

ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Rafael Alves de Souza

**Analista virtual de League of Legends
IZI Analytics**

**SÃO PAULO
2020**

Rafael Alves de Souza

Analista virtual de League of Legends

Trabalho de Curso submetido à
Universidade de São Paulo (USP)
como parte dos requisitos
necessários para a obtenção do
Grau de Bacharel em Engenharia
Elétrica

SÃO PAULO
2020

Rafael Alves de Souza

Analista virtual de League of Legends

Trabalho de Curso submetido à
Universidade de São Paulo (USP)
como parte dos requisitos
necessários para a obtenção do
Grau de Bacharel em Engenharia
Elétrica

Área: Engenharia Elétrica com
ênfase em Computação

Orientador: Professor Doutor Ricardo
Nakamura

SÃO PAULO
2020

Dedico esse projeto a todos
que me apoiaram ao longo
desses anos, tanto no ensino
com conhecimentos técnicos
quanto apoio emocional.

Agradecimentos

Ao Professor Doutor Ricardo Nakamura pela orientação e apoio durante a elaboração do projeto.

A empresa Riot Games pelo fornecimentos dos dados, aos amigos Pedro Doconski, Letícia Mantovani, Larissa Bittar, Thomas Doconski e Leonardo Alves de Souza pelo apoio e auxílio durante o projeto e a todos que colaboraram direta ou indiretamente no desenvolvimento desse projeto.

'Só você pode me ouvir invocador,
que obra-prima devemos tocar hoje?'
Sona, personagem de League of
Legends

Resumo

Na atualidade a indústria de jogos possui um enorme público e grande expressividade econômica. Dentre todos os jogos, o mais jogado da atualidade e com maior visibilidade já há alguns anos é o League of Legends, jogo desenvolvido pela Riot Games. O League of Legends possui um cenário competitivo com grande expressividade, que conta com diversos profissionais e, assim como clubes esportivos, possui treinadores, técnicos, analistas e muitos outros profissionais. Considerando esse fato, desejou-se desenvolver um sistema capaz de analisar os oponentes durante o início da partida com base em seu personagem e seu estilo de jogo, e de forma simplificada executar a função de um analista. Com isso as capacidades de análise, discernimento e aplicação de estratégias do usuário tendem a se tornar melhores. O projeto se utilizou de conceitos de aprendizado de máquina ao longo de seu desenvolvimento, visando realizar previsões com o melhor resultado possível através da interpretação da inteligência artificial desenvolvida.

Palavras-chave: Riot Games. Analista. Aprendizado de máquina. Inteligência artificial.

Abstract

Currently, the gaming industry has a huge audience and great economic expressiveness. Among all games, the most played today and with greater visibility for some years is League of Legends, a game developed by Riot Games. League of Legends has a competitive scenario with great expressiveness, this scenario has several professionals, and as sport clubs, they have coaches, technicians, analysts and much more professionals. Considering this fact, it was desired to develop a system capable of analyzing opponents during the beginning of the game based on their character and their style of play, and in a simplified way to perform the function of an analyst. With this the user's analysis, discernment and application capabilities in strategies tend to become better. The project used machine learning concepts throughout its development, aiming to make predictions with the best possible result through the interpretation of the developed artificial intelligence.

Keywords: Riot Games. Analyst. Machine learning. Artificial intelligence.

Lista de Figuras

Figura 1 - Mapa do League of Legends	15
Figura 1 - Modelo básico de uma rede neural	19
Figura 2 - Modelo básico de uma célula de rede neural	20
Figura 3 - Fluxograma do projeto	22
Figura 4 - Esquema das tabelas	26
Figura 5 - Parte da tabela “ChallengersId”	27
Figura 6 - Parte da tabela “MatchId”	28
Figura 7 - Parte da tabela “Champions”	29
Figura 8 - Parte da tabela “Itens”	30
Figura 9 - Parte da tabela “Perks”	31
Figura 10 - Tabela “SummonerSpells” completa	
32	
Figura 11 - Parte da tabela “MatchNotFound”	33
Figura 12 - Parte da tabela “MatchDetails”	35
Figura 13 - Parte da tabela “MatchPlayerTimeline”	36
Figura 14 - Parte da tabela “MatchEventTimeline”	38
Figura 15 - Exemplo de problema encontrado no registro de partidas	39
Figura 16 - Design da interface	42

Glossário

Atributos	Características básicas dos personagens, como força, quantidade de vida e agilidade
Dragão/Arauto	Monstros neutros que nascem em tempos regulamentados de dificuldade superior e que trazem melhores benefícios
Experiência	Valor numérico que altera o nível do personagem
Feitiços de invocador	Habilidades independentes do personagem e que precisam ser ativadas para que seus efeitos sejam realizados, e que são selecionadas antes do início da partida
Habilidades	Poderes que podem ser utilizados durante o jogo e que variam de personagem para personagem
Habilidades adicionais	Habilidades independentes do personagem que não necessitam de ativação e que são selecionadas antes do início da partida
Mana	Energia necessária para utilização de habilidades do personagem
Minions	Inimigos básicos que são criados em grupos de 6 ou 7, a cada 30 segundos
Nível	Atributo numérico que define os atributos e quantidade de habilidades do personagem
Ouro	Moeda básica de compra de itens que é adquirido no jogo por tempo e por cada abate realizado
Sentinelas	Item carregado pelos jogadores, que pode ser posicionado no mapa criando visão ao redor da região onde é posicionada, e que fica invisível após ser posicionada

- Sentinelas reveladoras** Item carregado pelos jogadores e que pode ser posicionado no mapa criando visão ao redor da região onde é posicionada, e revela personagens invisíveis e sentinelas invisíveis
- Torres** Unidades imóveis responsáveis pela defesa de uma região e que pertencem a um dos dois times

Sumário

1. Introdução	13
1.1. Objetivo	17
1.2. Motivação	17
1.3. Justificativa	18
1.4. Organização do Trabalho	18
2. Aspectos Conceituais	19
3. Tecnologias Utilizadas	20
4. Metodologia do trabalho	21
5. Especificação de Requisitos do Sistema	24
6. Projeto	25
6.1. Aquisição dos Dados	25
6.2. Análise dos Dados	26
6.3. Problema durante a análise	39
6.4. Solução do problema da análise	39
6.5. Criação dos modelos	40
6.6. Interface	42
7. Teste e avaliação	43
8. Considerações Finais	44
8.1. Conclusões do Projeto de Formatura	44
8.2. Contribuições	44
8.3. Perspectivas de Continuidade	44

1. Introdução

Os videogames se originaram por volta dos anos 70 e rapidamente adquiriram o caráter competitivo, isso devido a grande popularidade do jogo de arcade Pong. As competições com pessoas de diferentes regiões começou com disputas colocando desempenhos medidos pela pontuação dos jogos em em máquinas de fliperamas locais, disputando localmente com desconhecidos. Posteriormente passaram a publicar o desempenho em revistas para competir em escalas ainda maiores, ampliando o alcance das disputas entre jogadores que nem precisam se conhecer para competirem.

Em 1980 foi criada a primeira competição entre jogadores oficial, ela foi realizada pela empresa criadora do videogame Atari e era uma competição do jogo Space Invaders, onde o campeonato atraiu em torno de 10000 jogadores.

Nos anos 90, as disputas de campeonatos passaram a se focar mais em jogos de combate, sendo os mais populares Street Fighter e Mortal Kombat. Os jogos foram se desenvolvendo e originaram novos gêneros de jogos, como os jogos de tiro em primeira pessoa (First Person Shooter ou FPS) e jogos multijogadores em uma mesma arena de combate (Massive Online Battle Arena ou MOBA) (LYNCH; ANDREW, 2016) . Sendo que esses três últimos citados se consagraram nos cenários competitivos mais buscados tanto por jogadores quanto pela audiência, originando um ramo de trabalho completamente novo no qual os jogadores se tornam assalariados associados a organizações, tendo acesso a equipes técnicas, nutricionistas, psicólogos e tudo que for necessário para mantê-los com seu máximo desempenho.

Essas organizações recebem financiamento de investimentos de outras empresas que buscam a divulgação de suas marcas nos uniformes e apresentações dos jogadores. Algumas pesquisas indicam os e-sports como uma indústria de um bilhão de dólares, e sua popularidade atualmente alcança 3 bilhões de pessoas, considerando apenas Índia e China, sendo sua visibilidade muito maior devido a exibição através de streams (exibições de vídeo em tempo real através de sites como Twitch.tv e YouTube, contribuindo também para a divulgação de marcas patrocinadoras das equipes ou até mesmo patrocínios pessoais dos jogadores (RAPOZA; KENNETH, 2019).

A comissão técnica dos times de e-sports conta com diversos profissionais de diferentes áreas, desde profissionais da área da saúde como fisioterapeutas, nutricionistas e psicólogos, assim como profissionais que trabalham diretamente com o jogo, entre eles treinadores, técnicos e analistas.

O foco desse projeto será sobre os analistas. Os analistas são responsáveis por estudar as habilidades dos oponentes, verificando suas características de jogo, hábitos e padrões, muitas vezes acompanhados dos psicólogos da equipe para conseguir avaliar ainda mais profundamente o jogador e conseguir prever suas ações quando os times se enfrentarem. Também realizam análises mais gerais do time, identificando seu tipo de abordagem no jogo, por exemplo, se os jogadores costumam se juntar ou não, quando costumam se agrupar e até mesmo que tipo de objetivos visam quando se agrupam. Alguns profissionais dessa área são reconhecidos como tendo um grande impacto dentro da equipe, como é o caso da INTZ, uma equipe brasileira de League of Legends.

League of Legends (LoL) é um jogo eletrônico do gênero *Multiplayer Online Battle Arena* (MOBA), traduzido para arena de batalha de multijogadores online, gratuito desenvolvido pela Riot Games. No caso do League of Legends os jogadores são colocados em times de 5 e enfrentam outros jogadores em mapa pré-definido com 3 caminhos para confrontar diretamente os jogadores, indo de uma base até a outra, sendo um superior, um intermediário e um inferior. Além desses 3 caminhos existem uma selva que fica entre os caminhos e na posição intermediária do trecho há um rio. Os jogadores tradicionalmente se distribuem de forma que 1 vai pela rota superior, 1 pela rota do meio, 2 pela rota inferior e um dos jogadores se mantém na selva. O objetivo do jogo é alcançar a vitória ao destruir o núcleo central na base de seu inimigo, chamado nexus, passando por suas estruturas e enfrentando todos os oponentes que entrarem em seu caminho. O mapa pode ser visualizado na imagem abaixo:



Figura 1 - Mapa do League of Legends

Fonte:

<https://gaming.stackexchange.com/questions/143069/new-season-4-map-image>

A Riot Games é uma empresa fundada em 2006, seu primeiro jogo lançado foi o próprio League of Legends no ano de 2009. Posteriormente lançaram outros jogos derivados do mesmo universo, conhecido como Runeterra, sendo eles Teamfight Tactics (TFT), lançado em 2019 e Legends of Runeterra (LoR), lançado oficialmente em 2020, sendo o primeiro um jogo do gênero Auto Chess, e o segundo um jogo do gênero Collectible Card Game (CCG). Além dos jogos já lançados, o Teamfight Tactics teve sua versão mobile lançada em 2020 e outros jogos foram ditos como em produção através de um vídeo comemorativo que foi exibido na comemoração de 10 anos de League of Legends, que ocorreu dia 15 de Outubro de 2019. Dentre os jogos ainda em produção, alguns dos mais aguardados pelo público são League of Legends: Wild Rift (uma versão de League of Legends

para smartphones), um jogo do gênero First Person Shooter (FPS) chamado Valorant que será lançado em 2020 e um jogo de luta entre dois personagens conhecido atualmente como Project L.

