

INSTITUTO FEDERAL DO PARANÁ

ADRYELL NATHANN DA SILVA

JOÃO LENON LOPES MATTOSO

DALE.GG

UMA PLATAFORMA PARA JOGADORES DE LEAGUE OF LEGENDS

FOZ DO IGUAÇU, PR

2020

ADRYELL NATHANN DA SILVA
JOÃO LENON LOPES MATTOSO

DALE.GG
UMA PLATAFORMA PARA JOGADORES DE LEAGUE OF LEGENDS

Documentação do Projeto Integrador
apresentado ao Tecnólogo em Análise e
Desenvolvimento de Sistemas como requisito
parcial de avaliação.

Orientadores:
Humberto Beneduzzi

FOZ DO IGUAÇU, PR
2020

FOLHA DE APROVAÇÃO

ADRYELL NATHANN DA SILVA

JOÃO LENON LOPES MATTOSO

DALE.GG

UMA PLATAFORMA PARA JOGADORES DE LEAGUE OF LEGENDS

Documentação do Projeto Integrador apresentado como requisito parcial para a obtenção do título de Tecnólogo de nível superior, do curso Tecnologia em Análise e Desenvolvimento de Sistemas, aprovada pela seguinte banca examinadora:

Orientador(a): Humberto Martins Beneduzzi, Mestre em Engenharia Agrícola

Membro avaliador: Marcela Turim Koschevic, Mestre em Informática

Membro avaliador: Jefferson Oliveira Chaves

Foz do Iguaçu, 18 de março de 2020

RESUMO

Os E-Sports vem a cada dia mais ganhando espaço e mais pessoas entram nesse mundo de jogos. E em meio a tantos, League Of Legends é o que acaba se destacando nesse universo e muitas pessoas acabam ficando em dúvidas quanto ao seu desempenho ao longo do jogo e como é seu desenvolvimento nas partidas. Tendo em vista isso, foi criada pelos alunos Adryell Nathann da Silva e João Lenon Lopes Mattoso a plataforma “Dale.gg”. Essa plataforma tem em vista mostrar as estatísticas de um jogador em uma partida específica, com um campeão específico ou até mesmo de forma geral, podendo fazer uma análise de si mesmo e melhorando seu desempenho dentro do jogo. Foi criado ainda o aplicativo “Duozada”, um complemento ao “Dale.gg”, pois os jogadores terão mais vínculo com o jogo em si, buscando os aproximar e fazer com que os mesmos possam ter uma melhor experiência jogando ao lado de outra pessoa.

Palavras-chave: League Of Legends. Estatísticas. Aplicativo.

ABSTRACT

E-Sports is increasingly gaining space and more people enter this world of games. And among so many, League Of Legends is what ends up standing out in this universe and many people end up doubting about their performance throughout the game and how is their development in the matches. In view of this, the platform “Dale.gg” was created by students Adryell Nathann da Silva and João Lenon Lopes Mattoso. This platform aims to show the statistics of a player in a specific match, with a specific champion or even in general, being able to make an analysis of himself and improving his performance within the game. The “Duozada” application was also created, a complement to “Dale.gg”, as players will have more ties to the game itself, seeking to bring them closer and make them have a better experience playing with someone else.

Key words: League Of Legends. Statistics. App.

LISTA DE QUADROS

QUADRO 1	Requisitos Funcionais	18
QUADRO 2	Regras de Negócio.....	18
QUADRO 3	Requisitos não funcionais tecnológicos.....	19

LISTA DE ILUSTRAÇÕES

FIGURA 1	Diagrama de caso de uso.....	21
FIGURA 2	Diagrama de classes	22
FIGURA 3	Diagrama de Entidade Relacionamento.....	23
FIGURA 4	Diagrama de Sequência.....	24

SUMÁRIO

1	INTRODUÇÃO.....	9
1.2	OBJETIVO GERAL;.....	9
1.3	OBJETIVOS ESPECÍFICOS	9
1.4	METODOLOGIAS	10
2	TRABALHOS RELACIONADOS.....	12
3	TECNOLOGIAS UTILIZADAS.....	13
4	ESTUDO DE CASO.....	17
4.1	CONTEXTUALIZAÇÃO.....	17
4.2	LEVANTAMENTO DE REQUISITOS.....	17
4.2.1	REQUISITOS FUNCIONAIS.....	17
4.2.2	REGRAS DE NEGÓCIO.....	18
4.2.3	REQUISITOS NÃO FUNCIONAIS TECNOLÓGICOS.....	19
4.3	DIAGRAMAS DE ANÁLISE E MODELAGEM DO SISTEMA.....	21
4.3.1	DIAGRAMAS DE CASO DE USO.....	21
4.3.2	DIAGRAMA DE CLASSES.....	22
5	RESULTADOS	
	OBTIDOS.....	23
5.1	DIAGRAMAS DO PROJETO.....	23
5.1.1	DIAGRAMA DE ENTIDADE RELACIONAMENTO.....	23
5.1.2	DIAGRAMA DE SEQUÊNCIA.....	24
6	CONCLUSÃO.....	25
	REFERÊNCIAS BIBLIOGRÁFICAS.....	26

1 INTRODUÇÃO

O "Dale.gg" tem como objetivo apresentar os dados de um usuário do jogo League of Legends para uma análise específica de seus dados e uma melhor visão de si mesmo dentro do jogo.

