Instituto Superior de Engenharia de Lisboa
Licenciatura em Engenharia Informática e de Computadores 2023/2024 Unidade Curricular de Desenvolvimento de Aplicações Web Felipe Freitas

Área Departamental de Engenharia de Eletrónica e Telecomunicações e de Computadores

Fase 1

Autores: 48337 Daniel Antunes

49487 Ricardo Rovisco

49508 João Mota

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

20 – 01 – 2024

Índice

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. O domínio da aplicação é baseado em 5 entidades diferentes:

1. User: Cada user é caracterizado por um número único, um nome e um email único e uma password.

2. Game: Um game é caracterizado pelo seu id, os ids de ambos os jogadores integrantes da partida, o estado atual tanto do seu tabuleiro, sendo este controlado a partir da serialização e deserialização do mesmo na forma de uma linha de texto, o seu estado atual (*Starting*, *On Going* e *Finished*), e o vencedor da partida se a mesma já tiver acabado com a vitória de um dos jogadores.

3. Lobby: Um lobby é caracterizado pelo id do user que estiver à espera de um 2º jogador para iniciar um jogo, a rule escolhida para o jogo e também o tamanho do tabuleiro. O mesmo é constituído por dois estados, "*Full*" e "*Not Full*", que vão servir para a formação de pares entre os jogadores que procurarem por jogos com as mesmas regras e tamanho dos tabuleiros.

4. Token: Cada token representa os parâmetros necessários para autenticação de um user, o mesmo não pode estar associado a mais do que um user. O token é caracterizado pelo seu valor, o id do user a quem o mesmo pertence, bem como a data de criação e a data em que foi usado pela última vez.

5. Rankings: Esta entidade serve para guardar a informação estatística dos jogadores, sendo a mesma caracterizada pelo o id do user, quantos jogos o mesmo jogou, ganhou ou empatou. Esta informação é depois utilizada para calcular a taxa de vitória do jogador que o irá ajudar a organizar o jogador nos rankings.

Modelo Conceptual

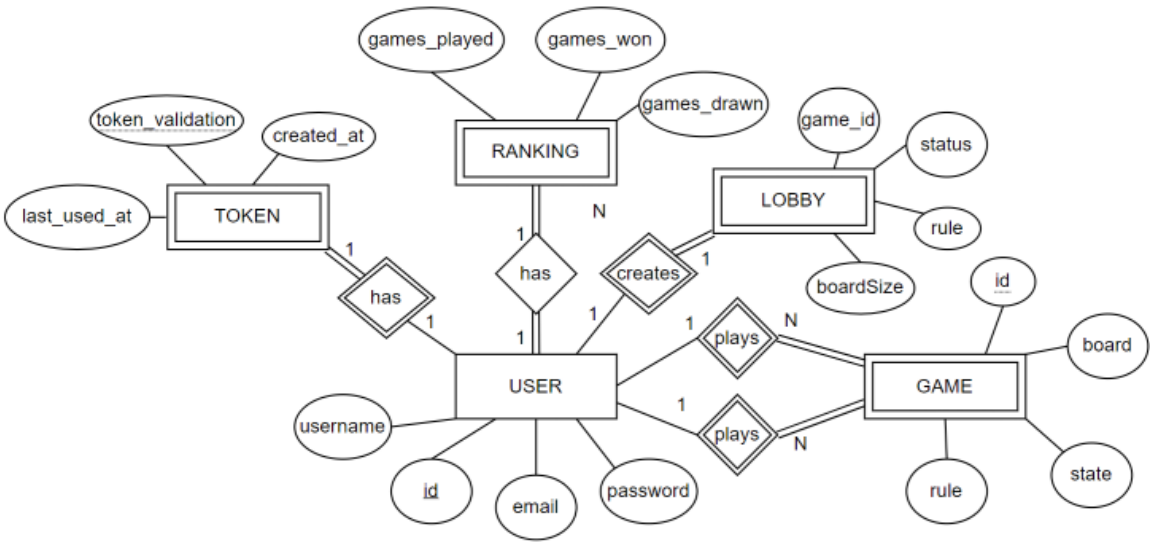

Figura 1 modelo EA

Como referido anteriormente o nosso modelo é composto por 5 entidades das quais 4 delas são fracas. Como demonstrado na figura 1, as entidades Ranking, Game, Lobbie e Token são fracas da entidade User uma vez que, todas essas entidades dependem da mesma para conter informação.

