Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Computação e Sistemas

Criação de um Dataset da Plataforma de Jogos Digitais da Epic Games para Análise Exploratória de Dados

Samuel de Souza Gomes

TRABALHO DE CONCLUSÃO DE CURSO

ORIENTAÇÃO:
Alexandre Magno de Sousa

Novembro, 2022
João Monlevade–MG

Samuel de Souza Gomes

Criação de um Dataset da Plataforma de Jogos Digitais da Epic Games para Análise Exploratória de Dados

Orientador: Alexandre Magno de Sousa

Monografia apresentada ao curso de Sistemas de Informação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina "Trabalho de Conclusão de Curso II".

Universidade Federal de Ouro Preto

João Monlevade

Novembro de 2022

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
INSTITUTO DE CIENCIAS EXATAS E APLICADAS
DEPARTAMENTO DE COMPUTACAO E SISTEMAS

FOLHA DE APROVAÇÃO

Samuel de Souza Gomes

Criação de um Dataset da Plataforma de Jogos Digitais da Epic Games para Análise Exploratória de Dados

Monografia apresentada ao Curso de Sistemas de Informação da Universidade Federal
de Ouro Preto como requisito parcial para obtenção do título de bacharel em Sistemas de Informação

Aprovada em 3 de novembro de 2022

Membros da banca

Prof. Me. Alexandre Magno de Sousa - Orientador (Universidade Federal de Ouro Preto)
Profa. Dra. Tatiana Alves Costa (Universidade Federal de Ouro Preto)
Profa. Dra. Helen de Cássia Sousa da Costa Lima (Universidade Federal de Ouro Preto)

Prof. Alexandre Magno de Sousa, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 20/01/2023

Documento assinado eletronicamente por **Alexandre Magno de Sousa**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 24/01/2023, às 18:12, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 , informando o código verificador **0462242** e o código CRC **9AC9B23B**.

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163
Telefone: (31)3808-0819 - www.ufop.br

Este trabalho é dedicado à toda minha família e amigos que contribuíram de alguma forma durante todo o período do curso.

Agradecimentos

Agradeço primeiramente à minha família, em especial aos meus pais Eunice e Geraldo, por todo apoio durante a graduação e dedicação com meu desenvolvimento. Pois, sua confiança em mim me motivou por todo esse tempo e me proporcionou alcançar muitos objetivos. Agradeço à minha família em geral por toda torcida e suporte durante esse processo.

Também agradeço aos meus amigos e irmãos da República DuBodi, no qual dividi durante todos esses anos meus desafios, dificuldades e, principalmente minhas conquistas.

Agradeço ao meu orientador, Alexandre, por todo seu tempo, dedicação e paciência investidos para a conclusão deste trabalho, bem como por todo conhecimento compartilhado.

"Science is more than a body of knowledge; it is a way of thinking."

— Carl Sagan (1934 – 1996),
in: The Demon-Haunted World: Science as a Candle in the Dark.

Resumo

Nos últimos anos, um mercado que vem crescendo cada vez mais, principalmente após a pandemia da Covid-19, é o mercado de jogos digitais, o qual é uma das indústrias que mais se destaca no mercado internacional em termos lucrativos. Nesse cenário, surgem as plataformas de jogos digitais que concentram grande parte das informações do mercado de jogos bem como registros de interações de jogadores. No entanto, algumas plataformas não tem um repositório estruturado e organizado de dados para facilitar pesquisas e investimentos para melhorias. Por exemplo, diferente da plataforma de jogos Steam, a Epic Games Store não possui uma API pública. Dessa forma, este trabalho tem como objetivo construir um dataset que contenha informações de todos os jogos disponíveis na Epic Games, informações de suas redes sociais e avaliações realizadas por jogadores. Para que isso seja possível, scripts que utilizam o método de *webscraping* foram implementados para coleta de dados diretamente no site da loja da Epic Games, das contas de jogos no Twitter e de seus respectivos tweets por meio da API do Twitter, bem como das avaliações através da API do OpenCritic. O dataset construído conta com 915 jogos, 17.584 avaliações e quase 1 milhão de tweets e os resultados da análise mostram que os 5 gêneros mais presentes na Epic Games são Ação, Aventura, Indie, Único jogador e RPG. Além disso, a média de preço dos jogos é R$ 23,23 e 95% dos jogos tem preço menor ou igual a R$ 60,00, o que mostra que os jogos tem preços bastante acessíveis para os jogadores. Também foi possível identificar que uma pequena parcela das contas de jogos (somente 2%) no Twitter possuem mais do que 1 milhão de seguidores. Essas contas concentram a maioria do engajamento dos usuários (jogadores). Como exemplo, um jogo bastante conhecido é o Fortnite que apresenta mais seguidores em sua conta no Twitter, contém cerca de 15 milhões de seguidores, além disso, é o jogo que mais lucrativo da empresa. Por fim, a Epic Games ainda é nova no mercado em relação à Steam e Origin, apesar disso, este trabalho mostra que há um crescimento da plataforma ano após ano e contribui com um dataset estruturado e organizada que visa colaborar com possíveis pesquisas futuras.

Palavras-chaves: Jogos Digitais. Plataforma de Jogos. Epic Games. *Webscraping*. Dataset.

Abstract

In recent years, a market that has been growing more and more, especially after the Covid-19 pandemic, is the digital games market, which is one of the industries that most stands out in the international market in terms of profit. In this scenario, digital gaming platforms emerge, which concentrate much of the information on the gaming market, as well as records of player interactions. However, some platforms do not have a structured and organized data repository to facilitate research and investments for improvements. For example, unlike the Steam gaming platform, the Epic Games Store does not have a public API. Thus, this work aims to build a dataset that contains information from all games available at Epic Games, information from their social networks and evaluations made by players. To make this possible, scripts that use the webscraping method were implemented to collect data directly on the Epic Games store website, from game accounts on Twitter and their respective tweets through the Twitter API, as well as from ratings through the OpenCritic API. The built dataset has 915 games, 17,584 ratings and almost 1 million tweets and the analysis results show that the 5 most present genres in Epic Games are Action, Adventure, Indie, Single player and RPG. Furthermore, the average price of the games is R$23.23 and 95% of the games are priced less than or equal to R$60.00, which shows that the games have very affordable prices for players. It was also possible to identify that a small portion of gaming accounts (only 2%) on Twitter have more than 1 million followers. These accounts account for the majority of user (player) engagement. As an example, a well-known game is Fortnite, which has more followers on its Twitter account, has around 15 million followers, and is the company's most profitable game. Finally, Epic Games is still new to the market in relation to Steam and Origin, despite this, this work shows that there is a growth of the platform year after year and contributes with a structured and organized dataset that aims to collaborate with possible future research.

Key-words: Digital Games. Gaming Platform. Epic Games. Webccraping. Dataset.

Lista de ilustrações

Lista de tabelas

Sumário

1 Introdução

Na década de 80, livros e artigos voltados para o tema de vídeo games eram primordialmente focados no público jovem (NYITRAY, 2019). Em torno desse tema, havia interesse de diversas áreas intelectuais como a ciência da computação, as artes, a humanidade e ciências sociais. Sendo assim, cada área contribuiu para a experiência do jogo e é válido que os jogos sejam estudados devido ao seu impacto na sociedade e na história cultural (NYITRAY, 2019). Ainda no trabalho apresentado por Nyitray (2019), algumas estatísticas sobre o impacto dos videogames na sociedade estadunidense são apresentadas. Sendo assim, estima-se que 60% dos norte americanos jogam videogames e/ou jogos de computador diariamente; 70% dos jogadores têm 18 anos ou mais, sendo que a idade média é 34 anos. Em termos financeiros, apenas 2017 foi estimado um valor de US$ 36 bilhões gastos em jogos.

Trazendo para o cenário dos dias atuais, a indústria de jogos continua crescendo bastante, é detentora de uma receita estimada de mais de 151 bilhões de dólares e é vista hoje mais do que um negócio de entretenimento. Atrelado ao crescimento das tecnologias e a popularidade dos jogos, os jogos online contemplam jogadores de todas as idades e não mais focados somente no público jovem. Dessa forma, entender o perfil do jogador se tornou ainda mais relevante nos campos acadêmicos e industriais (ZHAO et al., 2020).

A popularidade dos jogos está atrelada ao avanço das tecnologias e, para que os jogos possam cada vez mais melhorar o seu desempenho e, em alguns casos, se aproximar da realidade, o hardware utilizado precisa acompanhar esse processo de evolução. De acordo com Brownsword (2009), há 30 anos o hardware de videogame disponível era de um simples processador de de 8 bits, alguma lógica de exibição dedicada e poucos KBytes de memória. Atualmente, as empresas estão utilizando toda tecnologia de um computador para aplicar nos jogos digitais, dessa forma, é possível jogar em celulares, computadores e consoles mantendo uma alta performance (LIMA et al., 2022).

Nos últimos anos, os avanços das tecnologias de informação e comunicação têm melhorado a infraestrutura da Internet permitindo diversos tipos de serviços tais como as plataformas de música online (LastFM e Spotify), as plataformas de comunicação (WhatsApp e Telegram), *streaming* de vídeo (Netflix, Amazon Prime e Disney+) e, inclusive, plataforma de jogos digitais (e.g., Steam, Origin, Blizard e Epic Games). Além disso, aplicativos de voz sobre IP como Discord com recursos de comunicação para comunidades de jogos e plataforma de compra de jogos vem ganhando cada vez mais usuários. É importante ressaltar que, depois da indústria bélica, a indústria de jogos é a que mais se destaca no mercado internacional em termos lucrativos (O'NEILL et al., 2016).

1.1 Motivação e Justificativa

A qualidade dos serviços online vem incentivando os usuários a se socializarem cada vez mais no ciberespaço. Consequentemente, isso leva a uma quantidade massiva de dados pessoais serem registrados nesse ambiente virtual e atrai ainda mais o interesse e a atenção das grandes empresas pela compreensão do comportamento de usuários online. Esse cenário abre um amplo espaço para as áreas de análise de mídias sociais online, recuperação da informação, sistemas de recomendação e sistemas de busca online. Por exemplo, O'Neill et al. (2016) utilizou o conceito de *webscraping* para coletar e analisar dados da API da plataforma de jogos digitais da Steam. Eles conseguiram examinar o comportamento dos jogadores nas dimensões sociais, tempo de jogo, propriedade do jogo, afinidade de gênero e despesas monetárias. Além da API utilizada, as informações foram coletadas de fóruns e redes sociais relacionadas à Steam. Essas informações possibilitam a realização de análise de sentimentos dos usuários, que tem o objetivo de compreender o comportamento social e a opinião dos usuários (COOPER et al., 2007), bem como também realizar a análise da trajetória dos usuários para melhoria dos serviços para o usuário final. Ademais, as informações disponíveis também tornam viáveis a recomendação de jogos de forma personalizada na intenção de melhorar o engajamento dos jogadores e, consequentemente, aumentar as vendas de jogos nas plataformas.

O mercado de jogos digitais tem se mostrado com maior destaque dentre as indústrias criativas e culturais, tanto em termos financeiros quanto em perspectiva de crescimento para os próximos anos. Em relação ao faturamento, o mercado de jogos eletrônicos já ultrapassa as indústrias de música e cinema juntos (AMÉLIO, 2018).

A Epic Games, uma das maiores plataformas de jogos digitais do mundo, está revolucionando a indústria do entretenimento, principalmente a forma com que jogadores e desenvolvedores criam, publicam e consomem experiência de jogos. Essa empresa é pioneira no desenvolvimento de um ecossistema digital que dispõe da infraestrutura e serviços necessários para jogos em grande escala como o Fortnite (EPAM, 2021). O número de usuários ativos aumentou em 192%, o que gerou aumento nas compras pela plataforma Epic Games Store totalizando mais de US$ 700 milhões em 2020, no qual os jogos de terceiros representam 37% desse valor (GAMES, 2021b).

1.2 Definição do Problema

De acordo com KOTLER (2000), as pessoas modelam suas expectativas de acordo com informações coletadas ao seu redor, seja de comentários de amigos, vendedores ou até mesmo pesquisas na Internet. Por exemplo, quanto maior a diferença entre expectativa e desempenho, maior é a insatisfação de um consumidor de jogos eletrônicos. Dessa forma, obter informações sobre um determinado jogo de forma mais assertiva possível que esteja

alinhado aos gostos do usuário é de suma importância para a satisfação de compra em uma loja da plataforma de jogos.

A plataforma de jogos digitais da Steam disponibiliza uma API que contém dados de usuários e jogos a qual contém mais de 108 milhões de contas de usuários e mais de 1 milhão de jogos (O'NEILL et al., 2016). Por sua vez, a Epic Games, outra plataforma de jogos online, disponibiliza apenas uma referência a uma série de interfaces, cada uma lidando com um conjunto diferente de recursos relacionados às regras de negócio que a empresa utiliza em sua plataforma. Apesar disso, a Epic Games não disponibiliza os mesmos recursos que a Steam, como dados específicos do perfil de usuários ou jogos dentro de uma API pública. Dessa forma, toda essa gama de informações disponíveis não está organizada e estruturada em um único repositório de fácil acesso para análise exaustiva de dados e o uso dos mesmos para aplicação em um sistema de recomendação de jogos para seus usuários ou outras possíveis aplicações que poderiam se beneficiar desses dados.

1.3 Objetivo Geral e Específicos

O objetivo deste trabalho consiste na criação e desenvolvimento de um dataset da plataforma da Epic Games por meio de *webscraping*. Sendo assim, a ideia é recuperar e coletar dados diretamente da plataforma da loja, de fóruns de discussões e de redes sociais e armazená-los de forma estruturada e organizada. A partir desse dataset será possível compreender mais sobre o comportamento dos jogadores e melhorar o engajamento dos mesmos na plataforma da Epic Games, o que pode servir de base para melhoria de recomendações mais personalizadas de jogos. Pretende-se coletar e armazenar dados da rede de jogos favoritos, das categorias de jogos, da classificação e dos preços, além de dados de jogos específicos, suas avaliações, configuração mínima de hardware, dentre outras.

Para alcançar o objetivo geral, os seguintes objetivos específicos foram definidos:

- Revisão da literatura: estudar e identificar os processos nas quais envolvem *webscraping*, análise exploratória de dados e APIs de plataformas digitais;
- Estudar meios de implementar a coleta de dados e apresentar os dados coletados;
- Criar um dataset a partir dos fóruns e da plataforma da Epic Games;
- Complementar as APIs oferecidas pelas plataformas com dados focados nos jogadores e nos jogos para pesquisadores que tenham a intenção de melhorar o engajamento dos jogadores na plataforma.

1.4 Resultados e Contribuições

Durante o processo de desenvolvimento do trabalho foi possível construir um dataset com dados de todos os jogos disponíveis na plataforma da Epic Games até o momento da coleta. Além disso, dados de suas redes sociais e avaliações de empresas e jogadores também foram recuperados através das APIs do Twitter e OpenCritic para complementar o conjunto de dados, uma vez que a plataforma não disponibiliza informações do perfil dos jogadores. Ademais, algumas análises foram realizadas com intuito de abstrair informações com relação à média de preços dos jogos, frequência de avaliações realizadas e o engajamento das contas dos jogos no Twitter de um modo geral.

Ao comparar a Epic Games Store com outras plataformas de jogos digitais como a Origin e Steam, é notório que a loja da Epic ainda é muito nova no mercado e que, por exemplo, não possui um modo de interação entre os usuários em sua plataforma. Contudo, a Epic Games possui um grande potencial de crescimento e pode ser tema de muitas pesquisas para análise de sentimentos, recomendação de jogos e análise de redes complexas.

Neste trabalho foi possível entender que uma pequena parcela das contas no Twitter armazenadas pelo dataset possuem um grande engajamento em relação às demais. O top 1 da lista é o Fortnite, jogo mais importante para a Epic Games em termos de lucro e participação dos jogadores, além disso, possui 14.895.908 seguidores no Twitter. As análises também identificaram a participação dos usuários através de curtidas (máxima de 596.446), repostas (máxima de 119.217), citações (máxima de 71.698) e retweets (máxima de 566.550) em tweets de todas as contas obtidas, que incluem contas de jogos e plataformas parceiras da Epic Games. Além do Fortnite, jogos que também aparecem no top 10 com mais engajamento são: Sonic, League of Legends, Assassin's Creed, Genshin Impact, Valorant, Rainbow Six Siege, Radiohead, 2k e Cyberpunk 2077. Entre os gêneros de jogo que mais aparecem na loja da Epic estão: Ação (presente em 25% dos jogos), Aventura (presente em 20% dos jogos), Indie (presente em 18% dos jogos), Único Jogador (presente em 14% dos jogos) e RPG (presente em 13% dos jogos), lembrando que um jogo pode ter mais de um gênero. Em média, o preço dos jogos coletados pelos scripts são de R$ 23,23, sendo que 60% dos jogos são gratuitos ou possuem um valor de até R$ 20,00, bem como 90% dos jogos possuem preço inferior à R$ 60,00. Por fim, foi identificado que ao longo dos anos a quantidade de avaliações coletadas pelo OpenCritic crescem constantemente desde 2012, ademais, 80% das avaliações possuem uma nota maior que 60 em uma escala de 0 a 100 no site.

A construção desse dataset contribui, principalmente, no mercado de jogos digitais, pois, é notório que este mercado vem ganhando cada vez mais espaço. Contudo, a Epic Games não disponibiliza uma API pública com dados de seus jogos e jogadores, como a plataforma da Steam. Dessa forma, este trabalho entrega um conjunto de dados organizado e estruturado e fornece uma análise exploratória de dados e apresenta uma caracterização

inicial do dataset construído.

1.5 Estrutura da Monografia

Os próximos capítulos estão estruturados conforme descrito a seguir. O Capítulo 2 é composto por toda a pesquisa realizada dentre o mercado de jogos online, plataformas de jogos online, que inclui suas APIs e caracterização dos usuários. Ainda sobre os jogos online é abordado o impacto dos mesmos tanto em quantidade de usuários quanto em termos financeiros. Além disso, é descrito um pouco da evolução dos jogos e da plataforma Epic Games. Na definição de dataset foi apresentado o conceito de forma geral, na qual é aplicável a quase todas as áreas de conhecimento, mas que ainda necessita de um melhor entendimento por parte da comunidade devido aos diversos formatos existentes de um conjunto de dados. Por fim, na seção de trabalhos relacionados são descritos cinco projetos semelhantes ao da pesquisa atual, assim, essa última seção aborda os objetivos, metodologia utilizada, vantagens e desvantagens, e possíveis aplicações desses trabalhos.

Em seguida, no Capítulo 3 é descrito como o desenvolvimento do trabalho foi realizado. De início, foi necessário conduzir um estudo detalhado sobre a loja da Epic Games para identificar os dados que seriam coletados e como o site recomenda seus jogos. Além disso, o capítulo descreve como o dicionário de dados foi construído e apresenta um modelo entidade relacionamento no qual é detalhado cada entidade e seus respectivos atributos. Além disso, o capítulo também descreve o fluxograma que representa o funcionamento dos scripts implementados para *webscraping* e coleta de dados, os quais se encontram no Apêndice A. O Capítulo 4 são apresentados os resultados da pesquisa, desde quais dados foram coletados para o dataset até uma análise e caracterização dos mesmos. Por fim, no Capítulo 5 são descritas as conclusões finais, as limitações deste trabalho e possíveis trabalhos futuros para a partir desta pesquisa.

2 Fundamentação Teórica

Este capítulo apresenta uma revisão da literatura, bem como trabalhos correlatos. Sendo assim, a primeira seção descreve o cenário dos jogos digitais nos últimos anos, sua evolução e seus tipos. A seção 2.2 aborda sobre as plataformas de jogos digitais construídas devido à evolução da tecnologia e dos jogos digitais, ademais, cita e descreve 3 das principais plataformas atualmente. Já na seção 2.4, é descrito alguns conceitos e definições de dataset e sua importância, bem como seu processo de coleta, pré processamento e filtragem dos dados. Por fim, na seção 2.5 há resumos de todos os trabalhos relacionados utilizados como base para desenvolvimento deste trabalho.

2.1 Jogos Online

Dentre os serviços online, os jogos digitais estão sendo grandes destaques, com uma base cumulativa de usuários conquistando cerca de 40% da população online global e um mercado em termos globais avaliado em cerca de US $20 bilhões (LEE et al., 2020). Para garantir que os usuários se mantenham engajados e adquirir sucesso com estes jogos online, não basta que o jogo seja divertido, é preciso oferecer uma administração adequada de usuários. Além disso, Chambers et al. (2010) diz que para os desenvolvedores de jogos conquistarem uma receita saudável é necessário disponibilizar atualizações de conteúdo, tempo de jogo confiável e uma jogabilidade equilibrada para atrair novos jogadores e não perder os antigos.

Em um contexto geral, garantir a permanência de um cliente é mais vantajoso e econômico do que perdê-lo e tentar conquistar um novo. Dessa forma, as empresas costumam utilizar estratégias para evitar a perda de usuários. Com uma campanha de prevenção bem sucedida, a identificação de possíveis desertores torna-se cada vez mais crucial e, com o avanço contínuo de técnicas de mineração de dados, muitos pesquisadores estudam vários métodos de classificação (LEE et al., 2020).

2.1.1 Evolução dos Jogos Online

Os trabalhos relacionados à Internet vêm ganhando cada vez mais espaço a partir de que novas tecnologias são desenvolvidas e um grande exemplo disso são os jogos online. Essa expansão da informação possibilitou às pessoas obterem acesso a diversos meios de comunicação e relacionamento social (LESNIESKI, 2013).

Um dos grandes exemplos de evolução dos jogos online são os e-Sports, ou esporte eletrônico, que se tornaram populares no início da década de 2010 e demonstram muitas

características do esporte tradicional (LIMA et al., 2022). No entanto, muito antes dos e-sports, o primeiro Massive Multiuser Online Role Playing Game (MMORPG) lançado comercialmente foi feito pela Origin Systems em 1997, intitulado Ultima Online, o qual alcançou a marca de 250.000 jogadores em 2003 (LESNIESKI, 2013).

O mercado de jogos online cresceu bastante nos últimos anos. Em 2019 foi estimado uma movimentação de US $\$152,10$ bilhões com um crescimento de $9,6\%$ ao ano, considerando o Brasil como 13^{o} colocado no mercado mundial daquele ano (ANTONIOLLI, 2020). Já no contexto apenas de e-Sports, segundo Lima et al. (2022) a indústria gerou em 2018 aproximadamente US $\$905,6$ milhões. Desse valor US $174 milhões foram de publicidade, US $ 161 milhões em direitos de mídia, US $359 milhões em patrocínio, US $96 milhões de ingressos e mercadorias e o restante US $116 milhões de investimento das editoras de jogos.

Segundo dados da Newzoo, Super Data Research e Pesquisa Game Brasil, o mercado de games é o que mais cresce em entretenimento online. Sendo assim, até 2023 estima-se que a indústria de games fatura cerca de $US\$2,60$ trilhões no âmbito mundial, um crescimento média que chega a $4,3\%$ ao ano de 2019 a 2023 (ANTONIOLLI, 2020).

A indústria de jogos tem um destaque considerável em países como China, Estados Unidos e Japão. No entanto, a América Latina é a região que mais cresce nesse mercado nos últimos anos com uma taxa de crescimento de $10,4\%$ ao ano. Porém, ainda é o quarto maior mercado no mundo, atrás do Asiático, Norte Americano e Europeu (ANTONIOLLI, 2020).

2.1.2 Tipos e Gêneros de Jogos Digitais

Um jogo pode ser classificado de várias maneiras. No entanto, nesta subseção serão apresentados cinco tipos de jogos digitais e as diferenças entre os jogos não digitais de acordo com Lucchese e Ribeiro (2009):

- **Cooperatividade**: nesse modo é possível que os jogadores se unam em prol de um objetivo final para que todos saiam beneficiados. Dessa forma, há a possibilidade de modelar situações de conflito no decorrer do jogo. No entanto, a cooperatividade pode não acontecer a partir do momento em que um grupo de jogadores cooperativos seja representado como um único jogador, desde que esses jogadores obtenham pontuações distintas;

- **Simetria**: matematicamente falando, um jogo é visto como simétrico quando a matriz de pontuações é simétrica. Basicamente, isso quer dizer que se a identificação de um jogador pode ser alterada sem que altere o resultado final, este é um jogo simétrico.

Um exemplo disso são os jogos de xadrez, no qual o diferencial é a habilidade do jogador e não há muito o que o jogo possa fazer para compensar isso;

- **Soma Constante e Soma Zero**: em casos de jogos de soma constante pode acontecer que a soma das pontuações dos jogadores em qualquer resultado possível são iguais a um valor constante. Por outro lado, jogos de soma zero são aqueles em que se um jogador ganha pontos, quer dizer que outro jogador está perdendo pontos;
- **Dinâmica**: os jogos podem ser estáticos (simultâneos) ou dinâmicos (sequenciais). Sendo assim, nos jogos dinâmicos as sequências de jogadas é realizada por um jogador de cada vez e, a cada jogada, o jogador seguinte sofre os efeitos causados pelo jogador anterior;
- **Informação**: por fim, nos casos em que os jogos possibilitam que os jogadores vejam as ações uns dos outros é chamado de jogos de informação perfeita, do contrário informação imperfeita. Um outro modo de jogos de informação é a de jogos completos e incompletos. Dessa forma, os jogadores podem visualizar a estratégia e pontuação do oponente, mas não necessariamente sua jogada.

Dentro destes tipos citados, de acordo com Todor (2015) pode-se ir mais a fundo para classificá-los da seguinte forma: jogos educativos, de entretenimento, advergames, simuladores, exergames e serious games. Os jogos educativos servem para ensinar algo enquanto o jogador se diverte e, à princípio, é destinado ao público infantil. Entretenimento tem como o próprio nome diz divertir seus jogadores como possibilidade, por exemplo, de entrar em um mundo alternativo. Advergames servem para publicidade de algum produto específico, já os simuladores tem como objetivo exercitar uma determinada habilidade. Os exergames buscam evitar o sedentarismo de seus praticantes . E por fim, os serious games vai além do entretenimento e também tem como objetivo o treinamento de habilidades e aprendizado de seus jogadores (BLACKMAN, 2005).

A Epic Games conta com todos esses tipos de jogos. No entanto, para compor o dataset foram escolhidos os gêneros dos jogos. A seguir os principais gêneros identificados nos jogos presentes na loja da Epic Games.

Jogos de ação, geralmente, são aqueles que desafiam a velocidade, reflexo e raciocínio rápido do jogador, bem como geram conflitos estratégicos. Jogos de aventura são caracterizados pela exploração do cenário, enquanto que jogos Indie significam que são criados por uma pessoa ou pequenas empresas. Jogos *single player*, ou único jogador, são aqueles que permitem um único jogador por partida. RPG significa Jogo de Interpretação de Papéis em português e nesse tipo de jogo se reúne um grupo de pessoas para construir uma história. Jogos de estratégia são feitos para os jogadores tomarem decisões estratégicas e vai além da sorte. Jogos de simulação, normalmente, simulam uma situação da vida

real. Jogos *multiplayer*, ou multi jogadores, são aqueles que permitem vários jogadores simultâneos em uma mesma partida. Jogos de mundo aberto permitem que o jogador explore o mapa do jogo e, geralmente, contém missões para se realizar. Por fim, puzzle são jogos de quebra-cabeça (GAMES, 2021a).