O universo de League of Legends conhecido como Runeterra é extremamente rico e diversificado, tendo uma grande quantidade de roteiristas para seu desenvolvimento e até mesmo quadrinhos criados em parceria com a DC comics, usando uma teoria de multiverso com diversas versões dos mesmos personagens que justificam as mudanças e diferentes estilos para as skins (formas alternativas que são adquiridas por dinheiro real ou através de marcos alcançados no jogo). O universo principal, Runeterra, é um universo com um mapa bem estruturado contendo diversas regiões e reinos como Noxus e Demacia, cada um dos reinos tem características próprias que se refletem nos personagens que vivem ou nascem nos mesmos. A diversidade dos reinos vai até o nível ideológico e também ao estilo de vida de seus habitantes, como por exemplo, Noxus é um império que prega a força por qualquer meio, físico ou mágico, enquanto Demacia busca a justiça e considera a magia uma espécie de maldição, devido aos efeitos colaterais de uma guerra mágica ocorrida no passado desse universo. Uma opção alternativa brasileira para conhecer mais a fundo a história e mitologia do jogo é utilizar um canal do YouTube chamado Universo Lúdico, que sintetiza todos os textos em vídeos específicos ou genéricos e especula com base em analogias e referências reais, tanto mitologias do mundo real quanto fatos da própria realidade.

As histórias são tão bem escritas e detalhadas que até mesmo os personagens jogáveis possuem as suas, sendo que parte dessas histórias não são definitivas para o universo, são estruturadas como contos disseminados pelas pessoas comuns do universo acerca de seus feitos e personalidades, isso inclui até mesmo mitos e lendas, além de crenças e mitologias sobre eles.

Os analistas precisam fazer diversas análises ao observar o jogo, como identificar posicionamento de sentinelas (itens necessários para visão do mapa do jogo) e tempo de colocação das mesmas, identificar se o adversário costuma colocar ou não pressão em cada combinação de personagens possíveis. Essa dedução permite identificar se é viável outros membros da equipe ajudarem ou não e se cria condições de invasão de território inimigo nas proximidades daquela região ou não. Também fazem parte de suas responsabilidades a necessidade de identificar pontos a serem explorados do adversário e conhecimento dos

personagens, sendo que cada personagem possui habilidades diferentes, 46 ao todo, e há personagens que possuem vantagens ou desvantagens em relação a outros. O jogador necessita do conhecimento de como explorar essas vantagens e como controlar as desvantagens, para poder retirar o máximo potencial de seu personagem, além de aproveitar o máximo do combate no qual se encontra.

1.1. Objetivo

O projeto tem como objetivo desenvolver um programa que analisa o adversário do usuário e fornece dados que o auxiliem a derrotá-lo. A ideia é desenvolver as capacidades do jogador e aumentar suas chances de vitória através da adaptabilidade e de recomendações sobre como seria possível derrotar seu oponente a cada jogo, melhorando sua eficiência a cada partida e tornando-o mais consciente sobre seu oponente e sobre si mesmo.

1.2. Motivação

Esse projeto busca a melhoria da qualidade dos jogadores, trazendo uma melhor qualidade ao cenário competitivo, trazendo mais visibilidade e valor ao jogo em si. Além disso é uma proposta comercial relevante tanto adaptando-se para fornecer os serviços dentro do cenário competitivo, quanto para fornecer os serviços aos jogadores amadores.

1.3. Justificativa

A motivação para o desenvolvimento do projeto vem do fornecimento de dados oferecido pela Riot Games, empresa desenvolvedora do jogo, de algumas páginas que realizam análises estatísticas e da apresentação de estatísticas dentro do próprio jogo, para que o jogador consiga identificar de maneira relevante suas características para que consiga melhorar seu desempenho.

Em relação a projetos de outros jogos em situações similares, a inspiração foi o projeto OpenAI Five, desenvolvido pela OpenAI com uma proposta diferente. A ideia do projeto foi criar uma inteligência artificial capaz de jogar Dota, outro jogo do gênero MOBA que é similar ao League of Legends. No auge do treinamento da inteligência artificial criada, ela conseguiu obter a vitória contra um time contando com cinco jogadores profissionais. Com isso a vitória foi alcançada de maneira cada vez mais rápida ao longo dos jogos em que competiram.

1.4. Organização do Trabalho

Esse trabalho está organizado em oito capítulos, sendo o primeiro apenas uma introdução ao projeto, contendo os objetivos, as motivações e justificativas .

O segundo capítulo refere-se aos conceitos utilizados para a criação do projeto e desenvolvimento do programa.

O terceiro capítulo apresenta as tecnologias empregadas para o desenvolvimento do projeto, bem como os programas utilizados para sua criação, bem como a linguagem de programação utilizada.

O quarto capítulo contém explicações sobre a metodologia empregada no trabalho e uma descrição de sua aplicação.

O quinto capítulo apresenta as especificações de requisitos do sistema, como as necessidades de instalação e de definições esperadas para a execução do projeto.

O sexto capítulo descreve uma apresentação dos resultados obtidos ao longo do projeto e implementação do sistema.

O sétimo capítulo demonstra como foram realizados os testes e avaliações pertinentes ao projeto.

E o último capítulo encerra com as considerações finais alcançadas durante a elaboração do projeto, bem como possibilidades de melhorias do mesmo.

2. Aspectos Conceituais

Foram utilizados conceitos de criação e manutenção de tabela de modelo relacional para o armazenamento dos dados, para a manutenção e análise foram utilizados conceitos de utilização e manipulação de *data frames*, já para o desenvolvimento dos modelos foram utilizados conceitos de aprendizado de máquina, mais especificamente de rede neural.

A rede neural funciona utilizando-se de diversas camadas de células que produzem os resultados através da elaboração de equações definindo o peso de cada variável de entrada entrada dentro do sistema, uma imagem que pode exemplificar de maneira simples esse fato pode ser vista abaixo.

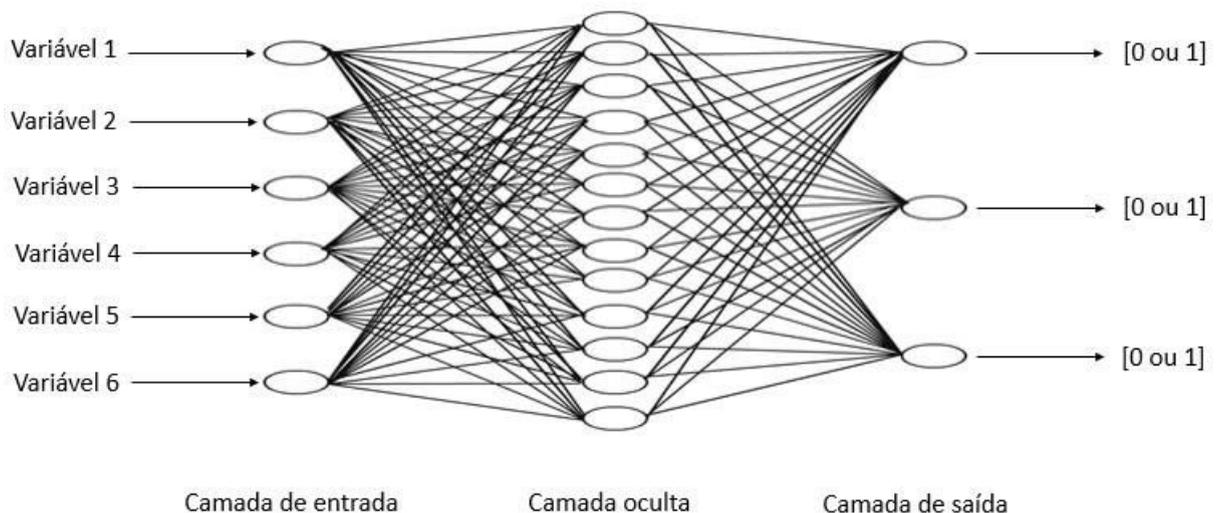


Figura 2 - Modelo básico de uma rede neural

Fonte:

https://www.researchgate.net/profile/Antonio_Penedo2/publication/324555121

Cada célula funciona recebendo as diversas entradas, identificando o peso de cada uma dentre as variáveis que são recebidas e somando todos esse valores,

depois disso passa por uma função multiplicativa conhecida como função de ativação, finalmente gerando um valor de saída que será utilizado pela próxima camada ou que será o resultado final alcançado pelo sistema¹, a estrutura de uma célula de rede neural pode ser vista abaixo.

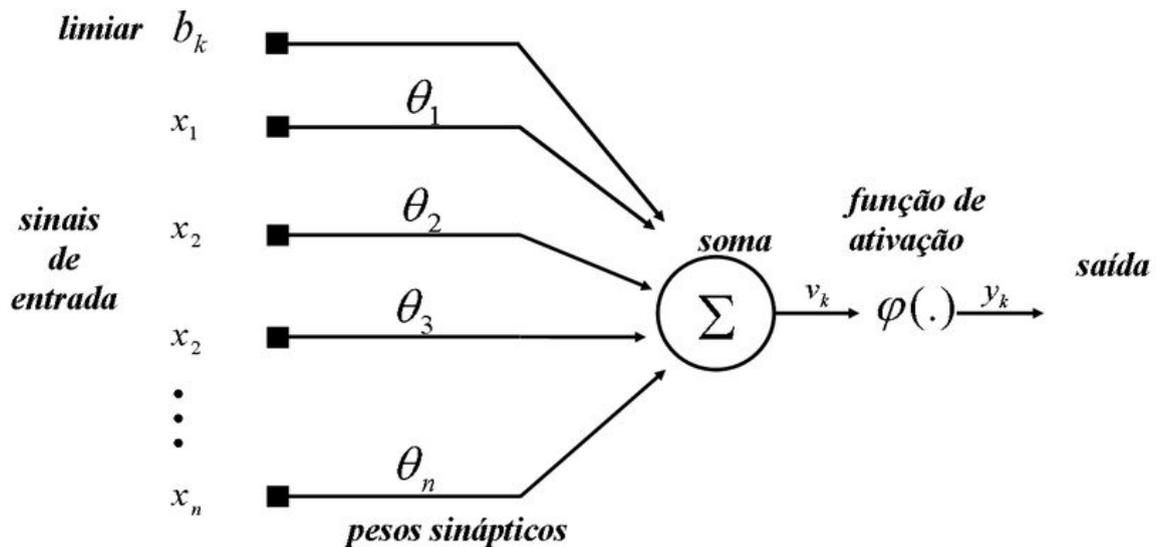


Figura 3 - Modelo básico de uma célula de rede neural

Fonte: https://www.researchgate.net/profile/Helaine_Furtado/publication/264735169

Para a interface foi utilizado um framework baseado em React Native, empregando de maneira eficiente e com um design simples, chamativo e funcional uma página de utilização para o usuário do programa.

3. Tecnologias Utilizadas

As tecnologias utilizadas durante o desenvolvimento do projeto foram Python 3.7.3 na versão de 64 bits, Sublime Text 3 para realizar o desenvolvimento dos programas na linguagem Python que seriam executados diretamente no prompt de comando do computador, Anaconda Navigator para utilização do Jupyter Notebook

¹ Informação fornecida por Paulo Salvatore durante o curso de Introdução ao Deep Learning com Python e Keras da Samsung Ocean, São Paulo, 2019

para facilitar as análises e elaboração dos modelos, e SQLiteStudio utilizado para a construção, visualização e manutenção da base de dados.

Para a criação da interface foi utilizado o Visual Studio Code, simplificando o desenvolvimento da página onde o jogador pode realizar a pesquisa para visualizar os resultados do modelo.

Todos esses programas foram selecionados tanto pela simplicidade de uso quanto pela familiaridade da equipe de desenvolvimento em utilizá-los que já vinha de experiências prévias relacionadas ao programa. No caso da opção por Python, entre as linguagens de utilização e aplicação de ciência de dados mais comuns (Python e R), ele é o mais eficiente para realizar a integração no conjunto do sistema.

Em relação às bibliotecas em python necessárias para a elaboração do projeto, foram utilizadas as bibliotecas requests 2.22.0, sqlite3, numpy 1.16.4, pandas 1.0.1, scikit-learn 0.23.2, tensorflow 2.3.1, matplotlib 3.1.3, seaborn 0.10.0, keras 2.3.1 e h5py 2.10.0.

