



ISEL
INSTITUTO SUPERIOR DE
ENGENHARIA DE LISBOA

Relatório de Projeto

FASE 1

Grupo 9 | DAW | 21-10-23

47179 – Afonso Cabaço

48264 – João Pereira

48292 – Tiago Neves

INTRODUÇÃO

Neste documento iremos reportar os aspetos mais significativos sobre o *design* e implementação do projeto de DAW. Este projeto tem como objetivo realizar uma aplicação inspirada no jogo *Gomoku*.

DATABASE MODELS

Conceptual Model

O seguinte modelo EA foi concebido em grupo e usado como base para a estruturação do nosso projeto.

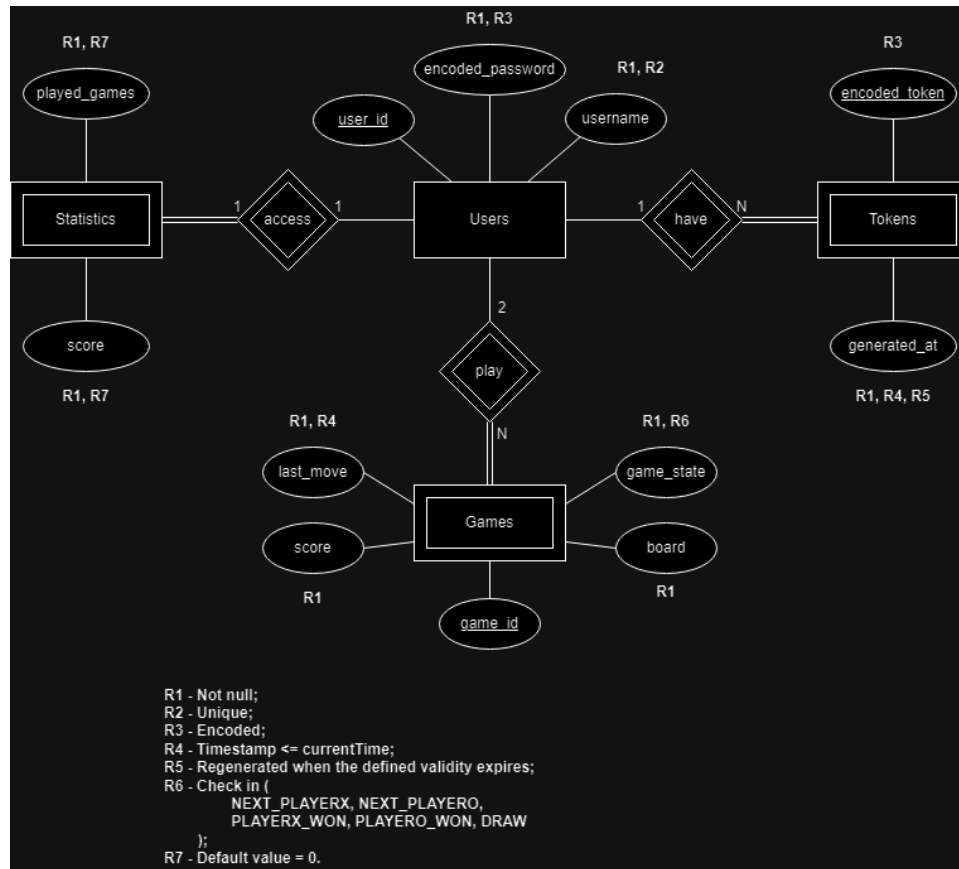


Figure 1 - Modelo EA

Physical Model

Via o seguinte [link](https://github.com/isel-leic-daw/2023-daw-leic53d-2023-daw-leic53d-gog/tree/main/code/sql) poderão encontrar os *scripts* usados para a concretização do modelo, concebido anteriormente, na nossa base de dados.

<https://github.com/isel-leic-daw/2023-daw-leic53d-2023-daw-leic53d-gog/tree/main/code/sql>

OPEN-API SPECIFICATION

As especificações do nosso projeto seguiram os conceitos da *OPEN-API*, para tal realizámos um ficheiro ‘.yaml’ para que todas especificações pudessem ser entendidas e testadas.

<https://github.com/isel-leic-daw/2023-daw-leic53d-2023-daw-leic53d-gog/blob/main/docs/GomokuAPI.yaml>

REQUEST DETAILS

Para efetuar o tratamento dos pedidos, começamos por filtrar o pedido para que este seja tratado no ficheiro certo do projeto. Para tal, recorreremos de diferentes serviços cada um dedicado a um espectro de recursos, por exemplo, no caso de se tentar obter um utilizador a partir do seu *ID*, este pedido irá apenas afetar os serviços dos utilizadores. Posteriormente, o serviço requisitado irá fazer um pedido à base de dados para obter o recurso pretendido, através de repositórios com funções definidas.

CONNECTION MANAGEMENT

Com uso da função *setUrl()* da biblioteca *BaseDataSource* de *Java*, definimos o caminho para a nossa base de dados e efetuar a ligação a base de dados.

DATA ACCESS

Para realizar a conexão entre os serviços e os dados da base de dados recorreremos de repositórios. Estes repositórios são implementados com base em interfaces onde ficam definidas as funções a serem utilizadas. Estas funções realizam então as *querys* necessárias para aceder à informação pretendida pelo utilizador da aplicação.

ERROR HANDLING/PROCESSING

Para o processamento de erros criámos uma classe com os possíveis erros que podem ocorrer. Estes erros são depois apanhados quando se efetua um pedido no caso de o pedido estar errado.

Também temos uma classe com possíveis exceções que o utilizador por incorrer quando tenta obter algum recurso como por exemplo realizar um *login*, porém com o *username* ou *password* errado.

CRITICAL EVALUATION

- Melhorar o gerenciamento dos erros e a forma de como os mostrar ao utilizador.
- Melhorar partes do código em geral.

