

Análise de dados e *Machine Learning* para a elaboração de uma plataforma de avaliação de equipes e jogadores no Jogo *Valorant*

Leonardo Pinheiro de Souza¹, Lucas Paulo da Rocha¹,
Luiz Octavio Tassinari Saraiva¹, Pollyana Notargiacomo¹

¹Ciência da Computação
Faculdade de Computação e Informática
Universidade Presbiteriana Mackenzie
São Paulo – SP – Brasil

{10388961, 10391076, 10374379}@mackenzista.com.br, 1112605@mackenzie.br

Resumo. Os jogos FPS táticos e o cenário competitivo online têm crescido, impulsionando a profissionalização de e-atletas – jogadores patrocinados que se dedicam às carreiras em jogos digitais. Um aspecto importante nesse cenário é a extração e análise de dados que permitam compreender o desempenho dos jogadores (se estão em ascensão, estagnação ou declínio). Este trabalho apresenta a criação de uma plataforma de estatísticas que, utilizando dados do site Valorant Esports Coverage (<https://www.vlr.gg>), busca realizar análises detalhadas. O projeto visa também fornecer informações que apoiem avaliações no mercado de jogadores profissionais. Espera-se que os resultados possam ser aplicados por equipes ao redor do mundo, permitindo uma análise eficaz do desempenho de seus jogadores. Também espera-se que a plataforma construída possa ser adaptada para cenários semelhantes, assim como a base de dados criada possibilite o desenvolvimento de novos trabalhos na área de ciência de dados.

Palavras-chave: redes neurais, inteligência artificial, jogos FPS Táticos, análise de dados, mineração de dados.

Abstract. Tactical FPS games and the online competitive scene have been growing, driving the professionalization of e-athletes – sponsored players dedicated to careers in digital games. An important aspect in this context is the extraction and analysis of data that help understand player performance (whether they are improving, stagnating, or declining). This work presents the creation of a statistics platform that, using data from the Valorant Esports Coverage website (<https://www.vlr.gg>), aims to conduct detailed analyses. The project also seeks to provide insights that support evaluations in the professional player market. The results are expected to be applicable for teams around the world, enabling effective analysis of their players' performance. Additionally, the platform developed is expected to be adaptable to similar scenarios, and the created database should facilitate the development of new work in the field of data science.

Keywords: neural networks, artificial intelligence, tactical FPS games, data analysis, data mining.

1. Introdução

Valorant é um jogo de Tiro em Primeira Pessoa (*First Person Shooter – FPS*), desenvolvido pela Riot Games, lançado em 2020. Desde o início, conquistou uma ampla base de jogadores, destacando-se pela jogabilidade estratégica e pelo ritmo dinâmico. Como explicado no website *The Enemy* (DEOLINDO; PEREIRA, 2020), as partidas são divididas em rodadas, com o objetivo principal de vencer 13 delas para garantir a vitória. As metas variam de acordo com o lado do jogador: no time de ataque, é necessário plantar e proteger a *Spike* (uma espécie de bomba) até a explosão; no time de defesa, o foco é desarmá-la antes que exploda. A Figura 1 ilustra uma imagem do jogo.

Figura 1. Imagem de um *round* do jogo Valorant



Fonte: Elaborada pelos autores.

Em maio de 2024, o *Valorant* alcançou mais de 19 milhões de jogadores ativos, com uma base mensal de 28 milhões de jogadores, dos quais 70% pertencem à geração Z, (KNUDSEN, 2024). Esses números evidenciam a força contínua do jogo no mercado, mesmo diante da concorrência de títulos como Counter-Strike 2. Além do sucesso em número de jogadores, Valorant é uma fonte significativa de receita para a Riot Games. Em 2023, a empresa arrecadou mais de US\$ 66 milhões com a venda de *skins*, evidenciando o potencial financeiro do jogo no mercado competitivo. Esse montante inclui US\$ 32 milhões obtidos com a venda de *skins* do *Champions*, o principal evento competitivo do jogo, entre agosto e setembro de 2022, mostrando um crescimento substancial na arrecadação anual (PEREIRA, 2023).

A Riot Games também distribuiu mais de US\$ 33 milhões em receita de itens digitais para as equipes parceiras em 2023, sendo que cada equipe nas ligas parceiras recebeu, em média, mais de US\$ 1 milhão, (XU, 2023), refletindo a estrutura financeira e o compromisso da Riot Games em sustentar e fomentar o crescimento do cenário competitivo de Valorant. Essa combinação de uma base de jogadores ativa, crescimento financeiro e apoio contínuo ao cenário competitivo faz de Valorant um estudo de caso pertinente à aplicação de modelos preditivos.

Para isso, por meio da utilização de informações disponíveis sobre partidas registradas no website (vlr.gg, 2024), busca-se fornecer dados para análises retrospectivas, comparativas e avaliações sobre o cenário atual, além de prever resultados de partidas. Assim, pretende-se apresentar um *dashboard* que combine diferentes dados e visualizações para apoiar estratégias e análises dos times em relação aos jogadores e formação de equipes.

O estudo está organizado nas seguintes seções: a seção 2 apresenta o referencial teórico, explorando aspectos dos jogos, em especial do gênero FPS, além da captura de informações via *web scraping* e dos métodos de tratamento e análise de dados; a seção 3 detalha os materiais e métodos utilizados para desenvolver a aplicação proposta; a seção 4 consolida e discute os resultados obtidos; e a seção 5 apresenta as conclusões e trabalhos futuros.

2. Referencial Teórico

2.1. *First Person Shooter* (FPS)

Os jogos digitais vêm se tornando cada vez mais populares na *Internet*, indicando que cerca de 73,9% dos brasileiros jogam jogos eletrônicos, um aumento de 3,8% no número de jogadores em relação ao ano anterior (CARDOSO; GUSMÃO; HARRIS, 2023). Em 2023, o Brasil possuía mais de 100 milhões de jogadores de videogame, número que ultrapassa a quantidade combinada de usuários na Coreia do Sul, Reino Unido e Canadá. O Brasil representa 3% dos gamers mundiais, ficando atrás apenas dos Estados Unidos e China (NEWZOO, 2023). Consequentemente, o mercado brasileiro já está entre os 10 maiores do mundo, gerando uma receita anual de US\$ 2,6 bilhões (aproximadamente R\$ 12,66 bilhões)(SOUZA et al., 2021). Em contrapartida, a pesquisa da Indústria Brasileira de Games, realizada pela Associação Brasileira das Desenvolvedoras de Jogos Digitais entre 2014 e 2022 (CARDOSO; GUSMÃO; HARRIS, 2023), mostra que o número de desenvolvedoras nacionais cresceu cerca de 160%. Se no começo da década o Brasil tinha em torno de 150 empresas de games, em 2022 esse número ultrapassou mil empreendimentos.

Jogos do gênero FPS podem ser tanto competitivos quanto focados em história e configuram uma proposta imersiva em que o jogador visualiza uma perspectiva tridimensional em primeira pessoa, sendo que estes podem não só dialogar, como correr e atirar (dentre outras possibilidades) (LIU, 2024).

Em *Valorant*, antes da partida começar, o jogador seleciona o agente que deseja utilizar; cada agente possui um conjunto de quatro habilidades que podem ser usadas em diferentes situações. Os jogadores são divididos em uma equipe atacante e uma defensora, a primeira tem como principal objetivo plantar a chamada *spike*, a condição de vitória de um *round* para a equipe atacante é eliminar todos os adversários ou conseguir proteger a *spike* plantada até que a mesma detone, (DEOLINDO; PEREIRA, 2020). Enquanto isso, a equipe defensora deve evitar que a *spike* seja plantada, e caso isso ocorra, ela deve desarmar antes da explosão, caso os jogadores da equipe atacante sejam eliminados antes de armar a *spike* a vitória da rodada é da equipe defensora.

Uma possibilidade para compreender aspectos de clusterização e especificidades do jogo em questão, requer também um estudo das formas de captura dos dados disponíveis on-line.

