

INSTITUTO FEDERAL DO PARANÁ

**ADRYELL NATHANN DA SILVA
JOÃO LENON LOPES MATTOSO**

DALE.GG

FOZ DO IGUAÇU, PR

2020

ADRYELL NATHANN DA SILVA
JOÃO LENON LOPES MATTOSO

DALE.GG

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

Orientadores:

Humberto Beneduzzi

Tassio Pinheiro

FOZ DO IGUAÇU, PR

2020

FOLHA DE APROVAÇÃO

ADRYELL NATHANN DA SILVA

JOÃO LENON LOPES MATTOSO

DALE.GG

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 Beneduzzi, com titulação

Colegiado de Informática, IFPR

Orientador(a): Tássio Pinheiro, com titulação

Colegiado de Informática, IFPR

Membro avaliador: Nome completo do Membro Avaliador 1, com titulação

Colegiado de {nome do colegiado}, IFPR

Foz do Iguaçu, XX de novembro de 2020

RESUMO

Elemento obrigatório, que evidencia os pontos mais importantes do trabalho, apresentando inclusive os resultados obtidos. Deve ser digitado em parágrafo único, com espaçamento simples, contendo de 150 a 500 palavras (teses, dissertações e outros), e de 100 a 250 palavras (artigos de periódicos). Utilizar a terceira pessoa do singular mantendo o verbo na voz ativa. O termo RESUMO deve ser escrito em letras maiúsculas, em negrito e centralizado. Devem ser indicadas de três a cinco palavras-chave, separadas entre si por ponto, que representem o assunto abordado no trabalho. O termo Palavras-chave deve ser escrito em letras minúsculas, em negrito, com a primeira letra em maiúscula. Deve ser deixada uma linha em branco entre a palavra RESUMO.

Palavras-chave: Termo 1. Termo 2. Termo 3.

ABSTRACT

Elemento obrigatório para teses e dissertações, escrito, preferencialmente, na língua inglesa. Deve ser digitado em folha separada, da mesma forma e contendo as mesmas informações do resumo apresentado na língua portuguesa, inclusive as palavras-chave. O termo ABSTRACT deve ser escrito em letras maiúsculas, em negrito e centralizado. Devem ser indicadas de três a cinco palavras-chave, separadas entre si por ponto, que representem o assunto abordado no trabalho.

Key words: Termo 1. Termo 2. Termo 3.

LISTA DE ILUSTRAÇÕES

<u>FIGURA 1 – Diagrama de casos de uso geral da dale.gg</u>	13
<u>FIGURA 2 – Diagrama de classes de implementação da dale.gg</u>	15
<u>FIGURA 3 – Exemplo de diagrama de sequência para requisição à api</u>	17
<u>FIGURA 3 – Exemplo de diagrama de sequência para encontrar seu duo</u>	19

LISTA DE QUADROS

QUADRO 1 - Requisitos Funcionais	16
QUADRO 2 - Regras de Negócio	17
QUADRO 3 - Requisitos não funcionais tecnológicos	18

SUMÁRIO

SUMÁRIO	8
1. INTRODUÇÃO	9
1.2. OBJETIVO GERAL	9
1.3. OBJETIVOS ESPECÍFICOS	9
1.4. METODOLOGIAS	9
2. TRABALHOS RELACIONADOS	11
3. TECNOLOGIAS UTILIZADAS	12
4 ESTUDO DE CASO	15
4.1 CONTEXTUALIZAÇÃO	15
4.2 LEVANTAMENTO DE REQUISITOS	15
4.2.1 REQUISITOS FUNCIONAIS	16
4.2.2 REGRAS DE NEGÓCIO	17
4.2.3 REQUISITOS NÃO FUNCIONAIS TECNOLÓGICOS	18

1. INTRODUÇÃO

A 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 esta 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

Ajudar jogadores a melhorarem seus resultados dentro do jogo League of Legends além de 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 campeonos
- 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 StarUML, 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.