A pesquisa traz uma vantagem para os jogadores observarem melhor tanto seus pontos fracos como seus pontos fortes, dessa forma o mesmo pode analisar melhor onde está errando, melhorando esses pontos fracos, e onde ele se sai melhor, para poder manter esses pontos altos.

O módulo "Duozada", visa ajudar os usuários a encontrarem sua dupla perfeita dentro do jogo, melhorando, assim, exponencialmente seus resultados e seu entretenimento acima de tudo.

1.2 OBJETIVO GERAL

Desenvolver uma plataforma para ajudar jogadores a melhorarem seus resultados dentro do jogo League of Legends além de uma aplicação mobile para auxiliá-los a encontrar sua dupla perfeita dentro do jogo através do módulo "Duozada".

1.3 OBJETIVOS ESPECÍFICOS

- Permitir a busca de jogadores do League of Legends
- Apresentar os gráficos estatísticos de um jogador em relação às partidas e campeões
- Apresentar o histórico de partidas assim como todos os detalhes de cada uma
- Permitir a visualização do perfil de outros usuários visando encontrar sua dupla dentro do jogo

1.4 METODOLOGIAS

O projeto se iniciará com a discussão dos requisitos do projeto e da implementação dos diagramas de UML utilizando a ferramenta AstahUML, que ajudarão durante todo o processo de planejamento e, posteriormente, no desenvolvimento da aplicação.

O desenvolvimento da aplicação começa com uma criação de conta no portal de desenvolvedores do jogo League of Legends, onde poderemos gerar uma chave de acesso a API da Riot Games. Com esse acesso, poderemos acessar uma API totalmente RESTful e fazer requisições de variados tipos, a fim de obter os dados dos jogadores.

A chave para API da Riot Games possui um limite de requisições por segundo e por minuto, onde será necessário a implementação de um banco de dados relacional utilizando o PostgreSQL para não excedermos o limite de requisições.

Com o acesso a API da Riot Games, partimos para o desenvolvimento de uma biblioteca usando o NodeJS, onde vamos organizar nossas requisições dentro de variados métodos, como a nossa chave da API tem um limite, a lib “ZedJS” vai se comportar como um *Gateway* entre os nossos sistemas e a API da Riot, ela vai controlar nossas requests para que não seja possível estourar o número de requests da nossa chave, sendo assim podemos usar vários *endpoints* da API e consumir todos os dados desencapsulados e organizados também pela “ZedJS.”

Digamos que estamos em um ambiente de desenvolvimento e talvez já tenhamos muitas funcionalidades implementadas, nesse caso, utilizamos o padrão Git Flow para desenvolver cada “feature” do nosso projeto e armazenar no repositório do GitHub, além do uso de uma tática de desenvolvimento ágil chamada “Scrum”. Sendo assim, teremos “sprints” de uma semana cada, onde será implementado pelo menos cinco “features” ou mais a cada semana.

Cada um dos nossos serviços podem ser encontrados no GitHub, onde estamos versionando o código, além disso estamos utilizando o Docker para deixar tudo containerizado e conseguir utilizar todos em conjunto de forma simples.

Ainda na questão GIT, iremos realizar um deploy automatizado da “branch master” no serviço do DockerHub e Amazon EC2 (VPS), ou seja, a cada “merge” das “pull request” ou “push” na “master”, nossos containers no DockerHub vão ser atualizados e depois nossa VPS no EC2 da Amazon irá se atualizar levantando a versão “latest” do nosso container.

Para o módulo “Duozada”, iremos precisar ir um pouco mais a fundo nas tecnologias, sendo necessário o uso de “socket.io” para ocorrer o “duozada” entre usuários e possibilitar também o uso do chat entre os dois, ambos em tempo real.

Inicialmente, o objetivo é finalizar uma API Web Service cem por cento RESTful, que irá servir o nosso próximo projeto em React Native.

2 TRABALHOS RELACIONADOS

2.1 OP.GG

O “op.gg” é um site de análise de dados de um jogador no jogo League of Legends. No “dale.gg” será um pouco semelhante ao “op.gg” onde você também poderá pesquisar por jogadores através da API da Riot Games e mostrar os dados desse jogador para o usuário.

2.2 League of Graphs

O “League of Graphs” é um site utilizado para análise de dados através de gráficos onde é possível ver os detalhes de uma partida ao vivo. O “dale.gg” apenas conseguirá mostrar os gráficos da partida assim como o League of Graphs, estamos estudando a possibilidade de implementar as partidas ao vivo.