2.2 Plataformas de Jogos Digitais

Como descrito anteriormente, os jogos digitais vem ganhando bastante espaço entre os serviços online. Por trás desse sucesso, as plataformas de jogos têm um papel fundamental na evolução desse mercado (SAKUDA, 2016). Sendo assim, nessa seção serão apresentadas três das principais plataformas de jogos digitais existentes.

As três plataformas referem-se a Steam, Epic Games Store e Origin. Atualmente, essas lojas de jogos têm ganhado o mercado de jogos e disputam entre a preferência dos usuários, bem como também dos desenvolvedores de jogos. Periodicamente, a Epic Games costuma disponibilizar alguns jogos populares de forma gratuita para todos os usuários. A Origin é uma vitrine para jogos da Eletronic Arts (EA), empresa desenvolvedora de jogos, que é uma boa alternativa e que, por bastante tempo, competiu com a própria Steam. Já a Steam, é detentora de um dos maiores catálogos de jogos entre as plataformas digitais e também possui muito mais jogadores cadastrados, devido ao seu sucesso e maior tempo de existência (TUDO, 2021).

Com o crescimento das tecnologias e popularidade das mídias sociais, as plataformas e comunidades de jogos têm papel fundamental no sucesso de um jogo. É notório que hoje é muito mais fácil divulgar e comentar sobre um jogo digital do que um jogo físico devido à rápida disseminação da informação por meio da Internet. Além disso, os jogos de múltiplos jogadores têm ainda mais impacto no relacionamento entre os jogadores (BANKOV et al., 2019). Dessa forma, torna-se crucial uma boa comunicação entre a equipe devido à evolução dos jogos que dão resposta de ações praticamente em tempo real. Sendo assim, duas das plataformas mais utilizadas para comunicação dos jogadores são Discord e Twitch, além disso, é possível assistir transmissões de partidas pelo Twitch (LIMA et al., 2022).

As empresas de jogos estão vendo novas oportunidades para divulgar seus produtos por meio das mídias sociais. Em 2015, o jogo World of Warcraft recebeu uma nova atualização que possibilitava seus jogadores fazer login utilizando a conta do Twitter (BANKOV et al., 2019). Dessa forma, os usuários poderiam compartilhar capturas de tela do jogo diretamente na rede social. Com o desenvolvimento em alta, uma das melhores maneiras de atrair o público desejado é por meio de divulgações em redes sociais (SAKUDA, 2016).

2.2.1 Plataforma Epic Games Store

A Epic Games é uma das maiores plataformas de jogos digitais do mundo e ano após ano busca revolucionar a indústria do entretenimento, a maneira como jogadores e desenvolvedores criam, publicam e consomem experiência de jogos. Ademais, a Epic Games é pioneira no desenvolvimento de um ecossistema digital que dispõe da infraestrutura e serviços necessários para jogos em grande escala como o Fortnite (EPAM, 2021). O número de usuários ativos aumentou em 192%, o que gerou aumento nas compras pela plataforma Epic Games Store totalizando mais de US $700 milhões em 2020, no qual os jogos de terceiros representam 37% do valor (GAMES, 2021b).

A empresa foi fundada em 1991 e é a criadora da Unreal Engine, Gears of War, Shadow Complex e da série de jogos Infinity Blade e tem sua sede em Cary na Carolina do Norte. A tecnologia Unreal Engine da Epic desenvolve entretenimento para várias plataformas como PC, console, dispositivos móveis, Augmented Reality (AR), Virtual Reality (VR) e Web (GAMES, 2021d).

Até o fim de 2020, a plataforma já contava com mais de 160 milhões de clientes de PC da Epic Games Store, mais de 56 milhões de usuários ativos só no mês de dezembro. Além disso, os jogadores gastaram cerca de US700 milhões em jogos da Epic, sendo que US $265 milhões foram em jogos de terceiros. Ademais, detém 77% de pontuação média geral de todos os jogos grátis com avaliação feita pela OpenCritic (GAMES, 2021b).

Alguns recursos foram lançados em 2020, como, por exemplo, a lista de desejos que foi um recurso importante e que continuou a receber melhorias adicionais ao longo do ano. Além disso, a Epic ainda lançou o Suporte a Modificações que seguem incorporando mais títulos que suportam modificações na Epic Games Store. Ademais, em 2020 a Epic ainda lançou mais algumas novidades (GAMES, 2021b):

- Avaliações da OpenCritic;
- Reembolsos por autoatendimento;
- 19 novas moedas;
- Faturamento direto via operadora em vários países;
- Início de sessão offline;
- Melhorias de desempenho para inicializador (muito útil para Mac);
- Navegação aprimorada para conteúdos adicionais;
- Mensagens de Acesso Antecipado;
- Filtros por preço.

O Fortnite foi lançado em 2017 e em pouco tempo se tornou um dos jogos mais famosos mundialmente. A Epic Games desenvolveu este jogo com uma proposta inovadora permitindo que o Fortnite fosse adquirido de forma gratuita, sendo o foco de lucro os itens disponibilizados no próprio jogo (NEVES et al., 2020). Dessa forma, o Fortnite se tornou um dos jogos mais lucrativos da história. Ademais, com o passar dos anos o jogo sempre disponibilizou conteúdo para seu público para mantê-los engajados (NEVES et al., 2020).

Gears of War é uma franquia de jogos eletrônicos criada e originalmente propriedade da Epic Games, desenvolvida e gerenciada pela The Coalition, e atualmente quem tem seus direitos é a Xbox Game Studios (FORBES, 2014). Outro jogo famoso é o Shadow Complex que foi desenvolvido com a Unreal Engine 3 e é vencedor de mais de 50 prêmios de Jogo do Ano e de Escolha do Editor (STEAM, 2016).

Do premiado estúdio Epic, ChAIR Entertainment, os jogos inovadores Infinity Blade elevaram os jogos portáteis a novos patamares com visuais deslumbrantes, batalhas de luta de espadas cheias de adrenalina e progressão avançada de personagens e personalização em um mundo 3D expansivo. O game é um RPG de ação desenvolvido pela Chair Entertainment e pela Epic Games e lançado na Apple App Store em 9 de dezembro de 2010. Foi o primeiro jogo iOS a rodar no Unreal Engine (GAMES, 2021c).

2.2.2 Plataforma Steam

Através da plataforma da Steam, além da possibilidade de baixar jogos demo, é possível participar de comunidades, interagir com outros usuários e ficar atualizado em relação aos lançamentos de jogos (O'NEILL et al., 2016). Ademais, jogos originais ficam disponíveis para compra na plataforma com preços mais baratos ou até mesmo de forma gratuita, o que é uma vantagem das plataformas de jogos digitais. A Steam também utiliza uma estratégia denominada "Acesso antecipado" que é uma forma de lançamento de software para permitir aos jogadores adquirirem uma versão inacabada do jogo. O objetivo dessa estratégia é receber o quanto antes feedbacks de seus produtos para que os desenvolvedores possam corrigir possíveis falhas ou melhorar alguma funcionalidade (LIN; BEZEMER; HASSAN, 2018).

As várias opções disponibilizadas pela Steam para comunicação entre os jogadores contribui para a popularidade do modelo de acesso antecipado (LIN; BEZEMER; HASSAN, 2018). Além disso, é possível compartilhar suas conquistas nessas comunidades da plataforma por meio de postagens com captura de tela. No perfil do jogador são exibidas informações como uma visão geral das estatísticas sociais e de algum jogo específico que foi jogado por este usuário. Ademais, há as divisões de amigos, grupos participantes e uma lista com jogos adquiridos e jogados recentemente (O'NEILL et al., 2016).

Um outro fator interessante da plataforma é que a Steam permite a evolução dos

jogadores em forma de níveis. Ou seja, a cada conquista, o jogador recebe um emblema e tem a possibilidade de subir de nível. Estes emblemas são conquistados de acordo com que o usuário joga utilizando a plataforma. Essas premiações dão aos usuários a permissão de desbloquear *widgets* adicionais para seus perfis e a possibilidade de aumentar a quantidade máxima de amizades (O'NEILL et al., 2016).

2.2.3 Plataforma Origin

A plataforma Origin, é uma vitrine de jogos da empresa desenvolvedora Eletronic Arts. Dessa forma, a plataforma contém todos os jogos somente desta desenvolvedora. Além disso, é possível conectar com amigos por meio da comunidade online disponibilizada na Origin e acessar sua conta por vários dispositivos que tenha compatibilidade com os jogos (ARTS, 2011).

Pelo fato de ter poucos jogos disponíveis em relação às outras plataformas anteriormente citadas, a Origin costuma oferecer ofertas de forma constante e, como são sempre os mesmos títulos, é possível ter bons jogos com preços interessantes constantemente (TUDO, 2015).

Além dos jogos já disponíveis na plataforma, a Origin tem um serviço chamado EA Play, que basicamente é um serviço de assinatura que permite aos usuários receberem mais recompensas, mais conteúdo exclusivo e acesso ilimitado a vários títulos. Ademais, assinantes recebem 10% de desconto em todas as compras digitais da plataforma (ORIGIN, 2022).

De acordo com a Origin (2022), pode-se citar algumas vantagens da plataforma:

- **Downloads rápidos**: a plataforma promete downloads otimizados com o objetivo de atingir a maior velocidade possível. Enquanto um jogo é instalado, é possível jogar paralelamente um outro jogo;
- **Modo offline**: a Origin disponibiliza alguns jogos de único jogador e quando o usuário estiver sem conexão com a Internet também é possível jogá-los;
- **Salvar na nuvem**: caso o usuário tenha algum problema com a memória local de seu computador, é possível salvar o desempenho do jogo na nuvem para até mesmo poder continuar de outro dispositivo;
- **Uma única biblioteca de jogos**: todos os jogos adquiridos pelo jogador estão disponíveis em um único local, independente de onde foi comprado.

Como foi descrito anteriormente, a Origin disponibiliza uma comunidade online para os usuários interagirem com seus amigos, sendo assim, pode-se citar mais alguns pontos sobre amigos e comunidade:

- **Lista de amigos**: a plataforma permite o usuário buscar seus amigos e montar uma lista com os mesmos;
- **Bate-papo de texto e voz**: além da comunidade que permite a comunicação por texto, enquanto se joga é possível manter uma comunicação de voz com os amigos;
- **Transmissão pelo Twitch**: a Origin ainda tem compatibilidade com a Twitch e permite a transmissão de partidas pela plataforma;
- **Outras comunidade**: além da comunidade disponível na plataforma, é possível ficar atualizado em relação as novidades pelo Facebook e Twitter.

Como todas as plataformas, a Origin também tem seus jogos destaque, entre eles estão Star Wars Battlefront e The Sims. Star Wars foi lançado em 2015 e foi bastante elogiado pelo crítica e jogadores, com apenas modo *multiplayer* há também missões offline. No entanto, o melhor modo do jogo é desbravando o universo de Star Wars que conta com os personagens do filme. Já The Sims, uma das séries de jogos mais populares nos últimos anos, o jogador pode criar seu perfil e simular a vida real no jogo, sua primeira versão foi lançada em 2000 (ARTS, 2022).

2.2.4 APIs para Coleta de Dados de Jogos de Plataformas Digitais

Pesquisas digitais de métodos qualitativos estão cada vez mais dependendo da coleta e armazenamento de dados de mídias sociais por meio do uso das *Application Programming Interfaces* (APIs). Nos últimos anos, esse processo tem sido simples, tendo desenvolvedores e pessoas do meio acadêmico usando APIs para encontrar dados para produzir análises de redes sociais (PERRIAM; BIRKBAK; FREEMAN, 2020). Por meio dessas APIs, plataformas como Youtube e Spotify podem disponibilizar informações como gênero musical que um usuário curtiu, total de visualizações ou até mesmo postagens vinculadas a uma hastag no caso do Twitter (D'ANDRÉA, 2021). Já APIs públicas de plataformas de jogos podem disponibilizar uma descrição de seus jogos como preço, resumo, hardware necessário para se jogar, além disso, informações de seus usuários como jogos mais jogados, jogos favoritos, grupo de amigos, entre outros (D'ANDRÉA, 2021).

De acordo com D'Andréa (2021), a popularização dessas ferramentas desenvolvidas a partir das APIs gerou oportunidades promissoras para realizar pesquisas nas áreas sociais e humanas, essencialmente em junção com outras áreas do conhecimento como a computação. No entanto, com o passar dos anos muitas dessas APIs estão passando a ter acesso restrito e regulamentado pelas corporações que comandam as mídias sociais (PERRIAM; BIRKBAK; FREEMAN, 2020).

Em relação às plataformas de jogos digitais citadas anteriormente, somente a Steam disponibiliza uma API pública para acesso aos dados de seus jogos e usuários (O'NEILL

et al., 2016). No caso da Epic Games, as APIs disponíveis servem para desenvolvedores de jogos e tem este tema como foco. Dessa forma, a construção deste trabalho focado na Epic Games tem ainda mais valor de agregação devido à falta de uma API pública com dados de seus jogos e jogadores (GAMES, 2022).

A API da Steam foi disponibilizada pela Valve Corporation, empresa proprietária da Steam. Um exemplo de aplicação desta API está em um dos trabalhos relacionados descrito na seção 2.5. O'Neill et al. (2016) utilizaram esta API para rastrear 108,7 milhões de contas Steam, juntamente com a lista de amizades, jogos de propriedade, tempos de jogo e associações de grupos relacionados a cada usuário da plataforma.

2.2.5 Caracterização de Usuários de Plataformas de Jogos Digitais

A popularidade dos jogos online vêm crescendo cada vez mais, sendo assim, é preciso entender o comportamento dos jogos e seus jogadores. De um modo geral, o comportamento dos jogadores é bastante diversificado e caracterizado por distribuições de cauda pesada (O'NEILL et al., 2016).

Contudo, pesquisas apontam que pessoas mais velhas jogam em um ritmo cada vez maior, inclusive, os adultos de hoje jogam mais do que as gerações anteriores. O resultado disso é uma população cada vez mais engajada nesse universo e, além disso, como a Internet é um recurso indispensável no cotidiano das pessoas hoje em dia, os jogos online também se tornaram um dos principais meios de entretenimento (WILLIAMS; YEE; CAPLAN, 2008).

De acordo com Williams, Yee e Caplan (2008), os jogadores têm em média 31 anos, sendo o limite mínimo 12 e o máximo de 65. Ao contrário do que diz os estereótipos, a maior parte dos jogadores estão nessa faixa de 30 anos, e não jovens adolescentes. Um outro fator é que jogadores mais velhos passam mais tempo jogando do que os mais novos.

Em relação ao gênero do jogador, em 2008 a distribuição era de $80,80\%$ masculino e $19,20\%$ feminino entre os jogadores norte americanas. Contudo, pessoas do sexo feminino jogam mais horas por semana do que pessoas do sexo masculino (WILLIAMS; YEE; CAPLAN, 2008).

No trabalho de O'Neill et al. (2016), é descrito que a maioria dos jogadores possuem um comportamento modesto em relação ao tempo jogado por dia e a quantidade de dinheiro gasto nos jogos, apesar de que exista uma grande cauda com resultados bem diferentes.

Com relação à saúde, os dados coletados no trabalho de Williams, Yee e Caplan (2008) apontam que, fisicamente, o jogador norte americano é mais saudável do que o restante da população. Seu Índice de Massa Corporal (IMC) é em média 25,19, o que pode ser considerado acima do peso, no entanto, a média é menor do que do restante

dos adultos norte americanos que é 28. Contudo, entre as crianças o IMC dos jogadores também é mais baixo, sendo 21,96 para os jogadores e 23,3 para o restante da população.

Já em relação ao mental, os jogadores possuem um nível de saúde abaixo na maioria dos indicadores. 22,76% da amostra da pesquisa aponta ter sido diagnosticado com depressão em algum momento, sendo que as mulheres relataram mais casos do que os homens. Uma exceção entre estes indicadores se trata da ansiedade, os jogadores relataram níveis ligeiramente mais baixos em comparação com a população geral (WILLIAMS; YEE; CAPLAN, 2008).

Por fim, o tempo de jogadores individuais varia bastante ao decorrer de uma semana. Entretanto, em jogos *multiplayer* os jogadores costumam passar um tempo constante e maior ao longo da semana, apesar do número de amigos ser baixo em relação a outras redes sociais (O'NEILL et al., 2016).

2.3 Plataformas Utilizadas

Para compor a pesquisa e tornar o trabalho mais completo foram utilizadas duas plataformas além da Epic Games Store, são elas: Twitter e OpenCritic. Através de suas APIs foram coletadas todas as contas no Twitter dos jogos que possuem um perfil no Twitter, com seus respectivos tuítes (API Twitter), bem como as avaliações dos jogos (API OpenCritic).

2.3.1 Twitter

Fundado em 2006, o Twitter é uma das maiores redes sociais do mundo, que é utilizada por uma grande parte da população para divulgação de notícias, artigos e para socializar com outras pessoas. Dessa forma, a massiva quantidade de dados gerados a todo momento abre espaço para diversos tipos de pesquisas analíticas e de interpretação (DOSHI et al., 2017).

Com isso, existem contas no Twitter de jogos digitais que utilizam seus perfis para divulgarem com mais facilidade seus produtos, tanto o jogo em si quanto os pacotes vendidos dentro do próprio jogo, bem como divulgam eventos. Além disso, o Twitter permite anúncios personalizados para obter o maior impacto possível em consumidores específicos (ADIBI; MAJIDI; ESHGHI, 2018).

Contudo, este trabalho utiliza dados disponibilizados pelo Twitter para realizar algumas análises como, por exemplo, quais são os jogos com mais seguidores na rede social, frequência de tuítes postados por ano, entre outras. Ademais, em trabalhos futuros um dos objetivos é realizar análise de sentimento com base nos tuítes postados pelos jogos e seus seguidores.

Figura 1 – Perfil Epic Games Store no Twitter.

Fonte: <https://twitter.com/EpicGames>.

A Figura 1 demonstra alguns dados que podem ser coletados de um perfil no Twitter. Sendo assim, pode-se coletar o nome de usuário, uma breve descrição do perfil, a data que ingressou na rede social, a cidade que está localizado, um site que o perfil possa ter, a quantidade de usuários seguindo e seguidores. Além desses dados de perfil, existem os dados dos tuítes que, inclusive, compõem o dataset deste trabalho, como a quantidade de curtidas, respostas, citações, retuítes e o próprio tuíte.

2.3.2 OpenCritic

Lançado em setembro de 2015, o OpenCritic é um site que agrega análises de jogos digitais, bem como gera pontuações com base em várias análises feitas em outras plataformas online por pessoas/empresas com veredicto e compila tudo em uma página. Dessa forma, seu objetivo é auxiliar consumidores a tomar decisões mais assertivas na hora da compra de um jogo (OPENCRITIC, 2022). A Figura 2 demonstra uma visão geral dos dados gerados pelo OpenCritic acerca de um determinado jogo, ou seja, sua média final, resumo das principais avaliações, plataformas disponíveis e nome do jogo.

Além disso, as próprias publicações podem enviar revisões utilizando o sistema de gerenciamento de conteúdo do OpenCritic. Ademais, o OpenCritic utiliza o conceito de web scraping para verificar novas avaliações de 15 em 15 minutos e quando identificado uma nova avaliação, os metadados necessários são coletados.

Por fim, de acordo com OpenCritic (2022), a avaliação é baseada na classificação percentil da média dos principais críticos de cada jogo, como demonstrado abaixo e na Figura 2:

- **Poderoso**: Jogos com média no percentil 90 ou acima;
- **Forte**: Jogos com média no percentil 60 a 90;
- **Razoável**: jogos com média no percentil 30 a 60;

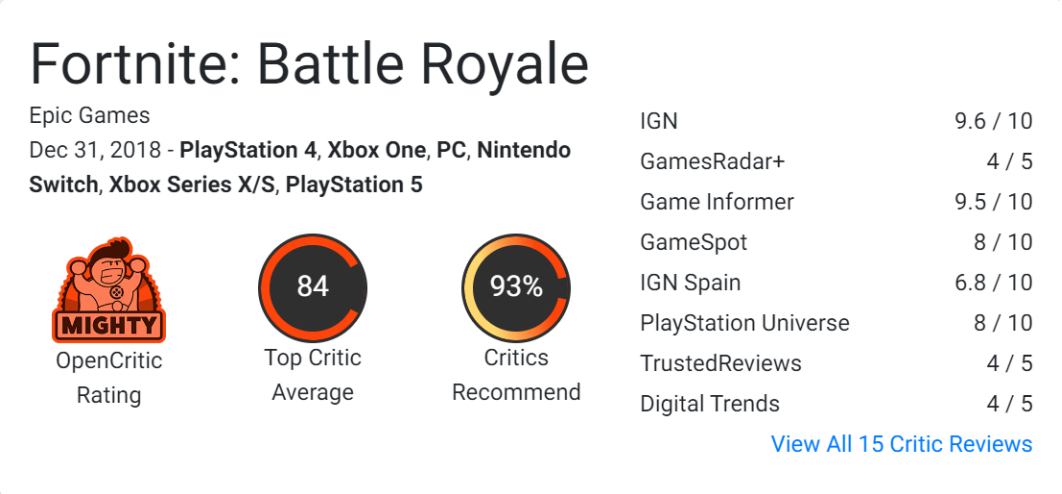


Figura 2 – Visão geral sobre Fortnite no OpenCritic.

Fonte: <https://opencritic.com/game/6228/fortnite-battle-royale>.

- **Fraco**: Jogos com média nos últimos 30 por cento dos jogos.

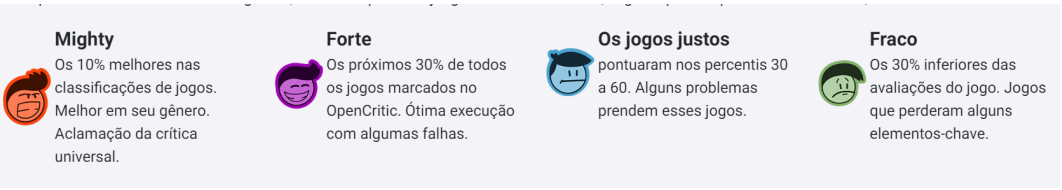


Figura 3 – Selos de classificação OpenCritic.

Fonte: <https://opencritic.com/>.

Este trabalho utiliza a API pública disponibilizada pelo OpenCritic para coletar todos as avaliações dos jogos da Epic Games Store disponíveis na plataforma. Dessa forma, foi possível verificar se os jogos da Epic Games possuem boas avaliações, a média de avaliações por jogo e a porcentagem de avaliações com veredicto.

2.4 Conceitos e Definições de Dataset

A inclusão de dados distintos em vários formatos de diferentes comunidades requer um melhor entendimento do conceito de conjunto de dados e dos principais conceitos relacionados, como formato, codificação e versão. Contudo, existem 4 características que podem ser utilizadas para definir um conjunto de dados: agrupamento, conteúdo, relacionamento e propósito (RENEAR; SACCHI; WICKETT, 2010).

Devido à onipresença da tecnologia digital nos processos de construção acadêmicos, cientistas de todo o mundo buscam abordar sobre a necessidade do aumento ao acesso global dos dados. Assim, é possível construir e compartilhar conhecimento de forma exponencial (SALES; SAYÃO, 2019).

Contudo, ao analisar grandes conjuntos de dados é possível revelar padrões, tendências e associações, principalmente aquelas que de alguma forma interferem indivíduos e empresas com o intuito de tomar alguma decisão. Portanto, não é tão simples extrair informações de *big data*, é preciso ter um bom planejamento para oferecer as melhores entradas possíveis e abstrair as informações necessárias (TALEB; DSSOULI; SERHANI, 2015).

2.4.1 Terminologia utilizada

No âmbito da ciência contemporânea, os dados de pesquisas estão ressurgindo como agentes principais para busca de novos conhecimentos. Isto ocorre devido à evolução das tecnologias que permitem a aparição do big data, que baseia-se na coleta, geração, processamento e análise de grandes quantidades de dados estruturados (SALES; SAYÃO, 2019).

Além do objetivo de simplesmente reunir uma grande quantidade de dados, os *datasets* são criados para agregar de alguma forma com a atividade científica. Isso pode ocorrer ao fornecer evidências para se analisar, refutações ou até mesmo confirmação de hipóteses já existentes (RENEAR; SACCHI; WICKETT, 2010). No entanto, ao contrário das publicações, os dados são heterogêneos e diversos, gerados para diferentes fins, diferentes tecnologias e áreas temáticas específicas. Devido a isso, é notório que existe uma dificuldade em gerenciar esses ativos informacionais (SALES; SAYÃO, 2019).

2.4.2 Processo de coleta de dados

Normalmente, o processo de coleta de dados é pensado para estruturar melhor uma gama de informações que estão dispersas em algum tipo de plataforma com o intuito de resolver possíveis problemas em um determinado campo (TALEB; DSSOULI; SERHANI, 2015). Dessa forma, principalmente com o avanço tecnológico, é possível realizar essa coleta de dados de algumas maneiras que facilite a filtragem de informações posteriormente (TIKITO; SOUISSI, 2020).

Diante desse contexto, podem ser citados alguns exemplos de aplicações que utilizam coleta de dados para compor sua pesquisa e também como é realizado. Esses exemplos serão descritos em detalhes na Seção seção 2.5. O primeiro exemplo é de um dataset construído por O'Neill et al. (2016) que visa coletar dados de usuários e jogos da Steam para auxiliar os pesquisadores a caracterizarem de forma mais precisa e direta o comportamento dos jogadores. Assim, o processo de coleta de dados passou por uma única etapa: Requisições REST da API pública da própria Steam que disponibiliza algumas informações relevantes para a pesquisa desses autores. Um outro exemplo é de um dataset criado por Vagavolu et al. (2021) que tem como objetivo facilitar a pesquisa de desenvolvedores de jogos.

Dessa forma, o processo de coleta de dados passou por duas etapas: A primeira foi por meio de mineração de dados do Github (Web Scraping), buscando algumas informações de repositórios da plataforma. E a segunda foi através da API REST do GitHub que disponibiliza informações não tão acessíveis sobre seus repositórios.

Por fim, um último exemplo é de um dataset criado por Zangerle et al. (2014) com a finalidade de alavancar as mídias sociais com a criação de um conjunto de dados diversificado e constantemente atualizado, que descreve o comportamento de escuta musical dos usuários. Para o processo de coleta de dados neste projeto, os autores passaram por algumas fases, sendo elas: coleta de dados da API do Twitter para obter tweets básicos, extração de trilha e artistas por meio dos tweets coletados e extração de dados da API do Spotify.

Em síntese, pode-se perceber um ponto em comum em todos esses exemplos. Todos os trabalhos utilizam de alguma API de uma plataforma disponível na Web para a coleta de informações mais precisas de uma entidade desejada e necessária na pesquisa. A partir disso, os filtros para direcionamento do objetivo são realizados.

2.4.3 Pré-processamento e filtragem de dados

A mineração de dados se tornou foco de várias aplicações envolvendo *Big Data*. Sendo assim, o pré-processamento dos dados é uma etapa importante durante este processo (HUANG et al., 2020). O tamanho, a velocidade e os formatos em que os dados são gerados e processados afetam a qualidade geral das informações. Portanto, a Qualidade do Big Data (QBD) se tornou um parâmetro determinante para assegurar que a qualidade dos dados seja mantida em todas as fases do processamento do Big Data (TALEB; DSSOULI; SERHANI, 2015).

Geralmente, as principais características do Big Data são descritas como: Volume, Variedade, Velocidade e Veracidade e comumente conhecido como definição “4Vs” de Big Data (TALEB; DSSOULI; SERHANI, 2015). Em sistemas de Big Data, os dados são a fonte final de conhecimento. Em seu ciclo de vida, os dados percorrem quatro diferentes fases como mostra a Figura 4: geração de dados, aquisição de dados, armazenamento de dados e análise de dados.