2.2. Web Scrapping e Clusterização

A Internet possui milhares de informações que podem ser acessadas pelos usuários, mas o processo de captura manual costuma ser inviável devido ao tempo necessário. Para realizar essa busca automatizada, é necessário conhecer como

funciona a infraestrutura da Internet, na qual a informação parte de um ponto e alcança o dispositivo que fez a requisição (MITCHELL, 2018). Dentre os vários passos que envolvem a ordenação dos IP's e das rotas de rede, ocorre o envio de uma requisição e a resposta de um servidor a ela. Nos arquivos dentro dessa resposta, é possível procurar e interpretar as informações HTML solicitadas que compõem a tela de um website que foi acessado inicialmente.

O processo de *web scrapping* pode ser complementado pela clusterização. Inclusive, o uso de um *heatmap* para identificar as maiores correlações entre variáveis fornece uma base sólida para a aplicação de técnicas de clusterização, como o *k-means*. Por meio do *heatmap*, é possível destacar variáveis que possuem correlação entre si, o que permite identificar agrupamentos mais homogêneos ao utilizar o *k-means*, melhorando a eficiência e a relevância dos *clusters* formados (HAMID et al., 2010). Essa abordagem facilita a interpretação dos dados, uma vez que as variáveis selecionadas representam os principais fatores de variação no conjunto, otimizando a análise e gerando agrupamentos que refletem as características principais dos dados.

Inclusive, um estudo vinculado ao basquete (CHEN; ZHANG; XU, 2023) utilizou a clusterização de dados para agrupar jogadores e entender melhor seus padrões. A proposta consistia em aplicar a clusterização para construir um modelo de classificação baseado no tipo de jogo de cada jogador. Foram usados dados, de 2017 a 2021, de 20 times da Associação de Basquete Chinesa. O objetivo era explicar como diferentes tipos de papéis dos jogadores nas partidas influenciavam a performance dos times. Para isso, os jogadores nativos e estrangeiros foram agrupados em diferentes funções ofensivas, empregando o método *k-means*, escolhido por sua capacidade de organizar os dados em grupos. O número ideal de *clusters* foi determinado usando o método do cotovelo, que identifica o ponto em que a soma dos erros quadrados (SSE) dentro dos *clusters* se estabiliza, sugerindo o número apropriado de grupos. Esse processo resultou em quatorze papéis ofensivos para jogadores nativos e cinco para estrangeiros, permitindo uma investigação detalhada sobre o impacto desses papéis no sucesso da equipe.

2.3. Trabalhos Relacionados

A predição de resultados em jogos competitivos constitui uma ferramenta relevante para apostadores, analistas, jogadores e equipes. Em um estudo envolvendo equipes do jogo *League of Legends* (HITAR-GARCÍA; MORÁN-FERNÁNDEZ; BOLÓN-CANEDO, 2023), os autores usaram o teste de correlação de Pearson para identificar a Taxa de Vitória do Jogador-Personagem e a experiência do jogador com esse personagem como características-chave para prever os resultados das partidas. Essas características, juntamente com variáveis derivadas como média, mediana, curtose, assimetria, desvio padrão e variância das taxas de vitória e pontos de maestria, foram usadas como entrada para diversos modelos de predição, incluindo *Support Vector Classifiers*, *k-Nearest Neighbors*, *Random Forest*, *Gradient Boosting* e *Deep Neural Networks*.

Inclusive, os modelos de aprendizado de máquina, como *Random Forest* e *Support Vector Classifiers*, também retornam precisão com variação entre 73% e 75%. O modelo mais eficaz foi o de Redes Neurais Profundas (*Deep Neural Networks*),

desenvolvido com a API Keras, configurada com uma Camada de entrada achatada com saída 1×44 , alternância de camadas de *dropout*, normalização e densas, totalizando 15 camadas (5 *dropout*, 5 normalização e 5 densas) com 160, 128, 64, 32 e 16 neurônios, respectivamente, Camadas de *dropout* com taxa de 0,69%, normalização por lote em cada camada de normalização, Ativação ELU e inicialização He em cada camada densa, Camada densa final 1×1 com ativação Sigmoid; o que alcançou uma precisão de 75,1% na predição dos resultados das partidas ([HITAR-GARCÍA; MORÁN-FERNÁNDEZ; BOLÓN-CANEDO, 2023](#)).

Outro projeto relacionado ao tema de predição aplicou modelos de aprendizado supervisionado aos dados de partidas de *Valorant* ocorridas no torneio *VCT North America* 2021. Esses dados foram obtidos por meio da API disponibilizada pela Riot Games, coletando informações de partidas profissionais. As informações abrangiam 11 equipes participantes e incluíam dados das partidas, equipes, rodadas, indicadores KDA (eliminações, baixas e assistências), *kills* (informação detalhada dos confrontos), dados de economia, estatísticas dos jogadores, localizações e eventos. No total, o *dataset* utilizado no projeto possuía 1302 registros, representando dados de apenas uma das equipes que jogou cada partida, com 651 registros de vitórias e 651 de derrotas ([SANHUEZA, 2023](#)). Isso envolveu um mapa de calor para identificar que variáveis de indicadores de ataque e economia (KD, KDR, ACS, *damage*, *avgEconRating*, *loadoutValue* e FKD) possuem alta correlação positiva com a variável *won*. Por outro lado, as variáveis *deaths*, *firstDeaths*, *clutchOpportunities* e *clutchOpponents* apresentaram a maior correlação negativa, nesse caso, com a derrota. Essas *features* foram utilizadas como entrada para os modelos. No entanto, por utilizar variáveis geradas durante a partida, não foi possível usá-las diretamente no modelo; dessa forma, optou-se por utilizar as médias das últimas três partidas como variáveis preditoras. Para realizar as previsões, foram utilizados os modelos de *Machine Learning Random Forest* e *XGBoost*.

O modelo *Random Forest* é um meta-estimador que ajusta uma série de classificadores de árvores de decisão em várias subamostras do conjunto de dados, quando a média para melhorar a precisão preditiva e controlar o sobreajuste, ajustando os valores de hiperparâmetros para gerenciar o consumo de memória, a complexidade e o tamanho das árvores. Já o *XGBoost* é um método de reforço de gradiente distribuído, projetado para ser eficiente, flexível e portátil, e requer a definição de três tipos de hiperparâmetros: gerais, de reforço e de tarefas de aprendizado. Utilizando a métrica de avaliação F1-score, foram obtidos os resultados de 0.68 para o modelo *XGBoost* e 0.64 para o modelo *Random Forest* ([SANHUEZA, 2023](#)).

Outra abordagem sobre a predição de resultados de partidas de *Valorant* fez uso de dados de partidas apenas da temporada inaugural, especificamente do circuito oficial conhecido como *Valorant Champions Tour* (VCT), que é uma competição global composta por seis regiões: América do Norte, Brasil, América Latina, Leste Asiático, Ásia-Pacífico e EMEA (Europa, Oriente Médio e Norte da África). A temporada de 2021 inclui dados de 1.471 mapas jogados por mais de 200 equipes em 133 eventos do VCT, ocorridos de 18 de janeiro a 12 de dezembro. A coleta desses dados foi realizada por meio de três sites que agregam informações de todas as partidas profissionais do VCT e se complementam em caso de dados faltantes: *vlr.gg*,

rib.gg e liquipedia.net. Este trabalho apresenta uma forma de comparar a força de dois times utilizando informações derivadas do ELO dos mesmos. Inicialmente, o conjunto de dados continha 1.471 mapas, mas, devido à exclusão dos qualificatórios abertos, onde as equipes geralmente não possuem dados confiáveis de Elo e estatísticas de equipe, o conjunto de dados foi reduzido para 1.144 mapas. Para a variável de diferença de rodadas (*RoundDiff*), foi utilizada a eliminação reversa para determinar as variáveis para a regressão. A eliminação reversa envolve executar a regressão com todas as variáveis possíveis e eliminar a variável com o maior valor p . A regressão é reexecutada sem a variável de maior valor p da regressão anterior e continua até que todas as variáveis alcancem um nível de significância viável e uma positividade ou negatividade sensata em seus coeficientes. O mesmo conceito de eliminação reversa foi aplicado à regressão logística. Em ambos os modelos, os conjuntos de dados de treinamento e teste foram divididos em 75% e 25%, respectivamente. As variáveis que representam a equipe 1 (x) e a equipe 2 (y) devem estar inversamente relacionadas na regressão; p.e., se a porcentagem de vitórias da Equipe 1 aumentar, a probabilidade de vitória deve aumentar e a diferença de rodadas deve aumentar. Assim, quanto maior for o Elo do oponente em comparação com o Elo da própria equipe, menor será a diferença de rodadas e menor será a probabilidade de vitória da Equipe 1. Como conclusão, ele obteve uma média de mais de 4 *rounds* de diferença no modelo de regressão linear e uma precisão de 73% no modelo de regressão logísticas, responsável por prever o vencedor de uma partida (EDMOND, 2023).

