

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

LEVI SPINA VIEIRA

PREDIÇÃO DE VITÓRIA NO LEAGUE OF LEGENDS DOIS VIZINHOS

DOIS VIZINHOS

2022

LEVI SPINA VIEIRA

PREDIÇÃO DE VITÓRIA NO LEAGUE OF LEGENDS DOIS VIZINHOS

**PREDICTION OF VICTORY IN THE LEAGUE OF LEGENDS TWO
NEIGHBORS**

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia de Software do Curso de Bacharelado em Engenharia de Software da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Rodolfo Adamshuk Silva

Coorientador: Prof. Dr. Yuri Kaszubowski
Lopes

DOIS VIZINHOS

2022



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

LEVI SPINA VIEIRA

PREDIÇÃO DE VITÓRIA NO LEAGUE OF LEGENDS DOIS VIZINHOS

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do
título de Bacharel em Engenharia de Software
do Curso de Bacharelado em Engenharia de
Software da Universidade Tecnológica Federal
do Paraná.

Data de aprovação: 25/novembro/2022

Rodolfo Adamshuk Silva
doutorado
Universidade Tecnológica Federal do Paraná

Francisco Carlos Monteiro Souza
doutorado
Universidade Tecnológica Federal do Paraná

Alinne Cristinne Corrêa Souza
doutorado
Universidade Tecnológica Federal do Paraná

DOIS VIZINHOS
2022

RESUMO

O mercado de e-sport vem crescendo em todo o mundo, trazendo patrocínio de empresas de diversos ramos da economia. Atrelado a isso, mais pessoas com pouco ou nenhum contato com jogos eletrônicos estão acompanhando suas transmissões. Com isso, um grande problema dos novos expectadores é entender como o jogo é jogado, visto que a dinâmica do jogo é complexa e os expectadores que nunca jogaram podem não entender o sistema de pontuação. Pensando em melhorar a experiência de todos os expectadores durante a transmissão, este trabalho propõe o desenvolvimento de uma aplicação que utiliza técnicas de reconhecimento de padrões de imagem para prever qual equipe tem a maior probabilidade de vitória na partida. A aplicação utiliza um modelo de predição previamente definido para fazer a predição. Esse modelo foi treinado a partir de uma base de dados gerada por meio da extração de dados de vídeos de partidas já realizadas. Como resultados, obteve-se uma aplicação que calcula a probabilidade de vitória dos times partir dos dados extraídos de uma partida do jogo de League of Legends ocorrendo em tempo real. Um estudo de caso foi realizado para avaliar a eficácia da aplicação em realizar a predição de vitória dos times e os resultados mostram que a aplicação consegue, em sua grande maioria, prever corretamente os times ganhadores, com uma taxa de eficácia entre 80% e 95%.

Palavras-chave: predição de vitória; e-sport; reconhecimento de padrão.

ABSTRACT

The e-sport market has been growing worldwide, bringing sponsorship from companies from different branches of the economy. Linked to this, more people with little or no contact with electronic games are following their broadcasts. Thereby, a big problem for new viewers is understanding how the game is played since its dynamic is complex and inexperienced viewers may not understand the game scoring system. To improve viewers experience, this work proposes the development of an application that uses image pattern recognition techniques to predict which team has the highest probability of winning the match. The application uses a previously defined prediction model to make the prediction. The model was trained using a database generated by the data extracted from videos of previous matches. As a result, an application was obtained that calculates the probability of winning for each team using the data extracted from a real-time match of the League of Legends. A case study was proposed to evaluate the effectiveness of the application in predicting winning teams, and the results show that an application can, for the most part, correctly predict the winning times with an efficiency rate between 80% and 95 %.

Keywords: victory prediction; e-sport; pattern recognition.

LISTA DE FIGURAS

Figura 1 – Mapa League of Legends.	13
Figura 2 – Modo espectador League of Legends.	16
Figura 3 – Modo espectador League of Legends.	16
Figura 4 – Histórico partida League of Legends.	17
Figura 5 – Imagem em escala de cinza.	17
Figura 6 – Representação de imagem como matriz.	18
Figura 7 – Representação de imagem como matriz.	20
Figura 8 – Representação de imagem como matriz.	21
Figura 9 – Módulos de trabalho do projeto	24
Figura 10 – Diagrama banco de dados	24
Figura 11 – Análise de similaridade entre imagens.	29
Figura 12 – Informações analisadas.	30
Figura 13 – Seleção do valor de QPSIAG	33
Figura 14 – Saída da ferramenta.	37
Figura 15 – Projeto do estudo de caso	39
Figura 16 – Taxa de predição ao fim da partida.	41

LISTA DE TABELAS

Tabela 1 – Pesos dos atributos nas etapas do jogo	26
Tabela 2 – Valores etapas do jogo	28
Tabela 3 – Análise de acertos de predição total nas partidas.	40
Tabela 4 – Análise early game.	42
Tabela 5 – Análise mid game.	42
Tabela 6 – Análise late game.	42

LISTAGEM DE CÓDIGOS FONTE

Listagem 1 – Comparacao Imagens	30
Listagem 2 – Conexão de banco de dados	32
Listagem 3 – Obtenção QSPIAG	33
Listagem 4 – Obtenção QTP	34
Listagem 5 – Código referente ao resultado das equipes vencedoras	34

SUMÁRIO

1	INTRODUÇÃO	9
1.1	Objetivos	9
1.1.1	Objetivo geral	9
1.1.2	Objetivos específicos	10
1.2	Estrutura	10
2	REVISÃO BIBLIOGRÁFICA	11
2.1	League of Legends	12
2.1.1	Reconhecimento de padrões	15
2.1.2	Imagem Digital	16
2.1.3	Reconhecimento de Padrões em Imagens	18
2.1.4	Técnica Subtract	19
2.2	Reconhecimento de padrão para jogos online	20
2.3	Banco de dados	21
3	PREDIÇÃO DE VITÓRIA NO LEAGUE OF LEGENDS	23
3.1	Introdução	23
3.2	MT-1: Criação da base de dados	23
3.2.1	Atributo ResultadoSomaVariaveis	26
3.2.2	Coleta manual	27
3.3	MT-2: Criação da ferramenta de extração	28
3.4	MT-3: Incremento da base de dados	31
3.5	MT-4: Criação da fórmula de predição	31
3.5.1	Resultado atual da equipe (RAE)	31
3.5.2	Porcentagem partida vencedoras abaixo (PPVA)	32
3.5.3	Distância de pontuação em relação a média geral (DPRM)	34
3.5.4	Exemplo de aplicação	34
3.6	MT-5: Criação da ferramenta de predição	36
3.7	Discussões	37
4	ESTUDO DE CASO	38
4.1	Planejamento	38
4.2	Execução	39

4.3	Resultados	40
4.4	Discussões	40
4.4.1	Análise sobre predição final	40
4.4.2	Análise por tempo de jogo	41
5	CONCLUSÃO	43
5.1	Trabalhos futuros	43
	REFERÊNCIAS	44

1 INTRODUÇÃO

O mercado dos jogos eletrônicos é um setor relevante da economia, considerando a quantidade de dinheiro e de pessoas envolvidas nesse setor. Segundo pesquisa realizada pela Newzoo (2022), estima-se que para o ano de 2022, o mercado de e-sports terá uma receita que girará em torno de 1,38 bilhão de dólares. E que para o ano de 2024, o setor de jogos como um todo, alcance uma receita de aproximadamente 218,7 bilhões de dólares (NEWZOO, 2021). Aproveitando-se desse sucesso, os esportes eletrônicos competitivos, ou e-sports, vêm ganhando importância na economia mundial. Além desse faturamento, estima-se que o público global de e-sports chegará em 532 milhões, deste número, 261 milhões será de pessoas que assistem e-sports mais de uma vez por mês. Em 2018 foram distribuídos cerca de 250 milhões de dólares em premiações em mais de 4000 mil eventos de e-sports em todo o mundo. Esse montante supera, por exemplo, toda a premiação do mesmo ano do tradicional circuito masculino do Tour de ténis (CONSULTING, 2019).

Com esse expressivo crescimento, o e-sport vem ganhando mais atenção de pessoas que possuem pouco ou nenhum contato com jogos eletrônicos. Desta forma, com o intuito de incluir esse grupo de pessoas, empresas responsáveis pelos campeonatos estão tentando apresentar informações e estatísticas que consigam informar a situação de cada time durante o jogo para qualquer espectador, do novato ao fanático. Porém, até o momento, não são todos os jogos que possuem um sistema que permite a extração de informações sobre a probabilidade que uma equipe tem de ganhar ou perder a partida, levando em consideração as informações apresentadas na tela. Visando esse problema, este trabalho apresenta uma aplicação, retorna em tempo real qual as chances de cada equipe chegar à vitória naquele momento da partida. Essa aplicação utiliza 1) informações extraídas da tela do modo espectador, que é a forma utilizada pela empresa responsável pela transmissão do campeonato, de exibir as partidas do e 2) uma base de dados para comparação das informações.

1.1 Objetivos

Nesta seção será apresentado o objetivo deste trabalho de conclusão de curso. Desta forma, a seguir é apresentado o objetivo geral e os objetivos específicos que norteiam a condução deste trabalho.

1.1.1 Objetivo geral

O objetivo deste trabalho consiste na criação de uma aplicação que usará os dados do modo espectador do jogo *LOL* para prever resultados da partidas e melhorar a experiência de

espectadores que assistem a partida, por meio da biblioteca openCV e da técnica subtract. Para prever os resultados, foram utilizados dados históricos de outras partidas.

Dada a ausência de trabalhos na área de predição de vitória para o jogo *LOL*, este trabalho contribui com o estado da arte, uma vez que apresenta o conceito de reconhecimento de padrão em jogos online com o objetivo de extrair dados para análises sobre o jogo.

1.1.2 Objetivos específicos

Para alcançar o objetivo do trabalho, definiram-se os seguintes objetivos específicos:

1. Apresentar um levantamento das técnicas de reconhecimento de padrões, para um melhor entendimento do assunto;
2. Implementar uma aplicação para processamento de imagem em tempo real com o intuito de obter e comparar dados do modo espectador e base de dados de partidas profissionais;
3. Validar a coleta de dados da interface gráfica a partir de um conjunto de dados de partidas oficiais de campeonatos de *LOL*;
4. Criação da aplicação para análise e predição de uma vitória em tempo real; e
5. Estudo de caso para validação da aplicação em partidas de jogos online.

1.2 Estrutura

O trabalho está dividido em cinco capítulos. No Capítulo 2, são apresentadas as principais definições e conceitos referentes ao contexto do trabalho. O Capítulo 3, descreve a ferramenta desenvolvida que utiliza métodos de reconhecimento de padrão para extrair dados da tela e com base nos dados, a ferramenta faz a predição de vitória dos times durante a partida. No Capítulo 4 é apresentado o estudo de caso realizado para a avaliação da ferramenta. Por fim, o Capítulo 5 realiza as considerações finais a respeito do trabalho.

2 REVISÃO BIBLIOGRÁFICA

O mercados de jogos eletrônicos, os jogos do gênero estão nas listas dos mais jogados mundialmente. Nesse gênero, 2 times disputam para ver quem consegue destruir mais rapidamente a base inimiga. Segundo dados divulgados pela empresa RIOT, desenvolvedora do jogo League of Legends, popularmente conhecido como *LOL*, todos os dias, cerca de 8 milhões de usuários acessam o jogo em horário de pico, o que o torna o jogo de computador mais jogado do mundo (ABREU, 2019).

Devido a atenção que os e-sports vêm recebendo, esse tipo de entretenimento passou a estar disponível para um maior número de pessoas. Os campeonatos são transmitidos em plataformas de streams ao vivo, geralmente em seus canais oficiais. Também existem empresas não ligadas à empresa organizadora do campeonato, que compram os direitos de transmissão e transmitem em seus respectivos canais. Em ambos os métodos, o número de espectadores é elevado.

Porém, a dinâmica desses jogos é, muitas vezes, confusa para espectadores que nunca jogaram o jogo. Dessa forma, o uso de ferramentas para a análise de jogos faz-se necessário para trazer um maior entendimento aos espectadores e, com isso, uma melhor experiência em relação a partida transmitida. No mercado de apostas, já há ferramentas dos próprios sites, que servem para calcular qual ganho um apostador teria em cima do time que apostou com base em tudo que aconteceu na um uma partida até tal momento.