O processo de mineração de dados consiste na extração de informações relevantes de grandes quantidades de dados. Dessa forma, é preciso preparar os dados fornecendo conjuntos qualificados, que normalmente necessita de um pré-processamento, bem como descobrir regras por meio de métodos de mineração de dados. Essas regras envolvem principalmente classificação, análise de regressão, agrupamento e regras de associação (HUANG et al., 2020).

A etapa de pré-processamento é obrigatória e essencial para refinar e avaliar os

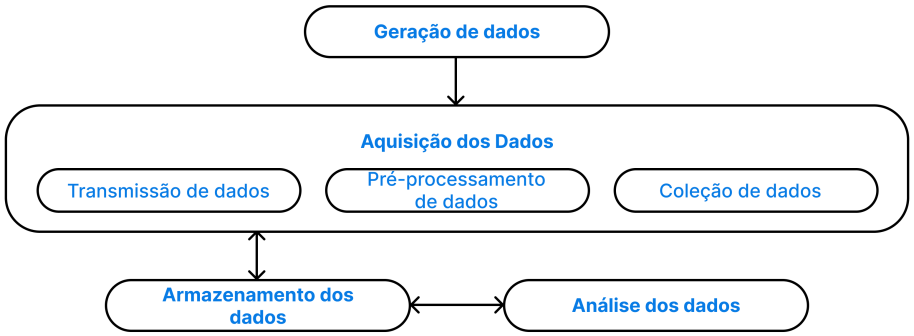


Figura 4 – Ciclo de vida de um Big Data.

Fonte: (TALEB; DSSOULI; SERHANI, 2015).

dados. Outras tarefas importantes de pré-processamento, como integração de dados e mesclagem de várias fontes heterogêneas, também estão ocorrendo e têm um impacto considerável nos dados transformados resultantes. Para rastrear o valor e a relevância dos dados, e a gravidade do impacto das transformações de pré-processamento mencionadas, é necessário um conceito de qualidade de dados (TALEB; DSSOULI; SERHANI, 2015). Além disso, existe uma correlação entre as informações obtidas nas análises e a qualidade do *dataset*, ou seja, uma classificação de alta qualidade obrigatoriamente depende de um conjunto de dados de muita qualidade (HUANG et al., 2020).

2.4.4 Importância da construção de um conjunto de dados

Está se tornando cada vez mais difícil e complexo explorar e analisar grandes quantidades de dados. A mudança geralmente ocorre mais rápido do que a capacidade de aprender e entender. Da mesma forma, há uma necessidade crescente de encontrar mecanismos para se comunicar de forma eficaz e eficiente com as pessoas e ajudá-las a entender os processos e conceitos que afetam a vida cotidiana (COSTA, 2017).

Construir um conjunto de dados de qualidade é de grande relevância para análise e extração de informações. Dessa forma, a mineração de dados ajuda a lidar com o alto fluxo de informações criadas dia após dia (HUANG et al., 2020). A vantagem da exploração de dados de forma visual é que o usuário está diretamente envolvido no processo de mineração de dados. Há um grande número de técnicas de visualização de informações que foram desenvolvidas ao longo dos últimos anos para apoiar a exploração de grandes conjuntos de dados (COSTA, 2017).

Com o avanço da tecnologia e a grande quantidade de informações sendo construída todos os dias, surge a necessidade de construção de um conjunto de dados em todas as áreas do conhecimento (COSTA, 2017). A seguir serão listados exemplos de aplicações que utilizam um conjunto de dados e possíveis aplicações:

- Conjunto de dados para auxiliar pesquisadores a caracterizar de forma mais precisa e diretamente o comportamento de jogadores de jogos disponíveis na loja da Steam (O'NEILL et al., 2016). Possíveis aplicações: trabalhos relacionados à análise de sentimento, análise de vícios em jogos e redes sociais;

- Dataset que armazena dados de 536 repositórios do GitHub sobre motores de jogos para facilitar a pesquisa dos desenvolvedores de jogos (ZANGERLE et al., 2014). Possível aplicação: trabalho focado em análise de frameworks de motores de jogos com o objetivo de verificar qual está sendo mais utilizado no momento e filtrar por tipo de jogo desenvolvido pela tecnologia;

- Dataset que busca alavancar as mídias sociais para a criação de um conjunto de dados diversificado e constantemente atualizado, que descreve o comportamento de escuta musical dos usuários (ZANGERLE et al., 2014). Possível aplicação: pesquisa que aprofunde na análise de músicas destaque no momento ou um trabalho que estude a influência da rede de amigos no Twitter em escolhas de músicas dos usuários.

2.5 Trabalhos Relacionados

Esta seção descreve os trabalhos relacionados que contribuíram como base para melhor entendimento de datasets, sua construção, coleta e análise de dados, conceitos estes os quais são utilizados neste projeto.

2.5.1 Plataforma de Jogos Steam

Estima-se que 60% dos estadunidenses jogam videogames, o que gera receita anual de mais de 25 bilhões de dólares só em jogos para PC (O'NEILL et al., 2016). Sendo assim, existem vários trabalhos focados na caracterização de jogadores que compõem esse mercado, examinando fatores como idade, gênero e status econômico. Sendo assim, o trabalho de O'Neill et al. (2016) tem como objetivo construir um dataset para auxiliar pesquisadores na caracterizarem mais precisa e direta do comportamento de jogadores. Diante disso, são coletadas informações de mais de 108 milhões de contas de usuários da Steam e mais de 384 milhões de jogos da plataforma, essa plataforma apresenta uma grande variedade de usuários com comportamentos diferentes.

Com o objetivo de coletar dados relacionados a informações de perfis, amizades, jogos, tempo jogado, grupos de associações e perfil dos usuários, foi utilizada a API REST disponibilizada pela Steam. Por conseguinte, para a validação dos dados, os autores passaram quase uma semana verificando manualmente se os dados sobre os usuários, que são públicos no próprio site da Steam, são realmente de contas reais. Sendo assim, a verificação foi feita analisando o nome, amigos e postagens no perfil público dos jogadores.

Por fim, a análise de dados permitiu aos autores identificar que a rede social da Steam se trata de uma distribuição de cauda pesada, ou seja, são uma classe de probabilidade de distribuições cujas caudas não são limitadas exponencialmente. Esse tipo de distribuição descreve comportamentos em que a probabilidade de observar um evento extremo é mais provável do que seria sugerido por uma distribuição gaussiana (normal). Dessa forma, os autores utilizaram a ferramenta *powerlaw 1.3* do Python projetada para classificar de forma empírica dados de cauda pesada para identificar com maior especificidade a natureza precisa dessas distribuições.

A Steam fornece a seus usuários a capacidade de estabelecer links sociais com outros usuários por meio de relações de amizade e afiliações de grupo. Em média, 9 milhões de jogadores $(88,06\%)$ adicionam dez ou menos amigos por ano e apenas 2500 jogadores $(0,02\%)$ adicionam mais de duzentos amigos por ano. Contudo, o número de jogadores em cada ano é muito semelhante, apesar do tamanho crescente da base de usuários da Steam ao longo dos anos, isso indica que o comportamento geral de amizade é independente do tamanho da rede do sistema (O'NEILL et al., 2016).

Também é importante destacar uma determinada situação que ocorre com os jogos, ou seja, os jogadores adquirem vários jogos mesmo que, depois, eles não joguem. Este fenômeno pode ser atribuído tanto à tendência dos jogadores em geral de adquirir jogos que não jogam quanto a um subconjunto "colecionador" específico da população Steam para a qual a aquisição de jogos é um objetivo em si. Ademais, para cada jogo na biblioteca de um usuário do Steam, o tempo de jogo é registrado diretamente de duas maneiras: (a) o tempo total de jogo para aquele jogo desde que foi adicionado à biblioteca do usuário; e (b) um valor de tempo de jogo contínuo abrangendo as duas semanas anteriores ao momento em que a consulta foi emitida. Além disso, essa pesquisa traz uma aproximação com relação ao dinheiro investido pelos os jogadores, já que a Steam não revela diretamente esse valor. Essa média foi assumida sobre o efeito do envelhecimento dos preços de 2014 da Steam, sendo assim, foi encontrado 1,1 milhão de anos cumulativos de tempo de jogo, bem como um valor de mercado de US $\$5.326.471.034,78$ de jogos Steam (O'NEILL et al., 2016).

Embora o Steam seja apenas parcialmente uma rede social, métricas de amizades da Steam podem ser aproximadas às métricas que foram realizadas em outras redes sociais. O principal resultado que pode ser observado é que a Steam é melhor caracterizada como uma rede de amigos, ao invés de uma rede onde contas de celebridades podem reunir um grande número de seguidores. Todas as amizades do Steam são recíprocas, a maioria dos usuários têm poucas amizades e o número máximo de amigos é limitado por várias políticas da Steam.

Em suma, as vantagens desse trabalho são os dados levantados relacionados aos usuários e jogos da Steam. Os autores chegam a comparar a Steam com uma outra rede

social comum relacionando o número de amizades com o tempo gasto em jogos *multiplayer*[1]. Além do tempo gasto na plataforma, os autores ainda trazem um valor aproximado em que cada jogador gera em dólares para a Steam, o que é extremamente importante para o mercado de jogos. Em contraste, a limitação do projeto está na caracterização de jogadores individuais, ou seja, que não possuem muitos amigos em sua rede de amizades da Steam e que passam mais tempo em jogos que necessitam de apenas um jogador.

Para concluir, a grande quantidade de jogos e jogadores da plataforma revelam uma enorme diversidade de comportamento entre os jogadores. Por fim, este trabalho pode ser aproveitado em projetos que visam análise de sentimento e análise de tempo e recursos monetários investidos em plataformas de jogos.

2.5.2 Dataset de Motores de Jogo *Open-Source*

Sobretudo, *game engines* são estruturas que fornecem uma plataforma para que os desenvolvedores criem jogos com uma interface feita sob medida para lidar com a complexidade do desenvolvimento de jogos. Embora haja uma extensa pesquisa empírica sobre estruturas de software, há uma necessidade de estudos empíricos sobre *game engines*, pois eles diferem dos frameworks de software tradicionais (VAGAVOLU et al., 2021).

Houve um aumento de mais de 385 mil jogos desenvolvidos durante a última década no Google Play Store a partir do ano 2020 (VAGAVOLU et al., 2021). Considerando o esforço relacionado ao desenvolvimento de recusos complexos a partir do zero, os desenvolvedores buscam por motores de jogos para um desenvolvimento mais rápido e sofisticado. Sendo assim, o objetivo deste trabalho é construir um dataset que armazena dados de 536 repositórios do GitHub para facilitar a pesquisa dos desenvolvedores de jogos. Sendo assim, os desenvolvedores teriam com mais acessibilidade informações sobre ciclo de lançamento, qualidade de código, usabilidade da API e lançamentos em breve.

Com a finalidade de construir o dataset, os autores passaram por duas fases: (1) mineração do Github e (2) coleta de dados na API do GitHub. Na primeira etapa, eles encontraram 3417 repositórios sobre *engine game*. Em seguida, os autores aplicaram alguns filtros nos resultados encontrados a fim de recuperar apenas os repositórios relevantes. Sendo assim, os filtros aplicados foram:

- Remoção de repositórios com menos de 1 estrela. As estrelas no GitHub significam salvar um repositório como favorito;
- Remoção de repositórios com rótulos de “listas”, “tutoriais”, “cursos” e “recursos” com a ideia de retirar repositórios com um falso positivo, assim, os autores alcançaram 990 repositórios;

[1] Jogos multiplayer, ou multijogador, são jogos que permitem que vários jogadores participem simultaneamente de uma mesma partida.

- Análise manual de cada repositório para remoção daqueles que não tinham o foco desejado, como resultado os autores conseguiram 656 repositórios;
- Por fim, foram removidos todos os repositórios com dados corrompidos, restando apenas 526 repositórios.

Na segunda fase da criação do dataset, a API do Github foi utilizada para obter acesso a informações não acessíveis anteriormente. Dessa forma, as informações recuperadas foram sobre as versões dos repositórios, a saber: *commits*, *issues*, *pull requests*, comentários nas *issues*, comentários nos *pull requests*, entre outras. Para cada informação citada anteriormente, foi criada uma entidade do banco de dados para armazenamento das mesmas.

Certamente, o trabalho facilita a pesquisa dos desenvolvedores no momento em que buscam por frameworks para trabalharem. Pois, além de terem acesso a vários repositórios relevantes do GitHub, cada repositório contém uma descrição objetiva sobre como os frameworks foram desenvolvidos e como podem ser utilizados. Além disso, ao acessar o dataset, é possível, por exemplo, filtrar os repositórios por estrelas recebidas, *forks*, descrição, entre outros. Porém, uma limitação do trabalho é que ele possui somente 7 arquivos descrevendo os repositórios e há a possibilidade de alguns repositórios estarem com ruídos nas informações, mesmo após a realização da filtragem dos dados.

Portanto, é notório a evolução do mercado de jogos e a qualidade do seu desenvolvimento impacta diretamente no mesmo. Sendo assim, esse trabalho poderia ser aproveitado para análise de frameworks de *engine games*, ou seja, qual está sendo mais utilizado no momento, qual o objetivo de cada framework e que tipo de jogo é ou pode ser desenvolvido com ele.

2.5.3 Rede Social do Youtube

Atualmente, o Youtube é o maior provedor de conteúdo em vídeo do mundo, uma plataforma que se tornou importante na divulgação de conteúdo multimídia. Um dos grandes diferenciais da plataforma em relação a outras emissoras de conteúdo tradicionais é a experiência social de usuário para usuário (WATTENHOFER; WATTENHOFER; ZHU, 2012). Sendo assim, esse trabalho tem como objetivo criar três conjuntos de dados para realizar análises com base em assinaturas, atividades e métricas de conteúdo enviadas pelos usuários a fim de entender o Youtube como uma rede social.

Dessa forma, a pesquisa foi dividida em duas categorias: (*i*) medições da Online Social Network (OSN) e (*ii*) aplicativos OSN baseados em aprendizado de máquina. Na primeira categoria, Wattenhofer, Wattenhofer e Zhu (2012) aproveitam trabalhos de outros autores que já haviam coletado informações na API do Youtube para obterem acesso

as métricas relacionados à popularidade dos vídeos. Com isso, eles complementam a pesquisa utilizada com métricas de conteúdo com topologia da rede social. Ademais, foi possível fazer a conexão entre a popularidade do conteúdo de vídeo e a popularidade social correspondente. Já na segunda categoria, os autores utilizam de várias métricas topológicas de rede calculadas a partir do grafo social em seu aplicativo por meio de coleta de dados, principalmente por MapReduce[2] em Python. Assim, eles basearam suas análises em três conjuntos de documentos principais: (*a*) o grafo social explícito (que descreve as assinaturas); (*b*) o grafo social implícito (que descreve atividades de comentários); e, por fim, (*c*) as métricas agregadas de conteúdo enviado pelo usuário.

Uma infinidade de trabalhos mostrou vários OSNs principais por meio de rastreamentos online e/ou uso de API. No entanto, poucos projetos de medição capturaram o grafo social completo sem compromisso (WATTENHOFER; WATTENHOFER; ZHU, 2012). Esse trabalho aproveita os dados e a capacidade de computação disponíveis no Google para pesquisar e obter *insights* sobre uma importante plataforma social. Os autores redigiram um grafo direcionado para representar as relações de assinatura de usuários registrados do YouTube. Cada nó representa um usuário, enquanto um link aponta de um assinante para o usuário inscrito. Portanto, este grafo é composto por usuários cadastrados que se inscreveram em pelo menos um canal ou receberam pelo menos uma inscrição em seu respectivo canal. Da mesma forma, o grafo de comentários é composto por usuários que postaram ou receberam pelo menos um comentário. Da mesma forma, os links apontam do comentarista para o usuário que recebe o comentário. Ambos os grafos contêm nós da ordem de centenas de milhões e links da ordem de bilhões, comparáveis às medidas de rede do Twitter.

Uma das vantagens desse trabalho é a análise feita por meio de medição e cálculo de recursos adicionais que representam a popularidade social e de conteúdo. Com isso, os autores demonstram em seus grafos como um usuário com mais assinantes obtém "influência" de um maior número de assinantes. No entanto, como este é um dos primeiros trabalhos que tratam de OSN, existem algumas limitações. Algumas delas são citadas pelos próprios autores e que inclusive podem ser feitas no futuro em possíveis novas aplicações. Como por exemplo, utilização de métricas de computação intensiva, como as que envolvem redes de dois saltos (*hops*), as quais não foram coletadas devido ao tamanho do conjunto de dados. Por fim, esse trabalho relacionado apresenta três conjuntos de dados em grande escala para melhor compreender a natureza da rede social do Youtube (descrevem as assinaturas, atividades e as métricas de conteúdo enviadas pelo usuário). Comparado às pesquisas recentes, os autores apontam este como um dos mais abrangentes estudos de medição de um grande OSN até o momento (WATTENHOFER; WATTENHOFER; ZHU, 2012).

[2] Modelo de programação com objetivo de processar grandes quantidades de dados em paralelo, dividindo o trabalho em tarefas menores e independentes.

2.5.4 Dataset de Músicas Online

A extração de informações de redes sociais online se tornou popular tanto na indústria quanto no meio acadêmico, uma vez que essas fontes de dados permitem aplicações inovadoras. No entanto, na área de sistemas de recomendação de música e recuperação de informação musical, os respectivos dados dificilmente são explorados (ZANGERLE et al., 2014). Dessa forma, o trabalho de Zangerle et al. (2014) teve como objetivo a criação de um dataset que alavanque as mídias sociais para a criação de um conjunto de dados diversificado e constantemente atualizado, que descreve o comportamento de consumo de música online de usuários.

Com a finalidade de construir o dataset, os autores passaram por algumas fases no desenvolvimento, sendo elas: (1) uso da API do Twitter; (2) coleta de tweets; (3) coleta da música e do artista; (4) coleta de dados do spotify; e, por fim, (5) conversão dos dados para formato de arquivo RDF[3] utilizaram a API pública do Twitter para filtrar tweets que continham as tags: "nowplaying", "listento" e "lisnteningto". Em seguida, na segunda fase, eles coletaram a data e horário que o tweet foi enviado, o serviço usado para publicar o tweet e o nome de usuário do responsável pelo tweet. Depois isso, na terceira fase o objetivo era combinar artistas e músicas que ocorrem nos tweets com as entradas correspondentes no banco de dados MusicBrainz[4]. Na quarta fase, um segundo extrator foi utilizado para aumentar a qualidade e quantidade de eventos de músicas consumidas no conjunto de dados #nowplaying, o Spotify Extractor, que aproveita os tweets que foram enviados através do Spotify. Por fim, na última fase foi feita a estruturação dos dados, com relação à descrição dos dados musicais foi utilizada a Ontologia Musical. Além disso, também foi incorporado elementos Dublin Core[5] para descrever metadados de eventos de escuta.

O número médio de novos tweets coletados por dia é de 61.985,67, sendo que, a mediana igual a 59.876 e o desvio padrão igual a 19.717,72. O artista mais popular no conjunto de dados foi a Rihanna com 237.108 vezes. Contudo, o número médio de eventos de escuta por artista é 333,54 e o valor mediano da distribuição de dados é 8 (desvio padrão igual 3.175,49). Da mesma forma, a distribuição da popularidade das faixas também apresenta um cauda longa. Para um total de 1.206.499 faixas de música, o número médio de ocorrências de cada faixa de música é 38,15, enquanto a mediana é 3 (desvio padrão igual a 503,57). Esta distribuição é fortemente cauda longa e enviesada para a esquerda, o que é comum para conjuntos de dados de música online. Quanto às fontes usadas para publicar #nowplaying tweets, Securenet Systems Radio Playlist Update (uma plataforma para fornecer *streaming* de rádio), o aplicativo Spotify, o site do Twitter e

[3] Documento escrito na linguagem *Resource Description Framework* que é usado para representar informações sobre um site em formato estruturado.

[4] Site oficial: <https://musicbrainz.org/>.

[5] Modelo de metadados que tem como objetivo descrever objetos digitais como vídeos, sons, imagens, textos e sites.

o cliente Twitter para iPhone estão entre as fontes mais populares (ZANGERLE et al., 2014).

Um dos pontos fortes desse trabalho é que ele apresenta uma base de pesquisa bastante interessante para trabalhos futuros, pois, com esses dados coletados é possível verificar, por exemplo, quais músicas estão em alta e quais estão sendo mais compartilhadas. Ademais, é possível verificar se há influência musical entre amigos no Twitter, ou seja, se uma música compartilhada por um usuário influenciar que alguém que o siga na rede social escute a música também. No entanto, como esta pesquisa é apenas o início de um trabalho, existem algumas limitações. Como por exemplo, a incorporação de outras fontes para inclusão no trabalho (além do Spotify) para garantir maior assertividade nos resultados. Portanto, nesse trabalho é apresentado o conjunto de dados #nowplaying que demonstra eventos de escuta que incorporam informações sobre os usuários ouvindo certas faixas de música e artistas ao longo de um período de dois anos. Essas informações foram extraídas do Twitter por meio de tweets, no total, foram 49.921.024 eventos de consumo de música online públicos de 4.150.615 usuários, contendo 144.011 artistas e 1.346.203 faixas. Contudo, o trabalho está em constante crescimento à medida que é atualizado diariamente.

2.5.5 Dataset YouTube-8M

Muitos avanços recentes em Visão Computacional são atribuídos a grandes conjuntos de dados. Pacotes de software de código aberto para aprendizado de máquina comum e barato reduziram a barreira de entrada para explorar novas abordagens em grande escala Dessa forma, é possível treinar modelos em milhões de exemplos em poucos dias. Embora existam conjuntos de dados em grande escala para a compreensão de imagens, como o ImageNet, não há conjuntos de dados de classificação de vídeo de tamanho comparável. Sendo assim, o trabalho desta seção apresenta o YouTube-8M, o maior conjunto de dados de classificação de vídeo até o momento de seu lançamento, composto por 8 milhões de vídeos, sendo 500 mil horas de vídeo (ABU-EL-HAIJA et al., 2016).

Para obter os vídeos e seus vários rótulos, foi utilizado um sistema de anotação do próprio YouTube, que rotula os vídeos de acordo com seus tópicos principais. Embora os rótulos sejam gerados por máquinas, eles têm alta precisão e são derivados de uma variedade de sinais, incluindo metadados e sinais de clique de consulta. Assim, eles representam um excelente alvo para abordagens de anotações baseadas em conteúdo. Além disso, Abu-El-Haija et al. (2016) filtraram os rótulos de vídeo usando estratégias de curadoria automatizadas e manuais, incluindo testes com pessoas para verificarem se os rótulos eram visualmente reconhecíveis. Por conseguinte, cada vídeo foi decodificado a um quadro por segundo utilizando um *Deep Convolutional Neural Networks* (DCNN) pré treinado em ImageNet para extrair a representação oculta imediatamente antes da camada de

classificação. Finalmente, os recursos de moldura foram comprimidos e disponibilizados para download. O conjunto de dados contém recursos de nível de quadro por mais de 1,9 bilhão de quadros de vídeo e 8 milhões de vídeos, o que o torna o maior conjunto de dados de vídeo multi-rótulo público.

Foram seguidos dois princípios principais ao projetar o vocabulário do dataset, sendo eles: (1) cada rótulo no conjunto de dados deve ser distinguível usando apenas informações visuais; (2) cada rótulo deve ter um número suficiente de vídeos para modelos de treinamento e para calcular métricas confiáveis no conjunto de teste. Para o primeiro, foi utilizado uma combinação de tópicos selecionados manualmente e avaliações humanas para reduzir o vocabulário em um conjunto visual. Para o último, foi considerado apenas entidades com pelo menos 200 vídeos no conjunto de dados.

Com o vocabulário inicial formulado, os seguintes passos foram seguidos para obter os vídeos:

- Coleta de todos os vídeos correspondentes às 10.000 entidades visuais e com pelo menos 1.000 visualizações, utilizando o sistema de anotação de vídeos do Youtube. Exclusão de vídeos muito curtos (menor do que 120 segundos) ou muito longos (maior do que 500 segundos);
- Amostragem aleatória de 10 milhões de vídeos entre eles;
- Obtenção de todas as entidades para a amostra de 10 milhões de vídeos usando sistema de anotação de vídeo do Youtube, o que completa as anotações;
- Entidades filtradas com menos de 200 vídeos e vídeos sem unidades restantes, o que reduz o tamanho dos dados para 8.264.650 vídeos;
- Divisão dos vídeos em três partições: treino, teste e validação com proporções de 70%, 20% e 10%, respectivamente.

Na pesquisa foram treinados vários modelos de classificação no conjunto de dados usando métricas de avaliação populares e que foram relatados como linhas de base. Apesar do tamanho do conjunto de dados, alguns dos modelos são treinados para convergir em menos de um dia em uma única máquina usando o framework Tensor Flow[6] disponível ao público.

Esperando que este trabalho seja útil para o desenvolvimento de compreensão de vídeo e técnicas de aprendizagem e representação, os autores apontam algumas vantagens em relação aos trabalhos relacionados:

[6] Site oficial: <https://www.tensorflow.org/>.

- Uma anotação de vídeo em grande escala e aprendizagem de representação, refletindo os temas principais de um vídeo;
- Um salto significativo no número e diversidade de classes de anotações - 4800 entidades do Mapa de Conhecimento versus menos de 500 categorias para todos os outros conjuntos de dados;
- Um aumento substancial no número de vídeos marcados - mais de 8 milhões de vídeos, sendo mais de 500 mil horas de vídeo;
- Disponibilidade de recursos de última geração pré-computados para 1,9 bilhões de quadros de vídeo.

Por fim, nesse trabalho foram abordados dois desafios principais: (1) coletar um grande conjunto de dados de vídeo rotulado, com rótulos de qualidade razoável; e (2) remover barreiras computacionais por pré-processamento de conjunto de dados fornecendo recursos de nível de quadro de última geração. Sendo assim, foram processados mais de 50 anos de vídeo e fornecidos recursos para quase 2 milhões de frames de mais de 8 milhões de vídeos, o que permite treinar um modelo razoável nesta escala dentro de 1 dia.

Tabela 1 – Resumo dos trabalhos relacionados.