3. Metodologia

A partir dos elementos dispostos anteriormente, foi realizado um estudo bibliográfico sobre os temas da pesquisa. Isso resultou na seleção da linguagem *Python*, que facilita a criação de ambientes virtuais, permitindo que as dependências dos projetos fiquem isoladas, evitando incompatibilidades de versões. Com o intuito de abstrair a complexidade desses processos, foram, então, utilizadas as bibliotecas da linguagem de programação *Python*: *urllib* e *BeautifulSoup*. A biblioteca *urllib* é capaz de realizar todo o processo que os navegadores realizam de preparar, enviar e receber pacotes de dados. Por meio da função *urlopen* e do parâmetro da URL desejada, a própria biblioteca realiza uma requisição para o servidor e prepara um objeto a ser utilizado pelo código que contém o arquivo HTML da página acessada. Além disso, é válido citar que existem outras bibliotecas que tem a mesma função da *urllib*, como a biblioteca *requests*. A diferença entre elas evidencia que a *urllib* é uma biblioteca nativa do *Python* e possui uma interface de mais baixo nível, sendo necessário ter mais controle das configurações e verbos anexados as requisições HTTP (GANESAN, 2024). Por outro lado, a biblioteca *requests* utiliza a *urllib*, mas aplica uma interface única de alto nível que lida com várias formas de *encoding* e abstrai configurações mais avançadas da requisição.

Já a biblioteca *BeautifulSoup* é capaz de facilitar as buscas e tratamentos feitos dentro de códigos HTML. Isso se deve à sua classe *BeautifulSoup*, que permite navegar pelas *tags* do arquivo utilizando comandos diretos. Essa simplificação do processo de leitura também permite a aplicação de expressões regulares para a limpeza de busca de palavras ou trechos específicos (MITCHELL, 2018).

Outra escolha tecnológica foi o *PostgresSql* e o *framework* Django, que foram usados para construir a aplicação *web* a partir dos dados coletados na etapa anterior. Apesar do Django usar por padrão o *SQLite*, por meio do *Object Relational Mapper (ORM)*, a configuração do banco de dados é simplificada, permitindo a utilização de outras opções, de bancos relacionais. No arquivo `settings.py`, as configurações essenciais são definidas, incluindo o banco de dados, os aplicativos instalados, e os arquivos estáticos e de *templates*. Além disso, o Django permite que cada aplicativo possa carregar seus próprios arquivos para a interface de usuário, permitindo a criação de telas personalizadas dentro do projeto (CHANDIRAMANI; SINGH, 2021).

Da mesma forma, os *endpoints*, que são os pontos de entrada para a interação com a aplicação *web*, são definidos no arquivo `urls.py` de cada aplicativo, mapeando URLs específicas para as respectivas visualizações. No `urls.py` do projeto principal, os *endpoints* dos aplicativos individuais são incluídos, organizando logicamente as URLs. Para definir os modelos de dados, o comando `from django.db import models` é utilizado, permitindo a criação de classes que representam as tabelas do banco de dados, com atributos que definem os campos e suas propriedades.

Além da infraestrutura oferecida pelo *framework* Django, é essencial preencher as páginas com informações relevantes para os usuários interessados. Com esse objetivo, buscou-se investigações que tratassem das principais bibliotecas de dados e análise de *Big Data*. O estudo selecionado (STANČIN; JOVIĆ, 2019) analisou as diferenças entre essas bibliotecas, destacando os principais usos de cada uma. As bibliotecas mais utilizadas para visualização de dados incluem *Plotly*, *Matplotlib*, *Seaborn*, *Bokeh* e *Ggplot*. *Bokeh* se destaca por ser a única biblioteca voltada exclusivamente para aplicações *web*, permitindo a criação de gráficos interativos em que os usuários podem explorar os dados de forma independente.

3.1. Web Scrapping

O conjunto de práticas descritas no item anterior foi usada para o processo de mineração de dados no site <<https://www.vlr.gg>>, o que envolveu o estabelecimento da conexão com as páginas desejadas para extrair os dados brutos. Em seguida, se fez uso de expressões regulares para analisar o HTML retornado, permitindo identificar e extrair as informações relevantes. Esse processo compreendeu a navegação e seleção de elementos específicos dentro das páginas. Para otimizar e acelerar a coleta de dados, foi implementado um sistema de paralelização que permite executar múltiplas requisições simultaneamente. Esse sistema foi desenvolvido para dividir a tarefa entre várias *threads*, cada uma responsável por acessar e coletar dados de diferentes conjuntos de partidas. Essa abordagem garantiu uma coleta de dados ágil e eficiente (MITCHELL, 2018; GANESAN, 2024).

A coleta de dados foi feita a partir da URL <<https://www.vlr.gg/matches/results/?page=>>, que ao receber um número em seu final, redireciona a requisição para páginas com listas de séries de partidas já concluídas. Dessa forma, é possível iterar sobre o total de páginas disponíveis e realizar uma busca pelos elementos que contém as informações desejadas, acessando os atributos necessários para coletar a URL de cada série e fazer outra requisição.

Na parte superior das páginas de partidas, é possível visualizar informações

básicas, como: nome dos times, placar, data da partida, tipo da série, nome do campeonato e o *patch* do jogo na época. Cada uma dessas informações foi capturada seguindo o processo de localização dos elementos relevantes no HTML. Ainda nessa página, se acessou a tabela de *Overview*, que guarda informações detalhadas separadas por partidas. Para capturar essas informações, se percorreu a tabela linha por linha, armazenando os dados de cada partida em estruturas organizadas. Apesar dos dados já estarem organizados, a tabela de *Overview* apresentou um desafio específico: se o tipo de partida for diferente de 'BO1', aparece a opção *all maps*, que traz uma média das estatísticas dos jogadores em todos os mapas da série. Esses valores foram descartados, pois podem ser calculados diretamente a partir das partidas individuais. A estrutura condicional foi implementada para evitar a coleta desses dados indesejados.

Durante a estruturação das informações coletadas, se aplicou tratamentos para lidar com cenários inesperados, como a ausência de uma informação em alguma tabela, atribuindo o valor *null* para aquele campo específico, ou aplicando a formatação necessária para alguns dados. Esse processo incluiu a utilização da classe *datetime* para os campos de data e a limpeza de valores indesejados, como '\n' e '\t'. Nos casos em que uma tabela estivesse completamente ausente, a série inteira seria descartada e sua *URL* seria salva em um arquivo de texto chamado `erros.txt`.

3.2. Armazenamento dos dados

Após a coleta dos dados, foi necessário criar uma estrutura que pudesse armazenar as informações de forma que fossem exportáveis para a etapa de tratamento dos dados. Para isso, se usou o processo de salvar dados em CSV ou em um banco de dados relacional (MITCHELL, 2018). Os dados obtidos foram enviados para um *DataFrame* da biblioteca *Pandas*, que disponibiliza um método para a criação automática de um arquivo CSV. Esse processo possibilitou a exportação dos dados para outras plataformas, mas ainda havia a necessidade de criar um banco de dados para ser utilizado na plataforma web de análise.