4. Metodologia do trabalho

O projeto foi desenvolvido seguindo diversas etapas, sendo elas:

1. Elaboração
2. Aquisição de dados
3. Análise dos dados e correlação com experiências de profissionais
4. Estruturação de uma interface
5. Adaptação e utilização dos dados para criação do modelo para personagens
6. Teste do modelo para personagens
7. Agrupamento dos jogadores por estilo de jogo
8. Estruturação do modelo para jogadores
9. Teste do modelo para jogador
10. Aplicação da API de busca em tempo real associada a interface
11. Conexão da interface com os modelos e elaboração do relatório final

12.Eventualmente pode se apresentar uma etapa final que seria a atualização da base

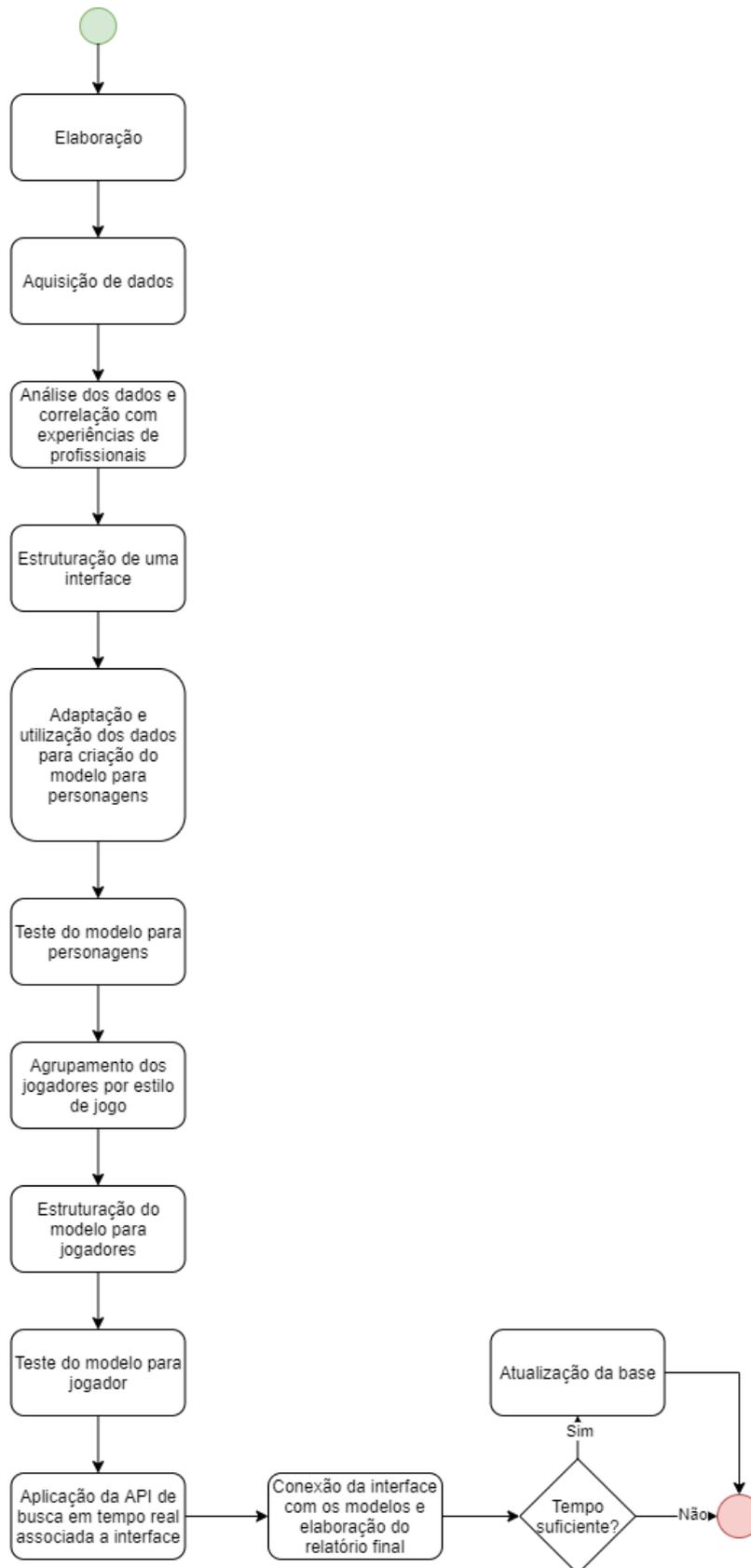


Figura 4 - Fluxograma do projeto

A etapa de aquisição de dados foi feita através da linguagem Python realizando requisições através das APIs fornecidas pela Riot Games, desenvolvedora do jogo.

A análise dos dados foi feita consultando profissionais do ramo e aplicando análises gráficas e de correlação entre os dados, definindo sua relevância para a obtenção dos resultados do projeto.

A estruturação da interface foi feita de maneira simples para que sua utilização seja intuitiva ao jogador, considerando um design simples e atrativo, além de buscando opinião de jogadores através de uma pesquisa feita com alguns indivíduos.

A adaptação e utilização dos dados para a criação do modelo de personagens foram feitos com tabelas contendo informações dos personagens e informações mais genéricas das partidas, obtendo uma análise que pode identificar as vantagens de cada confronto do jogo.

O teste do modelo de personagens foi feito validando-se os resultados obtidos tanto em termos absolutos, sendo definida uma “quantidade” de vantagem que existiria em cada caso, como o objetivo do projeto, que vem da definição de se existe uma vantagem ou uma desvantagem durante o combate que pretende-se analisar.

O agrupamento dos jogadores por estilo de jogo é uma maneira do programa interpretar de maneira mais eficiente as vantagens de estilos de jogo e os momentos do jogo onde essas vantagens podem ser exploradas, além de analisar as desvantagens para definir como evitá-las de maneira eficiente.

A estruturação do modelo de jogadores parte do conceito de definição do estilo de jogo e de estatísticas de combates entre os estilos, buscando como resultado se existem ou não vantagens no combate dos estilos e quando elas ocorrem.

O teste do modelo de jogador foi feito validando-se os resultados obtidos tanto em termos absolutos, sendo definida uma “quantidade” de vantagem que existiria em cada caso, como o objetivo do projeto, que vem da definição de se existe uma vantagem ou uma desvantagem entre os estilos de jogo durante o combate que pretende-se analisar.

A aplicação da API de busca em tempo real será utilizada para que o jogador insira apenas suas informações e através dela seja possível adquirir dados do seu oponente, sendo esses os dados necessários para a execução dos testes dentro do modelo.

A conexão da interface com os modelos é para que o usuário interagindo com a interface possa alcançar os resultados do modelo sem ter que executar códigos de programação diretamente.

A elaboração do relatório deve se originar a partir do resultado do modelo e considerando-se informações dos dados de entrada e de informações preexistentes jogadores e personagens.

Adicionalmente, caso seja possível realizar a atualização dos dados devido a uma grande atualização do jogo que ocorreu no dia 11 de Novembro seria necessário atualizar as bases para que a precisão seja mantida considerando a atualização do jogo.

5. Especificação de Requisitos do Sistema

Os requisitos de sistema vem da necessidade de executar os programas que foram citados anteriormente, para que seja possível a utilização da biblioteca tensorflow do Python é necessário um sistema operacional de 64 bits.

Já em questões opcionais, seria necessário utilizar um processador eficiente para acelerar os tempos de execução dos programas, uma placa de vídeo para melhorar a eficiência do processamento dos modelos, que preferencialmente seja da Nvidia, para que seja possível utilizar a CUDA Toolkit com máxima eficiência, melhorando a utilização da biblioteca tensorflow.

6. Projeto

6.1. Aquisição dos Dados

A aquisição dos dados utilizados no programa foi feita através de diversas APIs fornecidas pela desenvolvedora do jogo, Riot Games. Essas APIs foram acessadas através de códigos programados em Python e para realizar o acesso é necessário solicitar uma chave de acesso em uma página destinada a desenvolvedores interessados no jogo, página essa fornecida pela própria Riot. As APIs utilizadas para a aquisição dos dados foram:

- LEAGUE-EXP-V4: Utilizada para aquisição do identificador do jogadores de nível desafiante (classificação mais alta do jogo) por região do mundo.
- SUMMONER-V4 por identificador do jogador: Utilizado para adquirir o identificador da conta de cada jogador por região.
- MATCH-V4 por identificador de conta: Utilizado para adquirir uma lista de partidas na qual o jogador participou.
- CHAMPION: Utilizada para adquirir informações dos personagens do jogo, bem como identificar qual o personagem através da atribuição de um valor numérico.
- ITEMS: Utilizada para adquirir informações relacionadas aos itens utilizados pelos personagens, para identificá-los em casos onde os mesmos seriam mais relevantes para a predição da vitória.
- PERKS: Utilizado para adquirir informações sobre melhorias adicionais aos personagens selecionadas anteriormente ao início da partida.
- SPELLS: Utilizada para adquirir informações dos feitiços selecionados anteriormente ao jogo e do qual o jogador fará utilização
- MATCH-V4 por identificar de partida: Utilizada para adquirir detalhes das partidas identificando informações de cada jogador, bem como seus desempenhos durante a partida.
- MATCH-V4 linhas do tempo por identificar de partida: Utilizada para adquirir informações de eventos que ocorreram durante o jogo.

6.2. Análise dos Dados

A análise dos dados foi feita inicialmente com apoio de profissionais do cenário competitivo de League of Legends, oferecendo informações sobre dados e atributos relacionados a cada partida.

Para utilização simplificada dos dados necessários foi construída uma base de dados contendo diversas tabelas em linguagem SQL, utilizando o programa SQLiteStudio para fazer a visualização das mesmas. O programa foi selecionado devido ao conhecimento de utilização prévia e da gratuidade do sistema.

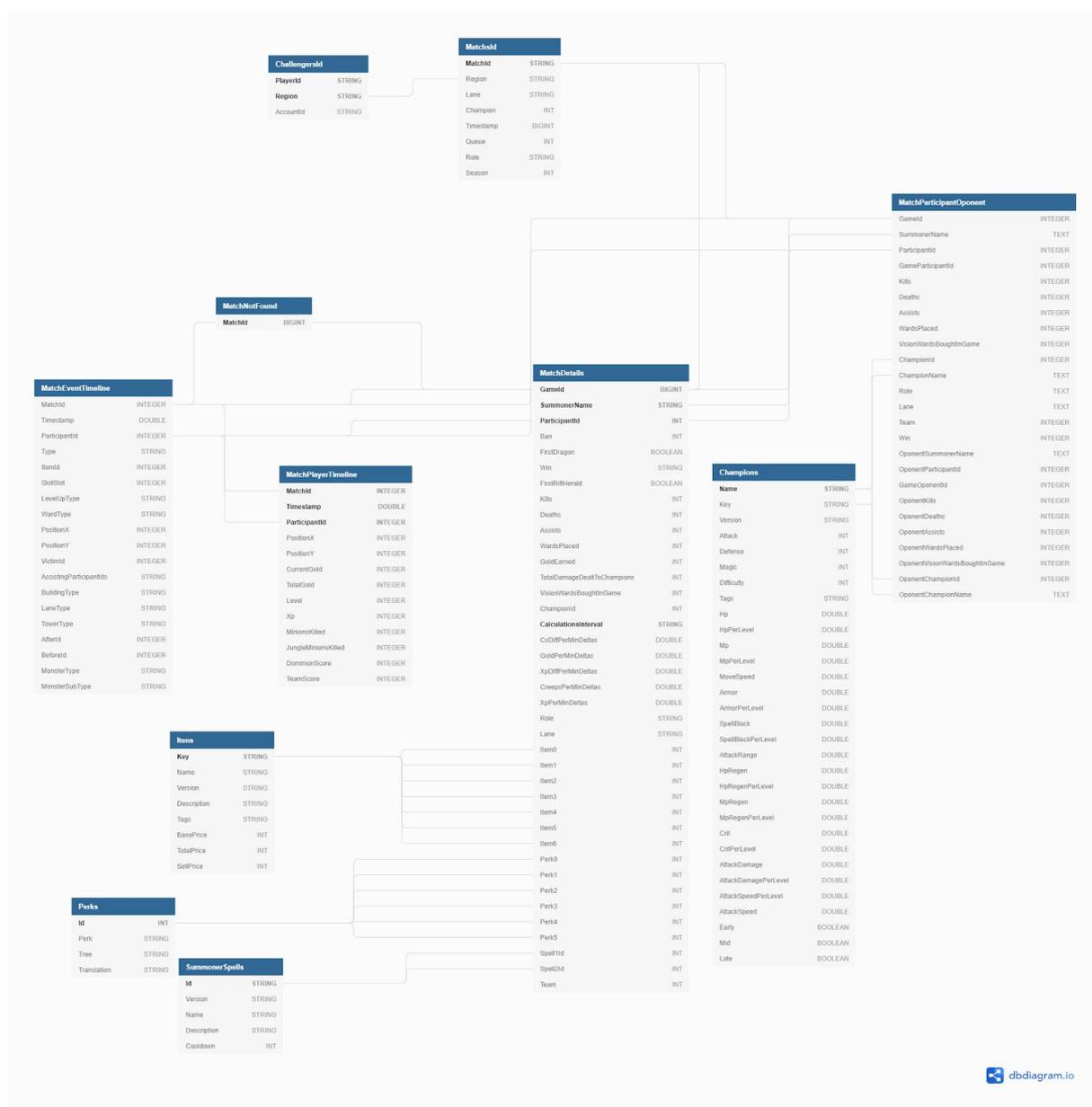


Figura 5 - Esquema das tabelas

A primeira tabela criada foi batizada “ChallangersId”, contendo informações do código de identificação do jogador, a região na qual ele joga e o código de identificação da sua conta, conforme imagem abaixo:

PlayerId	Region	Accountid
-1nUCoJ0M4U5L7HuDY64HXqUzS8VmhWak0M700bAnVIA1s	euw1	2ILUIMcB3gf2vgO-_LCgcsq75vnu4AAS-QwPykTu1rY8ppCo
-39bsTJFpi7wdHmX4N84zH4wZreU72krbPq4shibfnznSeE	na1	zoH_n94z70Brmcymcwa6XKe8ug2b6YwgJadeUB9uyn-6-g
-4q0lBnL5qkQm00aYfkKXcdlycpzO3x10F5HegYpKlIM9o9s	na1	l6yPjEwwAke4wFBA-6bsJwJXF1JemTHzdyDzAVVEQ6uXxA
-5c7-zDNFATNinjZb136YH6r0l62Yc5fEDrR8-qSuSg8k	euw1	vwBaHxLj4-Dvc0AmYs5EL5WWh9JfTyQTGbdm1ZDnbLgt3vxCKfHEeXRu
-7zdvQFPGR1755bRl8UCtKldwY-5L9yRLNGN8eNGjcQuhQ	la2	3R2RYqHpnAoUWKtvR5G2U_xMq6aApNmqsAApXUdAESyO_YE
-9dglWlwoHuCP-T9k7jy6dAmLqLkQvImCujahOwzhv2D-k	la1	yO31HtsPGFkUvFbQ4_GcrHKEP7nGg68TLE62471EFY8Ujs-BNRTNmGd3
-9y_StZFCanipdOdN5tQKD9vk4RmSXMPNART0lrZy7-3jk	la2	lmZKS8j-n2lL_5NB-zixoXtiJBkf-E_C0vf5TtW8MLduUeO
-BWHScleWYBCLU4A0Mh8MYmcrH9jr_n_sHxn7Q0VvbmCg	la1	2BYSNHBBhjwlt3VQM09oheTTSclfoa33MIUppqWcb6R5c5ek
-D9-A7cQQRNKDFixZWdrpXduw9rFjP6LrLryAwQrXfnXU0Fg8	euw1	SH9ySTVjJAZLORLmw_5uKZ7UHGBVubEdeHHzgE8x5Mi8uug
-PsmefP6GUhthPlwiRnNt_Vi4Y2lIHEOJ1h7-gfStcAPYAg	euw1	xcI7Wfqmi34z6NuJ94A2oAGF65KIZIsqAzEx3qEmg5dvDg
-SGuwomW-Fr7qu6z830QC3-778MdrktMBGNnrm1W9VvWSA	la2	hk3ZJ-WoEISEPmQqN0SY1P6p5EVdNU2bv0iOLTP-CAFFtBw
-T50ZkYtpp59vW876_oY97ndYrC-wff4j3eNd0QnT8_SLhpl	na1	vq89nF2Vkl1GGcVPKU_KU5ZE3WEb2zVxjN5wuDG5gBTRluBgPtlbseb
-THL5E7uU5ubAtUxjBba73uu5MqD8l4xD_7YNxyWEwq	ru	l4B61cT-Mfhg69U2jOb4-QESQKk7A4hquxRU1RzdvLt44U
-Wg6ysCTLWE9lo1L6L8nlWto_UqYn73i4Ourm60Rnm3yMvY	kr	3bso0Hml_TanrWmCyomyCVC_A0FNz22TjE5UEmvhGcJll6w
-XtRapaNgC11GITBBR6VT91mAyufXDfSfS18OaqqiFLCM	eun1	YOTg8w0NMQRx-ZardoEFxiX9LwZEIm6bQZiYsU5PLdBwwc
-ZFjx-F2HQF6G6GKQ_30h13vrX4-y_tl8IKJxsORxoSMI	euw1	zYkvaT3fKhQiwKqIs78HHIB0YU8PPRkIE2ks1vRQF3PZQ
-yOOpwU8PakU01B16bKbSfndFNPPYDfhsHXvtSfaXqTA	la1	KbIYL6x-mft8KKpG_1f9lyY7AXaUM474Lfa76q0RAXxKnO
-d4g4eBwWJ916FAX355At1RIMDUDICLuj9qm68gHV_wYtHs	eun1	CZGYFwNyxspfsjHsrjGAAR03Tktay-_00dYdtGtYQIKd0Q
-dK_Jw9vhHRjmv2nrd1ZNYbbUvChT9UP9ai6eKvkU0b4M	na1	PjnaAkNBCuSjtuH7tWCSJgm3c0QSD75CWaeOEnR_tYIsF30
-fvZY8Sualo4SyfuSBxFCde9lbWNd1NUDQZPfq-y5tpalo	euw1	BM3KuSnXFbIXbzPICeRU--onmca0arM-X71YAoxYylKUNcc
-gd8455FOIPIDFBQn5qG9wRIGcd6zh32ks1HmRgI4	la2	mhOAnKBaZWUtA4bYFcZuJLY04m7gsAKb832aWfp5rA
-h7fbWE4fD875N93fXzw4qbbGrMhRi9zlxTqePVMF281QsqU	la1	uLOVRXkDpd8q7LzZe19F7k_kgFYdREzX67Jgoq4ewyPusZE
-iptbjW5cMVJqcrLhhHmwEoBiElxXr95yfxA3ZrpmCcbdyY	eun1	QakwYZg1RuP3doq2zAe6NFYrLEpPMAKeCoKcpKms0SwT8w
-kQ7j-2ngml_4JTzCj-LzoEj9Ucix1XnKb2vsMaeuNrgA	tr1	HpybCkg2b0hQYEYTDMcgPVhPhMK5DstXbajngBbJ6l1Lkpk
-nYAym81NDmRjcwUxM5fcT5BhaSiK_YYWZBPCUOI2Z-p4TU	eun1	JsYvl19aybYPzylHBBV69ZDiDnEhPxLPM6HJRA7Isin6Qg
-sHADbBFHSj-ouCqGwyJf86iNwWx4L1_wktno8AN4Um8cYk	na1	gcHn0ldG300J3D0tdpQGEbB2rmNLnRAu7q3p61mCREnizo
01YwdGSAMDRZPMu395GHBJzDjKfEJOJDD3-vACw0WgFww	tr1	lu8NG58Xy7SA3YM-lie-5vtN83zQ4ge8Wa8pApPz0A6lGU
04HEL8lv_C7M92ph355Mh5tR-kKALzdvdJ7VhqK5a8U7tq22n	na1	pPtQR-h5sClo6tlw7-D1wu5EnW9smeHfB036iKpQ6sUecRMRrj7YGFsd
05buUr-1t6W0y8_Tr5g4GZgmKSp1JsoHh7YmfTPunBUUkQl	tr1	_GUoLbL8nvvbG3o6yqvFsU28r9IPpPF9ToKv3pBrva752eQ
05xaTCcxptz1ZV2bVgYKvGQdGOKRi6rPTHVrwnBPpFjg	la1	3LThmLX7BMHRD2T_yeGn8pd2fSwVYYSM4xoMp5AhULvQcH4
06sDfAf-wXZjmlq7YgKoSuu3Y9aCD20DR52zCj6zGT_n4l4	eun1	A1YqO8NgdM8uMBTWYS7DLWonK119nuiFmlwPPmBDXIONbw
08THYDhcCkfpCnwfgst14DlIditadwM6_L6HeWmWIZDG	ru	xnXsUq8MXOfGga6hOi07ah07upjbnls0Lfgok7-xvQf_EHU
09K1J1idL-R7AZcFYhaBCAtEoPXAkovFOYxufm5A1gwWg	tr1	wuZ_nqui8a9eweYyoBPyWeirQrLwVzhAgEQS4NXyVtdWRg
09TKS4NZOWr3pdH4B3lKXoy3un2MPx5n1Efp7feMBE8d4Q	eun1	KfIBzuNwk-ubHft0ztQtY3WE1tCKIMkUSILLTXgolNcKzQ
0BBYKhZlWQPQa8Xd2w4_dZ16CIBQNGm7Kbgetde-EERcJll	jp1	B1ReOSaqiTyrofKOK5nq6hQshqPt2co9RfRyjhjxabE3hXV6evv2d4R
0CBemr2duUmkjot9iWbavP2MBChgpxy7VR6Lu6cibVHthsU	kr	CSHVb9k5hfkeGpcJum3lphE3Ke5S7Je6-ZSsdF6Ghacayo
0D_U-XLbEj-Pjijys_7_3izsrFCeMllsfky2oIWDKW	br1	P_z3TkvWPIF_nXJK2RmO4DuJfAdVWpSbk0cdFlsQsrQ
0IDEEbdnJ6gxAmuRm8Fem94EKrUgtrsAPa9JER6mA09XMg	la1	0xouolbr0Nfr4IczubAtuW0wX7iOqvYyszWMLSMxcX5A0y0
0IIV4xuCue6PzrQr-Tx3FYKajwYrCXz6Z0b4rtBSE3C8yA	na1	zaNpi1qVNPz1MvvsqLwS7MW0VpiSj1xZlH1nvtzPTNltpg
0KHtZYTpNVUcc0ngPtDhhLs4pkSGOEF1boxMJOTgEu59QzWd	na1	O5WQQMj2a6duSAC0ZM0xPH5rAqz9kUe3ERQsohaHsu03AGidU98wF4X
0Vcu-gsOm_0AzowEDZxcelgjmQ93SXOdQbia0QtCXLaxTQ	na1	wPq9ZlWxmKb8DuPfh8zdfelTc3MSvohiB5phg5qL3dnaLQ
0XT4sm9ytLmkXbl4x5xI4inilHzmRFqEKOe7U5SwN_xSCP5	euw1	qNurP1w3uhNJ-LiWfWEBIDsluilbltFpjr3PqvrKrb3KA
0aTln5VaFbu--wbBwA-Zhe2l-ygahMcuY62LuA6gNhEGS4	kr	wYKDiMcp946VSkB8mX2oGiQkoL3usiAhOBjNUw3T3EU5rYY

Figura 6 - Parte da tabela “ChallangersId”

A segunda tabela criada foi “MatchsId”, que contém informações sobre o código identificador da partida, a região de atuação do jogador, a posição onde esse jogador estava jogando, o personagem que esse jogador utilizou, a data e hora de início da partida definido através de uma timestamp, o modo de jogo que o jogador estava jogando, o papel do jogador durante a partida e a temporada na qual ele estava jogando. Uma imagem dessa tabela pode ser vista abaixo:

MatchId	Region	Lane	Champion	Timestamp	Queue	Role	Season
4447267980	euw1	NONE	429	1583799567634	420	DUO	13
4447163610	euw1	BOTTOM	523	1583791640698	420	DUO_CARRY	13
4446913123	euw1	BOTTOM	81	1583781758406	420	DUO_CARRY	13
4445389204	euw1	NONE	134	1583696343209	420	DUO	13
4445265748	euw1	NONE	145	1583691803090	420	DUO	13
4445199660	euw1	BOTTOM	523	1583689771855	420	DUO_CARRY	13
4445121814	euw1	BOTTOM	21	1583683936457	420	DUO_CARRY	13
4445051834	euw1	BOTTOM	21	1583680776658	420	DUO_CARRY	13
4444943611	euw1	BOTTOM	81	1583677807145	420	DUO_CARRY	13
4439974303	euw1	BOTTOM	145	1583368057688	420	DUO_CARRY	13
4437616749	euw1	BOTTOM	21	1583243864179	420	DUO_CARRY	13
4436018835	euw1	BOTTOM	21	1583147302252	420	SOLO	13
4436015810	euw1	BOTTOM	45	1583144343077	420	DUO_CARRY	13
4434925741	euw1	BOTTOM	523	1583070730535	420	DUO_CARRY	13
4434374812	euw1	BOTTOM	81	1583017484138	420	DUO_CARRY	13
4434328051	euw1	BOTTOM	523	1583015265599	420	DUO_CARRY	13
4434313344	euw1	JUNGLE	30	1583013319424	420	NONE	13
4434136352	euw1	TOP	516	1583007569092	420	SOLO	13
4433959937	euw1	BOTTOM	134	1583005327581	420	DUO_CARRY	13
4432598196	euw1	BOTTOM	498	1582928355490	420	DUO_CARRY	13
4432538800	euw1	NONE	235	1582925256814	420	DUO_SUPPORT	13
4432460861	euw1	BOTTOM	235	1582919029299	420	DUO_SUPPORT	13
4432342788	euw1	BOTTOM	89	1582915520658	420	DUO_SUPPORT	13
4432274505	euw1	MID	81	1582912265735	420	DUO_CARRY	13
4430599356	euw1	NONE	21	1582822311639	420	DUO	13
4430651664	euw1	BOTTOM	81	1582819795240	420	SOLO	13
4430136767	euw1	BOTTOM	523	1582769021315	420	DUO_CARRY	13
4428650660	euw1	BOTTOM	7	1582679882780	420	DUO_CARRY	13
4428588528	euw1	BOTTOM	235	1582677918050	420	DUO_SUPPORT	13
4428516402	euw1	BOTTOM	21	1582671149989	420	DUO_CARRY	13
4428470773	euw1	BOTTOM	22	1582668701419	420	DUO_CARRY	13
4427438722	euw1	TOP	8	1582639646409	420	SOLO	13
4426766288	euw1	MID	523	1582576215956	420	DUO	13
4426648919	euw1	BOTTOM	81	1582571156906	420	DUO_CARRY	13
4426642024	euw1	TOP	266	1582568893713	420	SOLO	13
4426110438	euw1	BOTTOM	21	1582547257343	420	DUO_CARRY	13
4426175533	euw1	BOTTOM	523	1582545073445	420	DUO_CARRY	13
4425794639	euw1	BOTTOM	21	1582498157545	420	DUO_CARRY	13
4425424611	euw1	BOTTOM	235	1582482697550	420	DUO	13
4425343166	euw1	BOTTOM	523	1582479169643	420	DUO_CARRY	13
4425174585	euw1	BOTTOM	81	1582476549217	420	DUO_CARRY	13
4424764748	euw1	BOTTOM	498	1582470311416	420	DUO_CARRY	13
4424620907	euw1	BOTTOM	21	1582465352294	420	DUO_CARRY	13

Figura 7 - Parte da tabela "MatchId"

A próxima tabela chamada "Champions" foi feita para manter informações dos personagens do jogo. Os dados catalogados nessa tabela são nome do personagem, uma chave de identificação do personagem, a versão do jogo do qual são os dados desse personagem, o quão bom ele é como atacante, o quão bom ele é como defensor, o quão bom ele é como mago, o quão difícil é jogar com ele,

classificações definidas pelos desenvolvedores, a quantidade de vida básica do personagem, a quantidade de vida que o personagem adquire por nível, a quantidade de mana base do personagem, a quantidade de mana adquirida por nível do personagem, a velocidade que o personagem se move, a armadura física base, a armadura física por nível, a resistência a magia base, a resistência a magia adquirida por nível, o alcance de seu ataque básico, a capacidade de regeneração de vida base, a capacidade de regeneração de vida adquirida por nível, a quantidade de regeneração de mana base, a capacidade de regeneração de mana adquirida por nível, a capacidade de obter acertos críticos em seus ataques, a capacidade de obter acertos críticos em seus ataques adquiridas por nível, o dano de ataque base, o dano de ataque adquirido por nível, a velocidade de ataque adquirida por nível e a velocidade de ataque base.

Posteriormente, com ajuda de profissionais do cenário competitivo de League of Legends, foram acrescentados dados relacionados ao tempo onde a atuação de cada personagem é mais forte e/ou útil durante a partida. Uma parcela dessa tabela pode ser encontrada abaixo:

Name	Key	Version	Attack	Defense	Magic	Difficulty	Tags	Hp	HpPerLeve	Mp	MpPerLeve	Mov
Aatrox	266	10.5.1	8	4	3	4	Fighter,Tank	580	90	0	0	
Ahri	103	10.5.1	3	4	8	5	Mage,Assassin	526	92	418	25	
Akali	84	10.5.1	5	3	8	7	Assassin	575	95	200	0	
Alistar	12	10.5.1	6	9	5	7	Tank,Support	600	106	350	40	
Amumu	32	10.5.1	2	6	8	3	Tank,Mage	613.12	84	287.2	40	
Anivia	34	10.5.1	1	4	10	10	Mage,Support	480	82	495	25	
Annie	1	10.5.1	2	3	10	6	Mage	524	88	418	25	
Aphelios	523	10.5.1	6	2	1	10	Marksman	500	86	348	42	
Ashe	22	10.5.1	7	3	2	4	Marksman,Support	539	85	280	32	
AurelionSol	136	10.5.1	2	3	8	7	Mage	575	92	350	50	
Azir	268	10.5.1	6	3	8	9	Mage,Marksman	552	92	480	21	
Bard	432	10.5.1	4	4	5	9	Support,Mage	575	89	350	50	
Blitzcrank	53	10.5.1	4	8	5	4	Tank,Fighter	582.6	95	267.2	40	
Brand	63	10.5.1	2	2	9	4	Mage	519.68	88	469	21	
Braum	201	10.5.1	3	9	4	3	Support,Tank	540	98	310.6	45	
Caitlyn	51	10.5.1	8	2	2	6	Marksman	481	91	313.7	35	
Camille	164	10.5.1	8	6	3	4	Fighter,Tank	575.6	85	338.8	32	
Cassiopeia	69	10.5.1	2	3	9	10	Mage	575	90	350	40	
Chogath	31	10.5.1	3	7	7	5	Tank,Mage	574.4	80	272.2	40	
Corki	42	10.5.1	8	3	6	6	Marksman	518	87	350.16	34	
Darius	122	10.5.1	9	5	1	2	Fighter,Tank	582.24	100	263	37.5	
Diana	131	10.5.1	7	6	8	4	Fighter,Mage	594	95	375	25	
Draven	119	10.5.1	9	3	1	8	Marksman	574	88	360.56	39	
DrMundo	36	10.5.1	5	7	6	5	Fighter,Tank	582.52	89	0	0	
Ekko	245	10.5.1	5	3	7	8	Assassin,Fighter	585	85	280	50	
Elise	60	10.5.1	6	5	7	9	Mage,Fighter	534	93	324	50	
Evelynn	28	10.5.1	4	2	7	10	Assassin,Mage	572	84	315.6	42	
Ezreal	81	10.5.1	7	2	6	7	Marksman,Mage	500	86	375	50	
Fiddlesticks	9	10.5.1	2	3	9	9	Mage,Support	524.4	80	500	28	
Fiora	114	10.5.1	10	4	2	3	Fighter,Assassin	550	85	300	40	
Fizz	105	10.5.1	6	4	7	6	Assassin,Fighter	570	98	317.2	37	
Galio	3	10.5.1	1	10	6	5	Tank,Mage	562	112	500	40	
Gangplank	41	10.5.1	7	6	4	9	Fighter	540	82	282	40	
Garen	86	10.5.1	7	7	1	5	Fighter,Tank	620	84	0	0	
Gnar	150	10.5.1	6	5	5	8	Fighter,Tank	510	65	100	0	
Gragas	79	10.5.1	4	7	6	5	Fighter,Mage	600	95	400	47	
Graves	104	10.5.1	8	5	3	3	Marksman	555	92	325	40	
Hecarim	120	10.5.1	8	6	4	6	Fighter,Tank	580	90	277.2	40	
Heimerdinger	74	10.5.1	2	6	8	8	Mage,Support	488	87	385	20	
Illaoi	420	10.5.1	8	6	3	4	Fighter,Tank	585.6	95	300	40	
Irelia	39	10.5.1	7	4	5	5	Fighter,Assassin	580	95	350	30	
Ivern	427	10.5.1	3	5	7	7	Support,Mage	585	95	450	60	

Figura 8 - Parte da tabela “Champions”

A próxima tabela chamada “Itens” contém informações sobre os itens do jogo, sendo essas informações o código do item, o nome do item, a versão do jogo no qual esses dados funcionam, uma descrição sobre as funcionalidades do item, uma classificação da utilidade do item, o preço base do item, o preço total do item (que difere do preço base em caso de itens que são construídos a partir de outros itens), o preço de venda pelo qual o jogador pode “devolver” o item na loja. Uma parte dessa tabela pode ser vista abaixo:

Key	Name	Version	Description	Tags
1001	Boots of Speed	10.5.1	<groupLimit>Limited to 1 pair of boots.</groupLimit> <unique>UNIQUE Passive - Enhanced Move...	Boots
1004	Faerie Charm	10.5.1	<stats> <mana> +25% Base Mana Regen </mana> </stats>	Mana
1006	Rejuvenation Bead	10.5.1	<stats> +50% Base Health Regen </stats>	Heatl
1011	Giant's Belt	10.5.1	<stats> +380 Health</stats>	Heatl
1018	Cloak of Agility	10.5.1	<stats> +20% Critical Strike Chance</stats>	Critic
1026	Blasting Wand	10.5.1	<stats> +40 Ability Power</stats>	SpellE
1027	Sapphire Crystal	10.5.1	<stats> <mana> +250 Mana</mana> </stats>	Mana
1028	Ruby Crystal	10.5.1	<stats> +150 Health</stats>	Heatl
1029	Cloth Armor	10.5.1	<stats> +15 Armor</stats>	Armo
1031	Chain Vest	10.5.1	<stats> +40 Armor</stats>	Armo
1033	Null-Magic Mantle	10.5.1	<stats> +25 Magic Resist</stats>	SpellE
1036	Long Sword	10.5.1	<stats> +10 Attack Damage</stats>	Dama
1037	Pickaxe	10.5.1	<stats> +25 Attack Damage</stats>	Dama
1038	B. F. Sword	10.5.1	<stats> +40 Attack Damage</stats>	Dama
1039	Hunter's Talisman	10.5.1	 <unique>UNIQUE Passive - Tooth:</unique> Damaging a monster with a spell or attack steals 30 He	LifeSt
1041	Hunter's Machete	10.5.1	<stats> +10% Life Steal vs. Monsters</stats> <unique>UNIQUE Passive - Nail:</unique> Basic att	LifeSt
1042	Dagger	10.5.1	<stats> +12% Attack Speed</stats>	AttacL
1043	Recurve Bow	10.5.1	<stats> +25% Attack Speed</stats> <unique>UNIQUE Passive:</unique> Basic attacks deal an addit	AttacL
1052	Amplifying Tome	10.5.1	<stats> +20 Ability Power</stats>	SpellE
1053	Vampiric Scepter	10.5.1	<stats> +15 Attack Damage +10% Life Steal</stats>	Dama
1054	Doran's Shield	10.5.1	<stats> +80 Health</stats> <passive> Passive: </passive> Restores 6 Health every 5 seconds. <	Heatl
1055	Doran's Blade	10.5.1	<stats> +8 Attack Damage +80 Health +3% Life Steal</stats>	Dama
1056	Doran's Ring	10.5.1	<stats> +60 Health +15 Ability Power <mana> +5 Mana per 5 seconds</mana> </stats> <p...	Heatl
1057	Negatron Cloak	10.5.1	<stats> +40 Magic Resist</stats>	SpellE
1058	Needlessly Large Rod	10.5.1	<stats> +60 Ability Power</stats>	SpellE
1082	Dark Seal	10.5.1	<stats> +10 Ability Power +25% Increased Healing from Potions <mana> +100 Mana</mana> </stats>...	Heatl
1083	Cull	10.5.1	<stats> +7 Attack Damage +3 Life on Hit</stats> <unique>UNIQUE Passive:</unique> Killing a	Dama
1400	Enchantment: Warrior	10.5.1	<stats> +60 Attack Damage +10% Cooldown Reduction</stats> <groupLimit>Limited to 1 Gold...	
1401	Enchantment: Cinderhulk	10.5.1	<stats> +300 Health +15% Bonus Health</stats> <unique>UNIQUE Passive - Immolate:</unique>	
1402	Enchantment: Runic Echoes	10.5.1	<stats> +80 Ability Power +10% Cooldown Reduction <mana> +300 Mana</mana> </stats> ...	
1412	Enchantment: Warrior	10.5.1	<stats> +60 Attack Damage +10% Cooldown Reduction</stats> <groupLimit>Limited to 1 Gold...	
1413	Enchantment: Cinderhulk	10.5.1	<stats> +300 Health +15% Bonus Health</stats> <unique>UNIQUE Passive - Immolate:</unique>	
1414	Enchantment: Runic Echoes	10.5.1	<stats> +80 Ability Power +10% Cooldown Reduction <mana> +300 Mana</mana> </stats> ...	
1416	Enchantment: Bloodrazor	10.5.1	<stats> +50% Attack Speed</stats> <unique>UNIQUE Passive:</unique> Basic attacks deal 4% of th	
1419	Enchantment: Bloodrazor	10.5.1	<stats> +50% Attack Speed</stats> <unique>UNIQUE Passive:</unique> Basic attacks deal 4% of th	
2003	Health Potion	10.5.1	<groupLimit>Limited to 5 at one time. Limited to 1 type of Healing Potion.</groupLimit> <cons	Const
2006	Showdown Health Potion	10.5.1	<groupLimit>Limited to 2 at one time. Limited to 1 type of Healing Potion.</groupLimit> <cons	Const
2009	Total Biscuit of Rejuvenation	10.5.1	<consumable> Click to Consume:</consumable> Restores 80 Health and 50 Mana over 10 seconds.	
2010	Total Biscuit of Everlasting Will	10.5.1	<consumable> Click to Consume:</consumable> Restores 10% of missing Health and Mana over 5 seconds.	
2015	Kircheis Shard	10.5.1	<stats> +15% Attack Speed</stats> <unique>UNIQUE Passive - Energized:</unique> Moving and atta	AttacL
2031	Refillable Potion	10.5.1	<groupLimit>Limited to 1 type of Healing Potion.</groupLimit> <active>UNIQUE Active:</active>	Heatl
2033	Corrupting Potion	10.5.1	<groupLimit>Limited to 1 type of Healing Potion.</groupLimit> <active>UNIQUE Active:</active>	Activ

Figura 9 - Parte da tabela “Itens”

A próxima tabela, denominada “Perks” contém informações das habilidades adicionais selecionadas pelo jogador antes do início da partida, sendo esses dados o código de identificação da habilidade, o nome da Habilidade e qual a árvore de habilidades na qual se encontra. Uma parcela dessa tabela pode ser vista abaixo:

Id	Perk	Tree
8136	Zombie Ward	Domination
8126	Cheap Shot	Domination
8128	Dark Harvest	Domination
8135	Ravenous Hunter	Domination
8112	Electrocute	Domination
8124	Predator	Domination
8139	Taste Of Blood	Domination
8106	Ultimate Hunter	Domination
8120	Ghost Poro	Domination
8134	Ingenious Hunter	Domination
8105	Relentless Hunter	Domination
9923	Hail Of Blades	Domination
8138	Eyeball Collection	Domination
8143	Sudden Impact	Domination
8351	Glacial Augment	Inspiration
8359	Kleptomancy	Inspiration
8321	Futures Market	Inspiration
8410	Approach Velocity	Inspiration
8306	Hextech Flashtraption	Inspiration
8360	Unsealed Spellbook	Inspiration
8316	Minion Dematerializer	Inspiration
8339	Celestial Body	Inspiration
8352	Time Warp Tonic	Inspiration
8358	Master Key	Inspiration
8313	Perfect Timing	Inspiration
8304	Magical Footwear	Inspiration
8345	Biscuit Delivery	Inspiration
8347	Cosmic Insight	Inspiration
8473	Bone Plating	Resolve
8465	Guardian	Resolve
8446	Demolish	Resolve
8429	Conditioning	Resolve
8453	Revitalize	Resolve
8439	Aftershock	Resolve
8242	Unflinching	Resolve
8463	Font Of Life	Resolve
8472	Chrysalis	Resolve
8435	Mirror Shell	Resolve
8437	Grasp Of The Undying	Resolve
8401	Shield Bash	Resolve
8430	Iron Skin	Resolve
8451	Overgrowth	Resolve
8444	Second Wind	Resolve

Figura 10 - Parte da tabela “Perks”

Outra tabela, denominada “SummonerSpells” contém dados sobre os feitiços de invocador, selecionados antes do início de partida pelo jogador, e que são utilizados como habilidades adicionais. Essa tabela contém as informações do código de identificação da habilidade, a versão do jogo no qual foi extraído esse dado, o nome do feitiço, uma descrição do que o feitiço faz e o tempo de recarga do feitiço em segundos. Abaixo pode-se ver a tabela:

Id	Version	Name	Description	Cooldown
SummonerBarrier	10.5.1	Barrier	Shields your champion from 115-455 damage (depending on champion level) for 2 seconds.	180
SummonerBoost	10.5.1	Cleanse	Removes all disables (excluding suppression and airborne) and summoner spell debuffs affecting your	210
SummonerDot	10.5.1	Ignite	Ignites target enemy champion, dealing 70-410 true damage (depending on champion level) over 5 secon	180
SummonerExhaust	10.5.1	Exhaust	Exhausts target enemy champion, reducing their Movement Speed by 30%, and their damage dealt by 40%	210
SummonerFlash	10.5.1	Flash	Teleports your champion a short distance toward your cursor's location.	300
SummonerHaste	10.5.1	Ghost	Your champion gains increased Movement Speed and can move through units for 10 seconds. Grants a max	180
SummonerHeal	10.5.1	Heal	Restores 90-345 Health (depending on champion level) and grants 30% Movement Speed for 1 second to y	240
SummonerMana	10.5.1	Clarity	Restores 50% of your champion's maximum Mana. Also restores allies for 25% of their maximum Mana.	240
SummonerPororecall	10.5.1	To the King!	Quickly travel to the Poro King's side.	10
SummonerPorotrow	10.5.1	Poro Toss	Toss a Poro at your enemies. If it hits, you can quickly travel to your target as a follow up.	20
SummonerSmite	10.5.1	Smite	Deals 390-1000 true damage (depending on champion level) to target epic, large, or medium monster or	15
SummonerSnowURFSnowball_Mark	10.5.1	Mark	Throw a snowball in a straight line at your enemies. If it hits an enemy, they become marked, granti	80
SummonerSnowball	10.5.1	Mark	Throw a snowball in a straight line at your enemies. If it hits an enemy, they become marked, granti	80
SummonerTeleport	10.5.1	Teleport	After channeling for 4 seconds, teleports your champion to target allied structure, minion, or ward.	360

Figura 11 - Tabela “SummonerSpells” completa

As próximas duas tabelas são relacionadas aos dados de cada partida, sendo elas “MatchDetails”, “MatchPlayerTimeline” e “MatchEventTimeline”. Ao preenchê-las, devido a desistência prévia de alguns jogadores, algumas partidas terminam muito antes do previsto, o que resulta na falta de detalhes da partida e na falta de marcações temporais dos eventos que ocorrem nas partidas, para esse caso uma tabela chamada “MatchNotFound” foi criada para receber o código de identificação de partidas onde não foram encontrados dados de detalhes e de eventos temporais. Abaixo uma parte dessa tabela:

MatchId
3308809629
3308843913
3308865235
3308886453
3308915336
3308930424
3309409663
3309635616
3309638179
3309641590
3309750551
3309825675
3309826900
3309902942
3309910505
3309962831
3310415998
3310429816
3310563433
3310576712
3310607771
3310625136
3310630538
3310651168
3310711270
3310766505
3310789462
3310946119
3310992433
3311026329
3312828365
3312859716
3312872802
3319395505
3319408180
3319410036
3319453150
3319469572
3319476638
3319484893
3319488039
3319499776
3319529416

Figura 12 - Parte da tabela “MatchNotFound”

A tabela “MatchDetails” contém informações detalhadas das partidas, tanto genéricas quanto temporais, para a maioria das partidas possui mais de uma linha destinada à cada jogador na mesma partida, os dados armazenados são o código identificador da partida, o usuário do jogador, um número identificador do jogador

durante a partida, o personagem que foi escolhido para ser bloqueado durante a partida pelo jogador, um identificador que marca o time que matou o primeiro dragão que nasceu no jogo, o time que obteve a vitória na partida, o time que matou o primeiro arauto do vale da partida, a quantidade de vezes que o jogador matou na partida, a quantidade de vezes que o jogador morreu na partida, a quantidade de vezes que o jogador contribui em uma morte na partida, a quantidade de sentinelas colocadas pelo participante, a quantidade de ouro adquirida pelo jogador durante a partida, o dano total causado pelo jogador a personagens inimigos durante a partida, a quantidade de sentinelas reveladoras compradas durante a partida, o código de identificação do personagem escolhido pelo jogador, o intervalo de tempo para o cálculo de algumas das informações que virão a partir de agora, a diferença na quantidade de minions (personagens genéricos que nascem em intervalos regulares para cada time e que se direcionam a cada uma das rotas) mortos pelo jogador em relação a seu oponente por minuto no intervalo de tempo marcado, a quantidade de ouro adquirido por minuto no intervalo de tempo marcado, a diferença da quantidade de experiência ganha por minuto pelo jogador em relação a seu oponente no intervalo de tempo marcado, a quantidade de minions mortos por minuto no intervalo de tempo marcado, a quantidade de experiência adquirida por minuto no intervalo de tempo registrado, o papel do jogador durante o jogo, a rota do jogador durante o jogo, o itens que se encontram do primeiro espaço ao sétimo espaço, as seis habilidade adicionais escolhidas, os dois feitiços de invocador escolhidos e o time no qual o jogador estava. Uma pequena parte dessa tabela pode ser vista abaixo:

GameId	SummonerName	Participant	Ban	FirstDragor	Win	FirstRiftHei	Kills	Deaths	Assists	W
4447267980	Hirunn	1	875	0	Fail	0	3	5	1	
4447267980	Targamas	2	497	0	Fail	0	0	8	2	
4447267980	Porkchetta	3	246	0	Fail	0	0	4	4	
4447267980	aniwataK	4	350	0	Fail	0	1	4	0	
4447267980	Dzu	5	523	0	Fail	0	3	3	1	
4447267980	what a disaster	6	157	1	Win	1	7	2	5	
4447267980	Do not Forgive	7	89	1	Win	1	3	1	1	
4447267980	K1CK iBo	8	107	1	Win	1	0	2	4	
4447267980	ζζζ ι ζζζ	9	111	1	Win	1	6	2	11	
4447267980	FNC xMatty	10	516	1	Win	1	8	0	6	
4447163610	Kegemo	1	111	1	Win	1	2	2	12	
4447163610	Kegemo	1	111	1	Win	1	2	2	12	
4447163610	QLS Paolocannone	2	235	1	Win	1	4	5	20	
4447163610	QLS Paolocannone	2	235	1	Win	1	4	5	20	
4447163610	crazyadri	3	80	1	Win	1	1	4	6	
4447163610	crazyadri	3	80	1	Win	1	1	4	6	
4447163610	ζζζ ι ζζζ	4	38	1	Win	1	22	5	4	
4447163610	ζζζ ι ζζζ	4	38	1	Win	1	22	5	4	
4447163610	ChangeName 2	5	24	1	Win	1	9	7	1	
4447163610	ChangeName 2	5	24	1	Win	1	9	7	1	
4447163610	FNC xMatty	6	516	0	Fail	0	3	8	6	
4447163610	FNC xMatty	6	516	0	Fail	0	3	8	6	
4447163610	bigbuddhaboy	7	235	0	Fail	0	2	9	9	
4447163610	bigbuddhaboy	7	235	0	Fail	0	2	9	9	
4447163610	maraqia05	8	111	0	Fail	0	2	10	4	
4447163610	maraqia05	8	111	0	Fail	0	2	10	4	
4447163610	pls nerf ekko	9	64	0	Fail	0	11	8	6	
4447163610	pls nerf ekko	9	64	0	Fail	0	11	8	6	
4447163610	RevertSystemToS8	10	421	0	Fail	0	4	3	9	
4447163610	RevertSystemToS8	10	421	0	Fail	0	4	3	9	
4446913123	xeonerr	1	350	1	Fail	0	3	10	6	
4446913123	xeonerr	1	350	1	Fail	0	3	10	6	
4446913123	xeonerr	1	350	1	Fail	0	3	10	6	
4446913123	AMP Royal	2	115	1	Fail	0	6	6	5	
4446913123	AMP Royal	2	115	1	Fail	0	6	6	5	
4446913123	AMP Royal	2	115	1	Fail	0	6	6	5	
4446913123	uAreEzLpkek	3	246	1	Fail	0	5	10	5	
4446913123	uAreEzLpkek	3	246	1	Fail	0	5	10	5	
4446913123	uAreEzLpkek	3	246	1	Fail	0	5	10	5	
4446913123	Bladorthin	4	89	1	Fail	0	2	6	12	
4446913123	Bladorthin	4	89	1	Fail	0	2	6	12	
4446913123	Bladorthin	4	89	1	Fail	0	2	6	12	

Figura 13 - Parte da tabela “MatchDetails”

Já a tabela “MatchPlayerTimeline” contém informações detalhadas sobre como estava cada jogador durante a partida, esses registros são feitos a cada um minuto de partida, tendo o registro inicial quando a partida começa, os dados armazenados nessa tabela são o código de identificação da partida, o registro temporal do jogo feito através de uma timestamp, o código de identificação do jogador durante a partida, a posição horizontal na qual o jogador se encontra, a posição vertical na qual o jogador se encontra, o ouro que o jogador possui no momento atual, a quantidade total de ouro que o jogador juntou até o momento

atual, o nível no qual o jogador se encontra no momento, a quantidade de experiência que o jogador possui no momento, a quantidade de minions (personagens genéricos que nascem em intervalos regulares para cada time e que se direcionam a cada uma das rotas) que o jogador matou até o momento e a quantidade dos monstros da selva que o jogador matou até o momento. Uma parte dessa tabela pode ser vista abaixo:

MatchId	Timestamp	Participant	PositionX	PositionY	CurrentGold	TotalGold	Level	Xp	MinionsKille	JungleMinic
211936142	0	2	560	581	500	500	1	0	0	0
211936142	0	5	560	361	500	500	1	0	0	0
211936142	0	1	351	293	500	500	1	0	0	0
211936142	0	4	221	471	500	500	1	0	0	0
211936142	0	3	311	649	500	500	1	0	0	0
211936142	0	10	14486	14511	500	500	1	0	0	0
211936142	0	9	14486	14291	500	500	1	0	0	0
211936142	0	8	14277	14223	500	500	1	0	0	0
211936142	0	7	14147	14401	500	500	1	0	0	0
211936142	0	6	14237	14579	500	500	1	0	0	0
211936142	60026	2	1706	12971	0	500	1	0	0	0
211936142	60026	5	8238	3529	0	500	1	0	0	0
211936142	60026	1	6616	6882	0	500	1	0	0	0
211936142	60026	4	7217	3108	0	500	1	0	0	0
211936142	60026	3	7943	5201	0	500	1	0	0	0
211936142	60026	10	6804	11392	0	500	1	0	0	0
211936142	60026	9	9733	6347	0	500	1	0	0	0
211936142	60026	8	8696	6642	0	500	1	0	0	0
211936142	60026	7	11016	5561	0	500	1	0	0	0
211936142	60026	6	9704	6279	0	500	1	0	0	0
211936142	120032	2	2028	12202	41	541	1	88	1	0
211936142	120032	5	8209	2495	210	710	2	475	0	6
211936142	120032	1	6830	7307	125	625	1	264	6	0
211936142	120032	4	11981	2268	97	597	1	134	4	0
211936142	120032	3	12209	2031	35	535	1	134	1	0
211936142	120032	10	2150	12325	62	562	1	176	2	0
211936142	120032	9	12407	6454	120	620	2	285	0	4
211936142	120032	8	7972	7683	97	597	1	206	4	0
211936142	120032	7	12748	2705	97	597	1	134	4	0
211936142	120032	6	11888	2248	35	535	1	134	1	0
211936142	180075	2	1005	10940	283	783	2	588	7	0
211936142	180075	5	1938	12064	503	1003	3	717	0	12
211936142	180075	1	2999	11510	409	909	3	886	15	0
211936142	180075	4	11814	2117	402	902	2	460	14	0
211936142	180075	3	12137	1887	206	706	2	460	3	0
211936142	180075	10	1881	12337	413	913	3	886	13	0
211936142	180075	9	7050	11368	429	929	2	615	0	12
211936142	180075	8	7567	8058	395	895	3	886	15	0
211936142	180075	7	12913	2804	477	977	2	520	15	0
211936142	180075	6	12806	2573	245	745	2	520	3	0
211936142	240084	2	2353	12069	1047	1547	4	1297	18	0
211936142	240084	5	3550	9634	903	1403	3	1039	0	16

Figura 14 - Parte da tabela "MatchPlayerTimeline"

A tabela “MatchEventTimeline” é responsável por armazenar dados de eventos que ocorreram durante a partida, para cada um desses dados as informações de entrada variam, sendo seus dados o código de identificação da partida, o registro temporal feito através de uma timestamp de quando ocorreu o evento, o código de identificação do jogador durante a partida que executou a ação, o tipo de evento que foi registrado, o item envolvido durante o evento (no caso de ser relacionado a um item), a habilidade envolvida durante a ocorrência do evento (se o evento envolve habilidade), o tipo de melhoria realizada durante o evento (caso haja melhoria), o tipo de sentinela posicionado (caso o evento envolve sentinela), a posição horizontal onde o evento ocorreu, a posição vertical onde o evento ocorreu, o código de identificação do jogador durante a partida que sofreu o evento (caso algum jogador seja alvo do evento), os códigos de identificação dos jogadores durante a partida que participaram do evento (caso haja), o tipo de construção afetada pelo evento (caso haja), a rota na qual ocorreu o evento (caso seja a destruição de uma torre), o tipo de torre que foi afetada pela posição em relação a base do time (caso haja envolvimento de torre), o código de identificação do item antes (caso envolvam troca ou devolução do item), o código de identificação do item depois (caso envolvam troca ou devolução do item), o tipo de monstro (se for relacionado a algum dos monstros do jogo) e o subtipo do monstro relacionado (caso haja relação com um monstro do jogo). Abaixo pode-se ver uma parte dessa tabela de registros:

MatchId	Timestamp	Participant	Type	ItemId	SkillSlot	LevelUpType	WardType	Po
211936142	5937	5	ITEM_PURCHASED	1041	NULL			
211936142	6003	9	ITEM_PURCHASED	3340	NULL			
211936142	6498	5	ITEM_PURCHASED	2031	NULL			
211936142	7158	5	ITEM_PURCHASED	3340	NULL			
211936142	8742	1	SKILL_LEVEL_UP	NULL		1 NORMAL		
211936142	8808	4	ITEM_PURCHASED	1055	NULL			
211936142	8973	4	ITEM_PURCHASED	2003	NULL			
211936142	9171	4	ITEM_PURCHASED	3340	NULL			
211936142	10161	8	ITEM_PURCHASED	1055	NULL			
211936142	10623	8	ITEM_PURCHASED	2003	NULL			
211936142	11118	8	ITEM_PURCHASED	3340	NULL			
211936142	12933	3	ITEM_PURCHASED	3302	NULL			
211936142	13263	3	ITEM_PURCHASED	2003	NULL			
211936142	13494	3	ITEM_PURCHASED	2003	NULL			
211936142	15606	2	ITEM_PURCHASED	3340	NULL			
211936142	15969	6	ITEM_PURCHASED	3302	NULL			
211936142	16101	2	ITEM_PURCHASED	2003	NULL			
211936142	16134	6	ITEM_PURCHASED	2003	NULL			
211936142	16299	6	ITEM_PURCHASED	2003	NULL			
211936142	16464	2	ITEM_PURCHASED	1054	NULL			
211936142	16530	6	ITEM_PURCHASED	3340	NULL			
211936142	17025	5	SKILL_LEVEL_UP	NULL		2 NORMAL		
211936142	27420	10	ITEM_PURCHASED	2033	NULL			
211936142	43492	4	SKILL_LEVEL_UP	NULL		2 NORMAL		
211936142	60498	9	SKILL_LEVEL_UP	NULL		3 NORMAL		
211936142	78451	8	SKILL_LEVEL_UP	NULL		1 NORMAL		
211936142	84127	3	SKILL_LEVEL_UP	NULL		1 NORMAL		
211936142	84457	7	SKILL_LEVEL_UP	NULL		1 NORMAL		
211936142	89044	6	SKILL_LEVEL_UP	NULL		3 NORMAL		
211936142	92279	10	SKILL_LEVEL_UP	NULL		1 NORMAL		
211936142	101321	5	SKILL_LEVEL_UP	NULL		1 NORMAL		
211936142	107327	9	SKILL_LEVEL_UP	NULL		2 NORMAL		
211936142	109736	2	SKILL_LEVEL_UP	NULL		1 NORMAL		
211936142	126567	1	SKILL_LEVEL_UP	NULL		2 NORMAL		
211936142	127458	3	WARD_PLACED	NULL	NULL		YELLOW_TRINKET	
211936142	130890	2	ITEM_DESTROYED	2003	NULL			
211936142	135477	8	SKILL_LEVEL_UP	NULL		3 NORMAL		
211936142	135609	10	SKILL_LEVEL_UP	NULL		3 NORMAL		
211936142	137985	3	ITEM_DESTROYED	2003	NULL			
211936142	138975	6	WARD_PLACED	NULL	NULL		YELLOW_TRINKET	
211936142	140031	3	SKILL_LEVEL_UP	NULL		3 NORMAL		
211936142	140361	6	ITEM_DESTROYED	2003	NULL			
211936142	141681	4	ITEM_DESTROYED	2003	NULL			

Figura 15 - Parte da tabela "MatchEventTimeline"

Para realizar a análise mais aprofundada, optou-se por utilizar o Jupyter Notebook da Anaconda, que por experiência prévia e familiaridade, além da simplicidade de uso, foi determinado como uma opção viável de análise e aplicação de testes do sistema.