Com esse crescimento, estudos que visam analisar e melhorar o e-sport como um todo vêm sendo propostos. Na parte computacional, estudos que utilizam reconhecimentos de padrões e aprendizado de máquina visam melhorar tanto o entendimento do jogo por parte dos times, quanto por parte dos telespectadores. Na parte mais voltada para o jogo *LOL*, alguns estudos que podem ser citados são: Silva, Pappa e Chaimowicz (2018), Souza (2017) e Costa, Souza e Souza (2019).

A definição de reconhecimento de padrões é quando um padrão é atribuído a uma classe dentro de um número de classes (ERPEN, 2004). Boas praticas para um reconhecimento de padrões em imagens passa por 5 etapas: aquisição de imagem, pré-processamento, segmentação, extração de feições e classificação (ERPEN, 2004).

Esse capítulo apresenta o referencial teórico deste trabalho. Para isso, a Seção 2.1 apresenta os principais aspectos do jogo *LOL* que é o foco deste trabalho. A Seção 2.2 apresenta os principais conceitos sobre reconhecimento de padrões e a Seção 2.3 apresenta os principais conceitos sobre banco de dados. Ambas seções servirão como base para a implementação da abordagem proposta.

2.1 League of Legends

O *LOL* é um jogo de estratégia online no qual os jogadores são divididos em duas equipes. O objetivo de cada equipe é destruir construções da equipe inimiga. Para isso, o jogador utiliza os campeões: personagens disponíveis que os jogadores devem selecionar para iniciar as partidas. Cada campeão contém habilidades únicas e cada jogador pode utilizar apenas um por partida. O *LOL* resume-se a dois modos de jogo: *normal game* e *ranked*. No modo *normal game*, as partidas são feitas unicamente para diversão, pois uma vitória ou derrota não afeta a classificação do jogador no jogo. Já no modo *ranked*, a vitória ou derrota influencia diretamente em sua colocação do jogo. A classificação do jogador é definida pelo *rank*, ou elo, como é chamado no *LOL*. O *rank* representa um método para o cálculo de habilidade relativa de jogadores. Esse tipo de ranqueamento é originário das classificações de jogadores de xadrez. No *LOL*, são criadas faixas de pontuação e atribuídas a nove diferentes *ranks*, do menor para o maior: ferro, bronze, prata, ouro, platina, diamante, mestre, grande-mestre e desafiante. Dos níveis ferro ao diamante, há 4 divisões: I, II, III e IV, sendo IV a mais baixa e I a mais alta. Para subir de *rank*, por exemplo, saindo do ferro e subindo para o bronze, necessita-se passar por todas essas divisões. Já nos *ranks* mestre, grande-mestre e desafiante, a pontuação é corrida e só consegue-se subir de *rank* obtendo as melhores posições dentro do servidor da região em que o jogador estiver. Nesse sistema de classificação por *rank*, é possível tanto subir quanto descer de acordo com seu desempenho no jogo.

O *LOL* oferece 148 personagens, conhecidos como campeões. Os campeões são diversificados, cada um possuindo atributos únicos. Campeões tank são aqueles com maior vida e defesa. Os atiradores são focados em velocidade de ataque, ataques básicos e causadores de dano do tipo físico. Já os magos são campeões que majoritariamente causam danos com suas habilidades, ou *skills* e, com isso, causam danos mágico. Assassinos são focados em causar muito dano em pouco tempo, tendo como objetivo sempre matar alvos de maneira rápida, causando tanto dano físico quanto mágico. Já os suportes, utilizam *skills* para auxiliar o time, seja com escudos ou curas. Para obter determinado campeão, é necessário utilizar essências azuis, as quais se obtêm completando missões, subindo de nível, ou usando rp: moeda virtual que se compra utilizando moeda corrente de cada país.

Para jogar o *LOL*, o jogador precisa fazer o download do jogo no site oficial da RIOT games, a empresa mantedora do jogo. Após efetuar o download, é necessária a criação de uma conta. O nível inicial 1 será atribuído ao jogador ao criar uma conta. Este nível demonstra quanto o jogador já jogou. Consegue-se subir de nível jogando partidas ou finalizando missões que ficam disponíveis dentro do jogo. Até o jogador alcançar o nível 30, será apenas possível jogar o modo normal. Isso é feito para que jogadores iniciantes tenham como adquirir mais experiência e, assim, conseguir ingressar ao modo *ranked*. Ao alcançar o nível 30 e um total de 20 campeões, as partidas *ranked* serão liberadas. Quando se joga *ranked* pela primeira vez, é realizada uma melhor de 10 partidas. Conforme o número de vitórias e derrotas obtidas pelo

jogador nessas partidas, ele receberá um *rank* inicial para começar. Para subir de *rank*, torna-se necessário ganhar Pontos De Liga , pontos que se obtêm com vitórias e perde-se com derrotas. Entre os elos ferro e diamante, só é possível obter 100 PDL por divisão. Quando se alcança essa pontuação, uma série de 5 partidas é iniciada quando se está na divisão I. Sobe-se de divisão ou *rank* caso 3 vitórias sejam obtidas. Quando se chega a 0 PDL e perde-se muitas partidas seguidas, há a possibilidade de rebaixamento de divisão ou *rank* automaticamente. Não existe um limite de PDL nos elos mestre a desafiante. Nesses casos, a subida de *rank* funciona como um sistema de escalada, ou seja, quando um jogador obtém mais pontos que seu adversário, ele pega a posição do outro jogador.

No *LOL*, 10 jogadores são divididos em 2 times de 5 integrantes cada e são posicionados em lados extremos de Summoner's Rift, nome dado ao local no qual são disputadas as partidas, conforme mostrado na Figura 1. O mapa divide-se em três caminhos principais, popularmente conhecidos como lanes. As lanes existentes são: Top (lane superior do mapa) Mid (lane central do mapa) Bot (lane inferior do mapa). Além dessas 3 lanes, o mapa também contém a selva, que fica entre as lanes. Este é o padrão utilizado na maioria dos jogos competitivos, mas nada impede de uma equipe jogar de maneira diferente. O jogo conta com 5 posições:

Figura 1 – Mapa League of Legends.



Fonte: RIOT Games.

- Top laner : Jogador que fica na rota do topo;
- Mid laner : Jogador que fica na rota central;
- Atirador : Jogador que fica na bot lane com o Suporte;
- Suporte : Jogador que fica na bot lane com o Atirador;

- Jungler : Jogador que fica na selva;

Antes de iniciar uma partida, existe a etapa chamada de seleção de campeões, na qual cada equipe tem o direito de realizar até 5 banimentos de quaisquer campeões entre os 148 disponíveis, sendo 3 no começo e mais 2 no decorrer das escolhas. Após os 3 banimentos iniciais, cada equipe, de forma intercalada, escolhe 3 campeões. Com essa etapa de escolha concluída, mais 2 banimentos são realizados por cada equipe e depois mais 2 campeões novamente de forma intercalada são escolhidos por cada equipe. Além dos campeões, nessa etapa, é possível escolher os feitiços de invocadores, ou *spells*, que são habilidades que os campeões podem usar e tem um tempo de reutilização alto. Também é possível escolher as runas, que dependendo como são feitas, mudam a dinâmica de um campeão dentro da partida.

Com a conclusão da seleção de campeões, cada time inicia o jogo na sua base. Além dos campeões escolhidos, a partida conta com os minions: criaturas que, após um minuto e meio de partida, nascem a cada 30 segundos e andam em todas as três lanes. Os minions tem como função ajudar na destruição de estruturas (torres, inibidores e nexus) dentro da partida. Quando os minions são, dão experiência que ajuda o campeão a subir de nível, aumentando *skills* e ficando mais forte. Eles também proporcionam ouro para comprar itens que aumentam os atributos do campeão. Além dos minions, existem monstros que ficam espalhados pela selva. O jogador na posição de *jungler* pode matar os monstros com a *spell* de golpear (*smite*), obtendo experiência e ouro. Dentre os monstros da selva, existem 3 que se destacam:

- Dragão: Tem 5 tipos de elementos e a cada vez que uma equipe o mata, ela ganha uma melhoria (*buff*) dependendo de qual elemento o dragão pertence, e nasce aos 5 mim, e renasce a cada 5 minutos depois de morto. Quando uma equipe mata 4 dragões, aparece um novo tipo, o dragão ancião, que quando morto, dá ao time um *buff* para eliminar inimigos com uma certa quantidade de vida. O dragão fica localizado na parte inferior do mapa, dentro de um covil.
- Arauto: É um mostro que só fica até os 20 mim e posteriormente é substituído pelo baron. Quando morto, ele oferece um *buff* de invocação a um único jogador, que ao ser utilizado, invoca o arauto para seu time. Ele tem uma utilidade muito boa para causar uma grande quantidade de dano em estruturas. Ele só pode renascer uma única vez, 5 minutos após ser morto. O arauto fica localizado na parte superior do mapa, dentro de um covil.
- Baron: É um monstro que substitui o arauto aos 20 mim, ficando localizado no mesmo covil que o arauto. Seu *buff* atinge todos os membros do time, aumentando um pouco do dano físico e mágico do time e deixando todos os minions que estejam perto de algum integrante da sua equipe com o *buff* mais forte. Ele é extremamente útil para destruir estruturas.

Outra forma de conseguir experiência e ouro é matando campeões inimigos. Quando um campeão inimigo morre, ele renasce na base da equipe, porém precisa esperar um tempo para renascer. Esse tempo varia de acordo com o nível em que ele se encontrava quando foi morto.

O objetivo principal do jogo é destruir o nexus inimigo, objeto localizado no centro de ambas as base. Porém, para conseguir atacar o nexus e suas torres que o protegem, torna-se necessário destruir os inibidores, que são objetos que ficam na borda de cada base, no início de cada lane. Ao destruir um inibidor, ganha-se um minion especial, o super minion. Ele começa a aparecer junto com os minions normais a cada 30 segundos, porém só aparece na lane que o inibidor foi destruído. Ele só para de surgir quando o inibidor é reconstruído, a cada 5 minutos. Para conseguir destruir os inibidores, é necessário destruir todas as 3 torres que se encontraram no decorrer da lane.

Competitivamente, o *LOL* conta com um circuito fechado, ou seja, a própria empresa gerencia os campeonatos. Ela tem suas próprias ligas que acontecem simultaneamente, e com base nos resultados dos times, eles são classificados para campeonatos internacionais. Essas ligas e suas respectivas regiões são: LCK (Coreia do Sul), LPL (China), CBLOL (Brasil), LLA (América Latina), LCS (América do Norte), LEC (Europa), LJL (Japão), TCL (Turquia), OPL (Oceania), VCS (Vietnã), PCS (Taiwan, Hong Kong, Macau e Sudeste-asiático) e CIS (Rússia, Ucrânia e demais ex-repúblicas soviéticas).