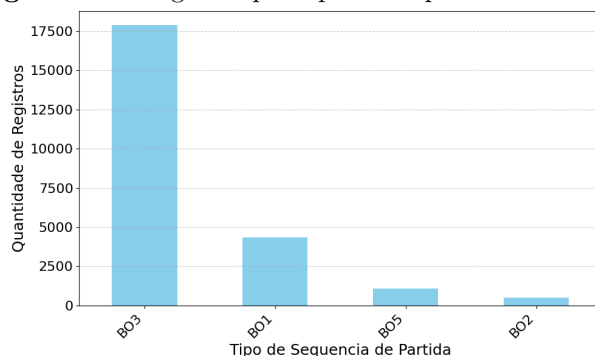
Após a instituição da instância do banco de dados, foi necessário desenvolver o *script* inicial para a criação das tabelas que armazenariam os dados. Dessa forma, foi criada uma aplicação denominada *adaptador-para-postgres*, que recebe como entrada o arquivo CSV gerado anteriormente e se conecta à instância do banco de dados para realizar automaticamente as inserções necessárias em cada tabela (PostgreSQL Global Development Group, 2023).

4. Resultados e Discussão

A última coleta de dados realizada no site <<https://www.vlr.gg>> ocorreu em 17 de outubro de 2024, abrangendo todo o meio competitivo disponível na plataforma. Vale ressaltar que os dados obtidos são aqueles que passaram pelos filtros da aplicação. Dentro desse conjunto, estão disponíveis informações de 4.439 times, 23.814 sequências de partidas, 51.329 partidas, 13.987 jogadores e 1.212 campeonatos. A distribuição de sequências de partidas por tipo pode ser visualizada na Figura 3, na qual a maioria dos dados se concentra em partidas do tipo BO3 (Best of 3, ou seja, Melhor de Três). Destaca-se que existem 9.047 partidas com

alguma informação faltante, com a inexistência de valores específicos e não de um conjunto inteiro de dados.

Figura 2. Histograma por tipo de Sequência de Partida



Fonte: Elaborada pelos autores.

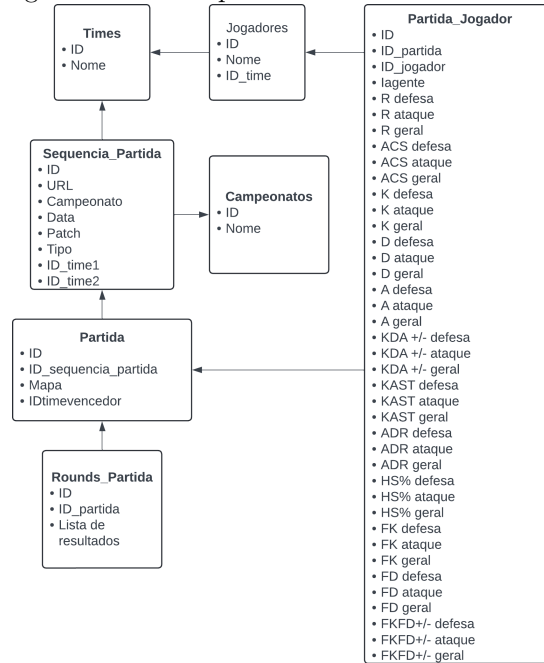
Concluída a exportação dos dados para o arquivo CSV, este foi importado no ambiente do Google Colab. Durante a análise, constatou-se que 3.906 séries de partidas apresentavam seções com dados faltantes; portanto, foram excluídas do conjunto de dados. Assim, manteve-se 19.908 séries de partidas, totalizando 42.285 partidas. Inicialmente, esses dados continham 14 colunas: a coluna *URL* refere-se ao endereço de onde os dados foram extraídos; a coluna *Campeonato* armazena o nome do campeonato ao qual a série pertence; a coluna *Data* registra a data em que a série foi jogada; a coluna *Patch* indica a versão do jogo na qual a série foi disputada; as colunas *Nome do Primeiro Time* e *Nome do Segundo Time* armazenam os nomes dos times que participaram da série; a coluna *Tipo Partida* define o número máximo de partidas que poderiam ser jogadas na série; a coluna *Mapas* lista os nomes dos mapas jogados durante a série; a coluna *Rounds* apresenta um catálogo de listas, onde cada sublista representa a forma como cada *round* foi vencido e o respectivo time vencedor.

A base construída conta com dados de jogadores de forma individual, os quais podem ser agregados para construir informações sobre históricos e desenvolvimentos de times. Essa estrutura permite análises detalhadas tanto no nível dos jogadores quanto das equipes, facilitando estudos sobre desempenho e evolução no cenário competitivo.

Além disso, a coluna *Estatísticas* contém listas de dicionários, cada dicionário armazena informações de cada jogador durante a série. Os dados de cada jogador incluem: *nome*, *agente*, *r*, *acs*, *k*, *d*, *a*, *kda+-*, *kast*, *adr*, *hs*, *fk*, *fd*, *fkfd+-* e *função* (sendo esta última atribuída no início do tratamento dos dados). Com exceção de *nome*, *agente* e *função*, que se mantêm constantes durante a partida, os demais valores numéricos são divididos entre *valor geral*, *valor na metade atacante* e *valor na metade defensiva*. Cada partida em uma série de jogos possui um dicionário específico para cada jogador. A coluna *Quadro All Kills*, por sua vez, contém uma matriz de confrontos diretos entre jogadores de times diferentes, detalhando, por exemplo, quantas vezes o jogador 1 do time 1 eliminou ou foi eliminado pelo jogador 1 do time 2, abrangendo todas as 25 combinações de confrontos possíveis entre os jogadores. Já a coluna *Quadro Jogadas* refere-se a outra matriz, na qual cada linha

representa um jogador e armazena informações como *nome* e *agente*, seguidos por dados de jogadas individuais, como o número de vezes que o jogador eliminou mais de um adversário em um único *round* (2k, 3k, 4k, 5k) e quantas vezes venceu rodadas sozinho contra múltiplos adversários, conhecido como *clutch* (1v1, 1v2, 1v3, 1v4, 1v5). Esta matriz ainda inclui dados sobre economia, o número de *spikes* plantados e o número de *spikes* desarmados. A modelagem de domínio pode ser visualizada na Figura 3.

Figura 3. Modelagem de domínio para armazenamento dos dados no PostgreSQL.



Fonte: Elaborada pelos autores.

A coluna *Quadro Compras*, por sua vez, apresenta um catálogo de listas que detalha informações sobre a economia dos times, especificando a quantidade de *rounds* em que cada time jogou com diferentes quantias de dinheiro e quantos desses *rounds* foram vencidos. Por fim, a coluna *Quadro Tipo Compra por Round* também é um catálogo de listas, mas este detalha, por *round*, o montante economizado para o próximo *round* e se os times realizaram algum tipo de economia. Essas colunas e suas respectivas descrições oferecem uma visão detalhada e abrangente dos dados analisados, sendo essenciais para a análise e interpretação dos resultados obtidos.

Com base na coluna *Estatísticas*, foram geradas duas colunas adicionais para separar as informações do *timeUm* e, em seguida, do *timeDois*. Além disso, foi incluída uma coluna *Vencedor*, destinada a definir o time vencedor com base na quantidade de *rounds* vencidos por equipe em cada partida, informação relevante para o treinamento dos modelos preditivos. O aumento no número de linhas dos dados também foi observado, pois as séries foram divididas em partidas únicas. Assim, uma série com três partidas jogadas passou a ter três linhas, permitindo que as informações específicas de cada partida fossem registradas separadamente.

No tratamento dos dados, foram desconsideradas séries e partidas sem informações completas, pois estas não poderiam ser utilizadas nas análises nem nas

predições. Além disso, os nomes dos jogadores originalmente vinham acompanhados da sigla do time em que jogaram a partida, então essa sigla foi removida para associar o histórico completo de cada jogador a ele, e não apenas às partidas em um time específico. Por exemplo, o jogador “aspas LLL” e “aspas LEV” refere-se ao mesmo jogador, mas jogando em diferentes times ao longo de sua carreira.

Três URLs específicas foram desconsideradas devido a registros incorretos na plataforma. O tipo de sequência da partida era incompatível com a quantidade de partidas jogadas. Embora estivessem marcadas como BO1 (*best of 1*, ou seja, uma única partida definia o vencedor), elas apresentavam mais mapas jogados.

