



Universidade do Minho

UMinho

Mestrado Engenharia Informática
Requisitos e Arquiteturas de Software
(2022/23)

RASBET

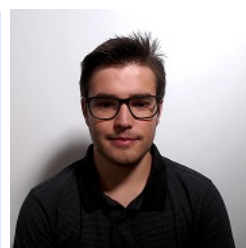
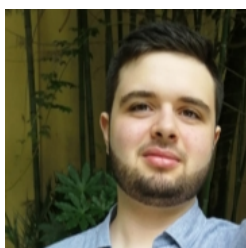
Grupo 7, PL5 - Análise as alterações

PG50404 - Gonçalo Ferreira

PG50369 - Gonçalo Santos

PG50003 - Hugo Nogueira

PG50560 - Luis Faria



Braga, 17 de janeiro de 2023

1. Análise das alterações realizadas ao Código

Aqui serão descritas todas as principais alterações realizadas ao código que permitiram um melhoramento da implementação dos requisitos não funcionais.

- **Introdução da funcionalidade "Seguir Jogos"** - funcionalidade seguir jogos, foi adicionada ao sistema. Agora na UI da aplicação, é visível em cada jogo uma pequena aba, que quando selecionada permite aos utilizadores passar a seguir um jogo, recebendo notificações sobre qualquer atualização que exista sobre o mesmo (finalização do jogo e atualização das Odds). Para a implementação desta feature, o grupo necessitou de associar uma listagem de ids de utilizadores interessados num evento, a esse evento, para que, sempre que este é atualizado, todos os seus seguidores sejam informados dessa alteração. A arquitetura desta funcionalidade é baseada no padrão arquitetural observer, onde cada utilizador (sob a forma da sua fila de notificações) é um observer de um dado evento.
- **Alterações na arquitetura da comunicação com as APIs** - a arquitetura da comunicação com as APIs fornecedoras das informações acerca dos eventos, foi completamente remodelada. Nesta entrega, as classes responsáveis por interagir com cada API foram isoladas das restantes, e todas implementam a interface "SportAPICommunication" que define os métodos para atualizar e criar novos eventos. Desta forma a adição de um novo desporto (ao nível do backend) apenas tornará necessário a adição de uma nova classe para contactar essa API e da sua inclusão na listagem de APIs disponíveis na classe APICommunicationHub; o restante sistema está pronto para receber novos desportos, sem que haja necessidade de qualquer alteração externa ao módulo APICommunication. De forma a demonstrar o dinamismo introduzido por esta mudança, o grupo decidiu também incluir a NFL (a liga Norte Americana de Futebol Americano); esta introdução pode ser realizada em menos de 10 min, sem que tivesse de existir qualquer mudança no restante código.
- **Implementação do sistema de controlo de sessões baseado em JWT** - Na fase inicial do projeto o grupo implementou a gestão de sessões de utilizador guardando num dicionário o token de acesso (key) , o email e role (value) de quem estava logged in. Sempre que fosse feito um pedido por parte da frontend, era enviado um token, o qual era verificado. Caso não existisse, era devolvida a resposta a declarar a rejeição do pedido, caso contrário o pedido era devidamente processado.

Com o propósito de melhorar a segurança na transmissão de informação e no controlo de sessões o grupo decidiu re-implementar, com recurso a JWT(JSON Web Token) - estes tokens foram projetados para serem compactos e seguros, o grupo considerou que seriam ideais para o efeito pretendido.

O novo processo de gestão de sessões funciona da seguinte forma:

- Sempre que um utilizador se regista ou faz login, são gerados dois tokens e ambos são enviados como resposta ao pedido. O access_token é gerado com um segredo próprio, e fica apenas guardado no objeto do jwt por um tempo determinado. Quanto ao refresh_token, este é também gerado com um segredo próprio, e é guardado na base de dados. Ambos os tokens contêm no payload o email e o role do utilizador.

- Quando um pedido é feito, é enviado o devido `access_token`, o qual vai passar por middleware que confirma se quem fez o pedido tem permissões para tal e se o token é válido. Caso não haja permissões, o pedido é recusado com essa indicação. Caso o token de acesso não seja válido (por o tempo de utilização ter expirado), então a frontend, com o `refresh_token` (sobre o qual é verificada a presença na base de dados), pede outro `access_token` para voltar a fazer o pedido.
 - Sempre que um utilizador faz logout, o `refresh_token` associado é eliminado da base de dados.
- **Encriptação Passwords**- No contexto de melhorias na segurança da aplicação, o grupo implementou também nesta fase, encriptação das passwords armazenadas na Base de Dados recorrendo à biblioteca `bcrypt`.
 - **Melhoria na comunicação com o utilizador** - foram melhorados os sistemas de comunicação com o utilizador, com a introdução de novas mensagens de alerta e confirmação para oferecer um serviço mais intuitivo e a todos os utilizadores.
 - **Melhoria na apresentação do serviço** - foram desenvolvidas algumas melhorias, no que toca à apresentação gráfica da aplicação, tornando-a um pouco mais apelativa para os seus utilizadores.
 - **Refactoring na qualidade de código por todo o projeto** - ao longo de todo o projeto foram realizados diversos pequenos "refactorings", que permitem que o código se torne mais legível e mais eficiente.