6.3. Problema durante a análise

Alguns problemas encontrados durante a análise dos dados foram a má alocação dos jogadores durante a partida, a Riot Games, empresa desenvolvedora do jogo, classifica de maneira incorreta a posição de alguns jogadores durante as partidas, o que causa problemas de identificação de seu oponente direto. Isso pode ser visto abaixo:

meld	SummonerName	ParticipantId	Kills	Deaths	Assists	WardsPlaced	VisionWardsBoughtInGame	ChampionId	Name	Role	Lane	Team
3610	Kegemo	1	2	2	12	2	1	59	JarvanIV	NONE	JUNGLE	0
3610	QLS Paolocannone	2	4	5	20	22	5	40	Janna	DUO_SUPPORT	BOTTOM	0
3610	crazyadri	3	1	4	6	8	2	21	MissFortune	DUO_CARRY	BOTTOM	0
3610	ççç ççç	4	22	5	4	5	3	11	MasterYi	NONE	JUNGLE	0
3610	ChangeName 2	5	9	7	1	5	0	55	Katarina	SOLO	MIDDLE	0

Figura 16 - Exemplo de problema encontrado no registro de partidas

Nesse exemplo é possível observar que em um mesmo time há dois jogadores cuja posição é na selva (jungle), dois jogadores na rota inferior, sendo um o suporte e um o carregador e um jogador na rota do meio, quando o padrão seria um jogador na rota superior, um na selva, um no meio e dois na rota inferior sendo um suporte e um carregador, logo a uma incoerência que dificulta a identificação do oponente direto do jogador que estava na rota superior. Levando em conta que as partidas analisadas são de jogadores dos mais altos níveis ao redor do mundo, a única opção existente para explicar a incoerência em relação a distribuição dos personagens seria uma incoerência de registro.

6.4. Solução do problema da análise

A solução escolhida para evitar perder muito tempo e considerando o grande volume de dados foi o descarte das partidas com problemas de identificação de rotas dos oponentes, das 51.667 partidas que estavam registradas restaram 25.376 partidas, o que ainda mantém um excelente volume para análise e aplicação do modelo.

Com as tabelas devidamente filtradas, foi criada uma nova tabela com a finalidade de registrar os oponentes diretos, registrando dados de um jogador e de seu oponente direto na mesma linha da tabela, essa tabela recebeu o nome de “MatchParticipantOponent”.

6.5. Criação dos modelos

Para a geração dos modelos foi escolhido utilizar rede neural, de maneira a explorar com máxima eficiência o fato de possuir diversas entradas numéricas e como um desafio visando o máximo potencial de precisão.

Para a utilização dos dados eles foram passados para a linguagem python através da biblioteca “sqlite3”, utilizando-se da biblioteca “pandas” para converter os dados das tabelas em dataframes mais facilmente manipuláveis para controlar os dados e criar os modelos desejados.

Utilizando-se dos recursos fornecidos pela biblioteca “sklearn” foi possível separar a base de dados em duas, sendo uma delas de teste e uma delas de treino, a base de treino é utilizada para realizar o treinamento do modelo e a base de testes é utilizada para validar a “acurácia bruta” do modelo. A escolha do nome “acurácia bruta” foi selecionado para indicar corretamente o percentual de acertos do projeto considerando a precisão do percentual de vantagem que poderia ser obtido. A acurácia mais condizente com o projeto, por outro lado, não diz respeito a uma quantidade de vantagem de cada embate, mas sim a uma análise de se a vantagem existe ou se é uma desvantagem, sendo um valor binário obtido através do quantidade de vantagem determinada, aumentando assim a precisão do sistema dada a necessidade que o sistema possui.

Para a criação do modelo foi utilizada a biblioteca “keras”, utilizando-se o modelo de análise de perdas baseado em mínimos quadrados, para o fator de otimização foram testadas algumas possibilidades, sendo elas SGD, RMSprop, Adam, Adadelta, Adagrad, Adamax, Nadam, Ftrl. Já para definir a quantidade de células por camada foi utilizado de pesquisa de eficiência, observando-se que a eficiência aumenta ao utilizar-se de valores que sejam exponenciais de dois, resultando em um aumento da velocidade de processamento dos resultados e na

qualidade dos resultados alcançados. Para verificar a “acurácia bruta” foi adicionado entre as métricas o fator de *accuracy* para realizar o treinamento e foi analisado o mesmo fator na hora de realizar o teste do modelo gerado. Para selecionar as funções de ativação foram feitos vários testes utilizando diferentes funções, sendo essas funções as funções de ativação linear retificada, sigmoidal, softmax, softplus, softsign, tangencial hiperbólica, exponencial linear em escala, exponencial linear e exponencial. Cada função de ativação foi testada em diversas camadas e com alterações de camadas até obter os melhores resultados possíveis.

Os modelos criados foram dois: um partindo-se dos dados relacionados aos personagens que realizam o combate, considerando os dados obtidos da desenvolvedora e dados de experiências de profissionais no jogo, partindo-se disso criou-se um modelo capaz de definir qual personagem possuiria vantagem nos embates e como seria a maneira mais eficiente de explorar essa vantagem; o segundo modelo utiliza-se de dados característicos do jogadores, classificando-os em alguns grupos e identificando quais grupos possuem vantagens sobre os outros grupos e quais os momentos onde essas vantagens serão exploradas.

Para a elaboração do modelo de personagens foram utilizadas as diversas informações sobre os personagens do jogo, além de dados adicionados a partir da experiência de profissionais que trabalham com o jogo, combinados a dados estatísticos mais genéricos de cada partida, definindo assim as entradas do modelo.

Para a elaboração do modelo de jogador foram utilizados dados dos eventos criados por cada jogador ao longo de diversas partidas, categorizando-os temporalmente e utilizando-os para que fosse possível categorizar o estilo de jogo do jogador, além de utilizando-os como entradas do sistema, bem como a classificação obtida pelo grupo.

A base de aprendizado da estruturação e elaboração dessa maneira foram adquiridos no curso Data Science for All Latin America, ministrado pelo professor doutor Natesh Pillai, professor de estatística da universidade de Harvard.

6.6. Interface

A interface foi feita de maneira simples e intuitiva, contendo apenas o logo, duas caixas sendo uma para escrever o usuário e outra para selecionar a região do jogador, e um botão para começar analisar.



Figura 17 - Design da interface

A análise se dá inicialmente obtendo os dados sobre o oponente direto do jogador através da API de tempo real fornecida pela Riot Games, a empresa desenvolvedora do jogo, partindo-se dele, analisa-se a capacidade do personagem do jogador ter vantagem sobre o personagem de seu oponente, identificando e informando ao jogador quando ele deverá tentar obter vantagem, considerando o tempo de jogo dos personagens e sua capacidade de ficar a frente do personagem oponente. Também será analisado dentro dos estilos de jogo se o estilo de jogo do jogador que está utilizando o programa é eficiente contra o estilo do oponente, bem como alguns hábitos que podem ser explorados e hábitos com os quais o jogador deve se preocupar, além de informações mais gerais sobre o tempo onde o jogador costuma posicionar suas sentinelas e sua capacidade de utilização das mesmas, prevendo vantagens em combate individual ou dependências relacionadas a outros jogadores.

O resultado das análises será apresentado ao jogador em forma de um documento de texto que será utilizado informando como o jogador poderia ter sua vitória facilitada, com isso depende apenas do jogador utilizar as instruções dadas como um guia para poder obter a vitória de maneira facilitada, levando-o a melhorar sua taxa de vitórias sobre seu oponente.

7. Teste e avaliação

O teste e avaliação da qualidade do programa foram feitas de duas maneiras. Durante a execução do modelo, uma métrica definida durante o treinamento do modelo, e em uma avaliação da base de teste realizado após o treinamento. Ambas definem a acurácia de maneira absoluta, definindo como um valor percentual a “quantidade” de vantagem ou desvantagem existente no combate. Entretanto a meta do projeto é definir apenas se existe vantagem ou desvantagem. Considerando esse fator, optou-se por validar considerando as estatísticas dos resultados obtidos nas partidas analisadas validando-se de forma binária entre existir vantagem ou desvantagem para cada confronto.

Utilizando-se desses métodos alguns dos testes realizados durante o processo se provaram ineficazes, obtendo-se sempre resultados próximos de 50%, demonstrando que a utilização de alguns dados em conjunto tornaram imprevisível a definição de se haveria vantagem ou desvantagem durante os embates.

Testado a possibilidade de avaliar as vantagens de combate antes mesmo de finalizar a escolha, considerando os tempos onde o personagem desempenha melhor papel também gerou resultados insatisfatórios.

Por fim, o projeto obteve bons resultados ao longo dos seus testes, sendo os resultados finais obtidos em torno de 80% de precisão considerando a avaliação do programa relacionada aos resultados obtidos estatisticamente e de maneira ponderada, provando-se um projeto eficiente.

8. Considerações Finais

8.1. Conclusões do Projeto de Formatura

Através do projeto desenvolvido foi possível concluir que de maneira mais superficial é possível que cada jogador consiga receber orientações pessoais sem a necessidade de se tornar um profissional para isso, adquirindo mais conhecimento analítico de jogos e conseguindo conquistar mais facilmente a vitória em suas partidas.

8.2. Contribuições

O projeto contribui com o desenvolvimento de um método de análise de vantagens eficiente, geração de relatórios que beneficia os jogadores que o utilizarem para que possa obter vantagem durante a partida, e com uma metodologia de filtragem dos dados de maneira relevante para o desenvolvimento do projeto.

8.3. Perspectivas de Continuidade

Para dar continuidade ao projeto seria possível melhorar o desempenho das análises, expandir para replicar outras funções encontradas no cenário profissional, como treinadores ou técnicos de equipes, solucionar de forma mais eficiente o problema de posições que é observado nos dados recebidos através da API da Riot Games, desenvolvedora do jogo, e criar um sistema adaptável para considerar as versões do jogo, conseguindo alterar o impacto de algumas variáveis do modelo ao longo do tempo.

Replicar a funções de treinadores e técnicos seria avaliar não mais a relação do jogador com o oponente, mas sim os resultados do jogador partida após partida, para determinar quais adaptações e melhorias seriam possíveis para que o jogador estaria seu potencial ao máximo.

Para solucionar o problema de posições recebidos nos dados seria possível fazer uma aplicação adicionando as estatísticas das variações de posições de cada personagem, de maneira a identificar o personagem mais provável a estar deslocado de sua posição.

Para criar um sistema atualizável por versões do jogo primeiramente seria necessário determinar se considera as pequenas atualizações semanais relevantes ou apenas a grande atualização que ocorre anualmente, após isso deveria utilizar um gerador de base demarcando os pontos de variação das versões para que o modelo reconheça a diferença nos pesos de cada partida analisada ao longo do tempo, sendo as mais recentes mais relevantes.

Referências

DATA SCIENCE FOR ALL LATIN AMERICA, 1 de Fevereiro de 2020, São Paulo.

Data Science. Disponível em: <<https://train.ds4a.com/auth/login>>. Acesso em: 7 de Dezembro de 2020

INTRODUÇÃO AO DEEP LEARNING COM PYTHON E KERAS, 6 de Junho de 2019, São Paulo. **Deep Learning.**

League of Legends: Universo. Disponível em: <https://universe.leagueoflegends.com/pt_BR/>. Acesso em: 7 de Dezembro de 2020

Lynch, Andrew . **Tracing the 70-year history of video games becoming eSports.**

Fox Sports, 6 de maio de 2016, Disponível em:

<<https://www.foxsports.com/stories/other/tracing-the-70-year-history-of-video-games-becoming-esports>>. Acesso em: 7 de Dezembro de 2020

OpenAI Five. Disponível em: <<https://openai.com/projects/five/>>. Acesso em: 7 de Dezembro de 2020

OpenAI Five blog. Disponível em: <<https://openai.com/blog/openai-five/>>. Acesso em: 7 de Dezembro de 2020

Rapoza, Kenneth. **Global Esports Popularity Give Gamer Companies Reason To**

Be Bullish. Forbes, 29 de maio de 2019. Disponível em:

<<https://www.forbes.com/sites/kenrapoza/2019/05/29/global-esports-popularity-give-gamer-companies-reason-to-be-bullish/#1fc2d9891bde>>. Acesso em: 7 de Dezembro de 2020

Riot Games Developer Portal. Disponível em: <<https://developer.riotgames.com>>.

Acesso em: 7 de Dezembro de 2020