

Relatório Final de Projeto



Grupo 10

Trabalho realizado por :

Tiago Silva, 8190614

Tiago Coelho, 8190340

Jaques Resende, 8190214

Maria Tavares, 8190709

Tiago Teixeira, 8190592

Índice

Introdução	3
Milestone 1	4
Milestone 2	5
Milestone 3	6
Dificuldades Encontradas	7
Aspetos a melhorar	9
Conclusão	10

Introdução

Este documento tem como objetivo descrever o trabalho desenvolvido ao longo do semestre, a organização, as dificuldades apresentadas, a separação de tarefas e estruturação do mesmo.

Este trabalho teve como objetivo uma WebApp que através de API's externas permite os utilizadores associarem a sua conta do videojogo "League of Legends" à sua conta na base de dados do website, e através disto permite o utilizador criar equipas e entrar em torneios realizados no website baseados no sua classificação no videojogo.

Este projeto foi realizado ao longo de 3 Milestones e 11 sprints.

Milestone 1

A primeira milestone foi focada no planeamento e organização do projeto de forma a ter uma visão mais concreta do que iríamos desenvolver, os prazos necessários, possíveis dificuldades e tecnologias e estratégias a usar.

O grupo foi dividido em FrontEnd e BackEnd, sendo que foi decidido que o BackEnd iria contar com 3 membros devido à densidade de trabalho que este teria pelo facto de usar API's externas e algoritmos de geração de brackets e o resto do desenvolvimento do mesmo no torneio.

A divisão ficou do tipo :

-FrontEnd :

- Tiago Coelho;
- Jaques Resende;

-BackEnd :

- Tiago Silva;
- Tiago Teixeira;
- Maria Tavares;

Como início foi feita uma recolha pormenorizada de requisitos funcionais e não funcionais e estruturado o nosso GitLab de acordo com o planeado, esta recolha foi baseada numa pesquisa atenta e estudo da RIOT Api para sabermos o que nos esperava no futuro.

Desenvolvemos Mockups que ilustravam pormenorizada a nossa aplicação de acordo com os requisitos desenvolvidos e por fim atualizamos o nosso gitlab com todas as informações necessárias e com o Product Backlog.

Foram criados também juntamente criados os diagramas de Use cases.

Em resumo a 1ª Milestone foi bem conseguida e trabalhada e provavelmente a melhor a nível de trabalho de equipa.

Milestone 2

A segunda milestone foi onde o desenvolvimento da aplicação em si começou tendo como ponto de partida um exemplo de uma WebApi com uma base de dados focada em memória.

Porém antes do desenvolvimento em si começar, foi pensado a criação do Diagrama de Classes, como guia para o desenvolvimento, saberíamos apartir do diagrama quais seriam os atributos de cada classe e que classes iríamos usar.

Após termos uma versão sólida do Diagrama de classes, começamos o desenvolvimento do BackEnd, enquanto isso, certos mockups tiveram de ser alterados e foram idealizadas versões dos mockups de forma Mobile/dinâmica.

O desenvolvimento do BackEnd foi complementado com a criação da nossa estrutura e da introdução da base de dados que viria a ser gerada automaticamente depois de definidas todas as entidades.

Como estrutura definimos, que existiria um diretório para os Models onde estariam as nossas classes de entidade, um diretório para os controllers onde seria feito o Handling dos nossos models com as suas funcionalidades e os respetivos endpoints, e o service que veria a servir os controllers e onde estaria a lógica e os algoritmos a ser chamados e implementados no Controller.

Assim que o BackEnd foi progredindo, os elementos do FrontEnd que estavam encarregues do Diagrama e dos Mockups, foram começando a desenvolver as páginas estáticas e progressivamente dinâmicas a trabalhar com os endpoints do BackEnd.

Nesta Milestone deparamo-nos com bastantes problemas tanto a nível de dificuldades do trabalho, como a nível interno com questões de horário e de organização, assim como problemas de comunicação.

Até à data final de entrega ficou desenvolvido a nível de funcionalidades :

- Os controlos de acesso (registo e login);
- A visualização dos dados de perfil;
- A comunicação com a API;
- Funcionalidades de Team, User e Tournaments mesmo que não implementadas a 100% no FrontEnd, devido aos problemas com JWT encontrados;

Milestone 3

A terceira milestone foi sem dúvida a mais complicada, trabalhosa e com mais problemas a nível de progresso dentro do grupo.

Apartir do feedback obtido na entrega da segunda milestone, começamos por corrigir os erros encontrados e as dificuldades enfrentadas.

Apartir daí conseguimos avançar progressivamente bem, desenvolvendo todos os pontos planeados e melhorar a qualidade dos anteriores, o FrontEnd ia acompanhando com a criação dos componentes que o BackEnd já tinha concluídos e fechados.

As partes mais complicadas foram desenvolvidas, RIOT API e Brackets, sendo que as Brackets não puderam ser desenvolvidas no frontEnd porque a base de dados começou a dar um erro, ao ponto que não conseguimos resolver devido à localização do erro tardia, mas confirmamos que o código estaria correto e funcionaria corretamente, vendo como a função corria quando chamada, ou seja o problema estaria localizado na Base de dados e não no código em geral.

Até ao prazo final foram desenvolvidos os seguintes pontos :

- Comunicação melhorada com a API incluindo a atualização dinâmica dos dados;
- Criar pedidos;
- Aceitar e recusar pedidos;
- Começar o torneio e gerar as brackets dinamicamente (apesar de não implementado no FrontEnd devido ao tal erro da Base de Dados);
- Criar equipas, editar equipas, apagar equipas, adicionar membros e remover membros;
- Criar torneios, editar torneios ,eliminar torneios, publicar torneios;
- Calcular estatísticas das equipas (apesar de não implementado juntamente com as Brackets);
- Documentação em Swagger;
- Criação de documentação específica;

Dificuldades Encontradas

Ao longo das 3 entregas fomos detetando dificuldades que nos levaram a pesquisar e trabalhar para poder ultrapassar, mesmo assim alguns não conseguimos resolver e afetaram a resolução final do nosso trabalho.

As dificuldades técnicas encontradas foram as seguintes :

- Curva de aprendizagem inicial de .NET;
- Implementação e correção de erros nas migrations automaticamente geradas pela base de dados;
- Erros com o JWT;
- Dificuldades com a Base de Dados e com o Model Builder que foi um erro fatal e crítico e impossibilitou a implementação das Brackets a certo ponto;
- Erros de git de última hora;

As dificuldades técnicas encontradas foram as seguintes :

- Falhas de comunicação;
- Incompatibilidades de horário;
- Má gestão de esforço;

Algumas das soluções que utilizamos para resolver os problemas foram através de pesquisas, testagem e contorno de obstáculos, por exemplo , no problema de modelação das Brackets usamos a criação da tabela intermédia como forma de usar as equipas e atribuir direito as posições.

Aspetos a melhorar

Como principais aspetos a melhorar no projeto foram definidos os aspetos que não conseguiram ser terminados ou alguns que podiam melhorar a qualidade do programa, tais como:

- Criação funcional das brackets com o FrontEnd e BackEnd;
- Atualização de aspetos visuais nas páginas automaticamente;
- Atualização da base de dados corretamente com o Model Builder;
- Desenvolvimento de funcionalidades extra como a de chat;

Foi evidenciado que a nossa dinâmica de grupo e organização precisava de ser também fortemente melhorada para futuros projetos do género.

Conclusão

Infelizmente não conseguimos terminar tudo que tínhamos planeado, não devido a erros de código mas da base de dados, o que nos impossibilitou de terminar a única issue que faltava, contudo desenvolvemos tudo o resto e tentamos trazer o maior esforço e qualidade no que ficou desenvolvido.

Estamos especialmente dececionados com o rumo final do nosso projeto, pois a issue fundamental não conseguiu ser terminada e o trabalho de grupo que começou tão bem, acabou desta forma.

No entanto tendo em conta tudo o que foi desenvolvido e terminado temos plena noção do esforço aplicado e da qualidade que introduzimos e daí achamos que fizemos o suficiente para passar nesta entrega, mesmo que não seja a nota que gente queira ou se contente, tendo em conta das boas notas nas entregas anteriores.