeto, escolhemos o MongoDB devido à sua natureza NoSQL, que oferece uma modelação de dados mais flexível e intuitiva. Esta característica elimina a necessidade de definir um esquema rígido previamente, permitindo ajustamentos contínuos à medida que a aplicação se desenvolve. Foi responsável pela gestão de utilizadores e das equipas/ligas favoritas dos utilizadores.

### **Bootstrap:**

Outro recurso adicional que utilizámos foi o Bootstrap, um framework que disponibiliza um conjunto abrangente de ferramentas e estilos para o desenvolvimento rápido e consistente de interfaces web. Este framework fornece componentes pré-estilizados, como botões, formulários e barras de navegação, o que acelera o processo de desenvolvimento e assegura uma experiência visualmente apelativa e funcional para o utilizador.

### **FreeSportsAPI:**

O último extra utilizado foi esta API fundamental no nosso projeto que fornece dados desportivos em tempo real. Ela permite integrar informações atualizadas sobre resultados, classificações, estatísticas tanto de futebol como de outros desportos. A utilização desta API facilitou a obtenção e exibição de dados relevantes de forma dinâmica, sem a necessidade de desenvolver um sistema interno complexo para a gestão dessas informações.

## **vii. Conclusão**

Este projeto representou um desafio enriquecedor, permitindo-nos aplicar de forma prática os conhecimentos adquiridos ao longo do curso. Inspirados no Flashscore, desenvolvemos uma aplicação que combina tecnologias modernas, como MongoDB, Bootstrap e a FreeSportsAPI.

Através da implementação de funcionalidades, como a gestão de utilizadores, favoritos e dados desportivos em tempo real, consolidámos competências em desenvolvimento full-stack e reforçámos a importância da integração de ferramentas externas para otimizar o seu desempenho.

Em suma, este trabalho não só nos permitiu consolidar os conteúdos lecionados, como também nos preparou para enfrentar desafios reais no desenvolvimento de aplicações web.