4.1. Previsão do resultado de partidas

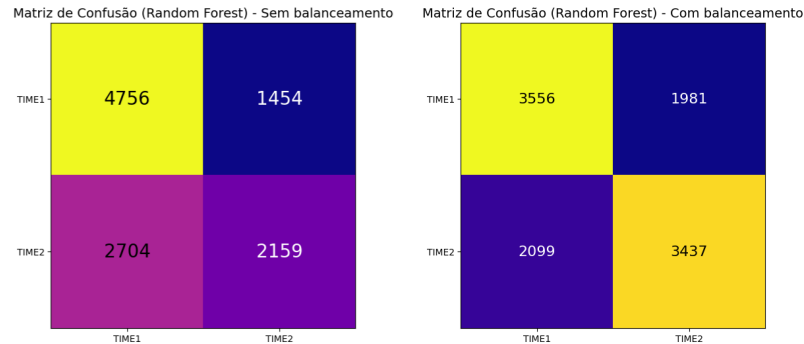
Visando prever o resultado das partidas, os dados previamente tratados foram divididos em seis *datasets* com o propósito de testar diferentes entradas para os modelos. Os testes realizados basearam-se no uso das médias das estatísticas das últimas três partidas como variáveis de entrada; porém, nesta pesquisa, também foram utilizadas as médias das últimas duas partidas, conforme outro estudo (SANHUEZA, 2023). Além dessa divisão, foram testados dois tipos de normalização dos dados: a normalização *min-max*, que ajusta todos os dados para um intervalo entre 0 e 1, e a padronização *z-score*, que centraliza a média dos dados em 0 e o desvio padrão em 1.

Os modelos de classificação utilizados foram: Regressão Logística, *Random Forest*, *Gradient Boosting*, *Support Vector Machines*, K-Vizinhos mais próximos, Redes Neurais (EDMOND, 2023) e Árvore de Decisão (SANHUEZA, 2023). A decisão dos modelos selecionados foi feita com base nos resultados analisados de autores de trabalhos correlatos e que tiveram resultados satisfatórios, como por exemplo o uso de alguns desses modelos para a predição pertinente ao jogo *League of Legends* (EDMOND, 2023).

Diferentes testes foram feitos com os seis *datasets* organizados, e para isso foram feitos alguns tratamentos situacionais. O primeiro tratamento que foi testado foi a respeito do balanceamento do número de resultados, enquanto o segundo foi referente aos *outliers* presentes nos dados. Em relação ao primeiro tratamento, os *datasets* tinham mais partidas que acabaram com a vitória do “TIME1” quando comparado com a quantidade de partidas com a vitória do “TIME2”, o que fazia com que os modelos acertassem mais partidas que acabavam em vitória para o “TIME1”. Na Figura 4 é possível ver um exemplo com o modelo *Random Forest*, enquanto para o “TIME1” o modelo acertou 3556 casos de teste, para o “TIME2” acertou 3437. Para ampliar o equilíbrio foi feita uma simples inversão entre os valores, o que melhorou a validação da qualidade dos modelos e também deixou as previsões mais equilibradas, no modelo de Redes Neurais por exemplo, esse tratamento melhorou os resultados em cerca de 2%. Já a respeito do segundo tratamento foram realizados testes removendo os dados que eram considerados *outliers*, a identificação desses *outliers* foi feita por meio da visualização do gráfico *boxplot* para os atributos dos *datasets*, foram então realizados testes removendo inicialmente poucos dados dos *outliers* e na sequência a maioria deles, para cada atributo foi removido uma quantia diferente dado os resultados dos gráficos *boxplot*, os resultados desse segundo tratamento não demonstraram um impacto significativo, apresentando uma diferença em torno de

1% nos melhores resultados.

Figura 4. Imagem da Matriz Confusão



Os testes realizados tem como saída diferentes métricas de avaliação, como a acurácia, precisão e *f1-score* dos modelos. A métrica utilizada para comparar a qualidade dos modelos foi principalmente a acurácia (EDMOND, 2023), dado que nesse ponto os *datasets* estavam balanceados, estratégia adotada para que não houvesse risco da métrica resultar em uma métrica errônea. Os resultados obtidos foram, então, organizados em uma planilha para possibilitar a comparação. A Figura 5, abaixo, dispõe o quadro com os resultados dos treinamentos com os dez melhores destacados em verde, sendo que é possível observar, independentemente do *dataset*, o modelo de Redes Neurais foi o melhor, com todos os seus seis resultados estando entre os dez melhores, variando entre aproximadamente 62% e 66%, com este maior valor sendo referente ao *dataframe* original, ou seja, sem normalização, com as médias das últimas duas partidas. Além disso é possível analisar que, no geral, os *datasets* com as médias das estatísticas das últimas duas partidas, se mostram melhores do que os *datasets* com as médias baseadas nas últimas três partidas, com o melhor resultado entre os *datasets* com as médias das últimas duas partidas estando em torno de 66%, utilizando o modelo de Redes neurais e o *datasets* sem normalização ou padronização, e o melhor resultado entre os *datasets* com as médias das últimas três partidas estando em torno de 62%, todos os *datasets* (original, normalizado e padronizado) alcançaram esse resultado quando utilizado o modelo de Redes Neurais.

Figura 5. Tabela dos resultados dos modelos de predição

Modelo	df original (medias 2 ultimas partidas)	df original (medias 3 ultimas partidas)	df normalizado (medias 2 ultimas partidas)	df normalizado (medias 3 ultimas partidas)	df padronizado (medias 2 ultimas partidas)	df padronizado (medias 3 ultimas partidas)
Regressão Logística	0.607288	0.60485	0.607423	0.607423	0.607559	0.605527
Árvore de Decisão	0.552289	0.551747	0.556624	0.547548	0.555134	0.550393
Random Forest	0.63709	0.608643	0.609049	0.609049	0.638038	0.608507
Gradiente Boosting Machines (GBM)	0.61379	0.613926	0.614061	0.614061	0.61379	0.613926
Máquinas de Vetores de Suporte (SVM)	0.610404	0.608914	0.61948	0.61948	0.62395	0.615145
K-Vizinhos mais Proximos (KNN)	0.582904	0.56841	0.559198	0.559198	0.576402	0.555811
Redes Neurais Artificiais (RNA)	0.662873685	0.622595489	0.621150553	0.623769522	0.651584923	0.624943554

Fonte: Elaborada pelos autores.

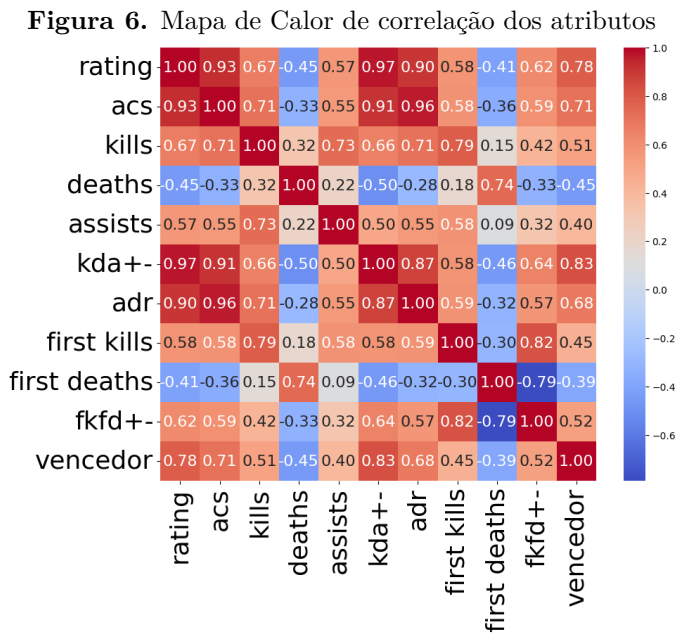
4.2. Clusterização

Para a etapa de clusterização, foi gerado um novo conjunto de dados a partir do *dataset* tratado anteriormente, focado nas equipes. O *dataset*, com as informações dos times, inclui as colunas de data, *timeUm* e *timeDois*, com as duas últimas

sendo representadas por dicionários com as estatísticas do respectivo time, conforme descritas acima. A coluna de *data* foi segmentada para obter informações específicas de ano e semestre. Algumas chaves foram removidas para otimizar as métricas, como ‘k’ e ‘d’, substituídas por ‘kda+-’, que representa a diferença entre abates e mortes de cada jogador. Assim, ‘fk’ e ‘fd’ foram consolidadas em ‘fkfd+-’, que indica a diferença entre primeiros abates e primeiras mortes. As métricas ‘kast’ e ‘hs’ foram desconsideradas devido ao formato percentual, o que exigiria tratamentos adicionais para cálculos de média e dificultaria a análise de dados sem essa transformação – um ponto a ser aprimorado em futuras iterações do projeto.

No *dataset*, foram considerados apenas os valores gerais dos jogadores, sem distinção entre desempenho em ataque e defesa. Cada partida foi dividida em duas linhas, cada uma dedicada a um time, com as colunas *nome*, *temporada* (*ano/semestre*), *r_media*, *acs_media*, *a_media*, *kda+-_media*, *adr_media* e *fkfd+-_media*, sendo as variáveis numéricas representadas pelas médias dos jogadores das equipes. Para a clusterização, os dados foram agrupados por temporada e nome do time, e as variáveis numéricas passaram a ser então a média da estatística do time ao longo da temporada.

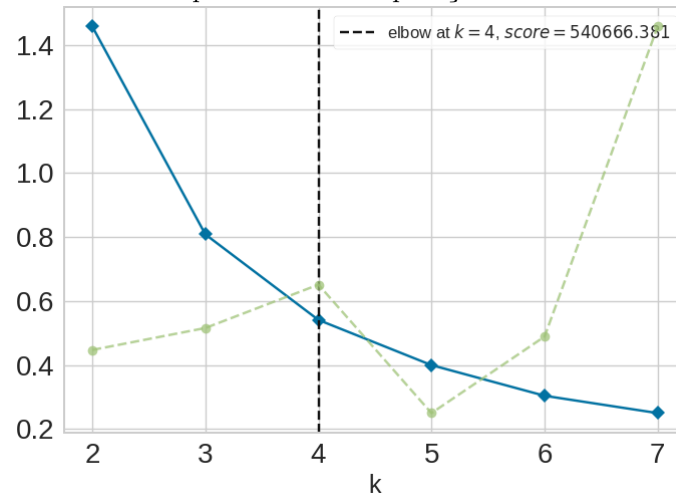
Para facilitar a criação das visualizações foi necessário escolher apenas dois atributos da base criada, o que envolveu um mapa de calor (HAMID et al., 2010) para explicar a correlação entre cada coluna e a chance de vitória (Figura 6). Analisando o mapa criado, é possível ver que as maiores correlações são relacionadas aos atributos R e o KDA+-. O R, é a métrica *Rating*, que é uma classificação que tenta calcular o desempenho de um jogador de acordo com todas as estatísticas disponíveis. O KDA+- é uma métrica de diferencial entre abates e mortes, crucial para estabelecer uma base de comparação entre jogadores de diferentes funções, permitindo uma análise equitativa do desempenho individual frente aos adversários.



Fonte: Elaborada pelos autores.

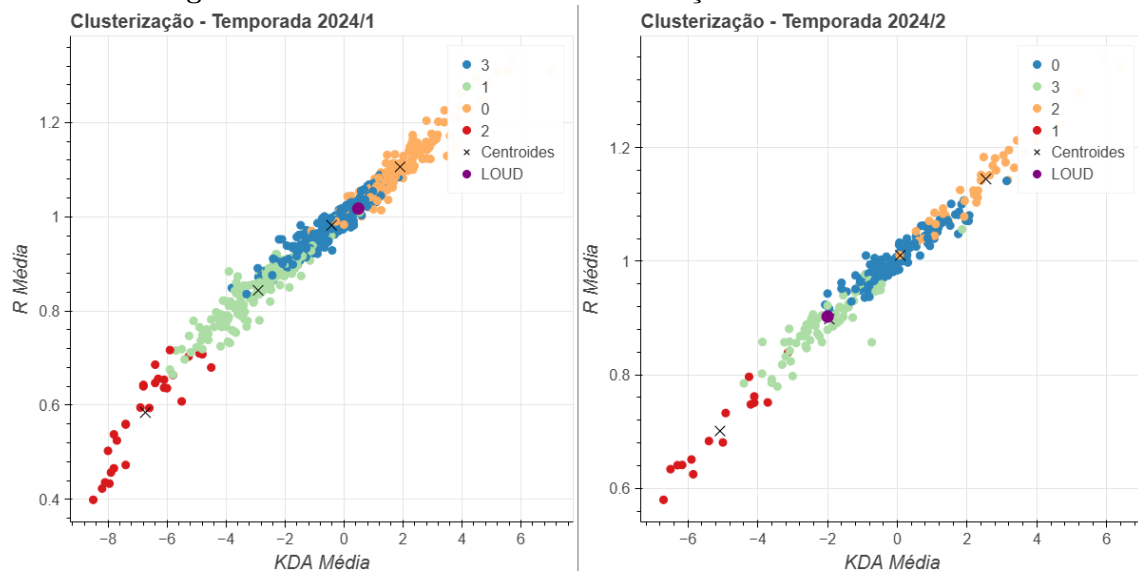
Foi aplicado o método do cotovelo para descobrir a quantidade ideal de *clusters* (CHEN; ZHANG; XU, 2023) (figura 7). Além disso, foi desenvolvido o modelo *K-means*. Os resultados obtidos são semelhantes aos obtidos pelo autor (CHEN; ZHANG; XU, 2023) quando o mesmo mostra os gráficos gerados com duas *features* para os jogadores de basquete, porém nesse caso para os jogadores de *Valorant*. A Figura 8 apresenta uma comparação entre as temporadas do ano de 2024 da equipe "Loud", na figura só é possível ver o ano de 2024 mas são gerados resultados de todos os anos, sendo possível realizar um comparativo de todos os anos daquela equipe.

Figura 7. Gráfico representativo da aplicação do método do cotovelo



Fonte: Elaborada pelos autores.

Figura 8. Gráfico do resultado da Clusterização com modelo *K-médias*



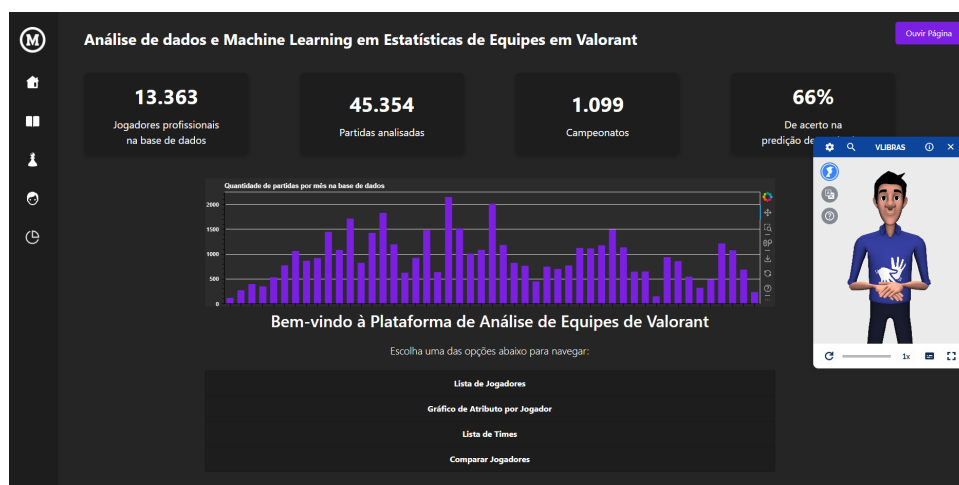
Fonte: Elaborada pelos autores.