Referência	Objetivo	Metodologia	Vantagens	Desvantagens	Possíveis Aplicações
(O'NEILL et al., 2016)	Construir um dataset para para auxiliar pesquisadores a caracterizarem de forma mais precisa o comportamento dos jogadores.	Coleta de dados da API da Steam e uso de uma ferramenta do Python para classificação empírica.	Os dados levantados relacionados aos usuários e jogos da Steam e a comparação feita com uma rede social comum.	Existe uma limitação na caracterização de jogadores individuais.	Pode ser utilizado para análise de sentimento, trabalhos relacionados a vícios em jogos e redes sociais.
(VAGAVOLU et al., 2021)	Construir um dataset que armazena dados de 536 repositórios do GitHub para facilitar a pesquisa dos desenvolvedores de jogos.	Mineração do Github utilizando alguns filtros e coleta de dados da API do Github.	Facilita a pesquisa dos desenvolvedores na busca de frameworks para desenvolvimento de jogos.	Mesmo após o refinamento, é possível que alguns repositórios estejam com ruídos.	Análise de frameworks, ou seja, qual está sendo mais utilizado no momento, qual o objetivo de cada um, etc.
(WATTENHOFER; WATTENHOFER; ZHU, 2012)	Examinar o aspecto da rede social do Youtube medindo o grafo de assinatura em escala real, o grafo de comentários e o corpus de conteúdo de vídeo.	Coleta de dados da API do Youtube e adição de métricas topológicas calculadas a partir do gráfico social.	A pesquisa demonstra como um usuário com mais assinantes obtém "influência" de um maior número de assinantes.	Falta de utilização de métricas de computação intensiva, como as que envolvem redes de 2 saltos.	A pesquisa pode ser aprofundada a partir dos pontos fracos citados.
(ZANGERLE et al., 2014)	Desenvolver um dataset que alavanque as mídias sociais e descreva o comportamento de escuta musical dos usuários.	Rastreio na API do Twitter, extração básica de tweets, extração de músicas e artistas, extração de dados do Spotify e conversão dos dados para documento RDF.	É possível verificar quais músicas estão em alta e sendo mais compartilhadas no momento.	Falta de incorporação de outras plataformas de música além do Spotify.	Análise de músicas destaque do momento e a influência da rede de amigos em escolhas de músicas.
(ABU-EL-HAIJA et al., 2016)	Construir o maior dataset de compreensão de imagens de vídeo e aprendizagem de representação em grande escala.	Utilizou-se um sistema de anotação de rótulos do Youtube e filtros de curadoria automatizada e manual.	Anotação de vídeo em grande escala; salto significativo no número e diversidade de classes de anotações; um aumento substancial no número de vídeos marcados.	Uma limitação do trabalho é a falta de recursos para análise de áudio e movimento, no qual os autores pretendem investir no futuro.	A pesquisa pode ser utilizada para acelerar trabalhos sobre compreensão de vídeo. Como por exemplo, na classificação de atividade no dataset do ActivityNet.

2.6 Considerações Finais

A Tabela 2.5.5 apresenta um resumo dos trabalhos relacionados e suas principais características, tais como objetivos, metodologia aplicada, vantagens, desvantagens e possíveis aplicações. É notório que as vantagens destes trabalhos destacam a importância da coleta dos dados para resultar em informações relevantes para pesquisas acadêmicas e/ou de mercado. Sendo assim, é possível utilizar o conjunto de dados para construir aplicações que façam análise de músicas destaque no momento ou análise de sentimento de usuário de uma determinada plataforma, por exemplo. Este projeto tem em comum com os trabalhos relacionados citados nessa tabela é que também utiliza de uma API pública para criação do dataset, nesse caso a API do Twitter e da OpenCritic[7]. Além disso, realiza a análise e sumarização dos dados coletados para que sirva de base para possíveis pesquisas e aplicações utilizando esta base de dados.

Este capítulo apresenta uma revisão bibliográfica para jogos online, sua evolução e alguns tipos de jogos online existentes no mercado. Pode-se observar que, o crescimento dos jogos online foi proporcional com a evolução da Internet e hoje se destaca como um dos principais meios de entretenimento. Além disso, foram apresentadas as principais plataformas de jogos online da atualidade, incluindo a Epic Games Store, a qual é foco deste trabalho. Nessas plataformas, observa-se que todas prezam por comunidades em mídias sociais, inclusive incorporadas em suas próprias plataformas. No entanto, percebe-se que a Epic Games ainda não é tão forte nesse quesito se comparada com a plataforma Steam, por exemplo, por não apresentar a possibilidade de interações de usuários em sua plataforma própria. Um outro ponto interessante para pessoas que gostam de jogos online, é a facilidade que essas plataformas permitem que os usuários adquiram seus jogos, muitas vezes alguns jogos estão disponíveis de forma gratuita.

Neste capítulo também foram abordadas APIs para coleta de dados de jogos de plataformas digitais. Neste trabalho são utilizadas duas APIs, uma do Twitter e outra do OpenCritic, plataforma que a Epic Games utiliza como referência para avaliações de seus jogos. Como foi descrito na subseção 2.2.4 sobre APIs, das plataformas de jogos citadas somente a Epic Games não possui uma API pública para acesso de dados de seus jogos e usuários. Em razão desse fato, a construção do dataset deste trabalho é realizada por meio da técnica de *webscraping* no site da própria loja da Epic Games, o que destaca a importância deste trabalho, o Capítulo 3 apresenta detalhes da abordagem utilizada para criação do dataset deste projeto.

Por fim, também foram discutidos conceitos de definições de dataset, terminologias, processos para uma coleta de dados, pré-processamento e filtragem de dados, além da importância da construção de um dataset e exemplos de aplicações. Tais conceitos formam

[7] Site oficial: <https://opencritic.com/>.

a base de conhecimento para a construção do dataset da plataforma de jogos da Epic games, suas redes sociais, contas no Twitter, tweets e avaliações desses jogos. O script desenvolvido para coleta dos dados é detalhado no Apêndice A. Sendo assim, as próximas seções demonstram como é feita a pesquisa de quais dados são coletados e de quais fontes eles são obtidos.

3 Desenvolvimento

Este capítulo descreve o desenvolvimento do trabalho que contém informações sobre os documentos desenvolvidos, pesquisas e análises realizadas. Dessa forma, na seção 3.1 é descrito um pouco do impacto da Epic Games na indústria de jogos e como funciona sua recomendação de jogos. Em seguida, será demonstrado os detalhes da construção do dicionário de dados, suas características, entidades e atributos na seção 3.2. Após a construção do dicionário de dados, foi desenvolvido um modelo Entidade Relacionamento que será detalhado e exibido na seção 3.3. Por fim, na seção 3.4 é demonstrado por meio de um fluxograma o funcionamento dos scripts para construção do dataset.

3.1 Plataforma da Epic Games

O mercado de jogos digitais tem se mostrado com maior destaque dentre as indústrias criativas e culturais, tanto em termos financeiros quanto em perspectiva de crescimento para os próximos anos. Em relação ao faturamento, o mercado de jogos eletrônicos já ultrapassaram as indústrias de música e cinema juntos (AMÉLIO, 2018).

A Epic Games, uma das maiores plataformas de jogos digitais do mundo, está revolucionando a indústria do entretenimento, a maneira como jogadores e desenvolvedores criam, publicam e consomem experiência de jogos. Essa empresa é pioneira no desenvolvimento de um ecossistema digital que dispõe da infraestrutura e serviços necessários para jogos em grande escala como o Fortnite (EPAM, 2021). O número de usuários ativos aumentou em 192%, o que gerou aumento nas compras pela plataforma Epic Games Store totalizando mais de US $700 milhões em 2020, no qual os jogos de terceiros representam 37% do valor (GAMES, 2021b).

3.1.1 Recomendações de Jogos

Em um trabalho futuro, um dos objetivos é desenvolver uma aplicação para recomendação de jogos com base no dataset construído neste trabalho. No entanto, esta pesquisa apresenta como é feita a recomendação de jogos na plataforma da Epic Games, o que pode ser útil para o trabalho futuro. Além disso, foi possível identificar quais dados seriam possíveis de coletar para compor o dataset.

A página principal da Epic Games Store é composta por algumas seções que listam os principais jogos do momento. Dentre essas seções estão: jogos em promoção, jogos gratuitos, novos lançamentos, mais vendidos, lançamento em breve, jogos atualizados

recentemente, jogos novos na loja, mais populares e a primeira seção é um misto de todas essas.

Figura 5 – Mapa do site Epic Games Store.

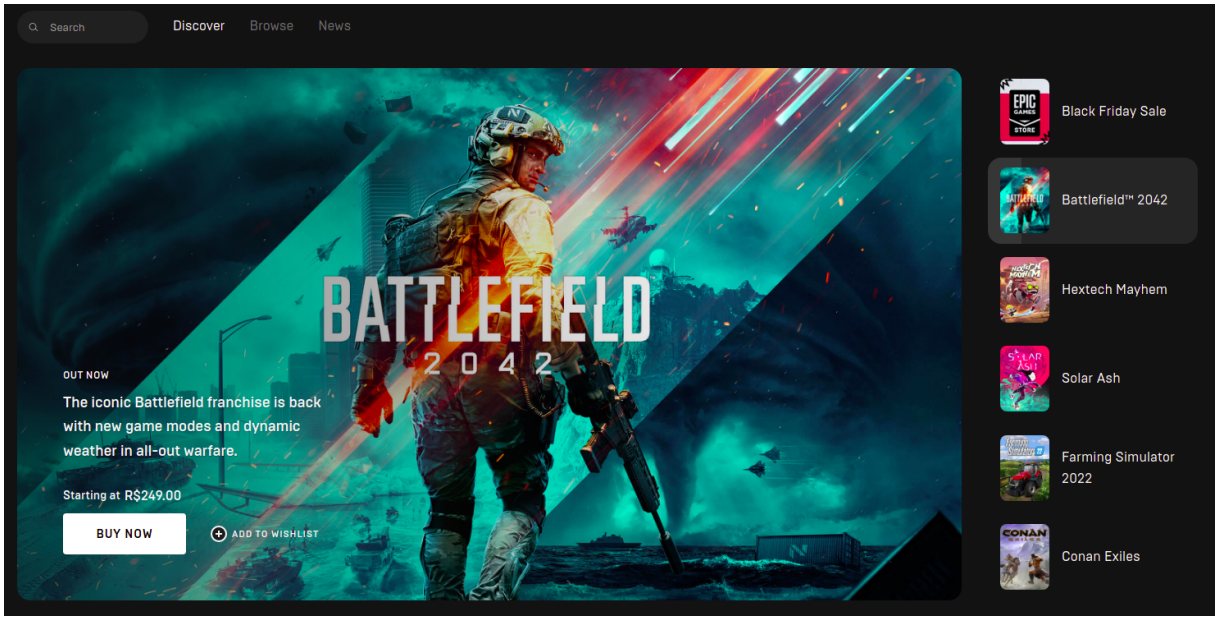


Figura 6 – Seção inicial.

Fonte: (GAMES, 2021a).

Antes de apresentar cada seção separada, a Figura 5 apresenta um mapa da página principal da loja da Epic Games com intuito de demonstrar onde cada seção pode ser encontrada no site. A página possui a opção de rolagem e vai desde a seção inicial representada pelo número 1 na Figura 5 até o número 9, última seção na Figura 5, que mostra a seção de jogos mais populares. Dessa forma, a página segue o seguinte fluxo: seção inicial (número 1 da Figura 5), seção de jogos em promoção (número 2 na Figura 5, seção de jogos gratuitos (número 3 na Figura 5), uma seção de novos lançamentos, mais vendidos e lançamentos em breve (número 4, número 5 e número 6 na Figura 5, respectivamente), seção de jogos atualizados recentemente que é o número 7 na Figura 5, seção de novos na Epic Figura 13, representada pelo número 8 na Figura 5, e, por fim, uma seção de jogos mais populares a qual é ilustrada pelo número 9 na Figura 5. Lembrando que de tempos em tempos a Epic Games altera um pouco este fluxo, mas no geral é bem parecido com o que foi apresentado. A seguir serão descritos os detalhes de cada seção.

A Figura 6 apresenta a primeira seção, também chamada de seção inicial, a qual alterna informações de descrição, preço e nome do jogo que está em maior destaque e é representada na Figura 5 pelo número 1. É a primeira chamada do site, dessa forma, a Epic busca deixar um jogo específico com um destaque maior por um certo período de tempo, no qual é alternado entre outros 5 jogos. A escolha desses 6 jogos para fazer parte dessa seção se trata de um resumo das demais seções ou destaque a algum jogo específico que não compreende nenhuma das demais seções.

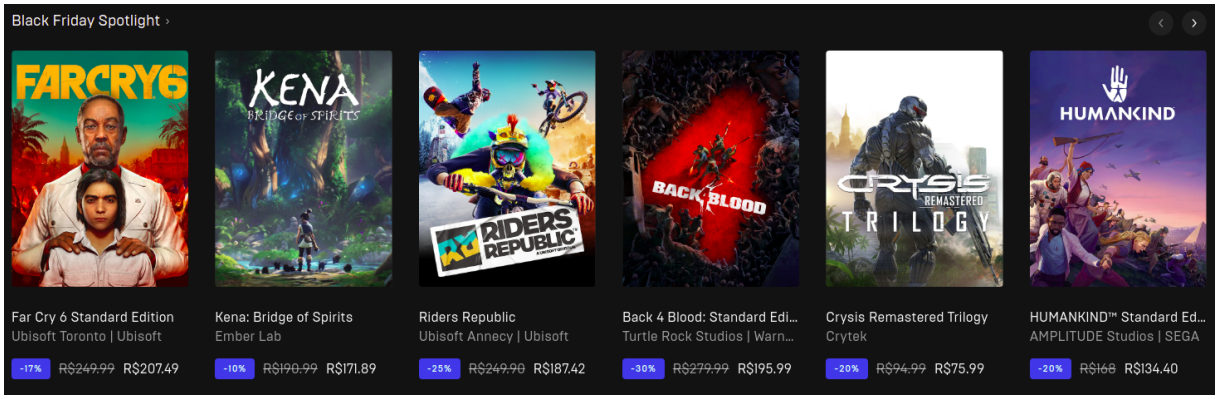


Figura 7 – Jogos em promoção.

Fonte: (GAMES, 2021a).

A Figura 7 contém uma lista dos jogos mais famosos e/ou recém lançados que estão em promoção, representada na Figura 5 pelo número 2. Nessa lista há informações do nome do jogo, empresa desenvolvedora e/ou publicadora, porcentagem de desconto, valor original e valor atual. Esta seção acompanha algum evento importante no mundo real, ou seja, o da imagem acompanha a Black Friday, dessa forma, a Epic seleciona alguns jogos, nesse caso com base nos jogos que estão em destaque no momento, para fazer parte desse evento. Outro exemplo ainda mais claro é o Halloween, nesse caso a Epic seleciona alguns jogos que tenham relação com o tema, jogos que tenham zombies, monstros, etc.

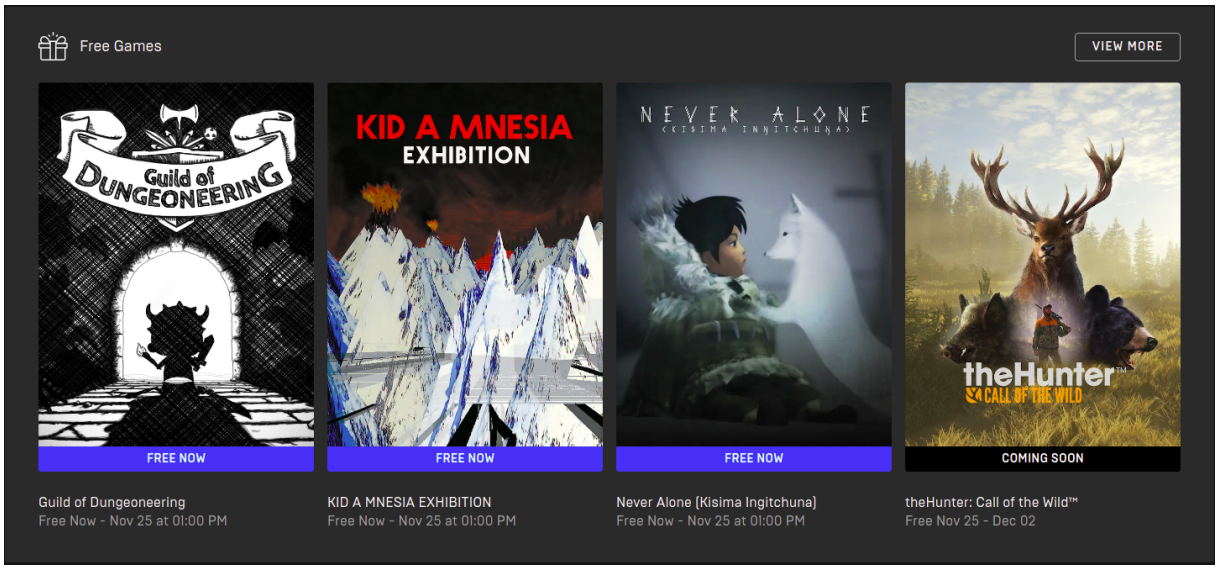


Figura 8 – Jogos gratuitos.

Fonte: (GAMES, 2021a).

Na Figura 8 é apresentada uma lista dos jogos em destaque no momento que estão gratuitos e/ou que ficarão gratuitos em breve, representada na Figura 5 pelo número 3. Nessa lista também há o nome do jogo e o período que ele ficará gratuito. Esta seção

também é bastante atrativa, pois pode conter jogos bem classificados ou não, no entanto, ainda sim o jogo está gratuito e pode ser adquirido para jogar quando quiser.

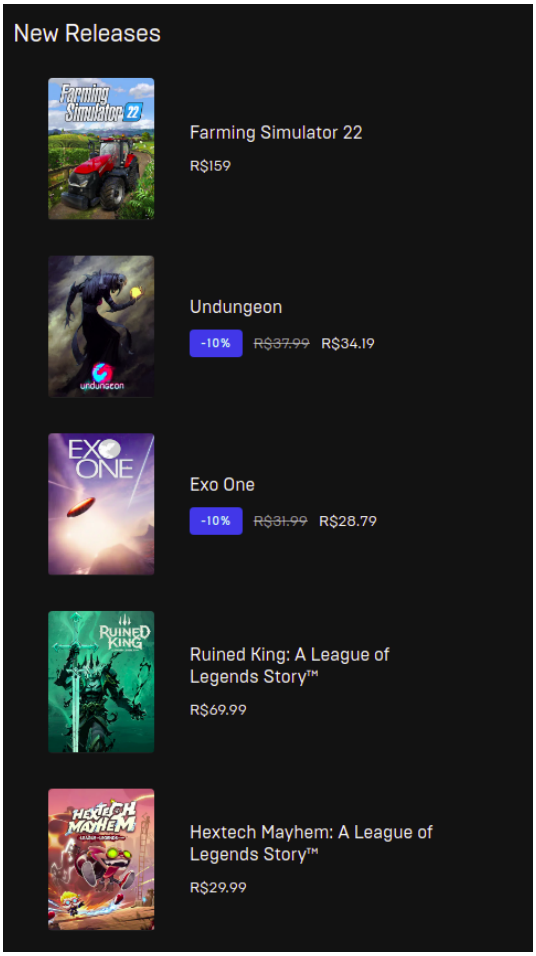


Figura 9 – Novos lançamentos.

Fonte: (GAMES, 2021a).

A Figura 9 exibe uma lista com jogos que acabaram de ser lançados pela Epic Games e as informações disponíveis são o nome, o preço e o desconto do jogo caso exista e é representada na Figura 5 pelo número 4. Esta seção pode ser atrativa para aqueles jogadores que gostam de novidades e estão sempre em busca de um jogo novo, ainda pode ter a chance de um desses jogos estarem disponíveis de forma gratuita ou com desconto.

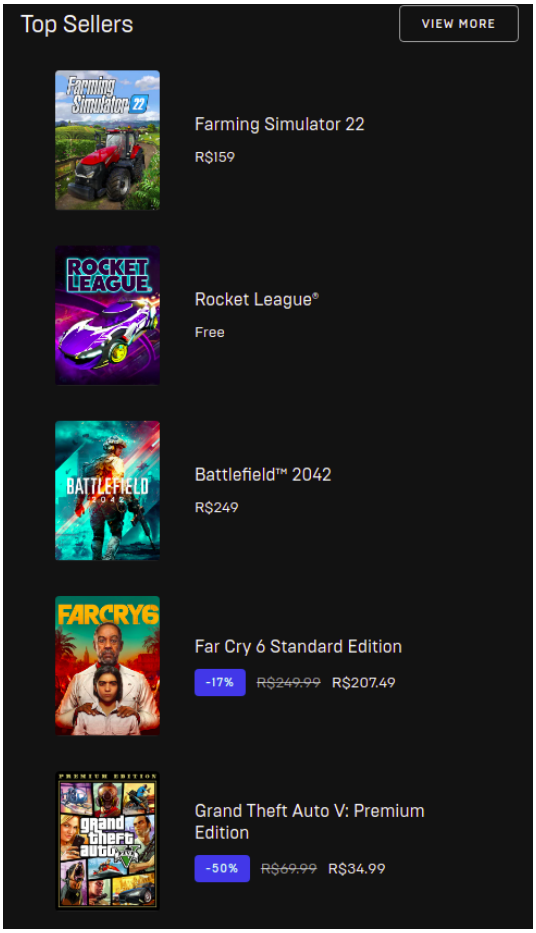


Figura 10 – Mais vendidos.

Fonte: GAMES (2021a).

Na Figura 10 é exibida uma lista com os jogos mais vendidos na plataforma e as informações apresentadas são o nome, preço e desconto do jogo, se esse último existir, representada na Figura 5 pelo número 5. Para aqueles que gostam dos jogos os quais a maioria das pessoas estão jogando, esta seção também é atrativa e também pode conter jogos gratuitos ou com desconto.

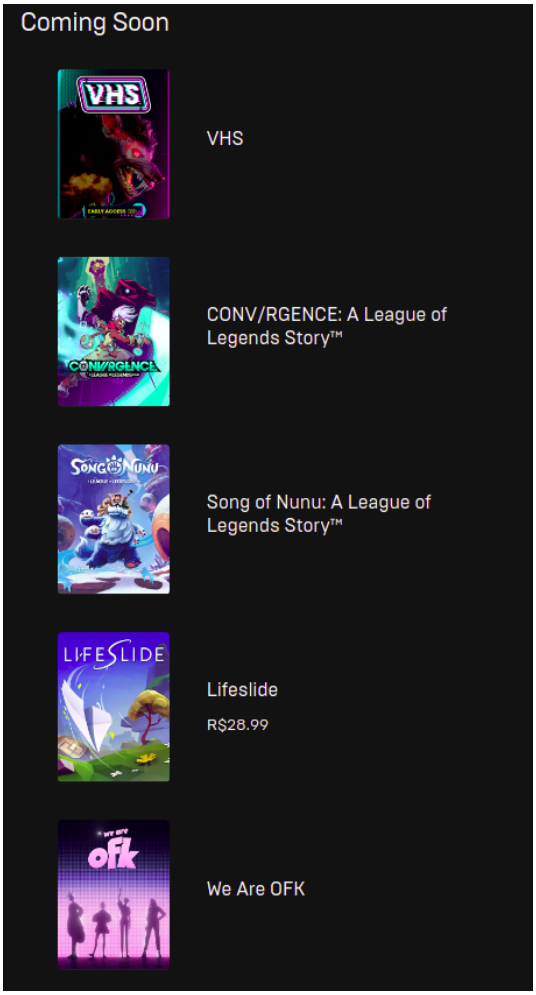


Figura 11 – Lançamentos em breve.

Fonte: (GAMES, 2021a).

A Figura 11 apresenta os principais jogos ou pacotes de jogo que serão lançados em breve e é representada na Figura 5 pelo número 6. As informações disponíveis são nome do pacote ou jogo e preço caso seja pago. Esta seção também pode servir para aqueles jogadores que gostam de novidades, além de saber quais jogos são lançamentos, pode ficar ciente de quais jogos serão os próximos a serem lançados.

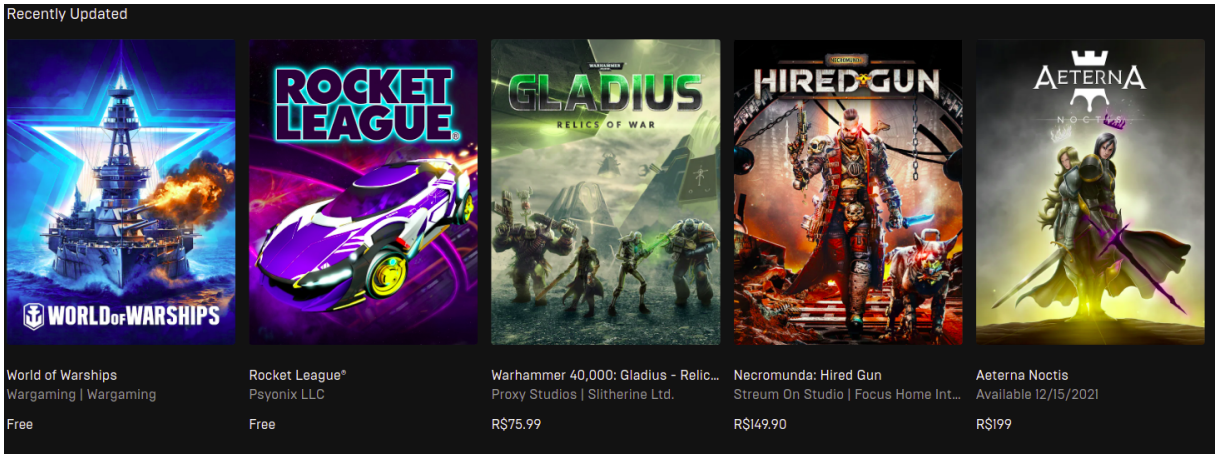


Figura 12 – Atualizados recentemente.

Fonte: (GAMES, 2021a).

Na Figura 12 contém uma lista com os jogos que sofreram atualizações nos últimos dias pela empresa desenvolvedora, representada na Figura 5 pelo número 7. As informações disponíveis nessa lista são nome, desenvolvedora e/ou publicadora e preço do jogo. Essa seção pode ser atrativa para os jogadores estão esperando que algum jogo que ele goste sofra alguma atualização, seja devido à alguma falha ou funcionalidade que ele acredite que possa melhorada. Também é uma forma da Epic deixar seus usuários informados sobre suas movimentações na plataforma.

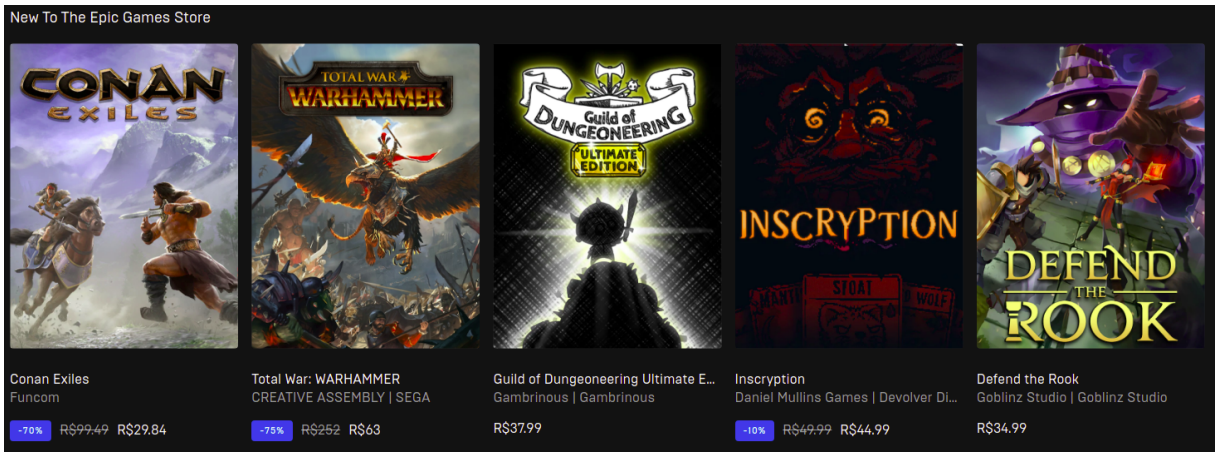


Figura 13 – Novos na Epic Games.

Fonte: (GAMES, 2021a).

A Figura 13 exibe uma lista de jogos que chegaram na loja recentemente, representada na Figura 5 pelo número 8. As informações disponíveis nessa lista são nome, desenvolvedora e/ou publicadora e preço do jogo. Diferente dos jogos que foram lançados na Epic Games, esta seção apresenta os jogos que acabaram de chegar na loja, mas que foi anteriormente lançada em outra plataforma. Sendo assim, pode ser interessante para

aqueles jogadores que preferem adquirir o jogo pela loja da Epic ou porque simplesmente não tem conta cadastrada na plataforma em que o jogo que foi lançado.

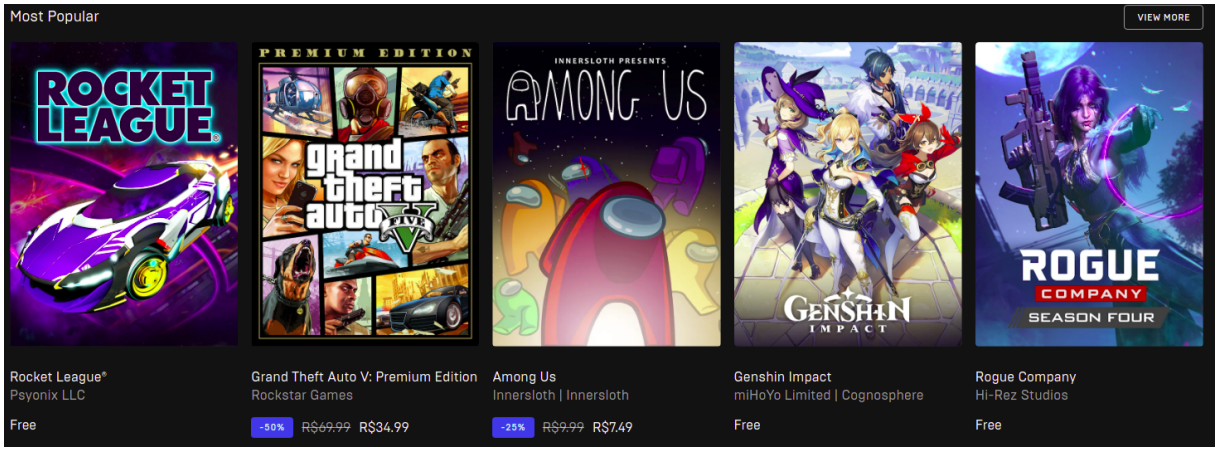


Figura 14 – Mais populares.

Fonte: (GAMES, 2021a).

Por fim, na Figura 14, última seção da página, contém uma lista com os jogos mais populares da Epic Games que engloba os que estão em destaque no momento e os que estão sempre no topo de jogos mais comentados. As informações disponíveis são nome, desenvolvedora e/ou publicadora e preço do jogo caso seja pago. Essa seção é representada na Figura 5 pelo número 9.

3.2 Dicionário de Dados

Durante a pesquisa realizada para entender melhor o contexto do trabalho e no projeto de desenvolvimento, foi criado um dicionário de dados que possa descrever as principais informações disponíveis no site da Epic Games. Essas informações são provenientes de dados não estruturados, neste projeto, esses dados serão estruturados na forma da construção de um modelo entidade relacionamento. Sendo assim, o dicionário de dados descreve todos os atributos que o dataset contém. Nas próximas subseções serão detalhadas as tabelas, que representam as entidades o conjunto de dados, e seus atributos (e.g., informações).

3.2.1 Tabela de Jogos

Na Tabela 2 é apresentada a entidade *games* composta por 10 atributos. Para construir o dicionário dessa tabela foi preciso analisar diretamente na plataforma da Epic Games as informações que estavam disponíveis sobre os jogos. Sendo assim, foram escolhidos os dados mais relevantes, no qual a maioria dos campos são obrigatórios na loja.

Tabela 2 – Games.

Tabela	**Notas da tabela**	**Campo**	**Definição**	**Valor exemplo**	**Notas de campo**
Games	Informações específicas de um jogo alocado na loja da epic games. Não costuma sofrer alterações	id	Identificador da tabela	1234	Atributo identificador da tabela game
		name	Nome do jogo	Grand Theft Auto V	Nome do jogo, campo não alterado
		gameSlug	Apelido do jogo	lego-batman	É o nome do jogo em forma reduzida que serve para facilitar uma possível requisição de dados extras
		price	Preço do jogo	R$34,99	Em caso de jogo gratuito o valor é 0
		releaseDate	Data de lançamento	Sep 17, 2013	Está disponível desde essa data até o momento visualizado
		platform	Plataforma que estará disponível	Windows	Nesse campo é exibido o sistema operacional em que o jogo está disponível
		description	Descrição do jogo	"*When a young street hustler, a retired bank robber and a terrifying psychopath land themselves in trouble...*"	Esse campo faz um resumo de como é o jogo.
		developer	Desenvolvedor do jogo	Ubisoft	Esse campo exibe o desenvolvedor ou desenvolvedores do jogo
		publisher	Publicador do jogo	Ubisoft	Esse campo exibe o nome da empresa que publicou o jogo.
		genres	Gênero do jogo	Ação, Aventura	Exibe os gêneros do jogo, pode ser mais de um

3.2.2 Tabela de Hardware Necessário

A Tabela 3 exibe a entidade *necessary-hardware* composta por 6 atributos. Assim como a tabela *games*, foi preciso analisar a loja da Epic Games e decidir quais dados seriam úteis e que estariam disponíveis na página da maioria dos jogos da loja. Portanto, foi decidido que salvar no dataset o sistema operacional, processador, quantidade memória RAM utilizada e gráficos seria muito importante para a análise dos dados e para um possível trabalho futuro.

Tabela 3 – Hardware necessário.

Tabela	**Notas da tabela**	**Campo**	**Definição**	**Valor exemplo**	**Notas de campo**
necessary hardware	Tabela que representa o hardware necessário que o jogador precisa ter em seu computador para instalar o jogo	id	Identificador da tabela	1234	Atributo identificador da tabela necessaryHardware
		operacional system	sistema operacional que o jogo suporta	Windows 10 64bit (min version 1809)	sistema operacional com arquitetura e versão mínima
		processor	processador que o jogo suporta	Intel i5-4590/AMD FX 8370	versão mínima de processador suportado
		memory	memória RAM necessária	8GB	memória RAM mínima necessária para jogar.
		graphics	gráficos necessários	(4GB VRAM) NVIDIA GeForce GTX 1050Ti (Legacy GPU: GeForce GTX 960) / AMD Radeon RX 470	gráficos mínimos necessários para jogar
		fkGameId	Chave estrangeira para o jogo	1234	Atributo chave estrangeira para o identificador da tabela game.

3.2.3 Tabela de Redes Sociais

Para a tabela *social-networks*, as avaliações são as mesmas das entidades anteriores. Foi decidido criar uma tabela somente para as redes sociais pelo fato de que alguns jogos possuem muitas redes sociais, poucas ou nenhuma – uma grande variação de redes sociais. Portanto, como demonstrado na Tabela 4, é armazenado no dataset a descrição da rede social (Twitter, Facebook, Twitch, dentre outros) e o link para a rede social.

Tabela 4 – Social Networks.

Tabela	**Notas da tabela**	**Campo**	**Definição**	**Valor exemplo**	**Notas de campo**
social networks	Tabela que representa as redes sociais do jogo caso tenha	id	Identificador da tabela	1234	Atributo identificador da tabela socialNetworks
		description	Nome da rede social	linkTwitter	serve para saber qual rede social está sendo armazenada
		url	Url da rede social	http://twitter.com/MetroVideoGame	Link para acesso da rede social
		fkGameId	Chave estrangeira para o jogo	1234	Atributo chave estrangeira para o identificador da tabela game.

3.2.4 Tabela de Contas no Twitter

Na Tabela 5 são apresentados 10 atributos e pode-se notar que são todos dados de fácil visualização ao abrir a página de um perfil no Twitter. Foi considerado que esses são dados importantes para o trabalho, em caso da realização de um trabalho futuro pode ser que seja necessário se aprofundar com mais detalhes a respeito da API do Twitter e coletar mais dados para complementar as análises deste trabalho.

Tabela 5 – Twitter Accounts.

Tabela	**Notas da tabela**	**Campo**	**Definição**	**Valor exemplo**	**Notas de campo**
twitter accounts	Tabela que representa as contas no Twitter	id	Identificador da tabela	1234	Atributo identificador da tabela twitterAccounts
		name	nome do perfil	Epic Games Store	representa o nome do perfil do twitter
		username	nome de usuário da conta	EpicGames	representa o nome de usuário da conta
		bio	biografia do usuário da conta	A curated digital storefront for PC and Mac, designed with both players and creators in mind. Focusing on great games and a fair deal for game developers.	uma descrição sobre aquela conta do twitter
		location	localização do usuário	Cary, NC	representa a cidade sede da empresa ou residência de uma pessoa
		website	link do website daquele perfil	epic.gm/ freegames	caso o usuário tenha um website ele pode inserir em seu perfil
		joinedDate	data que o usuário ingressou no twitter	Joined March 2010	representa a data que o usuário entrou no twitter pela primeira vez
		following	número de perfis que está seguindo	70	quantidade de contas no twitter que o perfil segue
		followers	número de pessoas que seguem o perfil	4.300.000	quantidade de contas no twitter que seguem este perfil
		fkGameId	Chave estrangeira para o jogo	1234	Atributo chave estrangeira para o identificador da tabela game.

3.2.5 Tabela de Tweets

Para a Tabela 6, foram selecionados os dados que mais impactam na audiência de um perfil no Twitter com o objetivo de verificar quais jogos estão sendo mais comentados e compartilhados na rede social, o que pode revelar o tipo de engajamento gerado pelo jogos e seus produtores. Esse se torna um fator importante em um possível trabalho futuro para construção de recomendação de jogos.

Tabela 6 – Tweets.

Tabela	**Notas da tabela**	**Campo**	**Definição**	**Valor exemplo**	**Notas de campo**
tweets	Tabela que representa os tweets dos jogos que possuem conta no Twitter.	id	Identificador da tabela	1234	Atributo identificador da tabela tweets
		text	texto do tweet	"texto do tweet"	texto que aparece na postagem do tweet
		urlMedia	url para acesso à mídia do tweet	https://pbs.twimg.com/ext_tw_video_thumb/892132393582157824/pu/img/5bAm_lkJ0Dl0qCol.jpg	link da imagem ou vídeo que aparece na postagem do tweet
		quantity likes	quantidade de likes no tweet	10	número de usuários que curtiram aquele tweet
		quantity retweets	quantidade de retweets	5	número de usuários que repostaram aquele tweet
		quantity quotes	quantidade de citações do tweet	3	número de usuários que postaram um novo tweet citando este tweet
		quantity replys	quantidade de respostas no tweet	20	número de usuários que postaram uma resposta ao tweet original
		inReplyToUserId	ID do usuário que postou o tweet pai	16516516	Se este Tweet for uma resposta, indica o ID de usuário do autor do Tweet pai
		timestamp	Horário da postagem do tweet	2017-08-09T20:55:29.000Z	Contém a data e hora da postagem do tweet
		fkTwitterAccountId	chave estrangeira para uma conta no twitter	1234	atributo chave estrangeira para o identificador da tabela twitterAccounts.

3.2.6 Tabela de Avaliações dos Jogos

Por fim, na Tabela 7 temos a entidade que contém informações sobre as avaliações dos jogos na plataforma da OpenCritic. Os principais dados sobre a avaliação foram

selecionados por meio de análise da plataforma. Assim como o engajamento no Twitter, os comentários e notas de críticos com veredicto são importantes para análise desses dados, como também para construção de recomendação.

Como já foi dito, a nota de uma avaliação no OpenCritic pode existir em vários formatos (8/10, 8.0/10.0, 4 estrelas, 80%, 80...). No entanto, ao coletar os dados diretamente pela API do OpenCritic, estas notas já vêm normalizadas em umas escala de 0 a 100, como informado na Tabela 7.

Tabela 7 – OpenCritic.

Tabela	**Notas da tabela**	**Campo**	**Definição**	**Valor exemplo**	**Notas de campo**
openCritic	Tabela que contém informações das avaliações realizadas pelos críticos com veredicto na plataforma da OpenCritic	id	Identificador da tabela	1234	Atributo identificador da tabela openCritic
		rating	Classificação dada pelo crítico	80	Avaliação do jogo normalizada na escala de 0 a 100.
		comment	Comentário dado pelo crítico na avaliação	"*Forza Horizon 5 is simply one of the top arcade racing games ever released!*"	O autor escreve um texto sobre o jogo e na listagem das avaliações é mostrado um resumo
		company	Empresa considerada no cálculo da média de avaliação do jogo	PC Gamer	O autor da avaliação posta seu comentário no site dessa empresa e o OpenCritic busca diretamente nesse site.
		author	Autor da avaliação	Jonathan Leo	Nome da pessoa que posta seu comentário com a avaliação do jogo
		date	Data da postagem da avaliação	nov. 4, 2021	Data em que a OpenCritic inseriu a avaliação em sua plataforma.
		description	Descrição do jogo	"*When a young street hustler, a retired bank robber and a terrifying psychopath land themselves in trouble...*"	Esse campo faz um resumo de como é o jogo.
		topCritic	Verifica se é uma top crítica	true	Flag que verifica se a crítica postada tem selo top critic ou não.
		fkGameId	Chave estrangeira para o jogo	1234	Atributo chave estrangeira para o identificador da tabela game

3.3 Modelo Entidade Relacionamento

Após a descrição do dicionário de dados, se tornou mais fácil construir o modelo entidade relacionamento para realização de uma representação mais estruturada dos dados. Sendo assim, como dito anteriormente, existem as tabelas *games*, *social_networks*, *necessary_hardware*, *critic*, *twitter_accounts* e *tweets*, como demonstrado na Figura 15.

Figura 15 – Modelo entidade relacionamento dos Scripts.

O modelo contém 6 entidades (conjuntos específicos de dados), na qual a Tabela *games* é a mais importante, pois, para recuperar os dados do restante do dataset, é preciso obter êxito na busca dos jogos. No modelo entidade relacionamento pode-se observar as chaves estrangeiras para referenciar esta entidade e os relacionamentos existentes com outras entidades. Nota-se que há uma relação 1 para muitos entre jogo e hardware necessário, isso ocorre porque o jogo possui o hardware mínimo e o hardware recomendado.

Além disso, as informações sobre os jogos são coletados diretamente da loja da Epic Games através da implementação de um script para realização da coleta por meio de

webscraping. Os dados das contas no Twitter e os tweets dos jogos são coletados da API do Twitter. E, por fim, os dados da Tabela *critic*, que são informações de avaliações dos jogos, são coletados da API da OpenCritic. Os detalhes sobre a coleta desses dados estão descritos no Apêndice A.

3.4 Fluxograma dos Scripts para Coleta de Dados

Esta seção descreve o funcionamento dos scripts por meio da representação de um fluxograma. O seu procedimento é bastante simples e precisa ser executado apenas 3 vezes para escolher as opções de busca de dados. Maiores detalhes estão disponíveis no Apêndice A.

Como demonstrado na Figura 16, a primeira opção escolhida deve ser a 1, pois, é por meio dessa opção que os principais dados da aplicação são requisitados, isto é, os dados dos jogos. Após a escolha da opção 1, o algoritmo irá realizar uma requisição HTTP ao servidor da Epic Games, a qual receberá como resposta os dados de quase mil jogos em formato JSON[1]. Esses dados contém informações necessárias para preencher 3 entidades do modelo entidade relacionamento (Figura 15): *games*, *necessary_hardware* e *social_networks*. Em seguida, o script formata os dados recebidos de jogos, redes sociais e hardware necessário e escreve em seus respectivos arquivos de extensão *.csv* para armazenamento dos dados coletados.

Posteriormente, é preciso executar o script novamente para realizar as próximas funcionalidades da coleta. Contudo, o próximo passo não precisa de uma ordem específica, ou seja, pode-se requisitar os dados do Twitter ou da OpenCritic. Como demonstrado na Figura 16, é possível escolher uma das duas opções. Após a escolha de um desses passos, as etapas seguintes são parecidas. Independente da escolha, é preciso realizar a requisição às API's (Twitter e OpenCritic) e formatar os dados recebidos para adaptar aos parâmetros exigidos em seus respectivos arquivos csv.

Uma informação importante é que é preciso criar uma conta de desenvolvedor na plataforma para obter acesso aos *endpoints* da API do Twitter. Ao criar a conta, o Twitter disponibiliza algumas chaves de autenticação de acesso à API. Mais informações em Twitter (2021).

Em caso de escolha da opção 2, o script busca os nomes de usuário do Twitter de todos os jogos que possuem conta na rede social. A API do Twitter não permite buscar mais de 100 contas em apenas uma requisição, pois apresenta paginação de dados como organização. Dessa forma, foi preciso construir um laço de repetição e utilizar o *token* de paginação retornado na resposta da requisição para buscar os próximos dados. Por

[1] Um acrônimo para notação de objeto JavaScript, com formato compacto para troca de dados simples e rápida entre sistemas.

conseguinte, os dados recebidos são formatados e escritos em seu arquivo de extensão *.csv*. Com os dados das contas armazenados em arquivo, utiliza-se o atributo identificador dessa conta recebida na requisição para buscar seus respectivos tweets. Sendo assim, no próximo passo, as requisições desses tweets são realizadas, os dados são formatados e, por fim, escritos em seu arquivo de extensão *.csv*.

Em caso de escolha da opção 3, os dados de todos os jogos que estão na OpenCritic são requisitados em sua API, na qual também precisa da varredura da paginação dos dados para realização da coleta dessas informações. Em seguida, é verificado quais desses jogos estão disponíveis na Epic Games para enfim buscar suas avaliações por meio de requisições na API da OpenCritic. No caso desta API não é preciso de chaves de autenticação. Por fim, o dataset está disponível para download na plataforma Zenodo[2].

[2] Dataset disponível em: <https://zenodo.org/record/7606569>

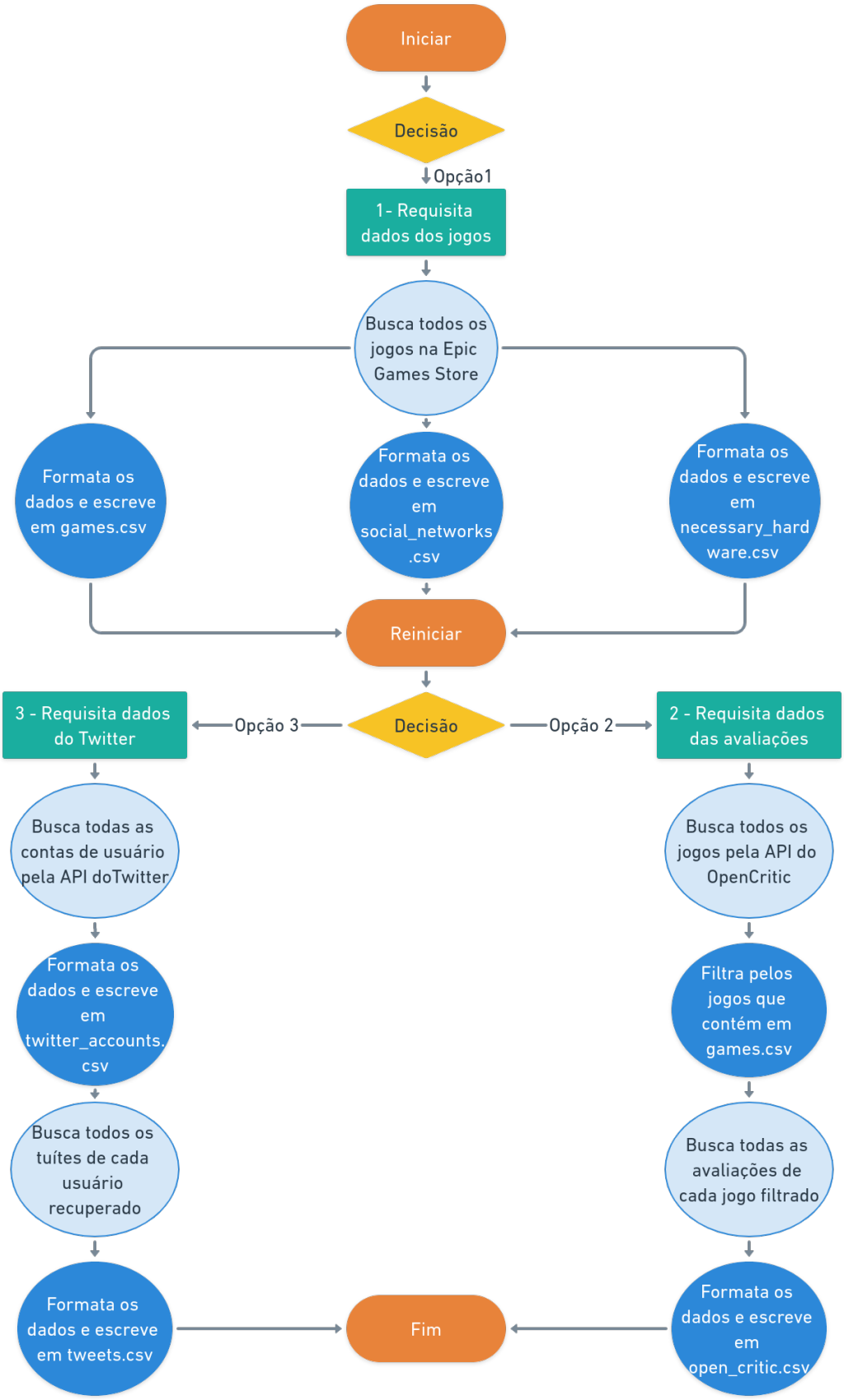


Figura 16 – Fluxograma dos Scripts para coleta de dados.

4 Resultados

Este capítulo descreve os resultados obtidos durante a análise do dataset. A análise preliminar é composta por uma sumarização estatística dos dados e suas possíveis correlações entre os atributos numéricos, como descrito na seção 4.1. Em seguida, na seção 4.2 foi feito a análise dos dados em si com base em atributos de todas as entidades.

4.1 Análise Preliminar

Nesta seção se da início à análise inicial realizada após a coleta dos dados. Em primeiro lugar, foi realizada a sumarização estatística desses dados com o objetivo de reunir, de maneira sintetizada, tudo que foi coletado pelos scripts.

4.1.1 Sumarização Estatística

Após a coleta dos dados, deu-se início à análise preliminar do dataset. Sendo assim, foi construído um algoritmo simples para obter a sumarização estatística dos dados. O resultado deste algoritmo contém a quantidade de registros de cada entidade e a média de avaliações por jogo coletadas do OpenCritic conforme é apresentado na Tabela 8.

Tabela 8 – Tabela quantidade de registros por entidade.

Total por Entidade	
Jogos	915
OpenCritic	17.584
Redes Sociais	3.045
Contas Twitter	522
Tweets	989.495
Média avaliações por jogo	25

Além da quantidade de registros, uma outra tabela foi construída para compor a sumarização estatística dos resultados. Como demonstrado na Tabela 9 e na Tabela 10, foram calculadas medidas de localização e de dispersão, tais como média, variação padrão, variação simples, coeficiente de variação, valor mínimo e máximo, 1º, 2º e 3º quartis, distância entre valor máximo e mínimo (*range*) e a soma de cada atributo numérico das entidades.

A Tabela 9, primeira parte da sumarização, indica que o atributo preço possui um desvio padrão baixo e que a maioria dos valores estão próximos da média, R$ 23,23. Ademais, existe uma tendência para valores mais baixos pois 75% dos jogos custam no máximo até R$ 29,99, isso é ilustrado pelo 3º quartil na tabela. Conclui-se que jogos são

Tabela 9 – Tabela sumarização estatística.

	Jogos (preço)	**Contas Twitter** (seguindo)	**Contas Twitter** (seguidores)	**OpenCritic** (avaliação)
Média	23,23	488,97	211.339,50	75,40
Desvio padrão	14,70	1122,85	1.143.075,85	13,80
Variância	216,19	1.260.799,14	1.306.622.398.974,76	189,70
Coeficiente de Variação	0,63	2,29	5,40	0,18
Mínimo	0,00	0,00	1,00	0,00
Primeiro Quartil (Q1)	14,99	35,00	3.095,50	70,00
Segundo Quartil (Q2)	19,99	144,50	10.014,00	80,00
Terceiro Quartil (Q3)	29,99	518,25	40.495,00	85,00
Máximo	109,99	17.726,00	14.895.908,00	100,00
Range	109,99	17.726,00	14.895.907,00	100,00
Maior	109,99	17.726,00	14.895.908,00	100,00
Menor	0,00	0,00	1,00	0,00
Soma	21.257,48	255.243,00	110.319.222,00	1.143.751,10
Total de amostras	915,00	522,00	522,00	17.584,00

em sua maioria mais baratos e, portanto, mais acessíveis aos jogadores. Além disso, os valores dos preços apresentados são dados em reais.

Partindo para a entidade "Contas no Twitter", seus atributos apresentam valores bem diferentes. Em média, as contas possuem bem mais seguidores (211.339,50) do que seguem outros usuários (488,97). No entanto, em ambos atributos a tendência é para valores mais baixos, observa-se que até 75% (3º quartil) do número de contas de jogos que seguem outras contas (jogadores) é menor ou igual a 518,25. Apesar do número alto de seguidores, 3º quartil de 40.495,00, vale ressaltar a grande diferença de valores de até 3 ordens de grandeza com relação à conta do jogo que possui o maior número de seguidores (14.895.908,00). Além disso, o desvio padrão das duas variáveis são bem altos, ou seja, a maioria dos valores estão distantes da média. No geral, existem poucas contas de jogos no Twitter com um grande número de seguidores e a grande maioria apresenta pouco seguidores, o que indica uma possível distribuição de cauda pesada, o que será melhor explorado na subseção 4.2.4.

Por fim, a variável "avaliação" indica as notas atribuídas aos jogos no site do OpenCritic. A média das notas é 75,40 e seu desvio padrão é baixo, 13,80. Dessa forma, é possível perceber pelos valores apresentados nos quartis que a maioria dos jogos possuem boa avaliação no site do OpenCritic, uma vez que o 1º quartil é de uma nota de 70,00, ou seja, a maioria das notas, mais especificamente 75%, é maior do que 70,00.

Em sequência, a Tabela 10 indica a sumarização dos atributos numéricos da entidade "Tweets". É notório que em todos os quartis de todas as variáveis os valores são 0 ou bem próximos de 0, dessa forma, é possível dizer que a maioria dos tweets não recebem muito engajamento dos usuários em relação à estes atributos, pois, o 3º quartil é de 17, 8, 0 e 1 para curtidas, retweets, citações e respostas, respectivamente. Se comparado com a quantidade máxima, são valores bem discrepantes atingindo diferenças de até 5 ordens

Tabela 10 – Continuação tabela sumarização estatística.

	Tweets (curtidas)	**Tweets** (retweets)	**Tweets** (citações)	**Tweets** (repostas)
Média	223,61	78,68	6,94	11,56
Desvio padrão	2.878,67	1.917,05	216,39	248,94
Variância	8.286.793,21	3.675.104,06	46.826,54	61.974.70,00
Coeficiente de variação	12,87	24,36	31,17	21,53
Mínimo	0,00	0,00	0,00	0,00
Primeiro Quartil (Q1)	0,00	0,00	0,00	0,00
Segundo Quartil (Q2)	0,00	1,00	0,00	0,00
Terceiro Quartil (Q3)	17,00	8,00	0,00	1,00
Máximo	596.446,00	566.550,00	71.698,00	119.217,00
Range	596.446,00	566.550,00	71.698,00	119.217,00
Maior	596.446,00	566.550,00	71.698,00	119.217,00
Menor	0,00	0,00	0,00	0,00
Soma	221.264.355,00	77.856.121,00	6.869.212,00	11.438.594,00
Total de amostras	989.495,00	989.495,00	989.495,00	989.495,00

de grandeza (veja retweets em que a média é de 78,68 e o valor máximo é de 566.550,00). Ou seja, as contas que possuem mais seguidores, tendem a obter um maior engajamento através de curtidas, retweets, citações e respostas em seus tweets, visto que os atributos de tweets possuem também um desvio padrão alto, o que pode ser observado pelos valores de coeficiente de variação que são muito maiores do que um (o que é um indicativo de cauda pesada).

Além disso, é possível perceber que, em média, as ações de curtir e retweetar são maiores do que citar ou responder um tweet, o que ressalta ainda mais a falta de engajamento dos usuários, uma vez que a ação de curtida e retweet é muito mais fácil de ser realizada. Esse comportamento para atividades do Twitter é padrão para ambientes de redes sociais online, o que reflete uma distribuição de cauda pesada. Porém, nesse caso, é perceptível a falta de engajamento dos jogadores com as redes sociais de seus respectivos jogos, o que demonstra o desinteresse das empresas em promover seus jogos nas redes sociais. Contudo, apesar de ser minoria, os jogos com o maior número de seguidores demonstram sua preocupação com o engajamento nas redes sociais pois eles possuem uma grande quantidade de seguidores e um grande número de atividades nas redes do Twitter. Mais detalhes sobre a distribuição desses dados são apresentados no final da seção 4.2.

4.1.2 Análise de Correlação

Além da quantidade de registros de cada entidade e a sumarização também foi construída uma matriz de correlação de Pearson e de Spearman para entidade de *Tweets*. Dessa forma, pode-se perceber que os atributos da entidade *Tweets* possuem correlações positivas e bastante aproximadas, no entanto, correlações um pouco menores para correlação de Pearson e um pouco maiores pela correlação de Spearman.

Além disso, o motivo de ter sido construído uma matriz somente com estes 4

atributos é porque, primeiramente, a entidade *Tweets* tem somente esses atributos que são do tipo numérico. Um outro fator é que as demais entidades não possuem mais que 2 atributos numéricos cada uma, tornando inviável a construção de uma matriz. Por fim, não foi possível construir uma matriz que correlacionasse todos os atributos numéricos do *dataset* porque cada entidade tem uma quantidade de registros diferente.

Contudo, é preciso entender o que as correlações significam, portanto, neste trabalho é definido que: (a) valores menores do que 0,50 indicam uma *correlação fraca*; (b) valores entre 0,50 e 0,70 apresentam uma *correlação moderada*; (c) enquanto que valores maiores que 0,70 apresentam uma *correlação forte*. Dessa forma, a Tabela 11 indica que a correlação entre curtidas e citações é forte (0,71) e, com isso, há uma tendência de que tweets que recebem mais curtidas também vão receber mais citações, já que o método de Pearson analisa o relacionamento linear entre essas duas variáveis.

Tabela 11 – Matriz de Correlação de Pearson.

	curtidas	**retweets**	**citações**
retweets	0,22	–	–
citações	0,71	0,20	–
respostas	0,36	0,11	0,42

Tabela 12 – Matriz de Correlação de Spearman.

	curtidas	**retweets**	**citações**
retweets	0,27	–	–
citações	0,59	0,47	–
respostas	0,66	0,24	0,54

Além da entidade *Tweets*, uma outra entidade que foi possível correlacionar seus atributos numéricos foi a *Contas no Twitter*. No entanto, como só havia a possibilidade de correlacionar apenas 2 atributos,a correlação entre “seguindo” e “seguidores” pelo método de Pearson foi igual 0,16 (fraca) e pelo método de Spearman foi igual a 0,13 (fraca). Dessa forma, observa-se que esses dados são fracamente correlacionados.

4.2 Análise Exploratória dos Dados

Após a análise preliminar dos dados apresentado na seção anterior, foram utilizadas as bibliotecas *matplotlib*[1], *numpy*[2], *pandas*[3] e *seaborn*[4] da linguagem python para apresentar de forma mais visual e de maneira intuitiva o que pode-se observar dos jogos partir dos dados coletados.

[1] <https://matplotlib.org/>.
[2] <https://numpy.org/>.
[3] <https://pandas.pydata.org/>.
[4] <https://seaborn.pydata.org/>.

4.2.1 Entidade Jogos

Neste momento, além dos atributos numéricos, também foram utilizados alguns atributos de texto das entidades do dataset. Dessa forma, a primeira informação que a Figura 17 demonstra se trata dos top 10 gêneros de jogos mais presentes na plataforma da Epic Games Store e a quantidade de jogos que possuem esses gêneros.

Aqui estão alguns exemplos de jogos para os 5 primeiros gêneros presentes na Epic Games, respectivamente: Assassin's Creed, Charon's Staircase, Paper Cut Mansion, Outlast e Fallout Tactics. Além disso, notório que o gênero mais presente na loja é o de ação com 237 (25%) títulos no momento em que os dados foram coletados.

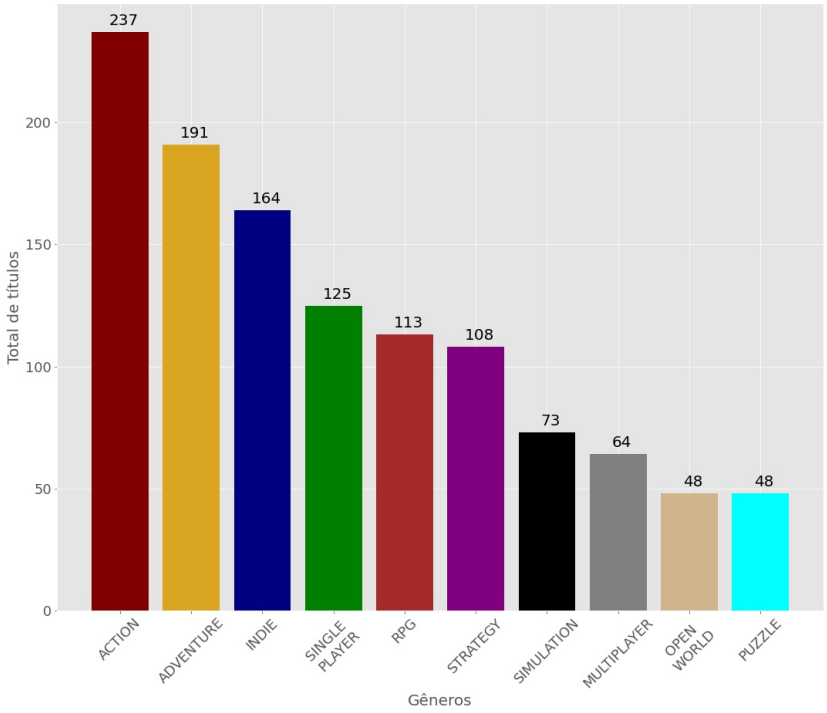


Figura 17 – Top 10 quantidade de jogos por gênero.

Em seguida, foram selecionadas as top 10 empresas que possuem jogos desenvolvidos por elas na plataforma da Epic Games Store, como também a quantidade de jogos de cada uma. De acordo com a Figura 18, a empresa com mais jogos na loja é a Ubisoft com 33 títulos, já que Ubisoft e Ubisoft Montreal são a mesma empresa.

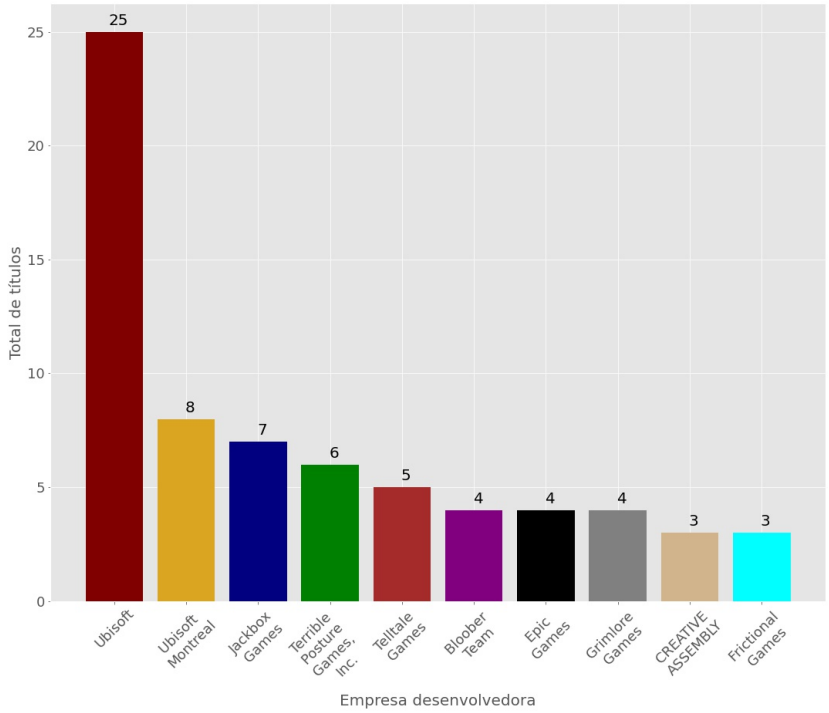
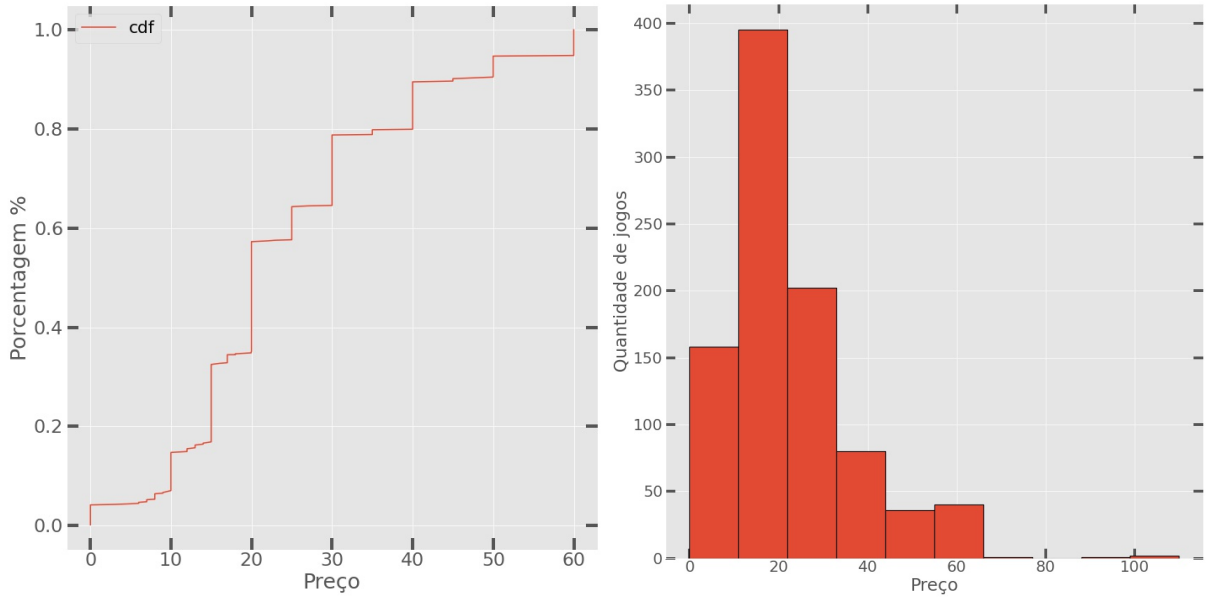


Figura 18 – Top 10 quantidade de jogos por empresa desenvolvedora.

Ainda na entidade de *Jogos*, foi construída uma Cumulative Distribution Function (CDF), em português, Função de Distribuição Acumulada, para verificar a probabilidade acumulada do preço de um jogo na Epic Games. Sendo assim, no momento em que os dados foram coletados, 60% dos jogos são gratuitos ou possuem um valor de até R$ 20,00 conforme mostra a Figura 19a. Além disso, 95% dos jogos possuem preço inferior à R$ 60. Por fim, o histograma da Figura 19b complementa essa informação, pois, a figura demonstra maior densidade de jogos por volta do valor de R$ 20.



(a) Probabilidade acumulada de preço dos jogos. (b) Quantidade de jogos por preço.

Figura 19 – Distribuição acumulada de preços e quantidade de jogos por preço.

4.2.2 Entidade OpenCritic

Dando continuidade às análises realizadas com os dados coletados, nesta subseção serão apresentados os resultados obtidos com a entidade OpenCritic . Sendo assim, na Figura 20 é exibida a quantidade de avaliações realizadas por empresa, novamente as top 10. Como já foi dito no Capítulo 2, a OpenCritic realiza o método de *webscraping* para buscar avaliações de jogos em alguns sites, e essas empresas citadas são as que possuem veredito para que suas notas atribuídas aos jogos contem para o cálculo de avaliação final do jogo.

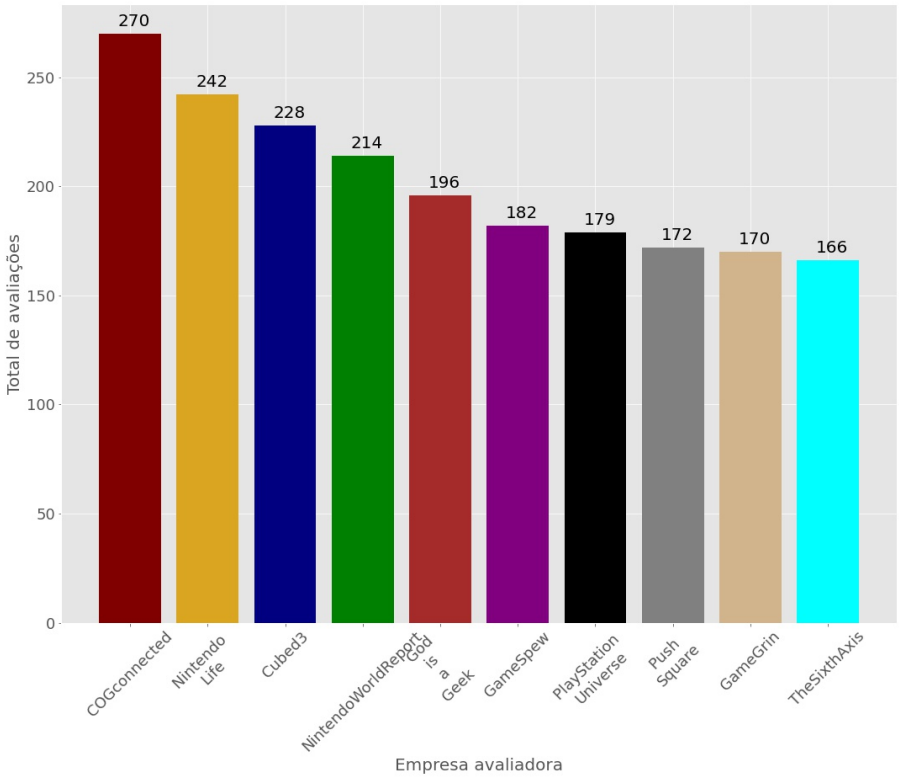


Figura 20 – Quantidade de críticas por empresa.

Sobre essas críticas com veredito, foi apurado na análise que 67% das avaliações que compõem o dataset possuem veredito e contam para avaliação final. O restante das avaliações pertencem ao público geral que podem avaliar diretamente pela plataforma do OpenCritic.

Além da quantidade de críticas por empresa, é possível perceber o aumento de avaliações sendo realizadas e coletadas pelo OpenCritic anualmente através da Figura 21, a curva na figura apresenta um crescimento constante desde 2012 até 2021. A frequência cai de 2021 para 2022 porque os dados foram coletados no início de 2022, dessa forma, não possui todas as críticas deste ano, portanto, pode ser que esse número seja maior.

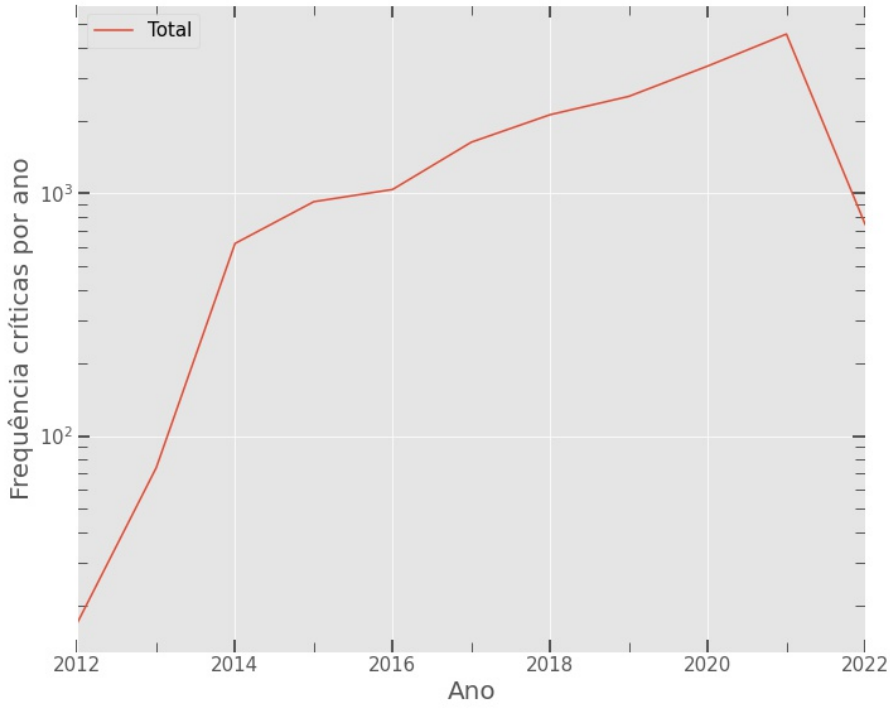
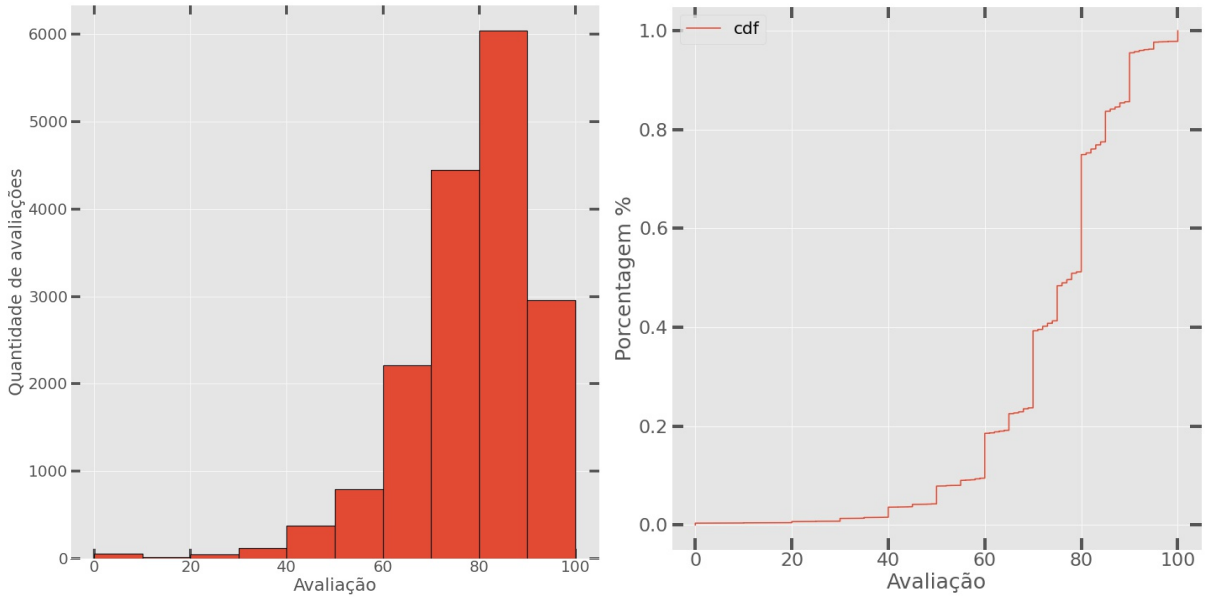


Figura 21 – Frequência de críticas por ano.

Já na Figura 22b é notório que a maioria dos jogos possuem boas avaliações, sendo que cerca de 80% das avaliações têm uma nota maior que 60 em uma escala de 0 a 100 no OpenCritic. O histograma da Figura 22a pode complementar esta informação e deixa mais evidente a distribuição das avaliações dos jogos que compõem o dataset são inclinadas para direita do gráfico, onde ele apresenta maior densidade e a maioria das notas dos jogos se encontram entre 60 e 100, conforme também mostra a Figura 22b.



(a) Histograma de avaliações OpenCritic. (b) Probabilidade acumulada das avaliações.

Figura 22 – Críticas por empresa e probabilidade acumulada das avaliações.

4.2.3 Entidade Redes Sociais

Na sumarização feita na análise preliminar dos dados na seção 4.1 foi observado que o total de redes sociais encontradas foram 3.045 contando todos os jogos. Contudo, foi construído mais um gráfico com o objetivo de ter uma noção mais clara de quais redes sociais eram mais populares dentre as mais de 10 encontradas. Dessa forma, percebe-se que o Twitter é o mais popular e apresenta 760 contas de jogos, aproximadamente 25, 00%, seguido do Facebook com 635 (21,00%) e do Discord com 444 (14,60%). Além disso, o gráfico da Figura 23 indica a quantidade de jogos por rede social.

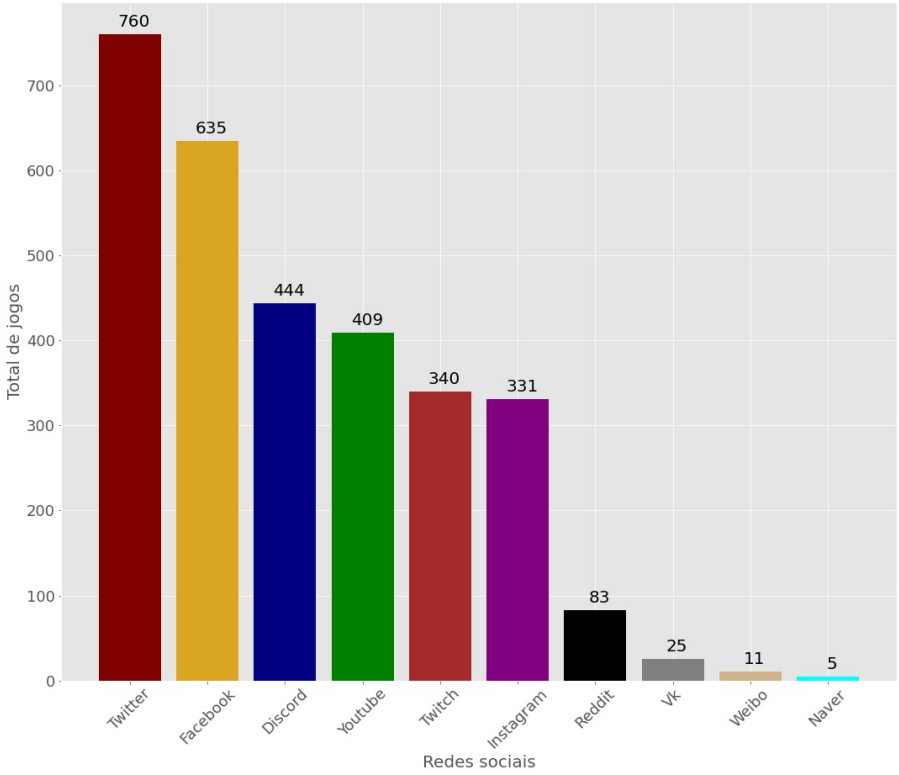


Figura 23 – Quantidade de jogos por rede social.

4.2.4 Entidade Contas no Twitter

No intuito de verificar quais jogos possuem um maior engajamento em suas redes sociais, foram feitas algumas análises nas entidades *twitter_accounts* e *tweets*. Primeiramente, foram apurados os top 10 jogos com mais seguidores no Twitter na entidade *twitter_accounts*, dessa forma, nota-se na Figura 24 as principais contas que compõem o dataset. A lista se inicia com Fortnite, Sonic the Hedgehhog, League of Legends, Assassin's Creed e Genshin Impact com 14.895.908, 5.990.318, 5.240.272, 4.449.399, 4.069.221 milhões de seguidores, respectivamente, seguido dos demais jogos. Contudo, é importante dizer que além de constas de jogos, o dataset contém contas de plataformas parceiras da Epic Games que também podem ser adquiridas pela loja como Discord e Rockstar Games.

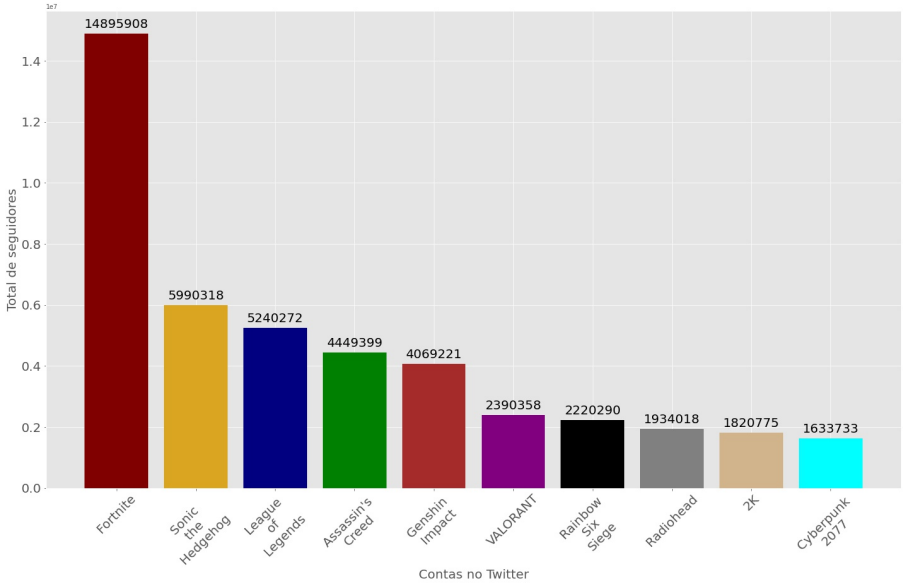
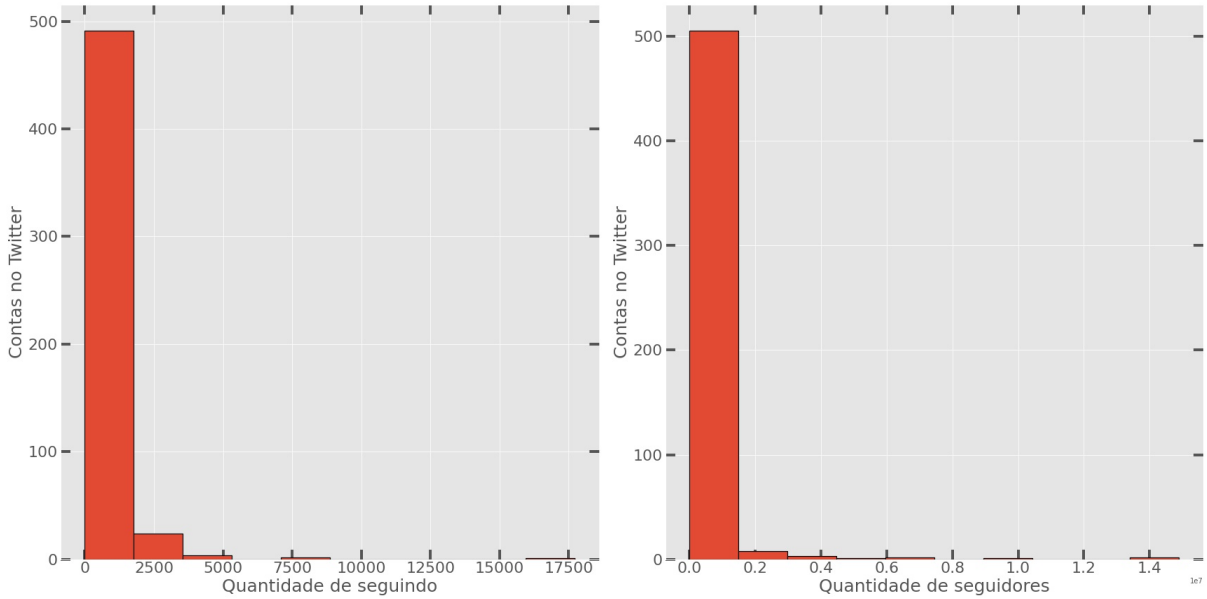


Figura 24 – Quantidade de seguidores por conta no Twitter

Ainda sobre os atributos *seguindo* e *seguidores*, histogramas e Complementary Cumulative Distribution Function (CCDFs) – em português, Função de Distribuição Acumulada Complementar – foram construídos para, dessa vez, mostrar de maneira geral a distribuição da quantidade de usuários seguindo e seguidores dessas contas. Observa-se que uma pequena porcentagem das contas possuem alto número de seguidores, em contraste, a maioria das contas possuem pouquíssimos seguidores. O mesmo acontece com o atributo *seguindo*, porém, em uma escala menor, como demonstrado nas Figura 25 e Figura 26.

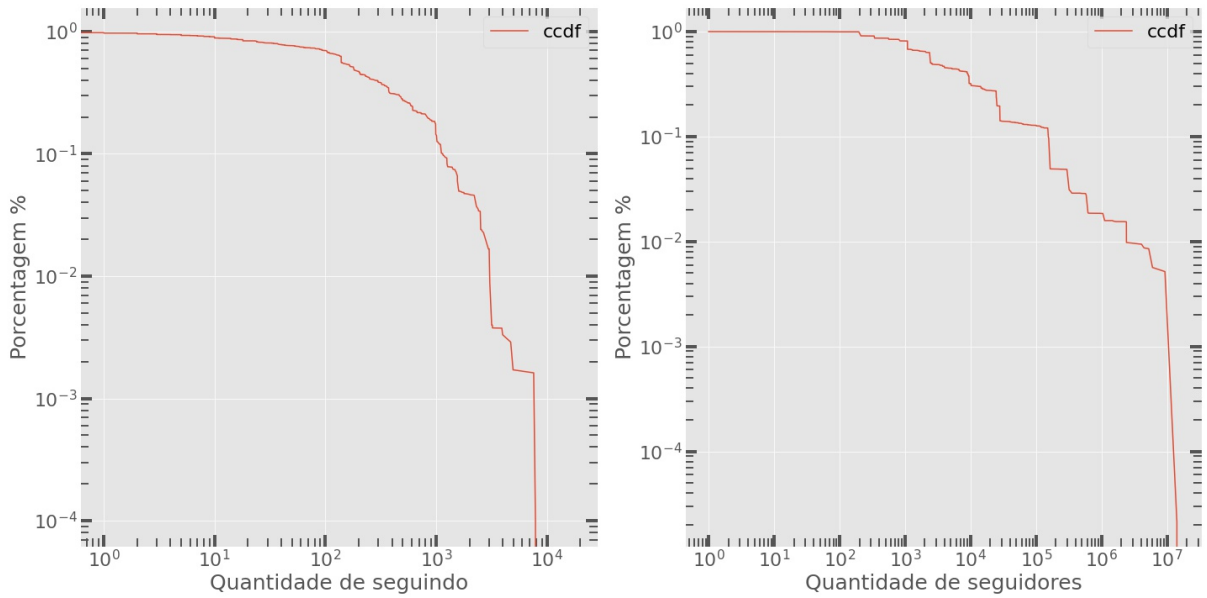
Por exemplo, na Figura 25a e Figura 25b pode-se perceber que a maior parte da densidade da frequência se concentra no primeiro *bin* de valores muito pequenas, e na medida em que os valores aumentam, a frequência se torna muito baixa na medida em que os valores do eixo x aumentam, o que indica uma possível distribuição de cauda pesada conforme será visto em seguida. A Figura 26a mostra que 90% dos jogos seguem um número de contas menor ou no máximo igual a 1000 contas do Twitter, o que implica que apenas 10% das constas de jogos seguem mais do que mil contas, algumas chegando a seguir até 8 mil contas no Twitter. Por sua vez, o número de seguidores para cada conta de jogos no Twitter supera em até 3 ordens de grandeza o número de contas as quais eles seguem, o que é bastante comum de acontecer no Twitter. A Figura 26b mostra que 60% dos jogos (aproximadamente 549) tem um número menor ou igual a 10 mil seguidores, e 30% (quase 274 jogos) possuem um número de seguidores entre 10 mil e 150 mil seguidores. Em contraste, apenas 2% dos jogos (cerca de apenas 18) possuem um número de seguidores maior do que 1 milhão (chegando até quase 15 milhões de seguidores), dos quais 10 deles estão listados na Figura 24. Por fim, as distribuições apresentadas nas Figura 26a e Figura 26b representam claramente uma distribuição de cauda pesada, o que é comum em uma distribuição de grau de amizades em uma rede social online e está de

acordo com estudos clássicos em redes sociais como os de Crane e Sornette (2008).



(a) Histograma de seguindo. (b) Histograma de seguidores.

Figura 25 – Distribuição de *seguindo* e *seguidores*.



(a) CCDF de seguindo. (b) CCDF de seguidores.

Figura 26 – CCDFs de *seguindo* e *seguidores*.

4.2.5 Entidade Tweets

Como última entidade para análise de atributos numéricos, a tabela *Tweets* também foi utilizada para verificar o engajamento das contas no Twitter. Dessa forma, os atributos escolhidos para análise foram: curtidas, retweets, respostas e citações. Contudo, antes

de começar as análises com base nestes atributos, foi verificado a frequência de tweets postados por ano somando todas as contas e tweets presentes no dataset.

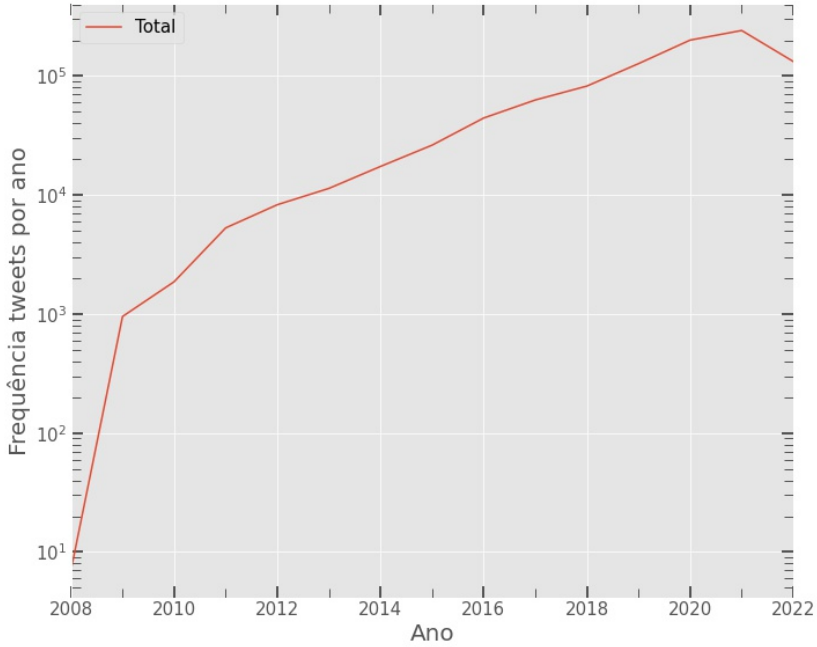
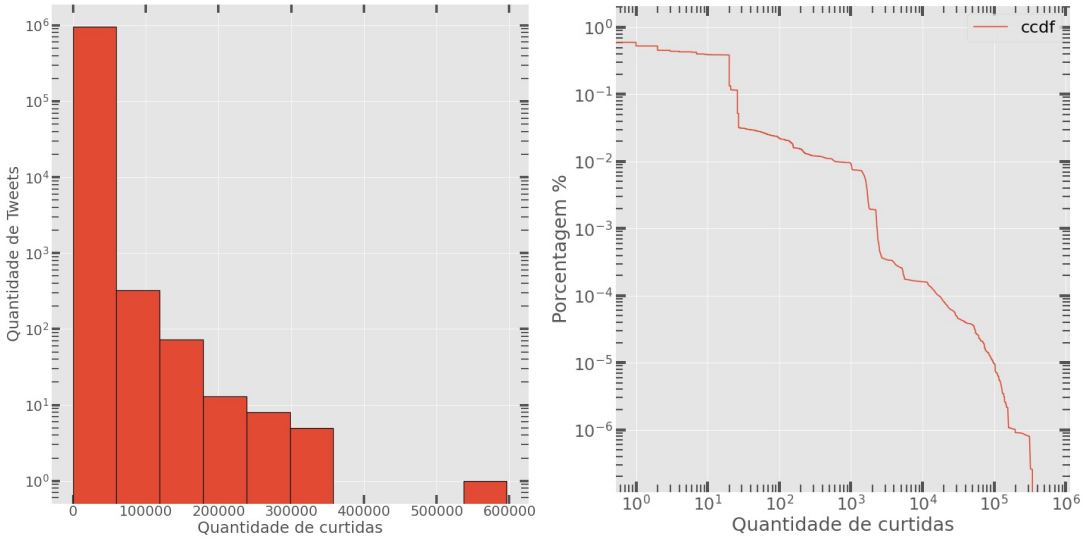


Figura 27 – Frequência de tweets por ano.

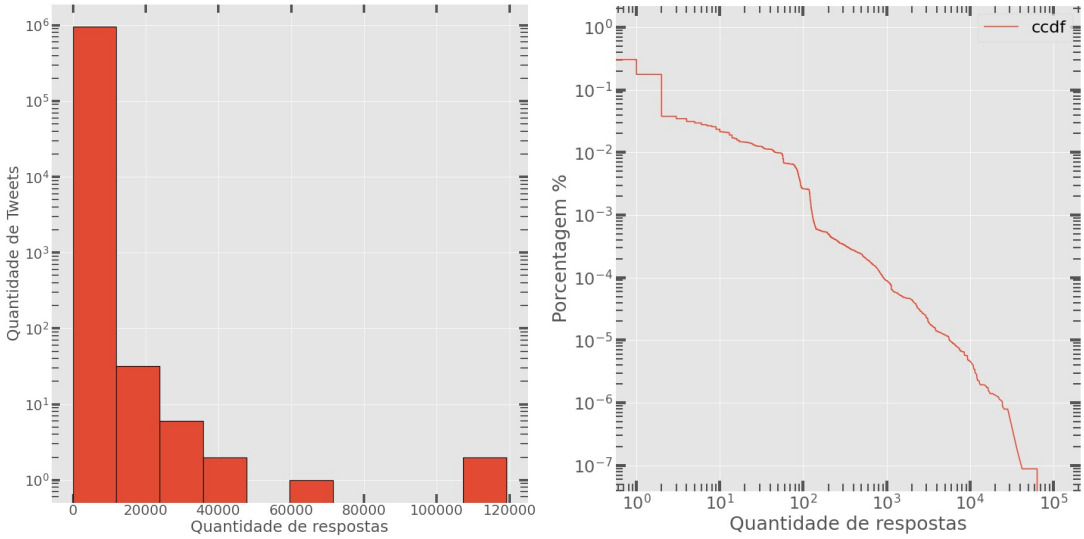
A Figura 27 mostra que desde 2008 a quantidade de tweets postados aumentou de maneira constante. No período de 2021 até 2022, a curva do gráfico indica uma queda na quantidade de tweets, no entanto, isso é devido ao momento em que os dados foram coletados que foi no primeiro semestre de 2022. Dessa forma, o dataset não contém todos os tweets do ano de 2022.

Por fim, foram construídos histogramas e CCDFs dos 4 atributos citados anteriormente para que o comportamento da distribuição dos atributos seja observado e o nível de engajamento das contas que compõem o dataset seja identificado. Sendo assim, percebeu-se a mesma tendência ao que foi apresentada anteriormente com o atributo *seguidores* na subseção 4.2.3. Ou seja, as poucas contas com um grande número de seguidores, são as mesmas que possuem bastante curtidas, respostas, citações e retweets em seus tweets. Ressalta-se que, ao todo, existem aproximadamente 990 mil tweets das contas de jogos no Twitter (vide Tabela 8).



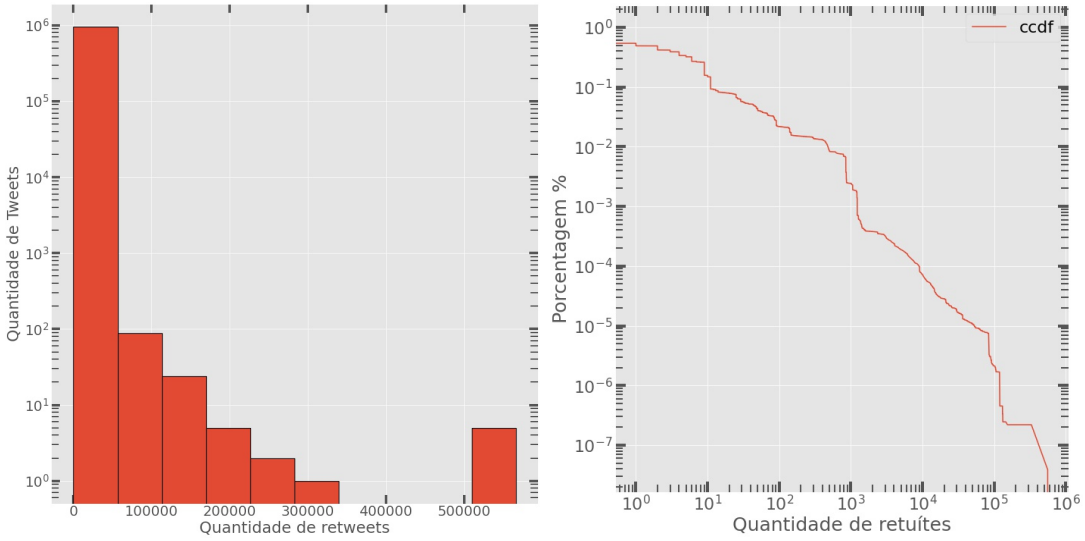
(a) Histograma de curtidas. (b) CCDF de curtidas.

Figura 28 – Distribuição de *curtidas*.



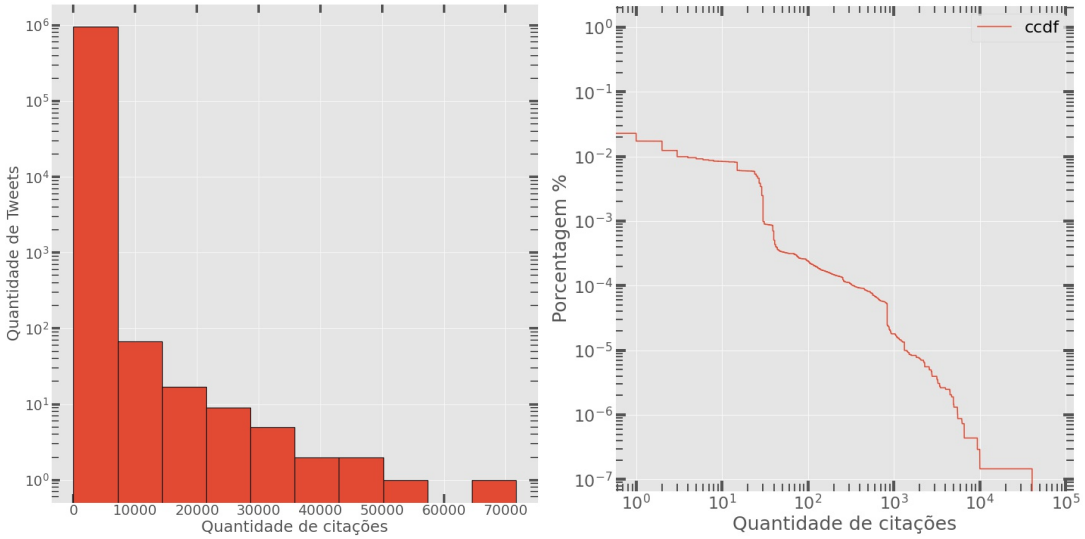
(a) Histograma de respostas. (b) CCDF de respostas.

Figura 29 – Distribuição de *respostas*.



(a) Histograma de retweets. (b) CCDF de retweets.

Figura 30 – Distribuição de *retweets*.



(a) Histograma de citações. (b) CCDF de citações.

Figura 31 – Distribuição de *citações*.

Nas Figura 28, Figura 29, Figura 30 e Figura 31 percebe-se uma certa semelhança na distribuição de ações dos usuários, seja com curtidas, retweets, respostas ou citações. Isso pode indicar a mesma situação demonstrada com a distribuição de número de seguidores por conta no Twitter, ou seja, poucos jogos possuem um maior engajamento, o que é refletido nos atributos de curtidas, retweets, respostas e citações. Além disso, os gráficos indicam que os usuários costumam curtir e "retweetar" mais do que responder ou citar algum tweet. Os histogramas presentes na Figura 28a, Figura 29a, Figura 30a e Figura 31a mostram uma maior frequência em valores menores em termos das ações dos usuários nas contas de jogos no Twitter. Um comportamento de uma distribuição de causa pesada é

observado na distribuição de todas essas ações atividades dos usuários nas contas de jogos. Por exemplo, a Figura 28b mostra que apenas 1% dos tweets – cerca de 10 mil – tem mais do que 1000 curtidas e no máximo até 300 mil curtidas. Na Figura 29b, apenas 0.1% dos tweets (quase mil tweets) apresentam mais do que 100 respostas, reflexo do baixo engajamento dos jogadores (usuários) com relação às contas de jogos.

Por sua vez, a Figura 30b mostra que 98% dos tweets das contas de jogos tem no máximo 100 retweets, novamente uma baixa atividade é percebida para quase todos os jogos, o que significa que apenas cerca de 20 mil tweets tem a maioria das ações de retweets, ressalta-se que essa grande atividade é proveniente de apenas 18 contas de jogos com muitos seguidores, veja Figura 26b da subseção 4.2.3. Finalmente, na Figura 31b mostra que apenas $0,2\%$ dos tweets tem mais do que 1 citação, isso representa somente cerca de 2 mil tweets, o que ressalta ainda mais que apenas poucas contas de jogos possuem muitas citações em seus tweets. Vale destacar que existem tweets com milhares de citações, um deles chegando até mais de 70 mil citações.

5 Conclusões e Trabalhos Futuros

Neste trabalho foi possível construir um dataset com dados de todos os jogos disponíveis na Epic Games Store até o momento da coleta, bem como dados de suas redes sociais, principalmente do Twitter, e avaliações encontradas no site do OpenCritic. Também foi possível realizar análises exploratória de dados e, com base nesses dados coletados, analisar tweets e avaliações relacionadas aos jogos, que representa o engajamento das contas de jogos no Twitter com seus respectivos jogadores (usuários).

Os dados foram coletados através do método de *webscraping*, primeiramente na plataforma da Epic Games Store para buscar todos os jogos disponíveis e seus principais dados. Em seguida, foram utilizadas duas APIs, a do Twitter para coletar todos os tweets de cada jogo que possui uma conta no Twitter e também do site do OpenCritic para coletar todas as avaliações dos jogos que possuem alguma crítica na plataforma. O dataset está disponível na plataforma Zenodo[1].

Além disso, as análises realizadas foram feitas por meio de algoritmos construídos na linguagem Python e com auxílio de bibliotecas disponíveis para a linguagem. Dessa forma, foi possível identificar que o dataset possui dados de 915 jogos, conta com 17.584 avaliações e mais de 1 milhão de tweets. Como resultado das análises percebeu-se um engajamento significativo em um número muito pequeno de contas de jogos no Twitter através da quantidade de seguidores e curtidas, respostas, citações e retweets em seus tweets. Ademais, foi possível identificar os principais gêneros de jogos presentes na Epic Games, bem como as principais empresas desenvolvedoras.

Vale ressaltar uma diferença do dataset construído neste trabalho com relação aos datasets disponíveis pela API da Steam[2], uma informação importante disponibilizada por essa plataforma é o número de horas jogado por cada jogador desde o ingresso dos jogadores na plataforma e o número médio de horas jogadas nas últimas semanas, bem como o número máximo de jogadores simultâneos até um determinado momento. Esses dados são importantes para analisar o envolvimento dos jogadores com os jogos da plataforma da Steam e, certamente, se esses dados estivessem disponíveis na plataforma da Epic Games agregaria ainda mais importância e valor ao dataset construído. Outras informações disponíveis pela Epic Games são as informações sobre os jogadores como conquistas nos jogos, que demonstram a evolução e o desempenho dos jogadores, e também informações de grupos e comunidades nos quais eles participam, os quais contém atividades das interações desses jogadores nos grupos. No entanto, entende-se que isso se deve ao fato de que a plataforma da Steam é mais antiga e, portanto, mais desenvolvida do que a da Epic Games.

[1] Dataset disponível em: <https://zenodo.org/record/7606569>
[2] Site da Steam: <https://store.steampowered.com/>.

Em relação às demais plataformas de jogos citadas no Capítulo 2, a Epic Games ainda é nova no mercado e isso reflete na quantidade pequena de jogos disponíveis na loja, bem como o baixo engajamento da maioria dos jogos em suas redes sociais, principalmente no Twitter. No entanto, foi identificado que a plataforma tem um grande potencial de crescimento e pode ser fonte de muitos trabalhos com temas como análise de sentimento e recomendação de jogos. As Figura 27 e Figura 21 demonstram um pouco desse crescimento, pois, é notório que a cada ano a frequência dos tweets vêm aumentando e implica em um maior engajamento dos usuários com os jogos da plataforma. Bem como a frequência de avaliações dos jogos sendo realizadas com muitas críticas positivas, o que dá ainda mais valor de mercado aos jogos da plataforma.

5.1 Limitações do Trabalho

Neste trabalho o conjunto de dados coletados foi limitado somente aos dados dos jogos encontrados na loja da Epic Games e não consta dados de jogadores. Devido à uma limitação da própria plataforma que não possui um método de interação entre os jogadores e também à API do Twitter que não possibilita coletar as respostas dos tweets dos usuários (jogadores). No entanto, há outros meios de coletar esses dados de jogadores e que podem superar essa limitação que podem ser explorados na continuidade deste trabalho.

5.2 Contribuições

Como já foi dito no Capítulo 2, a plataforma da Epic Games não disponibiliza uma API pública que nos permite coletar dados sobre seus jogos e jogadores como a plataforma da Steam. Dessa forma, este trabalho entrega de forma mais organizada e estruturada dados e informações sobre a plataforma e seus respectivos jogos. Sendo assim, a partir deste trabalho, é possível compreender os tipos de jogos que a Epic Games possui em sua loja, a média de preços, a classificação de um jogo bem avaliado ou não e, como já foi dito, o engajamento desses jogos em suas redes sociais.

Além disso, este trabalho traz a possibilidade de um comparativo entre a Epic Games e outras plataformas de jogos digitais, bem como detalha como é feita a recomendação dos jogos na loja. Contudo, demonstra que, apesar do crescimento em termos financeiros ano após ano, a Epic Games Store ainda é uma plataforma nova em relação às outras e tem como prioridade a sua vitrine de jogos do que uma possível interação entre os jogadores e ainda tem muito para evoluir.

Por fim, por meio de pesquisas e análises aqui realizadas, este trabalho apresenta a relevância do mercado de jogos, principalmente após a pandemia da Covid-19. E, com isso, existem várias possibilidades de pesquisas com relação à esse mercado, bem como a

construção de datasets e análises dos mesmos para, por exemplo, desenvolver melhores recomendações de jogos para determinados tipos de jogadores de acordo com seu perfil e engajamento com os jogos.

5.3 Trabalhos Futuros

Existem algumas possibilidades para uma futura melhora ou crescimento do trabalho tais como, por exemplo, coletar dados dos jogadores que adquirem jogos da loja da Epic Games, bem como coletar as respostas dos tweets dos jogos. Dessa forma, é possível abrir espaço para dois tipos de trabalho: recomendação de jogos e análise de sentimentos dos tweets e das críticas textuais realizadas no site da OpenCritic.

Ao coletar dados dos jogadores que adquirem os jogos pela plataforma, seria possível entender seu comportamento e suas preferências para indicar os melhores jogos para esse usuário (jogador). Além disso, ao coletar as interações entre as contas dos jogos e seus seguidores seria possível realizar análises de sentimento, análises para identificar o engajamento dessas contas e, também, análises mais profundas tais como a construção da rede social dos usuários envolvidos em atividades com os tweets das contas de jogos, o que permite a realização da análise de redes complexas.

Referências

ABU-EL-HAIJA, S. et al. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016. Citado 2 vezes nas páginas 40 e 43.

ADIBI, F.; MAJIDI, B.; ESHGHI, M. Personalized advertisement in the video games using deep social network sentiment analysis. In: *2018 2nd National and 1st International Digital Games Research Conference: Trends, Technologies, and Applications (DGRC)*. [S.l.: s.n.], 2018. p. 104–108. Citado na página 28.

AMÉLIO, C. de O. A indústria e o mercado de jogos digitais no brasil. *XVII SBGames, Foz do Iguaçu, Paraná, Brasil*, p. 1497–1506, 2018. Citado 2 vezes nas páginas 15 e 46.

ANTONIOLLI, D. *O IMPACTO DOS JOGOS ONLINE NA COGNIÇÃO E NAS RELAÇÕES SOCIAIS DO JOVEM ADULTO*. Tese (Doutorado) — PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL, 2020. Citado na página 20.

ARTS, E. *Electronic Arts lança Origin*. 2011. <https://www.ea.com/en-gb/news/electronic-arts-origin-launches>. Citado na página 25.

ARTS, E. *Jogos em destaque para PC*. 2022. <https://www.ea.com/pt-br/games/library/pc-download>. Citado na página 26.

BANKOV, B. et al. The impact of social media on video game communities and the gaming industry. *Varna: University of Economics in Varna*, 2019. Citado na página 22.

BLACKMAN, S. Serious games... and less! *ACM Siggraph Computer Graphics*, ACM New York, NY, USA, v. 39, n. 1, p. 12–16, 2005. Citado na página 21.

BROWNSWORD, A. *Reflecting on development processes in the video game industry*. 2009. 182–182 p. Citado na página 14.

CHAMBERS, C. et al. Characterizing online games. *IEEE/ACM Trans. Netw.*, IEEE Press, v. 18, n. 3, p. 899–910, jun 2010. ISSN 1063-6692. Disponível em: <https://doi-org.ez28.periodicos.capes.gov.br/10.1109/TNET.2009.2034371>. Citado na página 19.

COOPER, J. O. et al. Applied behavior analysis. Pearson/Merrill-Prentice Hall Upper Saddle River, NJ, 2007. Citado na página 15.

COSTA, F. G. d. Visualização de dados e sua importância na era do big data. 2017. Citado na página 33.

CRANE, R.; SORNETTE, D. Robust dynamic classes revealed by measuring the response function of a social system. *Proceedings of the National Academy of Sciences*, v. 105, n. 41, p. 15649–15653, 2008. Citado na página 76.

D'ANDRÉA, C. Para além dos dados coletados: Políticas das apis nas plataformas de mídias digitais. *Matrizes*, v. 15, n. 1, p. 103–122, 2021. Citado na página 26.

DOSHI, Z. et al. Tweeranalyzer: Twitter trend detection and visualization. In: *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)*. [S.l.: s.n.], 2017. p. 1–6. Citado na página 28.

EPAM. *EPIC GAMES: TRANSFORMING THE GAME INDUSTRY*. 2021. <https://www.epam.com/our-work/case-studies/epic-games-transforming-the-gaming-industry>. Citado 3 vezes nas páginas 15, 23 e 46.

FORBES. *Microsoft Acquires 'Gears of War' From Epic, Assigns Next Game To Black Tusk Studios*. 2014. <https://www.forbes.com/sites/danielnyegriffiths/2014/01/27/microsoft-acquires-gears-of-war-from-epic-assigns-next-game-to-black-tusk-studios/?sh=6fffb89f28e2>. Citado na página 24.

GAMES, E. *Epic Games Store*. 2021. <https://store.epicgames.com/en-US>. Citado 8 vezes nas páginas 22, 49, 50, 51, 52, 53, 54 e 55.

GAMES, E. *EPIC GAMES STORE 2020 - ANÁLISE DO ANO*. 2021. <https://www.epicgames.com/store/pt-BR/news/epic-games-store-2020-year-in-review>. Citado 3 vezes nas páginas 15, 23 e 46.

GAMES, E. *Infinity Blade*. 2021. <https://www.epicgames.com/infinityblade/en-US/home>. Citado na página 24.

GAMES, E. *Sobre*. 2021. <https://www.epicgames.com/site/pt-BR/about>. Citado na página 23.

GAMES, E. *Epic Online Services Developer Documentation*. 2022. <https://dev.epicgames.com/docs/game-services/services-overview>. Citado na página 27.

HUANG, H. et al. Data preprocessing method for the analysis of incomplete data on students in poverty. In: *2020 16th International Conference on Computational Intelligence and Security (CIS)*. [S.l.: s.n.], 2020. p. 248–252. Citado 2 vezes nas páginas 32 e 33.

KOTLER, P. *Administracao de Marketing (Bazán Tecnologia e Linguística, Trad.)*. 2000. São Paulo: Prentice Hall. Citado na página 15.

LEE, E. et al. Profit optimizing churn prediction for long-term loyal customers in online games. *IEEE Transactions on Games*, v. 12, n. 1, p. 41–53, 2020. Citado na página 19.

LESNIESKI, M. S. *A Evolução dos Jogos Online: Do RPG ao MMORPG*. 2013. Citado 2 vezes nas páginas 19 e 20.

LIMA, V. Z. de et al. E-sports: a evolução dos jogos online. *Revista Tecnologia e Sociedade*, v. 18, n. 54, p. 227–243, 2022. Citado 3 vezes nas páginas 14, 20 e 22.

LIN, D.; BEZEMER, C.-P.; HASSAN, A. E. An empirical study of early access games on the steam platform. *Empirical Software Engineering*, Springer, v. 23, n. 2, p. 771–799, 2018. Citado na página 24.

LUCCHESE, F.; RIBEIRO, B. Conceituação de jogos digitais. *São Paulo*, p. 7, 2009. Citado na página 20.

NEVES, R. et al. Estratégias publicitárias imersivas no game fortnite. Pontifícia Universidade Católica de Goiás, 2020. Citado na página 24.

NYITRAY, K. J. Game on to game after: Sources for video game history. *Reference and user services quarterly*, American Library Association, v. 59, n. 1, p. 7, 2019. ISSN 1094-9054. Citado na página 14.

O'NEILL, M. et al. *Condensing steam: Distilling the diversity of gamer behavior*. 2016. 81–95 p. Citado 11 vezes nas páginas 14, 15, 16, 24, 25, 27, 28, 31, 34, 35 e 43.

OPENCRITIC. *Frequently Asked Questions*. 2022. <https://opencritic.com/faq>. Citado na página 29.

ORIGIN. *Jogue grandes títulos para PC e conecte-se com seus amigos, tudo em um só lugar.* 2022. <https://www.origin.com/bra/pt-br/store/about#greatpcgames>. Citado na página 25.

PERRIAM, J.; BIRKBAK, A.; FREEMAN, A. Digital methods in a post-api environment. *International Journal of Social Research Methodology*, Taylor & Francis, v. 23, n. 3, p. 277–290, 2020. Citado na página 26.

RENEAR, A. H.; SACCHI, S.; WICKETT, K. M. Definitions of dataset in the scientific and technical literature. *Proceedings of the American Society for Information Science and Technology*, Wiley Online Library, v. 47, n. 1, p. 1–4, 2010. Citado 2 vezes nas páginas 30 e 31.

SAKUDA, L. O. *Plataformas como novo tipo de governança de cadeias globais de valor: estudo na indústria de jogos digitais.* Tese (Doutorado) — Universidade de São Paulo, 2016. Citado na página 22.

SALES, L.-F.; SAYÃO, L.-F. Uma proposta de taxonomia para dados de pesquisa. Universitat de Barcelona, 2019. Citado 2 vezes nas páginas 30 e 31.

STEAM. *Shadow Complex*. 2016. <https://store.steampowered.com/app/385560/Shadow_Complex_Remastered/>. Citado na página 24.

TALEB, I.; DSSOULI, R.; SERHANI, M. A. *Big data pre-processing: A quality framework*. 2015. 191–198 p. Citado 3 vezes nas páginas 31, 32 e 33.

TIKITO, I.; SOUISSI, N. Towards a systematic collect data process. *International Journal of Big Data Intelligence*, Inderscience Publishers (IEL), v. 7, n. 2, p. 72–84, 2020. Citado na página 31.

TODOR, R. *Taxonomia de Games Educativos*. Tese (Doutorado) — PUC-Rio, 2015. Citado na página 21.

TUDO, T. *Steam ou Origin? Descubra qual é a melhor plataforma de jogos para PC*. 2015. <https://www.techtudo.com.br/noticias/2015/10/steam-ou-origin-descubra-qual-e-melhor-plataforma-de-jogos-para-pc.ghtml>. Citado na página 25.

TUDO, T. *Steam, Epic Games Store ou Discord: veja qual loja de jogos é a melhor*. 2021. <https://www.techtudo.com.br/noticias/2019/02/steam-epic-games-store-ou-discord-veja-qual-loja-de-jogos-e-a-melhor.ghtml>. Citado na página 22.

TWITTER. *Documentação API Twitter*. 2021. <https://developer.twitter.com/en/docs>. Citado na página 62.

VAGAVOLU, D. et al. *GE526: A Dataset of Open-Source Game Engines*. 2021. 605–609 p. Citado 3 vezes nas páginas 31, 36 e 43.

WATTENHOFER, M.; WATTENHOFER, R.; ZHU, Z. *The YouTube social network*. 2012. 354–361 p. Citado 3 vezes nas páginas 37, 38 e 43.

WILLIAMS, D.; YEE, N.; CAPLAN, S. E. Who plays, how much, and why? debunking the stereotypical gamer profile. *Journal of computer-mediated communication*, Oxford University Press Oxford, UK, v. 13, n. 4, p. 993–1018, 2008. Citado 2 vezes nas páginas 27 e 28.

ZANGERLE, E. et al. *# nowplaying music dataset: Extracting listening behavior from twitter*. 2014. 21–26 p. Citado 5 vezes nas páginas 32, 34, 39, 40 e 43.

ZHAO, S. et al. *Multi-source Data Multi-task Learning for Profiling Players in Online Games*. 2020. 104-111 p. Citado na página 14.

Anexos

ANEXO A – Código Fonte

Esse anexo contém a documentação do código fonte do trabalho, aqui será demonstrado como o algoritmo de construção do dataset funciona.

A.1 Arquivo principal

Abaixo, no algoritmo 1 temos o pseudo código do arquivo principal da aplicação. Basicamente, este algoritmo irá servir para o usuário escolher quais dados serão requisitados. No entanto, é preciso que a opção 1 seja escolhida primeiro para o restante dos fluxos funcionarem corretamente.

Algoritmo 1: Escolha uma opção para requisitar os dados

Input: Entre com um número: $option = \{1, 2, 3\}$

```
1 Function main():
2     if option = 1 then
3         createGameService.execute()
4     if option = 2 then
5         createGameReviewService.execute()
6     if option = 3 then
7         createTwitterAccountService.execute(TwitterAuthenticate.headers)
8     else
9         print(Invalid option!)
```

A.2 Módulo de Jogos

Ao escolher a opção 1, o primeiro código executado é o algoritmo 2. Nessa primeira parte do algoritmo, foi utilizada a biblioteca *csv* para criar os arquivos que recebem os dados gerais dos jogos e alguns dados adicionais de hardware necessário. Além disso, o cabeçalho de cada arquivo também é escrito neste trecho de código.

Algoritmo 2: Fazendo requisição e formatação dos dados dos jogos

```
1  Function execute():
2      gamesFile ← open('games.csv',' w')
3      gamesWriter ← csv.writer(gamesFile)
4      headerGamesFile ← (id, name, gameSlug, price, releaseDate,
5      platform, description, developer, publisher, genres)
6      gamesWriter.writerow(headerGamesFile)
7
8      necessaryHardwareFile ← open('necessaryHardware.csv',' w')
9      necessaryHardwareWriter ← csv.writer(necessaryHardwareFile)
10     headerNecessaryHardwareFile ←
        (id, operacionalSystem, processor, memory, graphics, fkGameId)
11     necessaryHardwareWriter.writerow(headerNecessaryHardwareFile)
```

Em seguida, é utilizada a biblioteca *requests* para realizar uma requisição no servidor da Epic Games Store com intuito de recuperar dados de todos os jogos da plataforma.

Dessa forma, a resposta da requisição é salva em uma variável. A partir disso, algumas validações precisam ser realizadas. A primeira validação é feita para verificar se a requisição foi bem sucedida, ou seja, com status 200. Em caso verdadeiro, o algoritmo salva o objeto referente aos jogos na variável *games* e também inicializa outras variáveis que irão receber os dados formatados.

Após isso, um laço de repetição é iniciado para percorrer cada posição do array de jogos. Mais duas validações são realizadas para verificar se o array está preenchido. Na linha 16 uma nova requisição é realizada para buscar informações adicionais dos jogos, tais como hardware mínimo e recomendado para se jogar e links para as redes sociais, como demonstrado no algoritmo 3.

Algoritmo 3: Fazendo requisição e formatação dos dados dos jogos

```
1  gamesResponse ← requests.get(url)
2  gamesJson ← json.loads(gamesResponse.text)
3
4  if game ∈ gamesResponse.text then
5      games ← gamesJson[data][Catalog][searchStore][elements]
6      formattedGames ← []
7      formattedAddicionalGameInfos ← []
8      addicionalGameInfos ← []
9      gameIds ← []
10
11     for game ∈ games do
12         if game[catalogNs][mappings] ≠ ∅ then
13             if len(game[catalogNs][mappings]) > 0 then
14                 gameSlug ← game[catalogNs][mappings][0][pageSlug]
15
16                 platform ← ''
17                 genres ← ''
18                 developer ← ''
19                 publisher ← ''
20
21                 addicionalInfoResponse ← requests.get(addicionalInfoUrl)
22                 addicionalInfoJson ← json.loads(addicionalInfoResponse.text)
```

Com a requisição para dados adicionais realizada, é necessário novas validações. A primeira, verificar se o objeto com os dados existe, ou seja, a requisição foi bem sucedida. A próxima verificação é feita duas vezes, uma para a posição zero, caso o objeto desejado se encontre ali e outra na posição 1. Foi utilizado este método, pois nem toda requisição retorna o mesmo formato de dados. Estes passos são demonstrados nos algoritmo 4 e algoritmo 5

Algoritmo 4: Fazendo requisição e formatação dos dados dos jogos

```
1  if pages ∈ addicionalInfoJson then
2      if data ∈ addicionalInfoJson['pages'][0] then
3          addicionalGameInfo ← addicionalInfoJson['pages'][0]['data']
4
5          if platform ∈ addicionalGameInfo['meta'] then
6              platform ←','.join(addicionalGameInfo['meta']['platform'])
7
8          if tags ∈ addicionalGameInfo['meta'] then
9              genres ←','.join(addicionalGameInfo['meta']['tags'])
10
11         if developer ∈ addicionalGameInfo['meta'] then
12             developer ←','.join(addicionalGameInfo['meta']['developer'])
13
14         if publisher ∈ addicionalGameInfo['meta'] then
15             publisher ←','.join(addicionalGameInfo['meta']['publisher'])
16
17         formattedGame ← (
18         id : game['id'],
19         name : game['title'],
20         gameSlug : gameSlug,
21         price : game['currentPrice'],
22         releaseDate : game['releaseDate'],
23         platform : platform,
24         description : game['description'],
25         developer : developer,
26         publisher : publisher,
27         genres : genres
28         )
29         formattedGames.append(formattedGame)
30         formattedAddicionalGameInfo ← (
31         socialNetworks : addicionalGameInfo['socialLinks'],
32         gameId : formattedGame['id']
33         )
34         formattedAddicionalGameInfos.append(formattedAddicionalGameInfo)

35         addicionalGameInfos.append(addicionalGameInfo)
36         gameIds.append(game['id'])
```

Algoritmo 5: Fazendo requisição e formatação dos dados dos jogos

```
else if pages ∈ addicionalInfoJson then
    if data ∈ addicionalInfoJson[pages][1] then
        addicionalGameInfo ← addicionalInfoJson[pages][1][data]

        if platform ∈ addicionalGameInfo['meta'] then
            platform ←','.join(addicionalGameInfo[meta][platform])

        if tags ∈ addicionalGameInfo['meta'] then
            genres ←','.join(addicionalGameInfo[meta][tags])

        if developer ∈ addicionalGameInfo[meta] then
            developer ←','.join(addicionalGameInfo[meta][developer])

        if publisher ∈ addicionalGameInfo[meta] then
            publisher ←','.join(addicionalGameInfo[meta][publisher])

        formattedGame ← (
        id : game[id],
        name : game[title],
        gameSlug : gameSlug,
        price : game[currentPrice],
        releaseDate : game[releaseDate],
        platform : platform,
        description : game[description],
        developer : developer,
        publisher : publisher,
        genres : genres
        )
        formattedGames.append(formattedGame)
        formattedAddicionalGameInfo ← (
        socialNetworks : addicionalGameInfo[socialLinks],
        gameId : formattedGame['id']
        )
        formattedAddicionalGameInfos.append(formattedAddicionalGameInfo)

        addicionalGameInfos.append(addicionalGameInfo)
        gameIds.append(game['id'])
```

Por fim, caso a requisição para buscar os dados adicionais não seja bem sucedida, o algoritmo apenas insere os dados encontrados na primeira requisição no vetor já formatado de jogos. Ao sair do laço de repetição, outros algoritmos são chamados passando alguns parâmetros.

O primeiro algoritmo é o repositório de jogos que irá inserir no arquivo *games.csv* os dados recuperados. O segundo seria para finalizar a formatação dos dados coletados das redes sociais dos jogos. Por fim, o terceiro algoritmo também irá finalizar a formatação dos dados referentes ao hardware necessário de cada jogo. As funções chamadas no algoritmo 6 serão melhor detalhadas nas próximas seções.

Algoritmo 6: Fazendo requisição e formatação dos dados dos jogos

```
else
    formattedGame ← (
    id : game[id],
    name : game[title],
    gameSlug : game[productSlug]or'',
    price : game[currentPrice],
    releaseDate : game[releaseDate],
    platform :'',
    description : game[description],
    developer : game[developerDisplayName]or'',
    publisher : game[publisherDisplayName],
    genres : genres
    )
    formattedGames.append(formattedGame)

gameRepository.create(formattedGames, gamesWriter)
createSocialNetworkService.execute(formattedAddicionalGameInfos)
createNecessaryHardwareService.execute(
addicionalGameInfos,
gameIds,
necessaryHardwareWriter
)

gamesFile.close()
necessaryHardwareFile.close()
```

A.2.1 Repositório dos jogos

Seguindo a estrutura utilizada no script, foi apresentado o arquivo principal, um dos serviços utilizados e agora será mostrado um dos repositórios. No algoritmo 7 a única função construída serve para escrever os dados dos jogos no arquivo *games.csv*. Como foi dito anteriormente, a função recebe dois parâmetros. O primeiro parâmetro é um array com os dados dos jogos já formatado e o segundo é a variável responsável por executar a função que irá escrever os dados no arquivo.

Algoritmo 7: Escrevendo os dados no arquivo *games.csv*

```
   Parameters : formattedGames, gamesWriter
1  Function create():
2      for game ∈ games do
3          values ← (
4          game[id]
5          game[name]
6          game[game_slug]
7          game[price]
8          game[release_date]
9          game[platform]
10         game[description]
11         game[developer]
12         game[publisher]
13         game[genres]
14         )
15     gamesWriter.writerow(values)
```

A.2.2 Buscando Redes Sociais

Neste outro serviço, sua função é formatar os dados recebidos por parâmetro para enviar ao repositório. Dessa forma, a primeira parte do algoritmo demonstrado em algoritmo 8, serve para percorrer o vetor de informações adicionais dos jogos e remover os atributos que não serão utilizados. Por fim, é verificado as redes sociais existentes nos dados dos jogos recebidos e adicionado em um novo vetor.

Algoritmo 8: Buscando dados das redes sociais

```
Parameters : addicionalGameInfos
1  Function execute():
2      formattedSocialNetworks ← []
3      for addicionalGameInfos ∈ addicionalGameInfos do
4          socialNetworks ← addicionalGameInfo[socialNetworks]
5          gameId ← addicionalGameInfo[gameId]
6
7          if type ∈ socialNetworks then
8              deletesocialNetworks[type]
9          if title ∈ socialNetworks then
10             deletesocialNetworks[title]
11         if linkHomepage ∈ socialNetworks then
12             deletesocialNetworks[linkHomepage]
13
14         formattedSocialNetwork ← (
15         fkGameId : gameId
16         )
17
18         for key ∈ socialNetworks do
19             if socialNetworks[key] ≠ ∅ then
20                 formattedSocialNetwork[key] ← socialNetworks[key]
21
22         if len(formattedSocialNetwork)>1 then
23             formattedSocialNetworks.append(formattedSocialNetwork)
```

Na parte final do algoritmo é executado um novo laço de repetição no vetor criado anteriormente para enfim criar um objeto formatado contendo a descrição de qual rede social se trata, o link para acesso à rede social e o atributo identificador do jogo. Esses passos são demonstrados abaixo no algoritmo 9.

Algoritmo 9: Buscando dados das redes sociais

```
Parameters: addicionalGameInfos
Function execute():
    socialNetworks ← []
    id ← 1000
    if len(formattedSocialNetworks) > 0 then
        for socialNetwork ∈ formattedSocialNetworks do
            for key ∈ socialNetwork do
                url ←''
                fkGameId ←''
                description ←''

                if key ≠ fkGameId then
                    description ← key
                    url ← socialNetwork[key]
                    fkGameId ← socialNetwork[fkGameId']

                formattedSocialNetwork ← (
                id : id,
                description : description,
                url : url,
                fkGameId : fkGameId
                )

                if formattedSocialNetwork[fkGameId'] ≠'' then
                    socialNetworks.append(formattedSocialNetwork)

                id ← id + 1

    socialNetworksRepository.create(socialNetworks)
```

A.2.3 Salvando Redes Sociais

Após recuperar os dados das redes sociais e formatá-los de acordo com o que é esperado em sua tabela, é preciso escrever os dados no arquivo. Portanto, é demonstrado no algoritmo 10 que existe uma função para esta funcionalidade. Basicamente, a função recebe um vetor com esses dados formatados, um laço de repetição percorre esse vetor e escreve cada linha em sua posição.

Algoritmo 10: Escrevendo os dados das redes sociais

```
Parameters : socialNetworks
Function create():
    dataFile ← open('socialNetworks.csv','w')
    csvWriter ← csv.writer(dataFile)
    count ← 0

    for socialNetwork ∈ socialNetworks do
        if count = 0 then
            header ← socialNetwork.keys()
            csvWriter.writerow(header)
            count ← count + 1
        values ← socialNetwork.values()
        csvWriter.writerow(values)

    dataFile.close()
```

Ainda neste mesmo arquivo de código existe uma outra função para buscar todos os nomes de usuários de jogos que possuem conta no Twitter. Esta função será utilizada no momento em que os dados do Twitter forem solicitados. Basicamente, a função acessa a tabela de redes sociais (*social_networks*) e filtra os nomes de usuário removendo parte da string do link que não será utilizada. Por fim, retorna todos os nomes de usuário, como demonstrado no algoritmo 11.

Algoritmo 11: Buscando nomes de usuário do Twitter

```
1  Function getAllUsernames():
2      csvFile ← open('socialNetworks.csv','r')
3      csvReader ← csv.DictReader(csvFile)
4
5      formattedUsernames ← []
6      usernames ← []
7
8      for row ∈ csvReader do
9          if row[description] = linkTwitter then
10             usernames.append(row[url])
11
12     for username ∈ usernames do
13         username ←'' .join(username)
14         username ← username.replace('https://twitter.com/','')
15         username ← username.replace('http://twitter.com/','')
16         username ← username.replace('https://www.twitter.com/','')
17         username ← username.replace('http://www.twitter.com/','')
18         username ← username.replace('/','')
19         username ← username.replace('https:','')
20         username ← username.replace('http:','')
21         username ← username.replace('.comtwitter','')
22
23         index ← −1
24
25         if '?' ∈ username then
26             index ← username.index('?')
27         if index > −1 then
28             username ← username[: index − 1]
29         formattedUsernames.append(username)
30     csvFile.close()
31     returnformattedUsernames
```

A.2.4 Buscando hardware necessário

Para finalizar o módulo de dados recuperados através da loja da Epic Games, o último passo é formatar e escrever os elementos relacionados ao hardware necessário do jogo. O algoritmo 12 demonstra o primeiro trecho do código, no qual um laço de repetição

percorre os dados adicionais recebidos e verifica se os campos desejados existem. Por fim, algumas variáveis são inicializadas para receber os valores e realizar novas verificações nos próximos trechos.

Algoritmo 12: Buscando dados dos hardwares necessários

Parameters : addicionalGameInfos, gameIds, necessaryHardwareWriter

```
1  Function execute():
2      for index, addicionalGameInfo ∈ enumerate(addicionalGameInfos) do
3          if requirements ∈ addicionalGameInfo then
4              if systems ∈ addicionalGameInfo[requirements] then
5                  if details ∈ addicionalGameInfo[requirements][systems][0] then
6                      gameId ← gameIds[index]
7                      specifications ←
                        addicionalGameInfo[requirements][systems][0][details]
8
9                      operacionalSystemMinimum ← ∅
10                     processorMinimum ← ∅
11                     memoryMinimum ← ∅
12                     graphicsMinimum ← ∅
13                     storageMinimum ← ∅
14
15                     operacionalSystemRecommended ← ∅
16                     processorRecommended ← ∅
17                     memoryRecommended ← ∅
18                     graphicsRecommended ← ∅
19                     storageRecommended ← ∅
20
21                     hasMinimum ← False
22                     hasRecommended ← False
```

Nesse segundo trecho da função, como demonstrado no algoritmo 13, é o momento de percorrer as especificações do jogo. Dessa forma, as variáveis inicializadas anteriormente serão utilizadas para receber os dados do hardware mínimo e recomendado do jogo.

Algoritmo 13: Buscando dados dos hardwares necessários

Parameters : addicionalGameInfos, gameIds, necessaryHardwareWriter

```
1  Function execute():
2      for specification ∈ specifications do
3          if minimum ∈ specification then
4              hasMinimum ← True
5              if specification[title] = OS then
6                  operacionalSystemMinimum ← specification[minimum]
7              if specification[title] = Processor or specification[title] = CPU
                then
8                  processorMinimum ← specification[minimum]
9              if specification['title'] = Memoryorspecification[title] = RAM
                then
10                 memoryMinimum ← specification[minimum]
11             if specification[title] = Graphics or specification[title] =
                GPU or specification[title] = Video then
12                 graphicsMinimum ← specification[minimum]
13             if specification[title] = Storage or specification[title] = HDD
                then
14                 storageMinimum ← specification[minimum]
15
16     for specification ∈ specifications do
17         if recommended ∈ specification then
18             hasRecommended ← True
19             if specification[title] = OS then
20                 operacionalSystemRecommended ← specification[recommended]
21             if specification[title] = Processor or specification[title] = CPU
                then
22                 processorRecommended ← specification[recommended]
23             if specification[title] = Memory or specification[title] = RAM
                then
24                 memoryRecommended ← specification[recommended]
25             if specification[title] = Graphics or specification[title] =
                GPU or specification[title] = Video then
26                 graphicsRecommended ← specification[recommended]
27             if specification[title] = Storage or specification[title] = HDD
                then
28                 storageRecommended ← specification[recommended]
```

Por fim, as duas variáveis booleanas inicializadas anteriormente verificam se os dados que estão sendo buscados realmente estão na resposta da requisição. Caso exista, os objetos de mínimo e recomendado são preenchidos e enviados ao repositório responsável pelas redes sociais para escrever os dados em seu respectivo arquivo csv.

Algoritmo 14: Buscando dados dos hardwares necessários

Parameters : addicionalGameInfos, gameIds, necessaryHardwareWriter

```
1  Function execute():
2      if hasMinimum then
3          minimumFormatted ← (
4          operacionalSystemMinimum,
5          processorMinimum,
6          memoryMinimum,
7          graphicsMinimum,
8          storageinimum
9          )
10
11         minimum ←′ 1′
12         necessaryHardwareRepository.create(
13         minimumFormatted,
14         minimum,
15         gameId,
16         necessaryHardwareWriter)
17
18     if hasRecommended then
19         recommendedFormatted ← (
20         operacionalSystemRecommended,
21         processorRecommended,
22         memoryRecommended,
23         graphicsRecommended,
24         storageRecommended
25         )
26
27         recommended ←′ 2′
28         necessaryHardwareRepository.create(
29         minimumFormatted,
30         minimum,
31         gameId,
32         necessaryHardwareWriter)
```

O repositório referente aos hardwares necessários é bem simples. Basicamente, a função irá receber o objeto com os dados, uma flag para saber se os dados se tratam do mínimo ou do recomendado, o atributo identificador do jogo e a variável responsável por escrever os dados no arquivos csv. Portanto, a função insere os dados em uma lista e os escreve na tabela, como demonstrado no algoritmo 15.

Algoritmo 15: Escrevendo dados dos hardwares necessários

Parameters : hardware, minimumRecommended, gameId, necessaryHardwareWriter

```
1  Function create():
2      id ← gameId + minimumRecommended
3
4      values ← (
5      id,
6      hardware[operacionalSystem],
7      hardware[processor],
8      hardware[memory],
9      hardware[graphics],
10     hardware[storage],
11     gameId
12     )
13
14     necessaryHardwareWriter.writerow(values)
```

A.3 Módulo de Avaliações

Ao escolher a opção 2, como demonstrado na Figura 16, o script busca pelas avaliações dos jogos que se encontram na plataforma da OpenCritic. O algoritmo 16 exibe um trecho da primeira função executada. