Uma imagem com texto, Tipo de letra, Gráficos, design gráfico

Descrição gerada automaticamente

Instituto Superior de Engenharia de Lisboa

Licenciatura em Engenharia Informática e de Computadores 2023/2024

Unidade Curricular de Desenvolvimento de Aplicações *Web*

Docente Felipe Freitas

**Área Departamental de Engenharia de Eletrónica e** **Telecomunicações e de Computadores**

Autores:

47179 Afonso Cabaço

48264 João Pereira

48292 Tiago Neves

Grupo 09

Relatório para a Unidade Curricular de Desenvolvimento de Aplicações *Web* Licenciatura em Engenharia Informática e de Computadores

20-01-2024

Índice

[Introdução 4](#_Toc157082363)

[Modelo Conceptual 5](#_Toc157082364)

[Modelo Físico 6](#_Toc157082365)

[Especificação *Open-API* 7](#_Toc157082366)

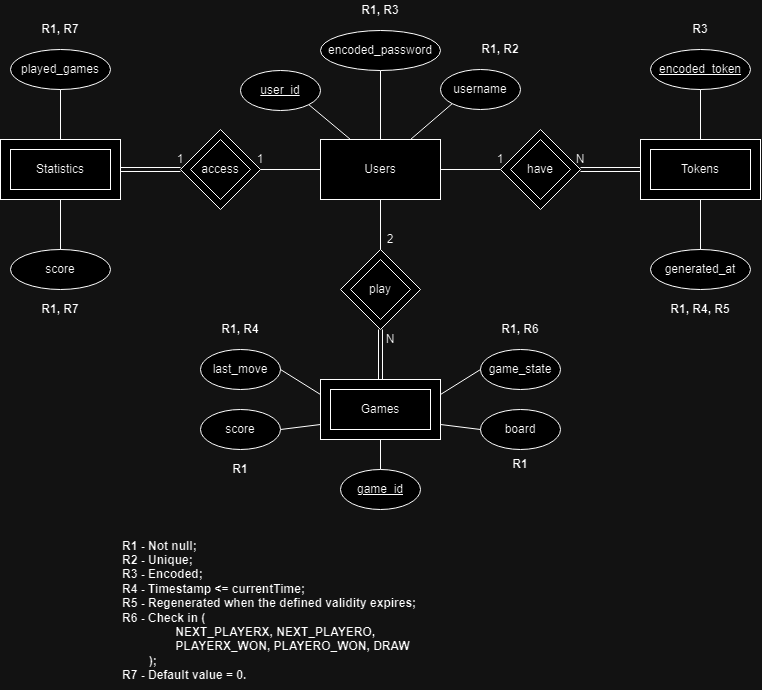
[Desafios de Implementação 7](#_Toc157082367)

# Introdução

O objetivo deste projeto é o desenvolvimento de um sistema baseado na ***web*** que permita a vários jogadores jogarem ***Gomoku***[. O](https://trello.com/) domínio da aplicação é baseado em quatro entidades diferentes:

1. ***User*:** Um *user* é caracterizado por um número único (*user\_id*), um *username* único e uma *password* única;
2. ***Game***: Um *game* representa um jogo, identificado por um número único, contendo informação sobre os jogadores que nele participam, o estado e o tamanho do tabuleiro com as posições ocupadas pelas peças, as regras e a variante do mesmo;
3. ***Statistics***: *Statistics* representa a qualificação de cada jogador, sendo identificado pelo número único de *user*, contendo também o número total de jogos, o número de vitórias e também o número de derrotas do jogador;
4. ***Lobby***: Um *Lobby* representa uma sala de espera para a qual os jogadores são inseridos quando manifestam vontade de jogar, sendo assim contém o identificador do jogador que procura um jogo e posteriormente também o do seu adversário, as regras, a variante, o tamanho do tabuleiro selecionados pelos jogadores e também o estado do jogo.

# Modelo Conceptual



**Figura 1 - Modelo EA**

# Modelo Físico

**Uma imagem com captura de ecrã, texto, diagrama, design

Descrição gerada automaticamente**

**Figura 2 - Modelo Físico**

Para a implementação do modelo físico recorremos a *REST API* que consiste na criação de serviços *web* com o objetivo de facilitar a comunicação e a integração entre sistemas distribuídos e heterogéneos.

* ***GomokuApp*** - Constitui o ponto de entrada para a aplicação;
* ***Interceptor*** - Interceta o pedido para verificar a existência de um *token*;
* ***GameController*, *UserController* e *StatistcController*** - Implementação das rotas *HTTP* que compõem a *REST API* da aplicação;
* ***GomokuServices*, *UserServices* e *StatisticServices*** - Implementação da lógica de cada funcionalidade da aplicação;
* ***LobbiesRepository*, *GamesRepository*, *UserRepository* e *StatisticsRepository*** - Acesso à informação da aplicação.

# Especificação *Open-API*

Toda a informação sobre a aplicação pode ser encontrada [aqui](https://github.com/isel-leic-daw/2023-daw-leic53d-2023-daw-leic53d-g09/blob/main/docs/GomokuAPI.yaml).

# Desafios de Implementação

Com esta implementação os maiores desafios que enfrentámos foram:

• Autenticação do utilizador e preservação da autenticação;

• *Matchmaking*.

Autenticação do utilizador e preservação da autenticação

Um dos maiores desafios que enfrentámos foi a realização da **autenticação** do utilizador. Este problema ocorreu devido a não entendermos, na altura, que deveríamos usar e como usar o componente ***httpOnly*** para as ***cookies*** do utilizador. Ao chegarmos a esta solução conseguimos resolver tanto o problema de **autenticação** como o da sua preservação, podendo assim navegar pela aplicação sempre autenticados.

*Matchmaking*

A ação de ***matchmaking*** foi algo trabalhosa devido à lógica do trabalho e tudo o que envolve. Porém, o verdadeiro desafio foi a conversão do que tínhamos originalmente nas respostas provenientes do ***backend*** para a sintaxe dos ficheiros “***.ts***”.