4.3. Criação de Website para análises

A criação de um website voltado para a análise de desempenho em jogos como *Valorant* permite centralizar, de maneira eficiente, dados relevantes sobre jogadores e equipes. Utilizando o Django (CHANDIRAMANI; SINGH, 2021), foi possível desenvolver uma plataforma que realiza o processamento e visualização dos dados de partidas, times e jogadores. Essa plataforma propõe uma maneira de facilitar/apoiar a análise estatística e a geração de *insights*, ajudando tanto jogadores quanto equipes a tomarem decisões estratégicas embasadas em dados concretos.

O website proposto exibe gráficos, tabelas e comparações de desempenho entre jogadores que já participaram exclusivamente de partidas competitivas do jogo *Valorant*. Essa ferramenta busca, portanto, melhorar o processo de tomada de decisão no ambiente competitivo, para identificar padrões e analisar resultados de partidas. Dessa forma, o website não só serve como um meio de análise retrospectiva, mas também como uma ferramenta preditiva, oferecendo suporte para ajustes táticos e avaliações detalhadas das campanhas de cada time, e possibilitar a fase dos principais membros do cenários competitivo. Na Figura 9, é apresentada a tela inicial do website, que exibe informações sobre o total de partidas analisadas, o número de jogadores profissionais, partidas registradas e campeonatos, além da precisão do modelo de predição de resultados de partidas. A tela conta ainda com um menu lateral para navegação entre as demais telas.

Figura 9. Tela principal da aplicação Web



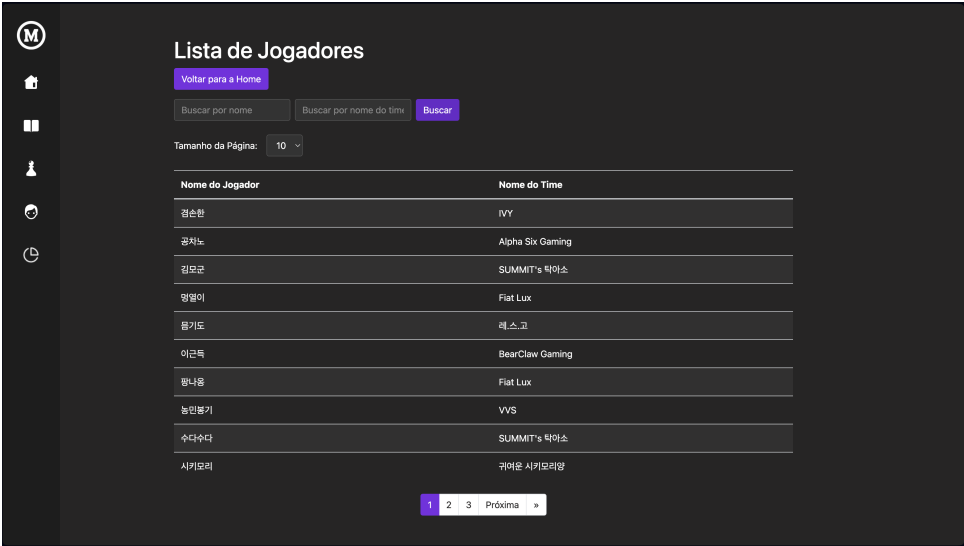
Fonte: Elaborada pelos autores.

Além disso, as telas deste projeto utilizam a *API Vlibras*, uma ferramenta desenvolvida pelo governo brasileiro para promover a acessibilidade digital. Essa API é capaz de traduzir automaticamente todo o conteúdo textual do website para a Língua Brasileira de Sinais (Libras), facilitando a compreensão de usuários surdos ou com deficiência auditiva, ampliando as possibilidades de acesso.

O uso da *API Vlibras* amplia a acessibilidade em plataformas digitais e promove a inclusão, permitindo que pessoas surdas acessem conteúdos de multimídia em sua língua natural de comunicação (Governo Digital, 2024).

Prosseguindo, a Figura 10 dispõe a tela que lista todos os jogadores, considerando-os como registros distintos para cada time em que atuaram. Dessa forma, um jogador com passagem por dois times diferentes aparecerá em duas linhas distintas. A tela conta com um filtro para facilitar a busca por jogador, time, ou um jogador específico em um time determinado.

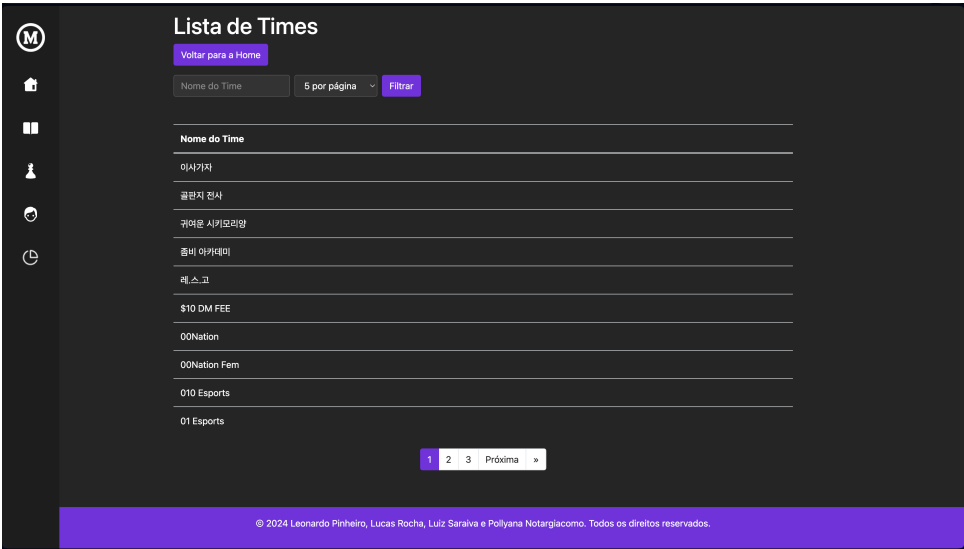
Figura 10. Tela de listagem de jogadores



Fonte: Elaborada pelos autores.

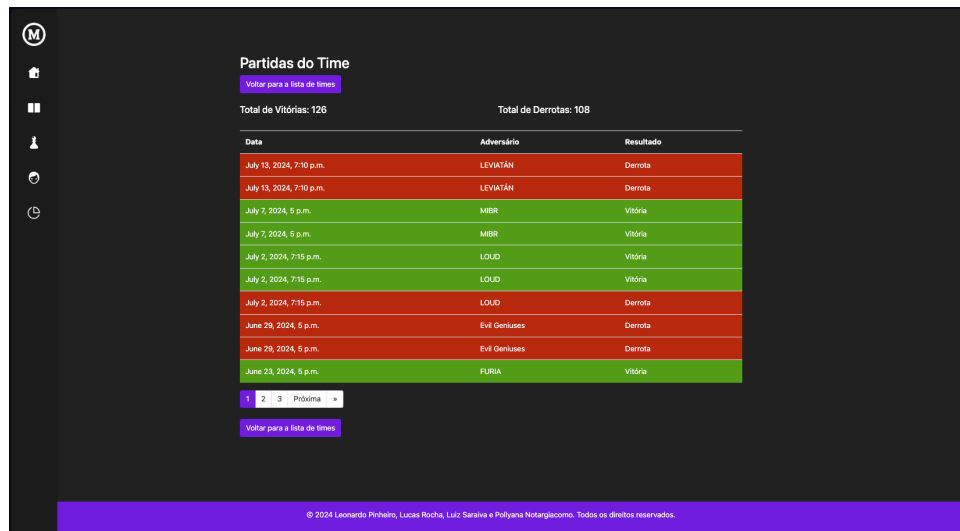
A Figura 11 apresenta a tela de listagem dos times. Nesta há um filtro para pesquisa de times específicos e os respectivos detalhes. A Figura 12 dispõe essa página de detalhes, onde se visualizam o total de partidas jogadas, ganhas e perdidas pelo time, bem como uma listagem completa dos resultados de partidas associadas.

Figura 11. Tela de listagem de times



Fonte: Elaborada pelos autores.

Figura 12. Tela de detalhes de times



Fonte: Elaborada pelos autores.