2.3 Tinder

No “dale.gg”, o Tinder é referenciado pois o projeto terá um módulo chamado “duozada” onde você pode encontrar sua dupla no jogo através de uma interação de arrastar para o lado para dar um “like” ou “deslike”.

3 TECNOLOGIAS UTILIZADAS

3.1 ESLINT v6.0.8 / PRETTIER v2.0.4

O ESLint e o Prettier são ferramentas de análise de código estática para identificar padrões problemáticos encontrados no código JavaScript. As regras no ESLint e no Prettier são configuráveis e regras personalizadas podem ser definidas e carregadas em seus respectivos arquivos.

3.2 TYPESCRIPT v3.8.3

TypeScript é um superconjunto de JavaScript desenvolvido pela Microsoft que adiciona tipagem e alguns outros recursos a linguagem. Anders Hejlsberg, arquiteto da linguagem C# e criador das linguagens Delphi e Turbo Pascal, trabalhou no desenvolvimento do TypeScript. Junto com ESLint e Prettier, o TypeScript vai conseguir garantir boas práticas dentro do nosso código.

3.3 JEST v26.0.1

Jest é uma estrutura de teste de JavaScript mantida pelo Facebook, Inc. com foco na simplicidade. Trabalha com projetos usando: Babel, TypeScript, Node.js, React, Angular, Vue.js e Svelte. Seu objetivo é funcionar fora da caixa e sem configurações. Será com ele que vamos realizar nossos testes unitários dentro da nossa aplicação.

3.4 TSYRINGE v3.8.3

O Tsyrringe é uma biblioteca para NodeJS que funciona como um container de injeção de dependência leve para TypeScript/JavaScript para conseguir injetar algo dentro do método construtor das classes, esse tipo de tecnologia é conhecida como IoC (Container de Inversão de Controle). Ele será de extrema importância no nosso projeto pois estamos utilizando DDD e TDD, então vamos conseguir fazer mock de classes e de dados, classes Fakes que vão conseguir substituir outras sempre que necessário, com um IoC se torna muito simples criar códigos desacoplados, atualmente nosso projeto consegue substituir o ORM TypeORM pelo Knex por exemplo.

3.5 NODEJS v12.16.1

O NodeJS é um runtime - aplicação que possibilita o processamento, a renderização e a execução de elementos escritos em linguagem não suportada nativamente pelo sistema - criada por Ryan Dahl em 2009. Usaremos ele em um ambiente Server Side, para usufruir de vários aspectos e elementos do Javascript.

3.6 EXPRESS v4.17.6

O Express, é uma estrutura de aplicativo da Web para o Node.js, lançada como software livre e de código aberto sob a Licença MIT. Ele foi projetado para criar aplicativos da Web e APIs.

3.7 ZEDJS v1.1.7

Desenvolvido pela equipe do Dale.gg, a biblioteca ZedJS se comporta como um Gateway entre os sistemas e a API da Riot Games/Data Dragon, ela vai controlar todas as requests para que não seja possível estourar o número de requests da chave de API fornecida a ela para realizar as requisições, sendo assim, com ela é possível usar vários endpoints da API e consumir todos os dados desencapsulados e organizados também pela ZedJS.

3.8 API DA RIOT GAMES v10.18.1

Essa API externa nos fornece todos os dados de um jogador dentro do jogo League of Legends. Construiremos nossos gráficos e dados dos jogadores usando a API da Riot games.

3.9 POSTGRESQL v13

O PostgreSQL é um sistema gerenciador de banco de dados objeto-relacional. Utilizaremos para armazenar os dados de nossa aplicação.

3.10 ASTAH UML v8.1.0

O Astah UML Community é um software de desenvolvimento de diagramas UML. Utilizaremos ele para criação de nossos diagramas.

3.11 GITHUB v2.24.3

O GitHub é uma plataforma de hospedagem de código fonte com controle de versões utilizando o GIT. Será utilizado em nosso projeto para controle de versionamento e armazenamento do código fonte.

3.12 SWAGGER v3.0.0

O Swagger é uma ferramenta que tem como objetivo testar e documentar aplicações RESTful e GraphQL por meio do envio de requisições HTTP e da análise de seu retorno. Será utilizado para realizar as requisições à API Web Service e manter nossos endpoints documentados.

3.13 VISUAL STUDIO CODE v1.48

O Visual Studio Code é um editor de código fonte. Ele inclui suporte para depuração, controle GIT incorporado, realce de sintaxe, complementação inteligente do código, snippets e refatoração de código. Utilizaremos para desenvolver o nosso código fonte da aplicação, depurar e controlar nosso repositório no GitHub.

3.14 AMAZON EC2 US-EAST-1

O EC2 é um serviço da Amazon de origem americana de infraestrutura em nuvem que prove ajudar a implantar e dimensionar aplicativos executados simultaneamente em vários computadores (VPS). Faremos o deploy de nossa “branch master” na Amazon EC2.

3.15 DOCKER v19.03.8

Docker é um software da empresa Docker, Inc, que fornece uma camada de abstração e automação para virtualização de sistemas operacionais como Windows e Linux. Com o Docker conseguimos criar containers para evitar a sobrecarga na nossa VM (Máquina Virtual). Vamos utilizá-lo para rodar todas as migrations do nosso banco de dados em conjunto de cada serviço.

3.16 DOCKER COMPOSE v1.25.5

Docker Compose é o orquestrador de containers da Docker. E como funciona um orquestrador em uma orquestra? Ele rege como uma banda deve se comportar/tocar durante

uma determinada apresentação ou música. Com o Docker Compose é a mesma coisa, mas os maestros somos nós, vamos utilizar ele para orquestrar todos os nossos containers dentro do EC2 da Amazon e deixar tudo funcional.

3.17 REDIS v6.0.8

Redis é uma base de dados em rede open-source, que armazena chaves com durabilidade opcional, patrocinado pela Pivotal Software e pela VMware, a partir de 2015 mudou para a Redis Labs. De acordo com o ranking mensal da DB-Engines.com, Redis é o banco de dados de valores-chave mais popular do mundo. Ele é excepcional para cache de dados, vamos utilizar ele para cachear IP's em nossos rate limiters e bloquear requisições dos usuários pelo IP.

3.18 AMAZON S3 US-EAST-1

O Amazon S3 ou o Amazon Simple Storage Service é um serviço oferecido pelo Amazon Web Services que fornece armazenamento de objetos por meio de uma interface de serviço da web. O Amazon S3 usa a mesma infraestrutura de armazenamento escalável usada pela Amazon.com para executar sua rede global de comércio eletrônico. Vamos utilizar ele para guardar as imagens dos nossos usuários e algumas imagens estáticas da aplicação.

3.19 AMAZON SES US-EAST-1

Serviço de envio e recebimento de e-mails em grande escala na nuvem. O Amazon Simple Email Service (SES) é um serviço de e-mail eficaz, flexível e dimensionável. Com ele, os desenvolvedores podem enviar e-mails de qualquer aplicação. Vamos utilizar esse serviço para fazer envio de email para nossos usuários.

3.20 DEDSEC v1.1.8

Desenvolvido pela equipe SecJS, a dedSec é uma biblioteca para NodeJS com várias funcionalidades que buscam deixar o código mais organizado sem muitas repetições nos projetos, ela possui responses padronizadas, gerador de exceptions, um simples IoC Container etc. Vamos utilizar ela para deixar nossas responses padronizadas dentro do nosso projeto, assim na hora de fazer o frontend vamos ter uma noção de como vamos receber as requests.

4 ESTUDO DE CASO

4.1 CONTEXTUALIZAÇÃO

O cenário do e-sports vem, na última década, passando por um crescimento exponencial, visto que sempre foi uma modalidade que dava indícios de sucesso provável, por acompanhar as novas tendências comportamentais das gerações que cresceram com acesso à internet. O maior destaque desse mercado, e que aliás é o jogo mais jogado do mundo, é o League of Legends, que, segundo o último levantamento feito pela empresa criadora do jogo (Riot Games), reúne hoje cerca de 100 milhões de jogadores pelo mundo todo.

Sendo assim, o “dale.gg” surgiu como uma opção para os jogadores da região brasileira poderem conferir tudo o que acontece no seu perfil do League of Legends e nas suas partidas, dando a eles a opção de conferir como estão se saindo em cada um dos seus jogos. Ou seja, o jogador vai poder ver quais são seus pontos fortes e seus pontos fracos, fazendo com que ele possa sempre melhorar no jogo e chegar a um nível PRO. Além disso, o “dale.gg” surge com uma opção um tanto quanto inusitada, um módulo chamado "Duozada" onde os jogadores vão poder encontrar seu par perfeito dentro do jogo. Essas pesquisas terão filtros para que os jogadores possam encontrar alguém que possa preencher os seus pontos fracos, sendo assim, seus índices de vitória iriam aumentar exponencialmente dentro do jogo!

4.2 LEVANTAMENTO DE REQUISITOS

4.2.1 REQUISITOS FUNCIONAIS

Após alguns dias de reuniões, chegamos aos requisitos funcionais que serão mostrados na tabela abaixo. Foi analisado alguns softwares para montagem desses requisitos além da decisão da melhor forma de implementação do projeto mas visando uma melhor usabilidade para o usuário.

Quadro 1 – Requisitos funcionais do sistema do dale.gg

Código	Requisito Funcional
RF 01	O sistema deve permitir o cadastro de usuários.
RF 02	O sistema deve permitir que um usuário já cadastrado faça o login na plataforma.
RF 03	O sistema deve permitir que um usuário recupere sua senha.
RF 04	O sistema deve permitir a visualização e a alteração do perfil.
RF 05	O sistema deve permitir que um usuário pesquise por Invocadores dentro da aplicação.
RF 06	O sistema deve permitir a visualização do histórico de partidas assim como todas as informações de cada uma.
RF 07	O sistema deve permitir a visualização do histórico de um jogador em relação às partidas jogadas: total de partidas, total vitórias, total derrotas, etc.
RF 08	O sistema deve permitir a visualização do histórico de um jogador em relação aos campeões jogados: campeão mais jogado, campeão menos jogado, campeão com melhor índice de vitória, etc.
RF 09	O sistema deve permitir que um usuário acesse o módulo "Duozada" para encontrar outros usuários.
RF 10	O sistema deve permitir que um usuário cadastre suas melhores Jogadas em seu perfil.
RF 11	O sistema deve permitir que um usuário possa encontrar e detalhar o perfil de outros jogadores.
RF 12	O sistema deve permitir que um usuário possa dar um Like no perfil de outros jogadores.
RF 13	O sistema deve permitir que aconteça um match entre dois usuários.
RF 14	O sistema deve permitir que aconteça um match perfeito entre dois usuários.
RF 15	O sistema deve permitir que os usuários conversem via Chat após ocorrer um match.

4.2.2 REGRAS DE NEGÓCIO

Abaixo é apresentado as regras de negócio discutidas em reuniões feitas entre os membros do grupos.

Quadro 2 – Regras de negócio do dale.gg

Código	RF	Regra de Negócio
RN 01	RF 01	O usuário deverá fazer um primeiro cadastro fornecendo informações básicas, e algumas informações do jogo como, 3 CAMPEÕES que mais joga e 2 ROTAS que mais joga.

RN 02	RF 04	O usuário só poderá fazer edições no seu perfil caso esteja autenticado.
RN 03	RF 03	O usuário poderá recuperar sua senha e confirmar sua conta via token no email.
RN 04	RF 05	O usuário poderá fazer requisições a API da Riot sem estar autenticado.
RN 05	RF 05	O usuário deve fazer as requisições a API da Riot pelo nome de algum Invocador apenas.
RN 06	RF 06 RF 07 RF 08	O usuário poderá atualizar o perfil de algum Invocador, para refazer a requisição à API da Riot.
RN 07	RF 09	O usuário deverá estar logado na aplicação para utilizar os recursos do “Duozada”.
RN 08	RF 10	O sistema deve fornecer para o usuário a opção de cadastrar suas melhores Jogadas quando acessar o “Duozada”, ou pular a opção.
RN 09	RF 11	O usuário poderá listar e detalhar o perfil de outros jogadores na plataforma, assim como suas Jogadas.
RN 10	RF 12 RF 13	O usuário poderá dar um Like no perfil de outros jogadores, caso o outro jogador também de um like no perfil do mesmo, irá ocorrer um Match.
RN 11	RF 13 RF 14	Quando ocorrer um Match entre usuários, o sistema deverá validar se, se trata de um Match comum (RN 10) ou um Match Perfeito (RN 12).
RN 12	RF 14	Após um Match, o sistema deverá validar caso haja a opção de ocorrer um Match perfeito, baseando-se pelas rotas de cada jogador e campeões jogados.
RN 13	RF 15	Apenas após um Match, os usuários terão a opção de conversar via Chat dentro da aplicação.
RN 14	RF 02	Um User que estiver com seu status setado como false, não poderá acessar a aplicação

4.2.3 REQUISITOS NÃO FUNCIONAIS TECNOLÓGICOS

Abaixo é apresentado os requisitos não funcionais tecnológicos discutidas em reuniões feitas entre os membros do grupos.

Quadro 3 – Requisitos não funcionais tecnológicos do dale.gg

Código	Requisito Não Funcional Tecnológico
RNFT 01	Um script será criado para realizar um pré cadastro dos Campeões na aplicação .

RNFT 02	O sistema deve fazer uso da API da Riot Games para buscar as informações dos Invocadores.
RNFT 03	O banco de dados a ser utilizado terá o padrão objeto-relacional
RNFT 04	A aplicação irá usar o design pattern MVC
RNFT 05	O sistema deve armazenar as informações vindas da API da Riot dentro do banco de dados local (PostgreSQL).
RNFT 06	Like e Dislike serão apenas funcionalidades usando o método Store para armazenar essas informações em um Usuário.
RNFT 07	O chat de texto pós match será implementado usando socket.io.

4.3 DIAGRAMAS DE ANÁLISE E MODELAGEM DO SISTEMA

4.3.1 DIAGRAMA DE CASO DE USO

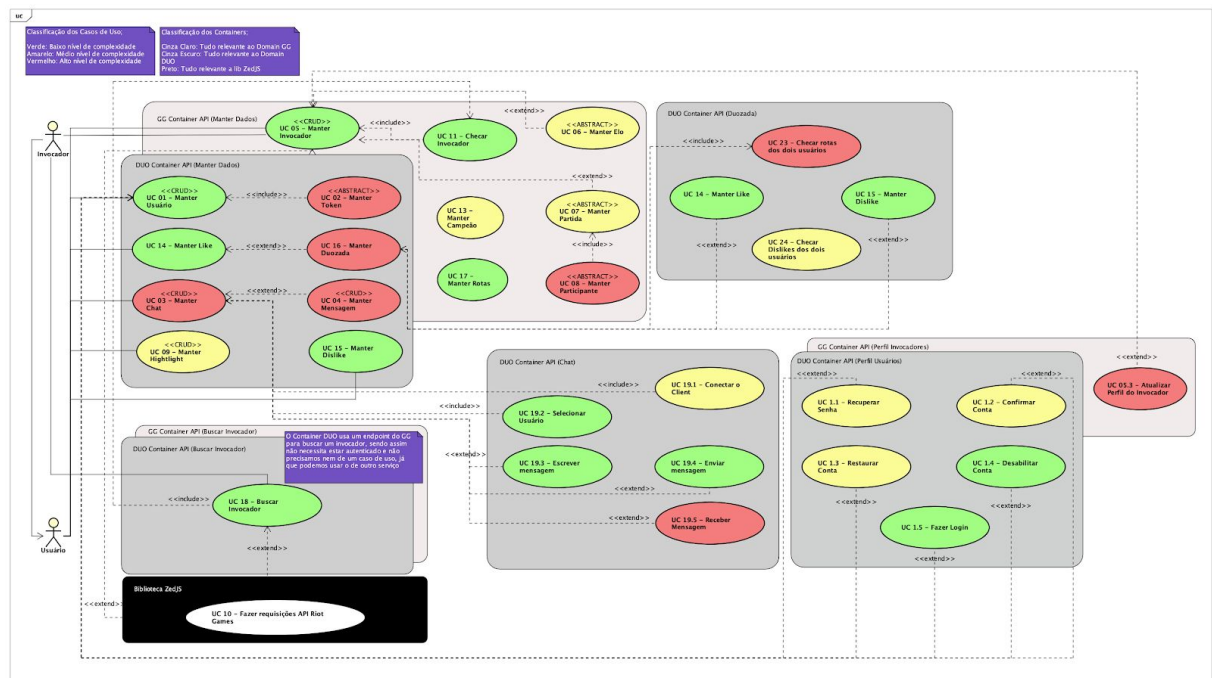


FIGURA 1: DIAGRAMA DE CASO DE USO

4.3.2 DIAGRAMA DE CLASSES

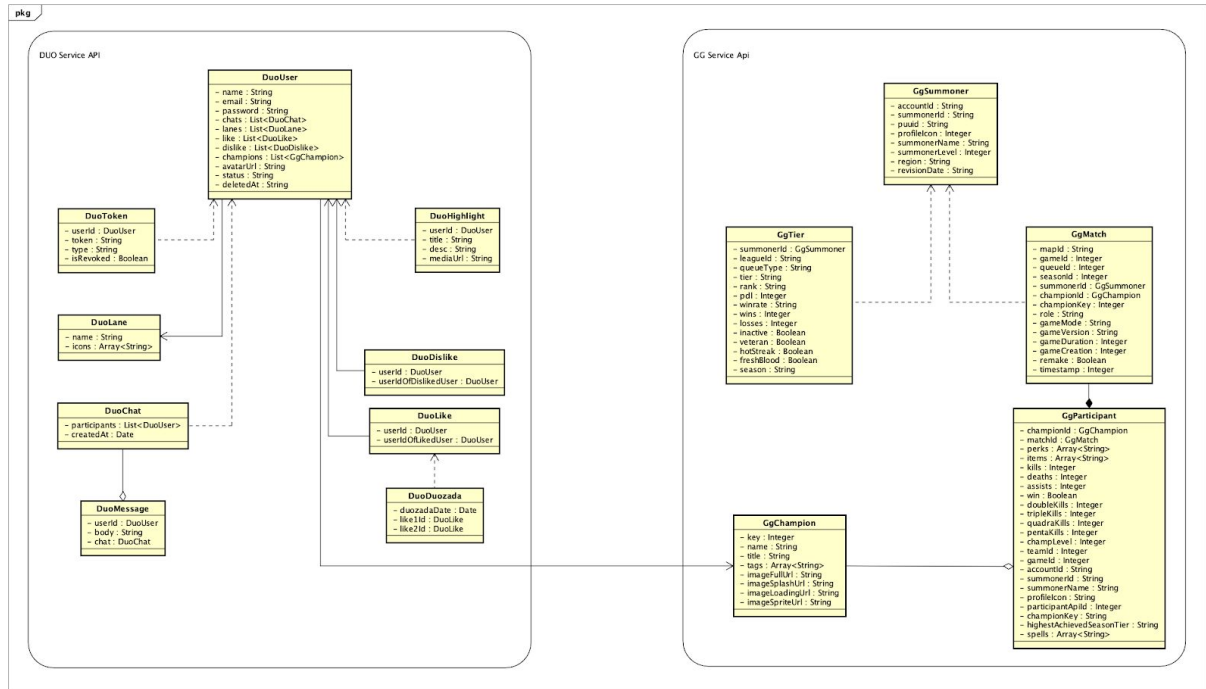


FIGURA 2: DIAGRAMA DE CLASSES DO “DALEGG”

5.1.1 DIAGRAMA DE ENTIDADE RELACIONAMENTO



5.1.2 DIAGRAMA DE SEQUÊNCIA

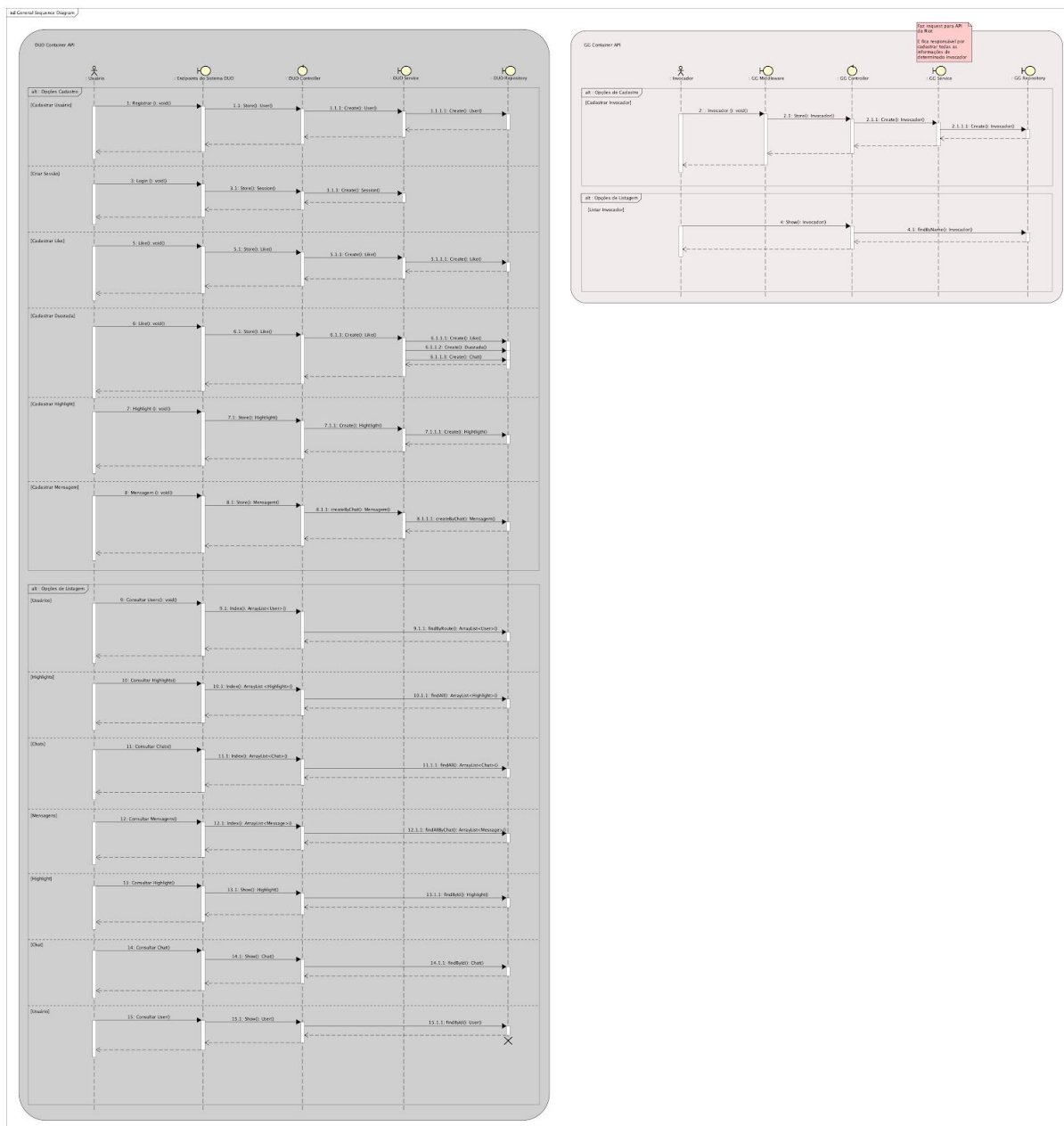


FIGURA 4: DIAGRAMA DE SEQUÊNCIA

6 CONCLUSÃO

O mundo dos games, indiscutivelmente, vem ganhando mais espaço a cada ano entre as mais variadas pessoas, fazendo-se necessário um maior suporte a elas para que possam ter uma melhor experiência nesse universo.