Modelo Físico

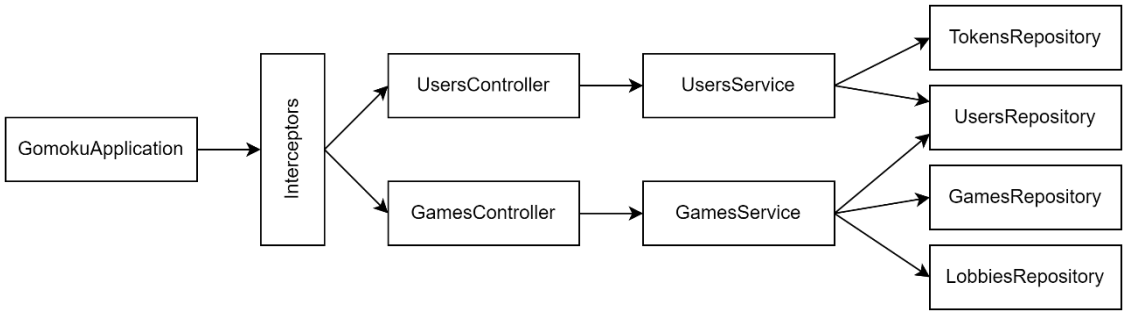

Figura 2 modelo físico

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 heterogêneos.

- Gomoku Application – Constitui o ponto de entrada a aplicação.
- Interceptors – Interceta o pedido para a verificação de existência de um token
- UsersController e GamesController – Implementação das rotas HTTP que compõe a REST API da aplicação
- UserServices e GameServices – Implementação da lógica de cada funcionalidade da aplicação
- TokensRepository, UsersRepository, GamesRepository e LobbiesRepository – Acesso à base de dados da aplicação

Para garantir que as funcionalidades da aplicação são executadas corretamente, como o acesso à base de dados, foram também implementadas 4 interfaces: UsersRepository, GamesRepository, LobbiesRepository e TokensRepository.

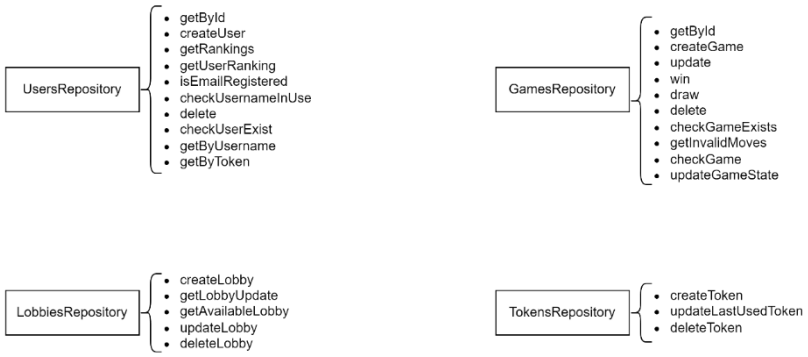

Figura 3 Interfaces da aplicação

Navegação na aplicação

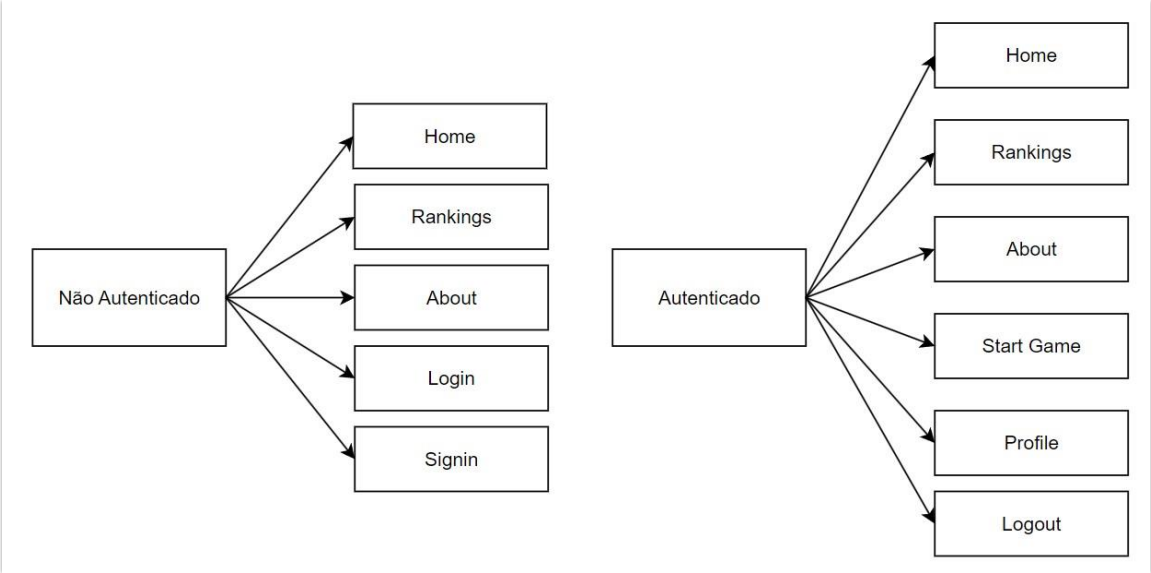

Figura 4 Front End

Para a navegação na aplicação representamos o utilizador com dois estados: "Não Autenticado" e "Autenticado" tendo assim acesso a diferentes funcionalidades da aplicação sendo eles: para "Não Autenticado" Home, Rankings, About, Login e SignIn; e para "Autenticado" Home, Rankings, About, Start Game, Profile e Logout

Especificação Open-API

Toda a informação sobre a aplicação pode ser encontrada aqui.

Detalhes do Pedido

Quando um utilizador efetua um pedido HTTP na aplicação, o Server envia a informação com a rota e método utilizados para que no Controller possa ser executada a funcionalidade correta com a informação passada no path, query ou no body. Antes dessa informação ser enviada para o módulo Services, a mesma é filtrada pelos Filters e Interceptors implementados. O módulo Services efetua as verificações necessárias para garantir que a informação seja enviada para a base de dados local efetuando a funcionalidade da aplicação no módulo Repository. Certas funcionalidades apenas podem ser efetuadas se o User estiver com sessão iniciada na aplicação, para isso desenvolvemos funcionalidades de login e logout para posteriormente efetuarmos essa verificação efetuando a criação de um token e associando o mesmo ao User que efetuou o login, este é removido após o User efetuar o logout.

Gerenciamento da Conexão

Utilizando a função setUrl() definimos o caminho para a base de dados, seguidamente efetuamos a conexão com a mesma usando a função create(). Utilizando um Handle criamos e executamos as queries necessárias para realizarmos a respetiva funcionalidade que desejamos. Foi também desenvolvida a função extensão configureWithAppRequirements() que efetua a configuração da conexão com a base de dados e adiciona os requerimentos necessários para a execução da aplicação.

Processamento/Tratamento de Erros

Para o tratamento de erros foram criadas duas classes, Error e Result. No processamento de um pedido, o módulo Services tem a classe Result como tipo de retorno para que caso não seja encontrado qualquer erro, o pedido retorna Success e caso seja encontrado um erro retorna Failure com o respetivo error. Posto isto, no módulo Controller, caso seja enviado Failure, o mesmo será tratado pelo envio do status do erro em questão e da mensagem de erro contida na classe Error, caso contrário é enviado o status de sucesso respetivo ao pedido efetuado bem como o envio da informação gerada pelo mesmo.

Representação de Responses

Para as respostas a pedidos utilizamos a representação em siren, para tal, foi necessário alterar os pedidos de forma a poder ser usada essa mesma representação. Algumas representações contêm links para que no front-end seja possível fazer a n para as páginas pretendidas.

Avaliação Crítica

Na realização desta fase do projeto, algumas das dificuldades encontradas foram o envio da informação do lobby criado para a página onde é feito o polling e a atualização da NavBar após o login ou o logout serem efetuados o que necessitou de alguma reestruturação do Front-End.