A tela seguinte, mostrada na Figura 13, é dedicada à visualização interativa do histórico de desempenho de um jogador em um atributo específico. A tela possui filtros para selecionar o jogador, o time, o atributo a ser analisado e o agente a ser considerado. O gráfico, construído com a ferramenta *Bokeh* (STANČIN; JOVIĆ, 2019), permite uma visualização interativa dos dados.

Figura 13. Tela de histórico de atributo por jogador



Fonte: Elaborada pelos autores.

Por fim, a última tela, ilustrada na Figura 14, traz um gráfico radial para comparar o desempenho de dois jogadores em atributos como *HS*, *ADR*, *KDA*, *ACS*, *Assists*, *Deaths*, *Kills*, *Rendimento*, *First Kills* e *KAST*. A tela inclui filtros para selecionar os jogadores (apenas um nome é obrigatório) e especificar se os atributos serão referentes aos registros de ataque, defesa ou geral. O gráfico exibe a média dos

registros do jogador de acordo com os filtros aplicados.

Figura 14. Gráfico radial comparativo entre os jogadores "aspas" e "Sacy".



Fonte: Elaborada pelos autores.

5. Conclusões e Trabalhos Futuros

Diante da popularidade e do conjunto de informações do jogo *Valorant*, foram desenvolvidos os seguintes entregáveis: um minerador de dados para o site <vlr.gg>, uma aplicação para inserir automaticamente os dados em um banco relacional, um modelo de *Machine Learning* para prever o provável vencedor entre duas equipes, um modelo de clusterização para agrupar times com base em performances semestrais, e uma aplicação web para análise de dados de jogadores e times.

Os dois primeiros entregáveis são complementares, permitindo uma democratização ágil dos dados disponíveis no site vlr.gg. A possibilidade de manter a base atualizada e trabalhar com os dados nos formatos CSV e SQL são fatores essenciais deste trabalho, dado o limitado número de bases de dados nesse domínio. Essa etapa do projeto possibilita que os dados sejam utilizados em diferentes domínios, permitindo que mais projetos surjam e complementem o segmento. Ao disponibilizar informações acessíveis e confiáveis, se incentiva o desenvolvimento de novas iniciativas que podem enriquecer e expandir a área de estudo, beneficiando não apenas os leitores deste artigo, mas toda a comunidade interessada.

Os últimos entregáveis da lista também desempenham papéis importantes no alcance do objetivo final deste projeto, fornecendo diferentes maneiras de gerar *insights* para o usuário. Embora o modelo preditivo tenha alcançado uma precisão inferior em relação a outros trabalhos sobre o mesmo tema (EDMOND, 2023), que obteve uma precisão de 75%, acredita-se que essa diferença se deve à quantidade e qualidade dos dados. O modelo desenvolvido neste projeto visa uma solução genérica e independente do torneio, considerando apenas o desempenho de cada equipe. Acredita-se que melhores resultados possam ser obtidos com uma análise de equipes com menor frequência de participação em partidas.

O modelo de clusterização, que permite marcar graficamente a posição dos times nos *clusters* semestrais, é um indicativo de desempenho comparativo entre os times, facilitando outras análises, como a identificação das razões para uma equipe ingressar em um *cluster* com melhores equipes ou cair para um *cluster* com equipes de menor desempenho.

Como sugestão de trabalho futuro, considera-se que a aplicação web poderia incluir uma página dedicada ao uso dos modelos de clusterização e predição, oferecendo ao usuário mais formas dinâmicas de análise.

Referências Bibliográficas

CARDOSO, M. V.; GUSMÃO, C.; HARRIS, J. J. *Pesquisa da indústria brasileira de games 2023*. 2023. Acesso em: 18 maio 2024. Disponível em: <https://www.abragames.org/uploads/5/6/8/0/56805537/2023_relato%u00e7%u00e3o_final_v4.3.3_ptbr.pdf>.

CHANDIRAMANI, A.; SINGH, P. Management of django web development in python. *Journal of Management and Service Science (JMSS)*, v. 1, n. 2, p. 1–17, Jul. 2021. Disponível em: <<https://jmss.a2zjournals.com/index.php/mss/article/view/10>>.

CHEN, R.; ZHANG, M.; XU, X. Modeling the influence of basketball players' offense roles on team performance. *Frontiers in Psychology*, v. 14, 2023. Disponível em: <<https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2023.1256796>>.

DEOLINDO, B.; PEREIRA, B. *Tudo o que sabemos sobre VALORANT, novo jogo de tiro da Riot Games*. 2020. Acesso em: 15 maio 2024. Disponível em: <<https://www.theenemy.com.br/valorant/valorant-tudo-sobre-beta-riot-games>>.

EDMOND, C. *Determining Win-Loss Probability and Round Differential in Professional Valorant*. Tese (Graduate Project), November 20 2023. Acesso em: 21 julho 2024. Disponível em: <<https://scholarworks.calstate.edu/concern/projects/pk02cj680>>.

GANESAN, M. Python requests vs urllib. *ProxiesAPI*, 2 2024. Acesso em: 18 maio 2024. Disponível em: <<https://proxiesapi.com/articles/python-requests-vs-urllib#:~:text=Urllib%20-%20If%20you%20need%20low,Great%20for%20most%20common%20cases>>.

Governo Digital. *VLibras — Governo Digital*. 2024. Acessado em: 2024-11-03. Disponível em: <<https://www.gov.br/governodigital/pt-br/acessibilidade-e-usuario/vlibras>>.

HAMID, J. S. et al. Cluster analysis for identifying sub-groups and selecting potential discriminatory variables in human encephalitis. *BMC Infectious Diseases*, BioMed Central, v. 10, p. 364, 2010. Disponível em: <<https://doi.org/10.1186/1471-2334-10-364>>.

HITAR-GARCÍA, J. A.; MORÁN-FERNÁNDEZ, L.; BOLÓN-CANEDO, V. Machine learning methods for predicting league of legends game outcome. *IEEE Transactions on Games*, v. 15, n. 2, p. 171–181, 2023.

KNUDSEN, C. *How many people play Valorant? Player count tracker: 2024*. 2024. Acesso em: 15 maio 2024. Disponível em: <<https://www.dexerto.com/valorant/how-many-people-play-valorant-player-count-tracker-2022-1668158/>>.

LIU, X. The application of artificial intelligence in fps games. *Applied and Computational Engineering*, v. 35, p. 201–206, 01 2024.

MITCHELL, R. *Web Scraping with Python*. USA: O'Reilly Media, 2018.

NEWZOO. *Top countries and markets by video game revenues*. 2023. Acesso em: 17 maio 2024. Disponível em: <<https://newzoo.com/resources/rankings/top-10-countries-by-game-revenues>>.

PEREIRA, W. *Valorant arrecadou mais de R\$ 324 mi com skins de esports*. 2023. Acesso em: 15 maio 2024. Disponível em: <<https://ge.globo.com/esports/valorant/noticia/2023/12/11/c-valorant-arrecadou-mais-de-r-324-mi-com-skins-de-esports.ghml>>.

PostgreSQL Global Development Group. *PostgreSQL Documentation*. [S.l.], 2023. Accessed: 2024-07-28. Disponível em: <<https://www.postgresql.org/docs/>>.

SANHUEZA, P. E. M. *Predicción de partidas en Valorant: utilizando algoritmos de Machine Learning, Random Forest y XGBoost*. Tese (Doutorado) — Universidad del Desarrollo. Facultad de Ingeniería, 2023. Acesso em: 3 Agosto 2024.

SOUZA, L. et al. Os grupos de gamers: Segmentação de mercado dos jogadores de jogos eletrônicos. *Brazilian Business Review*, v. 18, p. 177–195, 03 2021.

STANČIN, I.; JOVIĆ, A. An overview and comparison of free python libraries for data mining and big data analysis. In: *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. [S.l.: s.n.], 2019. p. 977–982.

vlr.gg. *vlr.gg*. 2024. Disponível em: <<https://www.vlr.gg/>>.

XU, D. *Riot Games offers insight into VALORANT esports revenue, 2024 plans in public letter*. 2023. Acesso em: 15 maio 2024. Disponível em: <<https://esportsinsider.com/2023/12/riot-games-valorant-esports-finances>>.