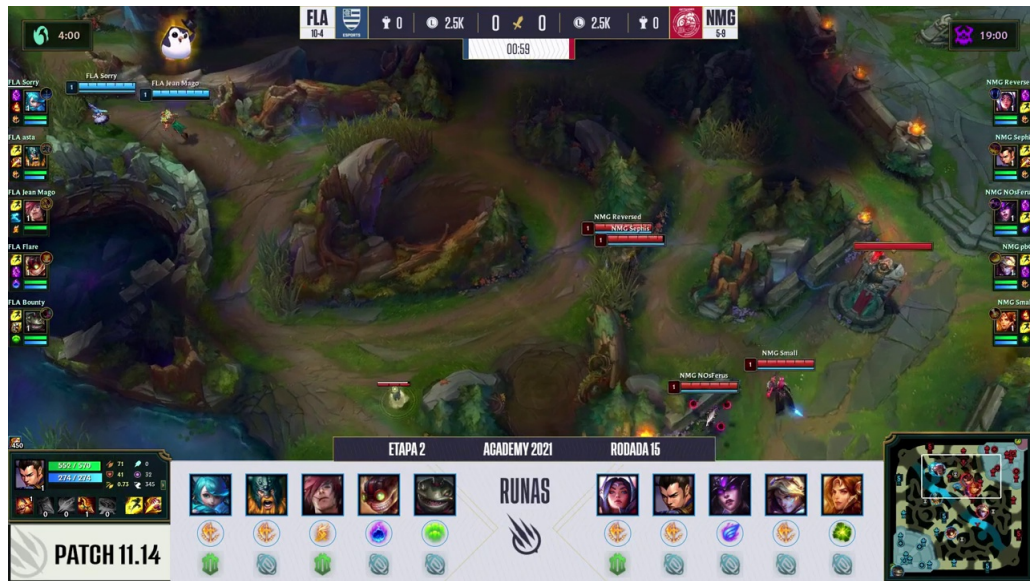
Conforme mostra na Figura 3, o modo espectador do *LOL* já traz informações para o telespectador, como placar de abates por equipe na parte superior da tela, e o tempo de jogo. Ao lado do placar de abates, há o ouro de cada equipe, o número de torres destruídas por equipe, sigla de cada time e abaixo o retrospecto de cada equipe no campeonato disputado. Nas laterais, há informações sobre os campeões utilizados por cada equipe, juntamente com as disponibilidades dos *spell* dos jogadores. Na parte inferior esquerda da tela, mostra detalhadamente os atributos, *skills* e *spell* de um campeão específico. Na parte inferior central, o número de minions mortos por cada jogador, o número de Abates/Mortes/Assistências de cada jogador, os itens que cada jogador tem naquele momento e o placar de visão de cada jogador. O placar de visão representa o número de vezes que um jogador colocou ou retirou do jogo uma sentinela de visão, item do jogo que ilumina uma parte do mapa, concedendo assim visão das ações do time adversário.

Ao fim das partidas, a RIOT disponibiliza todas as informações que aconteceram durante a partida (Figura 4), onde cada cor representa uma equipe e que mostram: diferença de ouro (eixo Y) entre as equipes baseado no tempo (eixo X), tempo de jogo em que as torres e inibidores foram destruídos e dragões, arautos e barons derrotados.

2.1.1 Reconhecimento de padrões

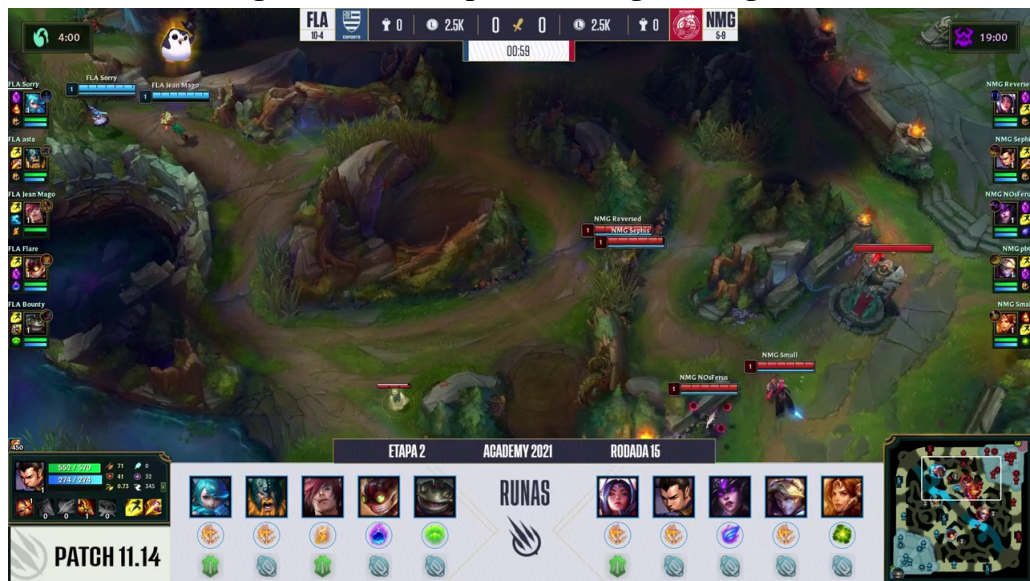
A área de processamento digital de imagens refere-se ao processamento de imagens digitais por um computador digital (GONZALEZ; WOODS, 2000). Inserido nesse contexto

Figura 2 – Modo espectador League of Legends.



Fonte: RIOT Games.

Figura 3 – Modo espectador League of Legends.



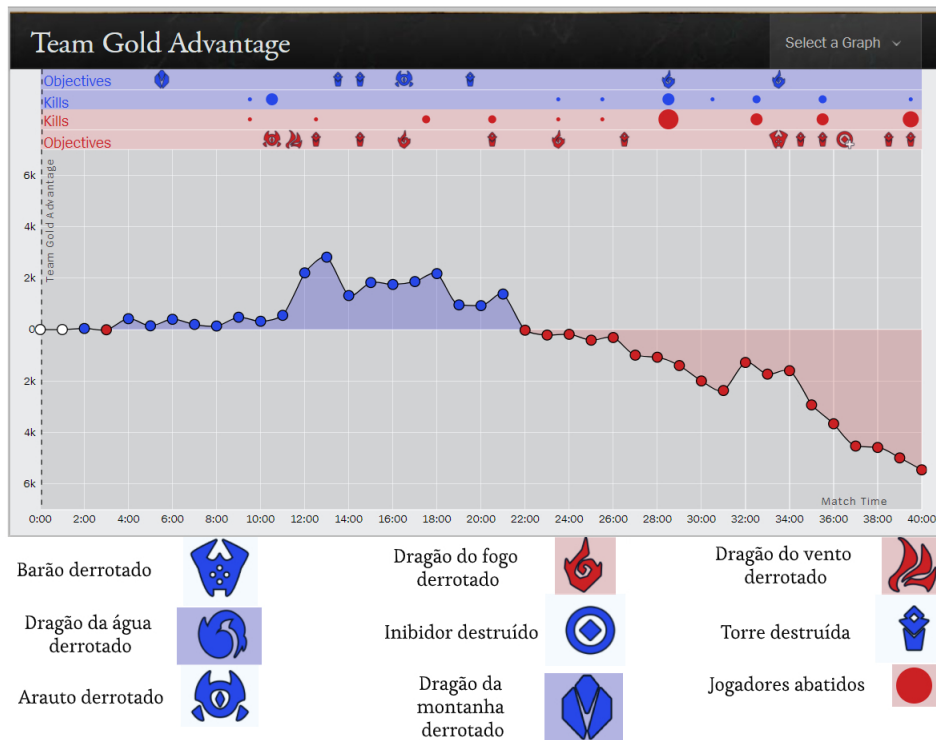
Fonte: RIOT Games.

está o reconhecimento de padrão em imagens digitais. Esta é um área em evolução da ciência da computação, sendo aplicada a diversas áreas de conhecimento como biologia (UCHIDA, 2013), medicina (ISASI; ZAPIRAIN; ZORRILLA, 2011), engenharia metalúrgica (BULGAREVICH *et al.*, 2018), entre outras.

2.1.2 Imagem Digital

Segundo Gonzalez e Woods (2000), uma imagem, formada por um único canal de cor, pode ser descrita por uma função bidimensional $f(x, y)$, em que x e y são coordenadas espaciais

Figura 4 – Histórico partida League of Legends.



Fonte: RIOT Games.

(plano), e a amplitude de f em qualquer par de coordenadas (x, y) é chamada de intensidade ou nível de cinza da imagem nesse ponto. Para imagens que possuam mais de um canal de cor, uma função f será atribuída para cada canal de cor. Por exemplo, o padrão de cor BGR (Blue, Green, Red) possui uma função f para a cor azul, um para o verde e um para o vermelho.

Quando x , y e os valores de intensidade de f são quantidades finitas e discretas, chamamos de imagem digital. Uma imagem digital é composta de um número finito de elementos, chamados de pixels. Cada pixel possui uma localização e um valor específico. A Figura 5 apresenta uma imagem em escala de cinza. Nesse exemplo, o nível de cinza de cada ponto é proporcional ao valor da intensidade da função f desse ponto.

Figura 5 – Imagem em escala de cinza.



Fonte: Autoria Própria.

Uma imagem digital é comumente representada por meio de uma matriz numérica de tamanho $N \times M$. Cada posição na matriz A representa a aplicação da função f em um pixel na posição i, j , de forma que $a_{ij} = f(x = i, y = j)$. Um pixel p na coordenada (x, y) tem quatro vizinhos horizontais e verticais cujas coordenadas são dadas por: $(x + 1, y)$, $(x - 1, y)$, $(x, y + 1)$, $(x, y - 1)$.

Figura 6 – Representação de imagem como matriz.

$$\mathbf{A} = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \vdots & \vdots & & \vdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{bmatrix}$$

Fonte: Gonzalez e Woods (2000).

2.1.3 Reconhecimento de Padrões em Imagens

O reconhecimento de padrões tem como objetivo categorizar ou classificar um grupo de dados de entrada. A priori, um bom projeto que usa reconhecimento de padrões utiliza 5 etapas, que são descritos em detalhes a seguir:

- **Aquisição da Imagem:** Nessa etapa, sensores ópticos são utilizados para captar sinais emitidos de luz que posteriormente são digitalizados. Exemplos de sensores são câmeras de vídeo e scanners.
- **Pré-processamento:** Essa é uma etapa opcional que tem como objetivo tentar reduzir possíveis falhas ou aperfeiçoar imagens captadas no processo de aquisição.
- **Segmentação:** Nessa etapa a imagem é segmentada levando em consideração a informação contida na imagem e como ela pode ser agrupada. O agrupamento pode ser feito de duas formas: agrupando pontos semelhantes ou determinando regiões a partir de uma borda.
- **Extração de feições:** Essa etapa, também chamada de seleção de características, lida com a extração de atributos que resultam em alguma informação quantitativa de interesse ou que possam ser utilizados para diferenciar uma classe de objetos de outra (GONZALEZ; WOODS, 2000). Essa etapa representa a imagem como um vetor numérico com objetivo de reduzir redundâncias de dados.
- **Classificação:** Etapa que se foca em reconhecer algum objeto na imagem.

Além dessas etapas, podemos citar os processamentos de imagem coloridas que se tem algumas técnicas principais: Processamento de imagens coloridas que são adquiridas imagens com utilização de sensores de cores reais, como televisões em cores. E pseudocores que atribui cores a valores de cinza com base em determinados critérios (GONZALEZ; WOODS, 2000).

Já métodos de processamento das imagens coloridas é dividido em duas categorias principais. Na primeira categoria, o processamento ocorre de forma individual de cada componente de imagem. Depois de todos os componentes serem processados, forma-se a imagem colorida. Na segunda categoria, em vez de fazer a separação, já se trabalha diretamente com os pixels coloridos. (GONZALEZ; WOODS, 2000).

Na fase de classificação, pode-se usar as duas formas de classificação: classificação supervisionada e classificação não-supervisionada (HUANG; KECCMAN; KOPRIVA, 2006). A classificação não-supervisionada se dá pelo processo de agrupar dados de entradas que tenham um certo nível de semelhança, no qual amostras semelhantes formam um grupo. Porém, tal processo é extremamente difícil, pois podem haver grupos de diversos tamanhos e formas e o número deles podem variar de acordo com a resolução que os dados são trabalhados (DUDA; HART; STORK, 2012).

Essa classificação é baseada em duas técnicas muito populares: clustering hierárquico que organiza os dados em grupos com base em suas semelhanças e o clustering particional que visa encontrar o máximo de semelhança dentro das amostras de um grupo (PAPA *et al.*, 2008).

Outra importante técnica que se utiliza da classificação não-supervisionada é a de redes neurais convolucionais. A técnica consiste em escolher arquitetura ou topografia para reduzir o tempo durante o aprendizado. Ela é projetada para processar dados em forma de múltiplas matrizes dimensionais e, dependendo do tipo de amostra a ser analisada, ela cria diferentes quantidades de dimensões, sendo uma para sinais e sequências, duas para imagens e espectrogramas e três para vídeos e imagens volumétricas (FERREIRA, 2017).

Já a classificação supervisionada ocorre quando já se tem classes de amostras de treinamento para realizar a comparação com a entrada de dados que será identificada. Existem diversas técnicas que utilizam a classificação supervisionada, entre as mais utilizadas está a redes neurais artificiais. Essas técnicas utilizam de métodos matemáticos e tentam simular sistemas de organismos inteligentes e, com isso, aprender conforme experiências adquiridas anteriormente. O principal objetivo é desenvolver uma técnica que possa criar uma regra de treinamento para tentar otimizar o resultado obtido como saída (PAPA *et al.*, 2008).