Com este acesso, partimos para o desenvolvimento de uma API Web Service usando o AdonisJS, onde faremos a requisição a qualquer link da API da Riot Games, tendo um retorno de um arquivo “JSON” com dados encapsulados.

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.

Tendo os dados dos jogadores, iniciamos um CRUD de Usuário com autenticação e envio de TOKEN no e-mail para confirmação de cadastro e restauração de senha, sendo assim, já possuímos informações o suficiente para inicialização dos testes unitários com o TDD e testar qualquer funcionalidade futura da aplicação. É seguido um padrão que, após cada funcionalidade implementada via teste unitário seja finalizada, é passado para um segundo tipo de teste utilizando o “Insomnia” para fazer requisições a nossa API Web Service.

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.

Ainda na questão GIT, iremos realizar um deploy automatizado da “branch master” na Digital Ocean (VPS), ou seja, a cada “merge” das “pull request” ou “push” na “master”, nossa VPS na Digital Ocean irá se atualizar com os dados recebidos na “branch” do Git Hub.

Para o módulo “duozada”, iremos precisar ir um pouco mais a fundo nas tecnologias, sendo necessário o uso de “socket.io” para ocorrer o “match” 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. Porém, gostaríamos de implementar, também, algumas simples telas usando ReactJS para demonstrar os retornos da API da Riot Games.

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.

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 joga através de uma interação de arrastar para o lado para dar um “like” ou “deslike”

3. TECNOLOGIAS UTILIZADAS

- ReactJS

O ReactJs é um framework front-end. Usaremos ele para fazer o front-end da aplicação.

- React Native

O React Nativa é um framework front-end para aplicações mobile híbridadas. Usaremos para o desenvolvimento mobile da aplicação

- Expo

O Expo é uma ferramenta utilizada no desenvolvimento mobile juntamente com o React Native. Usaremos ele para termos fácil acesso as APIs nativas do dispositivos. Exemplos: câmera, arquivos, microfone

- NodeJS

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. Usaremos ele um ambiente Server Side, para usufruir de vários aspectos e elementos do Javascript.

- AdonisJS

O AdonisJS é um framework para NodeJS inspirado no Laravel. Usaremos ele para a construção da nossa API RESTful.

- API da Riot Games

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.

- PostgreSQL

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

- StarUML

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

- GitHub

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.

- Insomnia

O Insomnia é uma ferramenta que tem como objetivo testar ferramentas RESTful por meio do envio de requisições HTTP e da análise de seu retorno. Será utilizado para realizar as requisições à API Web Service.

- Visual Studio Code

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.

- Postico

O Postico é um software gráfico para administração do SGBD PostgreSQL. Usaremos no gerenciamento de nosso banco de dados.

- Heroku

O heroku é uma plataforma em nuvem como um serviço que suporta várias linguagens de programação. Faremos o deploy de nossa “branch develop” e de nosso repositório front-end (ReactJS)

- Digital Ocean

A Digital Ocean é uma provedora americana de infraestrutura em nuvem que ajudam a implantar e dimensionar aplicativos executados simultaneamente em vários computadores (VPS). Faremos o deploy de nossa “branch master” na Digital Ocean.

4 ESTUDO DE CASO

4.1 CONTEXTUALIZAÇÃO

Durante os passos iniciais, startups estão inseridas em um ambiente de total incerteza e todos os esforços são no sentido de encontrar um modelo de negócio. Após a definição do modelo e com o surgimento e crescimento das receitas iniciais, novos investimentos serão necessários para tornar o negócio repetível e escalável a ponto de se tornar uma empresa altamente rentável. Possuímos hoje, apenas uma ideia do que o “dale.gg” pode se tornar no futuro, uma empresa altamente rentável. Para isso se faz necessário um gerenciamento constante do projeto em todos os seus níveis de gerenciamento.

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.

FONTE: AUTORIA PRÓPRIA (2020)

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.