Com isso, o projeto foi construído, como informando no início do trabalho, visando auxiliar os jogadores de League Of Legends a terem uma visão melhor de si mesmo dentro do jogo e como é seu desempenho nas partidas. Buscamos eles possam ter um avanço dentro do jogo, melhorando nos mais diversos aspectos além de garantir uma maior diversão junto de outras pessoas através do módulo “Duozada”.

Agradecemos aos professores que nos acompanharam por esse período, sem todo o conhecimento repassado por eles não conseguiríamos desenvolver esse projeto além de termos todo o suporte através do Campus.

REFERÊNCIAS BIBLIOGRÁFICAS

OP.GG. **About OP.GG.**

Disponível em: <<https://br.op.gg/about/>>

League Of Graphs.

Disponível em: <<https://www.leagueofgraphs.com/>>

Tinder. **O que é o Tinder?**

Disponível em:

<<https://www.help.tinder.com/hc/pt-br/articles/115004647686-O-que-%C3%A9-o-Tinder->>

EDUARDO RABELO. **Evitando erros com ESLint.** 2016.

Disponível em:

<<https://medium.com/@oieduardorabelo/evitando-erros-com-eslint-91b5a4bb9471>>

JULIO CARNEIRO. **Usando Prettier com VS Code e create-react-app.** 2018.

Disponível em:

<<https://medium.com/reactbrasil/usando-prettier-com-vs-code-e-create-react-app-385f2262d442>>

MATHEUS MARIANO. **O mínimo que você precisa saber sobre TypeScript.** 2016.

Disponível em:

<<https://medium.com/@matheusmariano/o-m%C3%A9nimo-que-voc%C3%AA-precisa-saber-sobre-typescript-58d1b418f78b>>

EDUARDO RABELO. **Jest: Escrever testes nunca foi tão divertido.** 2016.

Disponível em:

<<https://medium.com/@oieduardorabelo/jest-escrever-testes-nunca-foi-t%C3%A3o-divertido-5f0e1950ba10>>

PEDRO HENRIQUE. Utilizando injeção de dependências no NodeJs ♥ TypeScript. 2020.

Disponível em:

<<https://medium.com/@pedro.lg.cs/utilizando-inje%C3%A7%C3%A3o-de-depend%C3%AAnCIAS-no-nodejs-typescript-9ae4aa5dacdf>>

LENON. **Node.js – O que é, como funciona e quais as vantagens.** 2018.

Disponível em: <<https://www.opus-software.com.br/node-js/>>

Node.js Express FrameWork Tutorial - Learn in 10 Minutes.

Disponível em: <<https://www.guru99.com/node-js-express.html>>

RIOT GAMES. **About The Riot Games API.**

Disponível em: <<https://developer.riotgames.com/>>

DEVMEDIA. **Introdução ao PostgreSQL.** 2007.

Disponível em: <<https://www.devmedia.com.br/introducao-ao-postgresql/6390>>

BRUNO SEABRA. **O que é o Astah?.** 2015.

Disponível:

<<https://www.startupsstars.com/2015/10/o-que-e-o-astah-posttecnico-por-bruno-seabra/>>

BRENDON MARQUES. **O Que é GitHub e Para Que é Usado?.** 2019.

Disponível em: <<https://www.hostinger.com.br/tutoriais/o-que-github/>>

MONISE COSTA. **Swagger: Como gerar uma documentação interativa para API REST.** 2016.

Disponível em:

<<http://www.matera.com/blog/post/swagger-como-gerar-uma-documentacao-interativa-para-api-rest>>

MICROSOFT. **Why did we build Visual Studio Code?.** 2020.

Disponível em: <<https://code.visualstudio.com/docs/editor/whyvscode>>

AMAZON. **O que é o Amazon EC2?**. 2020.

Disponível em:

<https://docs.aws.amazon.com/pt_br/AWSEC2/latest/UserGuide/concepts.html>

AMAZON. **O que é o Amazon S3?**. 2020.

Disponível em: <https://docs.aws.amazon.com/pt_br/AmazonS3/latest/dev/Welcome.html>

AMAZON. **Amazon Simple Email Service**. 2020.

Disponível em: <<https://aws.amazon.com/pt/ses/>>

POSITIVO TECNOLOGIA. **Container docker: o que é e quais são as vantagens de usar?**. 2017.

Disponível em: <<https://www.meupositivo.com.br/panoramapositivo/container-docker>>

CRISTIAN TRUCCO. **Docker Compose: O que é? Para que serve? O que come?**. 2018.

Disponível em:

<<https://imasters.com.br/banco-de-dados/docker-compose-o-que-e-para-que-serve-o-que-com-e>>

AMAZON. **O que é o Redis?**. 2020.

Disponível em: <<https://aws.amazon.com/pt/elasticache/what-is-redis/>>