Outro método de classificação supervisionada utilizado é o , no qual todas as entradas de dados são utilizadas com o intuito de fazer uma separação utilizando características diferentes entre os dados. Esse método obtém um alto desempenho, porém com base no tamanho do conjunto de entrada de dados, seu custo computacional aumenta exponencialmente (PAPA *et al.*, 2008).

O método k-vizinhos também é bastante conhecido, cuja estratégia divide o conjunto de entradas com base em sua proximidade em relação ao conjunto de treinamento. Apesar de sua fácil ideia, ela pode vir a sofrer com dados irrelevantes, tendo em vista seu grande distanciamento do conjunto de treinamento. Seu custo operacional aumenta conforme o tamanho da entrada de dados utilizada (PAPA *et al.*, 2008).

2.1.4 Técnica Subtract

O subtract é uma técnica processamento de imagens que compara duas imagens. Para realizar essa comparação, as imagens são transformadas em matrizes numéricas, onde cada pixel representa uma posição na matriz. Em seguida, é feita a subtração de cada valor da matriz. Se

as imagens forem iguais, o resultado da subtração será uma matriz nula, ou seja, a qual todos os valores são 0.

Na linguagem de programação Python, a biblioteca `opencv` é usada para manipulações de vídeos e imagens. Ela recebe como parâmetro uma imagem e faz a conversão dela em uma matriz numérica. A Figura 7 demonstra uma parte de como é a matriz.

Figura 7 – Representação de imagem como matriz.

```
[[[61 45 46] [64 43 46]
 [60 44 45] [64 43 46]
 [60 44 45] [64 43 46]
 ...
 [59 44 42] [64 43 46]
 [58 43 41] [64 43 46]
 [56 41 39]]]
[[[62 43 46] [64 43 46]
 [60 44 45] [64 43 46]
 [60 43 46] [64 43 46]
 ...
 [58 43 41] [64 43 46]
 [57 42 40] [64 43 46]
 [56 41 39]]]
[[[63 41 46] [65 42 46]
 [61 42 45] [65 42 46]
 [61 41 46] [65 42 46]
 ...
 [60 42 41] [64 43 46]
 [58 40 39] [64 43 46]
 [57 39 38]]]
```

Fonte: Autoria própria.

Com duas imagens em matriz, a função `subtract` vai subtraindo os valores das matrizes posição por posição, e após realizar esse processo completamente, retorna em uma matriz o resultado das subtrações dos pixels, como demonstrado na Figura 8. Valores como 0, indicam que os pixels ali sobrepostos são exatamente iguais, e quanto maior esse valor, maior é a diferença entre os pixels comparados.

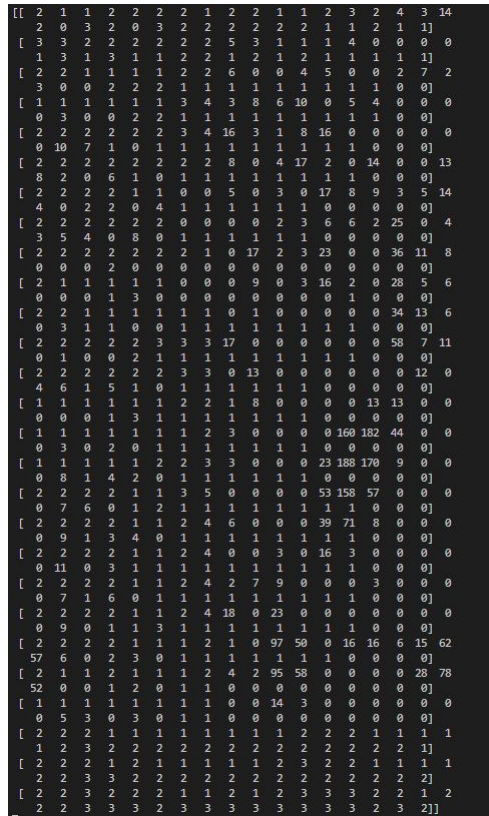
Quando maior o numero de valores perto ao valor 0 forem retornados, mais semelhantes as imagens são. E quanto maiores esses valores forem, mais diferentes as imagens vao ser.

2.2 Reconhecimento de padrão para jogos online

O universo de jogos eletrônicos envolvem muitas questões, desde melhorar a experiência dos telespectadores de e-sports, questões de segurança e fisionomia dos personagens do jogo. Nesse meio, o reconhecimento de padrões para jogos online vem sendo usado para áreas como identifica jogadores que abusam de ferramentas ilegais e até processos de melhoras de processos de modelações de personagens o mais próximo possível da realidade.

Segundo Yan e Randell (2005), trapaças (*cheating*) é o ato de usar algum recurso que esteja fora dos métodos normais do jogo e que aplique alguma vantagem que não poderia ser habilitada no jogo. Visando esse problema de segurança de trapaças, Gaspareto (2008) propôs um estudo que utilizou reconhecimento de padrões para identificação de trapaças no jogo `Hoverkill`. O estudo usou a técnica de Redes Neurais Artificiais (RNA) na qual unidades são treinadas por algoritmos supervisionado e não-supervisionado a fim de replicar o funcionamento de um neurônio. Como resultado do estudo, a aplicação da técnica obteve uma precisão de 60% na

Figura 8 – Representação de imagem como matriz .



Fonte: Autoria própria.

detecção da trapaça *Speed cheating*, que aumenta a velocidade do jogador dentro do game de uma forma fora dos métodos normais do game.

Já Teruel *et al.* (2019) utilizou reconhecimento de padrões para criar um software que agrupa representações da face humana para possibilitar a criação de faces que se assemelhem o mais próximo possível de uma face real em jogos 3D. Uma das técnicas mais usadas é a *Motion Capture* dividida em três categorias: sistemas ópticos, sistemas magnéticos e sistemas mecânicos. Depois de feita a captura por uma dessas categorias, é feita uma modelagem 3D. Outra técnica é o desenho livre no qual um artista cria representações 2D ou 3D da pessoa. A desvantagem dessa técnica é que, por se tratar de um desenho feito, traços do artista ficarão presentes mais visivelmente no produto criado.

2.3 Banco de dados

Banco de dados é um armazenamento de um ou múltiplos registros (DATE, 2004). Nele se é possível fazer operações como:

1. Acrescentar um ou vários registros;
2. Buscar um ou vários registros;

3. Editar um ou vários registros;
4. Remover um ou vários registros;

Tais operações são feitas utilizando a linguagem *sql*, uma linguagem inicialmente pertencente a IBM e que se tornou um padrão mundial (DATE, 2004). Existem duas estruturas amplamente usadas para banco de dados, os relacionais e os não relacionais.

Os banco de dados relacionais utilizam um conceito de relação chamada de tabelas, que organizam dados por um ou vários atributos, e esse conjunto de dados são organizados em registros. Uma característica marcante nesse tipo de banco de dados é a utilização de chaves primarias que identifica um registro como único dentro de uma relação e, com isso, aumentando o desempenho de consultas, e chaves estrangeiras que permitem a realização de relações entre as tabelas (SOUZA; SANTANA, 2017).

Já os bancos de dados não-relacionais funcionam mesmo sem um estrutura pré-definida de tabelas. Por conta disso, é muito mais flexível para diversas situações de armazenamento, assim ganhando mais performace, flexibilidade e escalabilidade do bando de dados em situações específicas(BRITO, 2010).

Além dessas duas estruturas citadas, um banco de dados também conta um sistema de gerenciamento de bando de dados (SGBD) que é um conjunto de programas de computador que gerenciam uma coleção de dados e que tem como principal objetivo tirar a aplicação cliente de gerenciar e manipular dados (MAGALHÃES *et al.*, 2015).

O postgresql é um SGBD de banco de dados relacional que possui recursos versáteis e seguros, além de possuir código aberto. Com isso, torna-se uma ótima opção para empresas e acadêmicos que visam ter confiabilidade na aplicação que utilizam (MAGALHÃES *et al.*, 2015).

3 PREDIÇÃO DE VITÓRIA NO LEAGUE OF LEGENDS

Este capítulo apresenta o projeto conduzido neste trabalho de conclusão de curso. A Seção 3.1 apresenta uma breve introdução aos módulos de trabalhos (MT) no qual o projeto foi dividido para uma organização maior em seu desenvolvimento. A Seção 3.2 apresenta o MT-1 e como foi feita a criação da base de dados que serviu como base para a aplicação. A Seção 3.3 apresenta o MT-2 e o processo de criação da ferramenta de extração de dados das partidas do jogo. A Seção 3.4 mostra o MT-3 e como a base de dados foi aumentada para utilizar na aplicação. A Seção 3.5 explica o MT-4 e como foi pensado e desenvolvido o fórmula de predição usado pela aplicação. A Seção 3.6 explica o MT-5 e o funcionamento da aplicação e o que ela apresenta ao seu usuário. Por fim, a Seção 3.7 demonstra uma discussão a cerca da aplicação como um todo.

3.1 Introdução

Este projeto seguiu o esquema técnico apresentado na Figura 9, e é composto por cinco módulos de trabalhos (MT). Cada módulo representa uma atividade que foi conduzida para atender aos objetivos específicos deste trabalho. São eles:

1. MT-1: Criação da base de dados;
2. MT-2: Criação da ferramenta de extração;
3. MT-3: Incremento da base de dados;
4. MT-4: Criação da fórmula de predição; e
5. MT-5: Criação da ferramenta de predição.

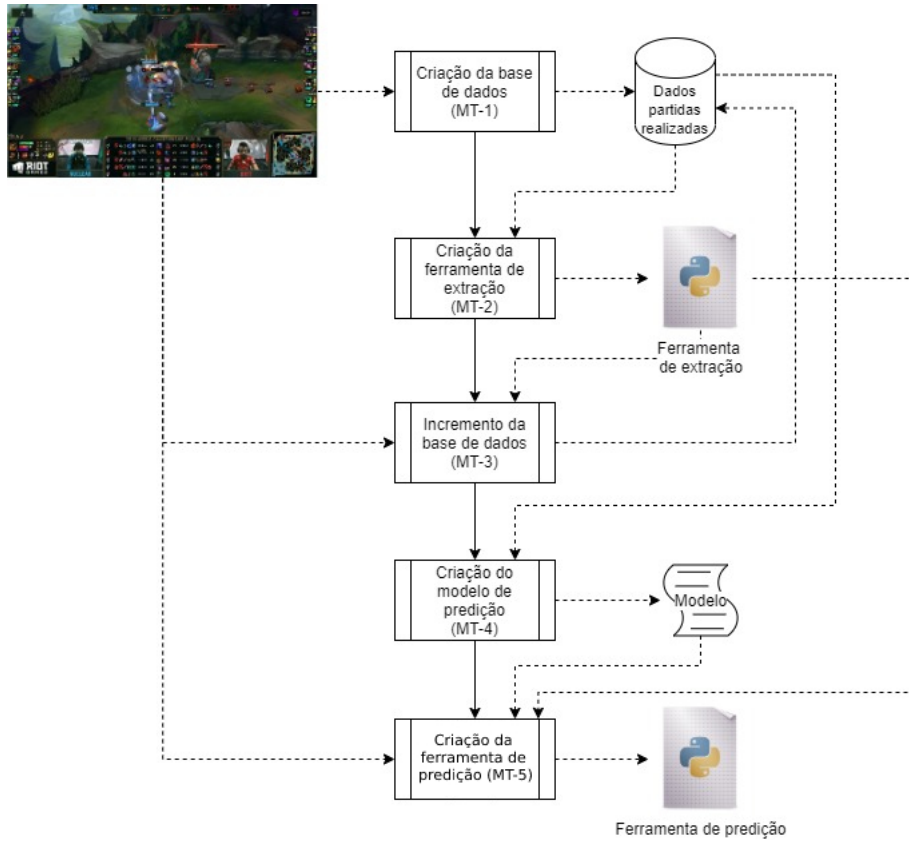
3.2 MT-1: Criação da base de dados

O primeiro módulo de trabalho consistiu na criação de uma base de dados inicial com partidas de *LOL* já realizadas. Esse módulo de trabalho tem como objetivo criar uma base de dados com valores reais que serão utilizado pela ferramenta de predição. Quanto mais dados houver na base de dados, mais assertiva será a predição de vitória da ferramenta.

Durante esse módulo, foi realizada a modelagem do banco de dados tipo relacional com PostgreSQL¹ a ser utilizado durante todo o trabalho e que seguiu o diagrama apresentado na Figura 10, composto por 6 Tabelas:

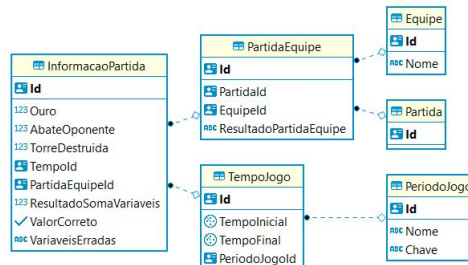
¹ <https://www.postgresql.org/>

Figura 9 – Módulos de trabalho do projeto



Fonte: Autoria própria.

Figura 10 – Diagrama banco de dados



Fonte: Autoria própria.

- **Equipe:** Tabela que armazena os dados das equipes que disputam as partidas. Composta pelas colunas:
 1. Id: chave primeira da tabela.
 2. Nome: nome da equipe.

- **Partida:** Tabela que armazena os dados das partidas que foram jogadas. Composta pela coluna:
 1. Id: chave primeira da tabela.

- **Período Jogo:** Tabela que armazena os dados dos períodos do jogo utilizados na fórmula de predição. Composta pelas colunas:
 1. Id: chave primeira da tabela.
 2. Nome: nome do período.
 3. Chave: coluna de identificação em texto.

- **Partida equipe:** Tabela que armazena e relaciona equipes às partidas que elas jogaram. Composta pelas colunas:
 1. Id: chave primeira da tabela.
 2. PartidaId: id da partida jogada.
 3. EquipeId: id da equipe que jogou a partida.
 4. ResultadoPartidaEquipe: resultado de vitória ou derrota que a equipe teve na partida.

- **Tempo jogo:** Tabela que armazena e relaciona os tempos de jogo ao período de jogo que ele pertence. Composta pelas colunas:
 1. Id: chave primeira da tabela.
 2. TempoInicial: armazena o tempo inicial dentro do intervalo de um minuto. Exemplo de dado gravado 01:00.
 3. TempoFinal: armazena o tempo final dentro do intervalo de um minuto. Exemplo de dado gravado 01:59.
 4. PeriodoJogoId: id do período do jogo.

- **Informação partida:** Tabela que armazena e relaciona as equipes e partidas as informações que foram obtidas delas durante as partidas. Compostas pelas colunas:
 1. Id: chave primeira da tabela.
 2. Ouro: informação extraída via ferramenta de extração do total de ouro da equipe durante o intervalo de tempo analisado.
 3. AbateOponente: informação extraída via ferramenta de extração do total de abates de oponentes da equipe durante o intervalo de tempo analisado.
 4. TorreDestruída: informação extraída via ferramenta de extração do total de torres destruídas pela a equipe durante o intervalo de tempo analisado.
 5. TempoId: id do intervalo de tempo de jogo em que as informações foram obtidas.
 6. PartidaEquipeId: id de relação da partida e equipe que teve os dados extraídos.

7. ResultadoSomaVariaveis: resultado da equipe durante o intervalo analisado obtido através da fórmula de predição.
8. ValorCorreto: coluna responsável por informar se os dados de ouro, abate ou torre foram extraídos de forma correta. Esse valor pode ou não ser utilizado na fórmula de predição. Caso haja inconsistência na obtenção dos valores, esses valores são descartados na utilização da fórmula de predição.
9. VariaveisErradas: coluna que armazena qual das variáveis entre ouro, abate ou torre veio de forma inconsistente.

3.2.1 Atributo ResultadoSomaVariaveis

Para auxiliar no processo de predição de vitória posteriormente no MT-4, a cada entrada no banco de dados em que os valores forem corretos, é feito um cálculo do resultado atual da equipe. Esse cálculo é uma forma de medir o desempenho da equipe naquele momento do jogo. Esse valor é calculado seguindo a Equação 1:

$$RAE = (abate * peso) + (ouro * peso) + (torre * peso) \quad (1)$$

O peso de cada atributo da fórmula é definido considerando o conhecimento dos autores do trabalho em relação ao jogo e o período do jogo. Isso se deve ao fato de que a importância desses atributos no jogo varia dependendo do período de jogo decorrido. Os períodos do jogo são popularmente conhecidas como *early*, *mid* e *late*. O valor dos pesos são definidos conforme mostrado na Tabela 1.

Tabela 1 – Pesos dos atributos nas etapas do jogo

	Early	Mid	Late
Abate	3	2	3
Ouro	2	3	1
Torre	1	3	1

Fonte: Autoria Própria.

Early

Para esse período, foi considerado o tempo de 00:00 min até 14:59 min. Os valores dos pesos para esse período são:

- Ouro: peso 2, tendo em vista que quanto mais ouro, mais itens é possível comprar e, consecutivamente, ficar mais forte.
- Torre destruída: peso 1, visto que nesse período os jogadores estão em suas rotas e longe das torres. Dessa forma, torres são bem difíceis de serem destruídas.

- Abate oponente: peso 3, considerando que todos os jogadores estão em suas rotas. Dessa forma, quando se mata um oponente, se ganha experiência para subir de nível e ouro para comprar itens. Isso acaba gerando uma vantagem grande sobre o oponente.

Mid

Para esse período, foi considerado o tempo de 15:00 até 34:59 ou fim da partida. Os valores dos pesos para esse período são:

- Ouro: peso 3, visto que quanto maior o ouro, mais forte o jogador é, causando uma grande disparidade em termos de poder entre os jogadores
- Torre destruída: peso 3, pois nessa etapa, o mapa começa a ser mais explorados pelos jogadores. Com isso, se o time tem menos torres a seu favor, ele fica desprotegido de possíveis jogadas que podem ser feitas pelo time adversário.
- Abate oponente: peso 2, pois os ganhos de experiência e ouro ainda são importantes. Porém, nessa parte do jogo, abates muitas vezes não conseguem trazer vantagens expressivas para algum time.

Late

Para esse período, foi considerado o tempo de 35:00 até o fim da partida. Os valores dos pesos para esse período são:

- Ouro: peso 1, pois nessa etapa da partida, todos os jogadores responsáveis por aplicar danos em seus oponentes já estão praticamente com todos os itens que conseguem comprar. Com isso, o ouro acaba virando um recurso inútil para eles.
- Torre destruída: peso 1, considerando que nesse momento do jogo, o perigo que as torres representam é extremamente baixo tendo em vista seu baixo dano causado a jogadores. Com isso, são facilmente destruídas.
- Abate oponente: peso 3, visto que o tempo de renascimento de um jogador morto é grande, a equipe adversária consegue uma boa vantagem para tentar alcançar objetivos como torres, inibidores, dragões ou baron e consecutivamente acabar com a partida. Nesse momento, a experiência e ouro obtidos ao realizar um abate são inúteis, tendo em vista que os jogadores já podem estar totalmente equipados e terem alcançado o nível mais alto dentro da partida.

3.2.2 Coleta manual

A primeira coleta de dados foi realizada de forma manual. Isso se deve ao fato da API oficial do *LOL* não disponibilizar os dados das partidas profissionais. Dessa forma, 5 vídeos do

canal oficial do Youtube ² foram utilizados para a inclusão dos dados no banco de dados. Os dados das partidas extraídas manualmente são mostradas na Tabela 2:

Tabela 2 – Valores etapas do jogo

Partida	Time A	Time B	Tempo de Jogo	Vencedor	Link
1	Flamengo	Netshoes Miners	39:49	Flamengo	Link
2	Netshoes Miners	Kabum	42:45	Netshoes Miners	Link
3	INTZ	Furia	30:14	INTZ	Link
4	Rensgar	Furia	39:07	Furia	Link
5	Netshoes Miners	Vorax	30:58	Vorax	Link

Fonte: Autoria Própria.

Além do tempo de jogo, para cada time de cada partida, foram extraídas as seguintes informações:

1. Número de jogadores abatidos;
2. Ouro total; e
3. Número de torres destruídas;

Um histórico dos jogos de uma equipe foi obtido de forma manual no MT-1 para se ter uma análise mais assertiva sobre os dados, assim diminuindo a chance de resultados discrepantes. Os jogos foram obtidos em formato de vídeos para coleta de dados. Isto se faz necessário já que API do *LOL* não retorna os dados de partidas profissionais. Foram coletados inicialmente 30 vídeos de partidas profissionais de *LOL* (Figura 3). Dessas partidas, apenas 5 foram selecionadas e seus dados foram extraídos manualmente a cada intervalo de um minuto do começo até o fim da partida..

3.3 MT-2: Criação da ferramenta de extração

Como citado anteriormente, a API do *LOL* não fornece os dados das partidas oficiais para que possam ser utilizados no projeto. Para extrair os dados das partidas, é necessário um trabalho manual de visualização dos vídeos e extração manual dos dados. Essa extração manual é demorada e suscetível a erros.

Por esse motivo, o módulo de trabalho 2 consiste em otimizar essa tarefa de extração de dados de partidas profissionais do *LOL*. Para isso, uma ferramenta de extração foi desenvolvida para realizar a extração automática de dados da partida a partir dos vídeos das partidas do *LOL*.

A ferramenta de extração foi desenvolvida em linguagem de programação Python, juntamente com a biblioteca *opencv* e é composta por 3 fases representada pela Figura 11, apresentadas a seguir.

² <https://www.youtube.com/c/CBLLOL>

1) Captura de um frame da tela: Nessa fase, a ferramenta inicia a execução do vídeo e faz a captura de um frame da tela do modo espectador do jogo. Os vídeos utilizados nessa fase precisam estar em um formato de 720P com uma resolução de 1280 x 720.

2) Tratamento da imagem: Nessa fase, os quadros (a) e (d) da Figura 11 apresentam os frames onde ocorre o tratamento da imagem para considerar somente a parte da imagem onde há as informações sobre o jogo. A parte destacada em vermelho representa o que será utilizado para a comparação no passo 3. Os quadros (b) e (e) da Figura 11 apresentam as imagens presentes em um diretório previamente construído e que consta com 1522 imagens extraídas de outras partidas. Cada imagem no diretório foi classificada e tem os seus valores salvos em pastas dentro da pasta principal do projeto.

Figura 11 – Análise de similaridade entre imagens.



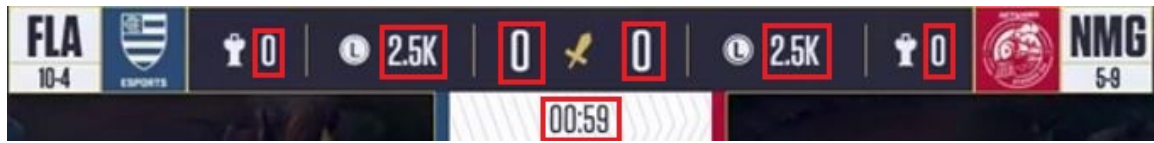
Fonte: Autoria Própria.

3) Aplicação da técnica de processamento de imagem subtract: Nessa fase, a biblioteca opencv é utilizada para aplicar a técnica de Image subtract. A técnica de subtract é utilizada para coletar a diferença de pixels em cada posição entre a imagem de entrada, que se deseja extrair os dados, e uma imagem modelo, a qual os dados são conhecidos. A técnica subtract calcula então a diferença entre as duas imagens pixel-a-pixel e depois é feita a conversão da imagem resultante dessa comparação em tons de cinza usando o método cvtColor. Com essa conversão, então são validados quantos pixels estão acima de uma determinada escala de cor. Caso a quantidade de pixels for maior que 30, número obtido depois de diversos testes para que se englobasse um número grande de imagens como corretas sem deixar muitos pixels diferentes das imagem de modelo, as imagens são consideradas iguais, caso contrário, são consideradas imagens diferentes.

O resultado do processo citado acima é apresentado nos quadros (c) e (f). No quadro (c) é possível visualizar o resultado da comparação das imagens dos quadros (a) e (b). O resultado é uma imagem predominante preta, indicando que as imagens comparadas têm grandes probabilidades de serem iguais. Porém, se executarmos a mesma comparação entre os quadros (a) e (e), o resultado será diferente, conforme mostrado no quadro (f) que apresenta uma grande quantidade de pixels, ficando nítido vários tons diferentes de preto. Isso indica que as duas imagens comparadas têm alta probabilidade de serem diferentes. Esse processo é feito para todas as 7 informações que se deseja extrair, como mostra a Figura 12.

A função que realiza essa comparação é apresentada no Algoritmo 1.

Figura 12 – Informações analisadas.



Fonte: RIOT Games.

Listagem 1 – Comparacao Imagens

```

1 def comparacaoImagem(image1, image2, diferenca=30):
2     global listCountPixel
3     listCountPixel = []
4     diff = cv2.subtract(image1, image2)
5     diff = cv2.cvtColor(diff, cv2.COLOR_BGR2GRAY)
6
7     rows, cols = diff.shape
8     count = 0
9
10    cv2.imwrite(r"C:\Users\levis\Desktop\tcc" +
11              "\\RESULTADOCOMPARACAO.jpg", diff)
12
13    for i in range(rows):
14        for j in range(cols):
15            if diff[i, j] > 15:
16                count += 1
17
18    listCountPixel.append(count)
19    if count > diferenca:
20        return False
21
22    return True

```

Fonte: Aatoria própria (2023).

Após a execução, foram aplicados testes para verificar a eficácia da ferramenta de extração dos dados. Para verificar a aplicação, os 5 vídeos utilizados na fase MT-1 foram utilizados como entrada e a saída obtida foi comparada com os dados obtidos manualmente. Um conjunto de 30 frames escolhidos de forma aleatórias entre todos os que foram extraídos desses 5 vídeos foi utilizado para verificar a eficácia da aplicação em extrair corretamente os dados presentes nas imagens. Cada um dos frames foi submetido aos 3 passos da aplicação e ao final da execução, foi verificado se o resultado salvo no banco de dados é igual ao valor mostrado na imagem. Como resultado, a ferramenta conseguiu identificar os dados de 80% dos frames utilizados. os 20% restantes foram considerados errados, pois o número de pixels apresentados no final da ferramenta passou do limite estipulado.

3.4 MT-3: Incremento da base de dados

O terceiro módulo de trabalho consiste em incrementar a base de dados criada no MT-1. Utilizando a ferramenta de extração criada no módulo anterior, foi possível incrementar a base de dados de forma automatizada e mais rapidamente. Nessa fase, a ferramenta fez a extração de dados de 10 partidas. O processo de coleta tornou-se simples, pois todos os dados a serem extraídos ficam sempre no mesmo lugar e seguem um padrão de aparência.

A classe de acesso ao banco de dados foi criada para a persistência dos dados extraídos. Ela utiliza a biblioteca `psycopg2` para criar uma conexão com o banco de dados passando dados de identificação de usuário do banco juntamente com o host e porta em que o banco de dados está hospedado.

Por meio dessa classe, é possível executar as operações como adicionar, atualizar e obter dados do banco. O trecho de código apresentado no Algoritmo 2 mostra o processo de obter os dados do time na tabela de times através do seu nome e depois faz a inserção dos dados na tabela que relaciona a equipe com a partida.

Com os valores das variáveis definidas, foi utilizado o MT-2 para fazer as extrações dos dados dos pontos mostrados na Figura 12 que são do modo espectador das partidas profissionais do *LOL*. Essas extrações ocorrem a cada intervalo de 1 minuto, e todos os seus retornos são salvos no banco de dados. Porém nem sempre é possível identificar no frame qual é o valor das informações. Isso ocorre, por exemplo, quando o placar usado pela transmissão da partida para o modo telespectador sobrepõe o mapa do jogo, com isso, pode ocorrer que pixels fiquem mais fortes ou mais fracos, dependendo da área que foi sobreposta pelo placar. Nesse caso, o valor 0 é persistido no banco de dados. Como sabemos que esse valor está incorreto, o valor `false` é inserido no campo “valor correto” da tabela “Informação partida”. Com isso, esses valores são desconsideradas na hora de realizar as previsões de vitória em MT-4.

3.5 MT-4: Criação da fórmula de predição

Em MT-4, foi criada uma fórmula de predição para realizar a predição de vitória de um jogo de *LOL* usando os dados de um momento do jogo. Na fórmula de predição, cada variável foi planejada considerando o conhecimento dos autores do trabalho em relação ao jogo. Essa fórmula realiza três cálculos: 1) Resultado atual da equipe, 2) Porcentagem Partida Vencedoras Abaixo e 3) Distância de Pontuação em Relação a Média.

3.5.1 Resultado atual da equipe (RAE)

A fórmula de predição recebe como entrada as informações de uma partida em um determinado momento. Esses valores são obtidos da ferramenta de extração apresentada em

Listagem 2 – Conexão de banco de dados

```

1 import psycopg2
2
3 connection = psycopg2.connect(user="postgres",
4                               password="postgres",
5                               host="192.168.99.100",
6                               port="5432",
7                               database="TCC")
8 cursor = connection.cursor()
9 # closing database connection.
10 psycopg2.extras.register_uuid()
11
12 sql_select_query = """
13 SELECT e."Id"
14 FROM "Equipe" e
15 WHERE e."Nome" = %s
16 """
17 cursor.execute(sql_select_query, (nomeEquipe,))
18 record = cursor.fetchone()
19
20 postgres_insert_query = """
21 INSERT INTO "PartidaEquipe" ("Id", "PartidaId", "EquipeId")
22 VALUES (%s, %s, %s)
23 """
24 record_to_insert = (partidaEquipeId, partidaId, record[0])
25 cursor.execute(postgres_insert_query, record_to_insert)
26
27 connection.commit()

```

Fonte: Autoria própria (2023).

MT-2. Os valores recebidos são: Tempo de Jogo, Número de jogadores abatidos, Ouro total e Número de torres destruídas.

A partir desses valores, é feito o cálculo do resultado atual da equipe. Esse cálculo usa a mesma fórmula apresentada na Equação 1. *rae* representa o desempenho de uma equipe no determinado momento em que a captura do frame foi feita.

3.5.2 Porcentagem partida vencedoras abaixo (PPVA)

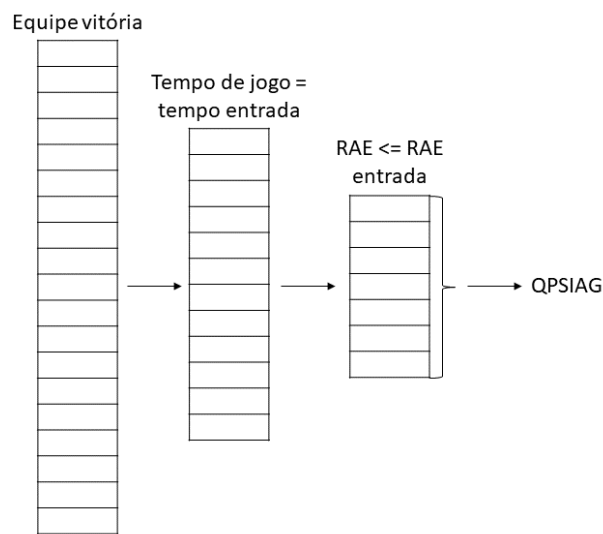
O segundo cálculo é referente à Porcentagem Partida Vencedoras Abaixo. *ppva* tem como objetivo identificar, para o conjunto de entrada qual é a porcentagem de partidas no banco de dados que tem o valor da coluna ResultadoSomaVariaveis na tabela InformaçãoPartida menor ou igual ao valor de RAE obtido na extração. Essa fórmula considera duas variáveis: *qpsiag* e *qtp*. A Porcentagem Partida Vencedoras Abaixo é calculada pela Equação 2.

$$PPVA = (QPSIAG/QTP) * 100 \quad (2)$$

Quantidade partidas de soma igual ou abaixo de ganhadas (QPSIAG)

Essa variável corresponde à quantidade partidas de soma igual ou abaixo de ganhadas. Esse valor é obtido através de uma busca no banco de dados considerando as entradas das equipes que obtiveram vitória e considera o mesmo período de jogo da entrada. Para essas entradas, é feita a comparação entre o atributo ResultadoSomaVariaveis que representa RAE e o valor do RAE da entrada, calculado anteriormente. O valor da variável consiste no número de entradas no banco em que o time possui o RAE menor ou igual ao RAE da entrada. A Figura 13 representa como o valor dessa variável é retirado do banco de dados e o Algoritmo 3 apresenta a consulta SQL realizada para retornar o resultado da variável.

Figura 13 – Seleção do valor de QPSIAG



Fonte: Autoria Própria.

Listagem 3 – Obtenção QSPIAG

```

1 (SELECT count(ip2."Id")
2 FROM "InformacaoPartida" ip2
3 JOIN "PartidaEquipe" pe2 ON ip2."PartidaEquipeId" = pe2."Id"
4 JOIN "TempoJogo" tp2 ON ip2."TempoId" = tp2."Id"
5 WHERE tp2."Id" = %s AND ip2."ValorCorreto"
6 AND pe2."ResultadoPartidaEquipe" = 'Vitoria'
7 AND ip2."ResultadoSomaVariaveis" <= %s )
8 AS quantidadePartidasSomaIgualAbaixoGanhada

```

Fonte: Autoria própria (2023).

Quantidade Total de Partidas (QTP)

Essa variável corresponde à quantidade total de partidas. Esse valor é obtido através de uma busca no banco de dados que conta todas as partidas gravadas de acordo com o tempo de jogo analisado, conforme demonstrado no Algoritmo 4.

Listagem 4 – Obtenção QTP

```

1 (SELECT count(ip1."Id")
2 FROM "InformacaoPartida" ip1
3 JOIN "PartidaEquipe" pe1 ON ip1."PartidaEquipeId" = pe1."Id"
4 JOIN "TempoJogo" tp1 ON ip1."TempoId" = tp1."Id"
5 WHERE tp1."Id" = %s
6 AND ip1."ValorCorreto")
7 AS qtdTotalPartidas

```

Fonte: Autoria própria (2023).

3.5.3 Distância de pontuação em relação a média geral (DPRM)

O terceiro cálculo da fórmula é referente à distância de pontuação em relação a média geral. *dprm* tem como objetivo encontrar a distância do valor de RAE que a equipe está no tempo de jogo analisado em relação ao média geral de RAE das equipes ganhadoras das partidas armazenadas no banco de dados nesse mesmo tempo de jogo, obtendo assim uma análise da performance geral da equipe em relação a partidas que já ocorreram. Esse valor é obtido por meio da Equação 3.

$$DPRM = RAE - MREV \quad (3)$$

Média do resultado das equipes vencedoras (MREV)

Essa variável corresponde à média do resultado das equipes vencedoras. é obtido através de uma busca no banco de dados selecionando a média de todas as partidas gravadas de acordo com o tempo de entrada analisado e que tenham ganhando a partida, conforme demonstrado no Algoritmo 5.

Listagem 5 – Código referente ao resultado das equipes vencedoras

```

1 (SELECT AVG(ip3."ResultadoSomaVariaveis")
2 FROM "InformacaoPartida" ip3
3 JOIN "PartidaEquipe" pe3 ON ip3."PartidaEquipeId" = pe3."Id"
4 JOIN "TempoJogo" tp3 ON ip3."TempoId" = tp3."Id"
5 WHERE tp3."Id" = %s
6 AND ip3."ValorCorreto"
7 AND pe3."ResultadoPartidaEquipe" = 'Vitoria')
8 AS mediaResultadoEquipesVencedoras

```

3.5.4 Exemplo de aplicação

Para exemplificar a utilização da fórmula de predição, um cenário será demonstrado a seguir. Para uma equipe que queremos verificar a pedição de vitória, a fórmula recebe como valores de entrada do jogo as seguintes informações por meio da ferramenta de extração:

- Tempo = 01:59
- Abates = 0
- Ouro = 2.5
- Torre = 0

O primeiro cálculo a ser feito é o do valor de RAE que representa o desempenho da equipe no determinado momento do jogo. Como o tempo do valor do tempo é 01:59, utilizam-se os pesos referente ao Early Game.

$$RAE = (abate * 3) + (ouro * 2) + (torre * 1)$$

$$RAE = (0 * 3) + (2.5 * 2) + (0 * 1)$$

$$RAE = 5$$

Em seguida, os cálculos a serem feitos são os da QPSIAG e QTP. Para esse exemplo, considere que a base de dados tem 5 partidas em que o time foi vitorioso. Dessas 5 partidas, no intervalo de tempo analisado, entre 01:00 e 01:59, 4 partidas têm os valores do atributo “ResultadoSomaVariáveis” da tabela “Informação partida” com o valor igual a variável REA calculada para os dados de entrada. Dessa forma, o valor da quantidade partidas de soma igual ou abaixo de ganhada é:

$$QPSIAG = 4$$

E o valor da quantidade total de partidas é:

$$QTP = 5$$

O próximo cálculo é o da MREV. Das 5 partidas consideradas no banco no tempo de partida sendo tempo da entrada, 4 têm o valor do “ResultadoSomaVariáveis” igual o valor do REA da entrada. Para a partida que não segue esse padrão, o valor do atributo “ResultadoSomaVariáveis” é igual a 6. Então, o cálculo para obter o valor de MREV é:

$$MREV = (5 + 5 + 5 + 5 + 6)/5$$

$$MREV = 5,2$$

Com essas variáveis definidas, é possível calcular o valores de PPVA e DPRM que fazem parte da fórmula de predição, as quais trazem as informações finais da predição. Para esse exemplo o valor da Porcentagem Partida Vencedoras Abaixo é calculado:

$$PPVA = (QPSIAG/QTP) * 100$$

$$PPVA = (4/5) * 100$$

$$PPVA = 80$$

A partir desse valor, a fórmula de predição retorna que a chance dessa equipe analisada tem de vencer é de 80% nesse momento específico do jogo.

Por fim, calculamos a distância de pontuação em relação à média geral (DPRM).

$$DPRM = RAE - MREV$$

$$DPRM = 5 - 5,2$$

$$DPRM = -0,2$$

Como resultado, a diferença de pontuação que o time analisado tem no momento da partida comparado com a média das equipes que ganharam as partidas é de -0,2. Isso significa que a equipe de forma geral está atrás em pontuação das equipes que ganharam os jogos das outras partidas já finalizadas.

3.6 MT-5: Criação da ferramenta de predição

Com todos esses passos realizado, uma ferramenta foi desenvolvida (MT-5) com o objetivo de fornecer suporte à aplicação da fórmula de predição em jogos do *LOL*. A ferramenta recebe como entrada um link de um video da plataforma Youtube³, então basicamente se coloca o link do video na aplicação e a executa.

A ferramenta possui 3 atividades principais. A primeira consiste em identificar o tempo 00:59 da partida. Nessa atividade, a ferramenta recebe como entrada o frame da captura do vídeo e, por meio da ferramenta de extração de informações (MT-2), extrai as informações da tela. Esse processo acontece a cada 1 segundo até se achar o frame que referente ao valor 00:59. Ocorre de a ferramenta não conseguir identificar exatamente o valor 00:59, assim podendo pegar valores anteriores a ele. Quando isso ocorre, apesar de não ser o valor desejado, a ferramenta ainda continua com seu funcionamento padrão. Agora, caso nenhum valor seja encontrado, a ferramenta vai continuar em execução, porém não vai fazer mais nenhum dos processos seguintes dela.

A partir disso, a ferramenta faz a captação do frame a cada intervalo de um minuto e faz a extração das informações. Com as informações, ocorre o cálculo da fórmula de predição (MT-4). Na fórmula de predição, é calculado o valor do RAE para cada uma das equipes considerando os valores de abate, ouro e torre considerando o peso referente ao tempo da partida.

Em seguida, por meio de consultas ao banco de dados utilizando a linguagem de SQL, são feitas a contagem de quantas partidas já foram coletadas (QTP), quantas partidas o vencedor tinha uma pontuação menor do que a do time analisado atualmente (QPSIAG) e a média da pontuação de todos os vencedores das partidas (MREV). Esse processo se repete a cada intervalo de um minuto e dura até o encerramento da partida. Com essas informações, são realizados os cálculos do valor da Porcentagem Partida Vencedoras Abaixo (PPVA) e da distância de pontuação em relação à média geral (DPRM).

³ <https://www.youtube.com/>

Com esse cálculos realizados, a Figura 14 mostra o que é exibido como saída da ferramenta. Para cada equipe é apresentada a porcentagem de chance de vitória com base nas partidas que já ocorreram, a cor que está sendo representada pelo time e qual é a diferença da pontuação da equipe atual em relação a pontuação média das equipes vencedoras.

Figura 14 – Saída da ferramenta.

```

-----AZUL-----
Resultado equipe azul 27.0
Porcentagem de vitória: 36.36
Diferença pontuação atual em relação a média da pontuação vencedora: 2.36
-----VERMELHO-----
Resultado equipe vermelha 23.8
Porcentagem de vitória: 31.82
Diferença pontuação atual em relação a média da pontuação vencedora: -0.48

```

Fonte: Autoria Própria.

Durante a execução do programa no vídeo, a ferramenta de predição calcula e apresenta os resultados sempre que se passa o intervalo de 1 minuto, e durante todo o tempo em que a partida esta ocorrendo, sendo finalizada quando a partida acaba.

3.7 Discussões

Ambas ferramentas de extração de valor da tela quanto a da fórmula de predição conseguiram se mostrar eficazes em suas tarefas. A de extração possui uma confiabilidade de 80% em suas extração, no qual se da por imagens que o gral de confiabilidade fica fora do valor pre-estabelecido em virtude do alto numero de pixels diferentes da imagem utilizada como base da comparação.

Já a de predição mostrou uma taxa de acerto de 70%, demonstrando uma taxa aceitável para confiança. Um fator que influencia bem nesses 30% de erro, é a falta de conseguir identificar os valores da tela com a ferramenta de extração em momentos finais do jogo, com isso, não é possível ter a conclusão real da partida, tornando a predição um valor incorreto.

As ferramentas podem ser utilizadas por qualquer pessoa, com um intuito de conseguir um melhor entendimento de como funciona uma partida de *LOL* e uma melhor satisfação com o entretenimento que ela possa trazer. Os conceitos aqui utilizados podem ser estendidos para outros jogos, já que ambas as ferramentas podem ser adaptáveis para outros contextos, sendo necessário apenas modificar a base de dados e a fórmula de predição que se quer utilizar.

4 ESTUDO DE CASO

Esse capítulo apresenta um estudo de caso no qual a ferramenta de predição de vitória é aplicada. Dessa forma, a Seção 4.1 apresenta o contexto em que o estudo de caso foi feito como também as especificações do que foi utilizado para se fazer um ambiente pseudo controlado para mostrar a aplicação. A Seção 4.2 mostra como foi feita a execução do estudo de caso. A Seção 4.3 apresenta os resultados obtidos no estudo de caso. Já a Seção 4.4 apresenta ideias e pontos que podem ser explorados em possíveis trabalhos futuros visando melhorar ainda mais a aplicação aqui demonstrada.

4.1 Planejamento

O objetivo desse estudo de caso é demonstrar a eficácia da ferramenta de predição de vitória desenvolvida nesse projeto. Para isso, um estudo de caso será conduzido de forma a analisar se os resultados da predição de vitória apresentados pela ferramenta condizem com o resultado final do jogo. Como resultado será obtida a porcentagem de acerto da ferramenta durante a sua execução.

A partir do objetivo, pode-se formular a seguinte questão de pesquisa: "Qual é a eficácia da ferramenta em prever corretamente o resultado do jogo?". Para analisar o resultado, deve-se coletar todas as predições e comparar com o resultado final do jogo. Quanto mais a ferramenta prever que o time vencedor vencerá a partida, melhor é a eficácia da ferramenta.

Para esse estudo de caso, foram selecionadas 10 partidas do jogo *LOL* disponíveis na plataforma de vídeo Youtube. As partidas foram selecionadas no canal do YouTube do campeonato brasileiro de *LOL*¹. Os vídeos apresentam qualidade 720p com uma resolução de 1280 x 720. A duração de cada vídeo varia, pois a partida pode durar o tempo que for necessário para uma equipe conseguir a vitória. A seleção dos vídeos foi feita usando uma amostragem aleatória.

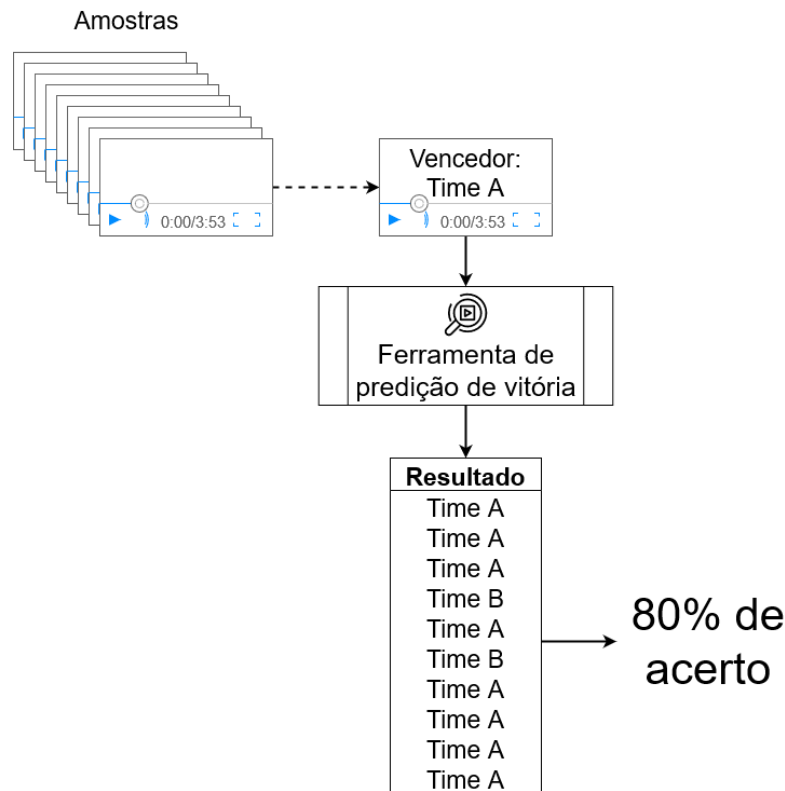
O projeto do estudo de caso é apresentado na Figura 15 e segue os seguintes passos:

1. Um vídeo é selecionado do conjunto de amostras.
2. A ferramenta de predição é executada e a predição de vitória é salva em uma base de dados. Durante a execução da ferramenta, os dados da partidas são coletados a cada 1 minuto e é feito o cálculo de predição de vitória para ambos os times. A ferramenta mostra a chance de vitória da equipe no tempo analisado e a diferença de pontuação da equipe, em relação a média das equipes vencedoras em partidas que já foram analisadas anteriormente.

¹ <https://www.youtube.com/c/CBLOL>

- Após a finalização da execução é feita uma análise dos resultados obtidos pela execução da ferramenta. Nesse passo, é calculado a eficácia da ferramenta, ou seja, quantas previsões corretas ela fez.

Figura 15 – Projeto do estudo de caso



Fonte: Autoria Própria.

4.2 Execução

Para a execução do estudo de caso o mesmo ambiente foi utilizado para executar a ferramenta em todos os vídeos. O ambiente de execução utilizado foi um notebook com as seguintes configurações: processador I5 de 8ª geração, placa de vídeo GTX 1050TI, com 24GB de memória RAM.

Cada um dos 10 vídeos de partidas do campeonato brasileiro do jogo *LOL* selecionados para a amostra foi executado uma única vez com a ferramenta de predição de vitória. Para cada vídeo analisado, os dados foram extraídos da tela da partida usando a ferramenta de extração apresentada em MT-2 e sendo manipulados pela fórmula de predição apresentada em MT-4. Os valores obtidos em uma partida foram salvos no banco de dados e, dessa forma, já foram utilizadas no cálculo de predição das partidas seguintes.

Após a execução da ferramenta com o vídeo, foi calculado o percentual de acerto da predição. A obtenção dos resultados foi feita de forma manual, através da análise dos dados gravados no banco de dados.

4.3 Resultados

Os dados coletados durante a execução do estudo de caso são apresentados na Tabela 3. Nessa tabela é possível observar que a ferramenta obteve um percentual de acerto acima de 80% em todas as partidas. Ou seja, para cada partida, a ferramenta conseguiu prever corretamente o vencedor da partida em no mínimo 80% das vezes. Também é possível observar que a ferramenta conseguiu alcançar 95% de eficácia na Partida 4.

Tabela 3 – Análise de acertos de predição total nas partidas.

	Predições certas na partida
Partida 1	85%
Partida 2	90%
Partida 3	82%
Partida 4	95%
Partida 5	85%
Partida 6	80%
Partida 7	85%
Partida 8	85%
Partida 9	92%
Partida 10	90%

Fonte: Autoria Própria.

A ferramenta apresentou uma porcentagem de 80% em seu menor valor de acertos de predição nas partidas, este é um excelente valor que passa uma confiança elevada que a ferramenta consegue realmente predizer com precisão o time ganhador da partida.

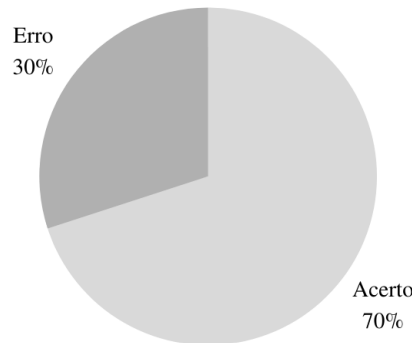
4.4 Discussões

4.4.1 Análise sobre predição final

Um ponto a ser discutido com relação à aplicação da ferramenta de predição de vitória é o resultado apresentado pela última predição. Essa análise é interessante, visto que nem sempre a ferramenta conseguiu prever corretamente a vitória no último momento da partida. Das 10 partidas, em 3 a ferramenta não conseguiu prever corretamente a vitória do time campeão no último momento do jogo. A Figura 16 apresenta o grafo com o resultado da eficácia da ferramenta no último momento do jogo.

Tal diferença entre acertos nas partidas ocorre pois a ferramenta utiliza a técnica *subtract* para a extração das informações. Dessa forma, foi observado que algumas vezes as informações não são extraídas corretamente da tela. Isso ocorre pois o placar usado pela transmissão da partida para o modo telespectador sobrepõe o mapa do jogo, com isso, pode ocorrer que pixels fiquem mais fortes ou mais fracos, dependendo da área que foi sobreposta pelo placar, assim afetando diretamente o resultado da análise de valor da imagem.

Figura 16 – Taxa de predição ao fim da partida.



Fonte: Autorial Própria.

Um ponto importante a ser analisado em relação às 3 partidas que não foi possível obter corretamente o time vencedor ao final da partida, é que em duas partidas, não foi considerado o resultado correto de que a equipe que a ferramenta apontou como vencedora nas últimas faixas de tempo analisadas de forma correta, realmente ganhou, pois, a última faixa de tempo a ser analisada não se obteve o resultado correto dos valores da tela com o processamento de imagem.

Já a terceira partida que teve o resultado diferente da predição, a equipe que estava atrás em todos os critérios analisados ganhou a partida, pois, por se tratar de um e-sport onde além de fatores de objetivos dentro da partida, a execução realizada pelos jogadores dos times ainda tem um enorme impacto na partida. Fazendo uma relação dessa situação com um esporte tradicional, seria o mesmo que em uma partida de futebol, no qual time A e time B estivessem de enfrentando, o time A tivesse todos os dados de chute a gol, escanteios, posse de bola a frente do seu adversário, porém o time B conseguisse a vitória mesmo tendo todos esses indicadores inferiores em relação ao seu adversário, pois conseguiu fazer um gol.

4.4.2 Análise por tempo de jogo

Outro critério importante a ser analisado é o número de vezes que a equipe vencedora tinha a pontuação maior que a sua adversária quando era realizado o cálculo de pontuação das equipes nos intervalos de um minuto. A Tabela 4 exibe período do early game que conta com 15 intervalos. Nota-se que das 7 partidas, 5 obtiveram mais de 50% de vitória nessa etapa do jogo. Isso mostra que as equipes que conseguem obter vantagens na maior parte do começo do jogo, tendem a sair vitoriosas das partidas.

Tabela 4 – Análise early game.

	Intervalos jogados	Intervalos ganhos	% Ganhos
Partida 1	15	13	86%
Partida 2	15	11	73%
Partida 3	15	03	20%
Partida 4	15	09	60%
Partida 5	15	06	40%
Partida 6	15	09	60%
Partida 7	15	08	53%

Fonte: Autorial Própria.

A Tabela 5 mostra a parte do mid game, é importante ressaltar que 3 das 7 partidas acabaram nessa etapa. Com isso, tem partidas que nem chegaram a entrar nos 20 intervalos de tempo disponíveis para análise. As partidas 1 e 2 exemplificam que quando se tem um grande domínio da parte de early game, em que ganharam em mais de 80% do tempo, a chance de obtenção da vitória de uma maneira mais rápida e dominante no mid game é grande.

Tabela 5 – Análise mid game.

	Intervalos jogados	Intervalos ganhos	% Ganhos
Partida 1	05	05	100%
Partida 2	09	09	100%
Partida 3	20	10	50%
Partida 4	20	15	75%
Partida 5	20	20	100%
Partida 6	17	13	76%
Partida 7	20	04	20%

Fonte: Autorial Própria.

No late game, como mostra a Tabela 6, das 4 partidas que chegaram a esse ponto, todas acabaram em tempos distintos. Na partida 5 a equipe vencedora teve um domínio de 100% já no mid game, e estendeu esse domínio para o late game, sendo a partida com o menor tempo de finalização nessa etapa. Já na partida número 4, a equipe mesmo não tendo dominado 100% do mid game, conseguiu também finalizar a partida de maneira rápida nessa e ganhando todas os intervalos disputados.

Tabela 6 – Análise late game.

	Intervalos jogados	Intervalos ganhos	% Ganhos
Partida 3	15	10	66%
Partida 4	05	05	100%
Partida 5	03	03	100%
Partida 7	10	03	30%

Fonte: Autorial Própria.

5 CONCLUSÃO

O público que assiste ao e-sport varia de pessoas que já acompanham o jogo quanto curiosos que estão a assistir pela sua primeira vez. Com isso a aplicação criada vem com intuito deixar essa experiência de assistir ao jogo melhor. Para isso ela exibe a cada 1 minuto a porcentagem de chance de vitória de cada time com base em informações que estão presentes na transmissão da partida e exibe na tela para que isso traga um maior entendimento e possa tornar ainda mais agradável e imersivo a experiências das pessoas em relação a partida que estão acompanhando.

Porém, a ferramenta nem sempre conseguirá prever o resultado final da partida com precisão. Isso se deve ao fato de que o jogo envolve fatores humanos: os jogadores que estão jogando a partida. É possível que a experiência dos jogadores faça o jogo virar, indo contra às predições de vitória apontadas pela formula. Mesmo nesses casos, a ferramenta ainda serve de um ótimo parâmetro de explicação para o entendimento das vantagens que algum time têm durante a partida.

5.1 Trabalhos futuros

Como trabalhos futuros pode-se colocar mais variáveis na formula de predição, como exemplo dragões abatidos por cada equipe. Isso pode melhorar o cálculo do RAE, uma vez que com mais dados que estão dentro da partida para ser analisados, os cálculos podem ser feitos com maiores precisões e ter resultados mais assertivos. Além de tentar colocar outra técnica para realizar a extração de valores da tela, tentando melhorar a taxa de confiabilidade em relação as informações que são obtidas.

Outra melhoria na ferramenta seria a criação de um módulo para análises mais detalhadas de informações para análise dos times. Com isso, além de servir para expectadores, os próprios times podem utilizar para conseguir informações e análises sobre como chegara as vitórias com informações mais precisas e reunidas em uma única ferramenta.

Por fim, há a possibilidade de melhorar o processo de extração de valores da tela, deixando a ferramenta mais assertiva e confiável, colocando mais imagens nos diretórios de comparação para se ter mais opções de comparação da mesma imagem.

REFERÊNCIAS

- ABREU, V. de. **League of Legends é o jogo de computador mais popular do mundo**. 2019. <https://www.techtudo.com.br/noticias/2019/09/league-of-legends-e-o-jogo-de-computador-mais-popular-do-mundo-entenda-esports.ghtml>. Acesso em: 14 de agosto. de 2020.
- BRITO, R. W. Bancos de dados nosql x sgbd's relacionais: análise comparativa. **Faculdade Farias Brito e Universidade de Fortaleza**, 2010.
- BULGAREVICH, D. S. *et al.* Pattern recognition with machine learning on optical microscopy images of typical metallurgical microstructures. **Scientific reports**, Nature Publishing Group, v. 8, n. 1, p. 1–8, 2018.
- CONSULTING, F. **Esports to Break the Billion-Dollar Barrier in 2020**. 2019. <https://www.futuresource-consulting.com/the-source-newsletter/esports-to-break-the-billion-dollar-barrier-in-2020/>. Acesso em: 14 de agosto. de 2020.
- COSTA, L.; SOUZA, A.; SOUZA, F. C. An approach for team composition in league of legends using genetic algorithm. *In: . [S.l.: s.n.]*, 2019.
- DATE, C. J. **Introdução a sistemas de bancos de dados**. [S.l.]: Elsevier Brasil, 2004.
- DUDA, R. O.; HART, P. E.; STORK, D. G. **Pattern classification**. [S.l.]: John Wiley & Sons, 2012.
- ERPEN, L. R. C. Reconhecimento de padrões em imagens por descritores de forma. 2004.
- FERREIRA, A. d. S. Redes neurais convolucionais profundas na detecção de plantas daninhas em lavoura de soja. 2017.
- GASPARETO, O. B. Redes neurais artificiais aplicadas ao reconhecimento de speed cheating em jogos online de computador. 2008.
- GONZALEZ, R. C.; WOODS, R. E. **Processamento de imagens digitais**. [S.l.]: Editora Blucher, 2000.
- HUANG, T.-M.; KECCMAN, V.; KOPRIVA, I. **Kernel based algorithms for mining huge data sets**. [S.l.]: Springer, 2006. v. 1.
- ISASI, A. G.; ZAPIRAIN, B. G.; ZORRILLA, A. M. Melanomas non-invasive diagnosis application based on the abcd rule and pattern recognition image processing algorithms. **Computers in Biology and Medicine**, Elsevier, v. 41, n. 9, p. 742–755, 2011.
- MAGALHÃES, L. B. *et al.* Análise comparativa dos algoritmos de otimização de consultas do postgresql. *In: Colloquium Exactarum*. ISSN: 2178-8332. [S.l.: s.n.], 2015. v. 7, n. 1, p. 01–21.
- NEWZOO. **Newzoo Global Games Market Report 2021 | Free Version**. 2021. <https://newzoo.com/insights/trend-reports/newzoo-global-games-market-report-2021-free-version>. Acesso em: 01 de julho de 2022.
- NEWZOO. **Newzoo's Global Esports Live Streaming Market Report 2022 | Free Version**. 2022. <https://newzoo.com/insights/trend-reports/>

newzoo-global-esports-live-streaming-market-report-2022-free-version. Acesso em: 05 de julho de 2022.

PAPA, J. P. *et al.* Classificação supervisionada de padrões utilizando floresta de caminhos ótimos. [sn], 2008.

SILVA, A. L. C.; PAPP, G. L.; CHAIMOWICZ, L. Continuous outcome prediction of league of legends competitive matches using recurrent neural networks. *In: SBC-Proceedings of SBCGames*. [S.l.: s.n.], 2018. p. 2179–2259.

SOUZA, F. J. d.; SANTANA, P. H. A. d. Estudo de caso: Análise entre banco de dados relacional e não relacional. 2017.

SOUZA, R. T. d. Aplicação de algoritmos classificadores para previsão de vitória em uma partida de league of legends. 2017.

TERUEL, V. P. *et al.* **Reconhecimento de padrões na investigação dos olhos da face humana aplicada em jogos**. 2019. Dissertação (Mestrado) — Universidade Tecnológica Federal do Paraná, 2019.

UCHIDA, S. Image processing and recognition for biological images. **Development, growth & differentiation**, Wiley Online Library, v. 55, n. 4, p. 523–549, 2013.

YAN, J.; RANDELL, B. A systematic classification of cheating in online games. *In: Proceedings of 4th ACM SIGCOMM workshop on Network and system support for games*. [S.l.: s.n.], 2005. p. 1–9.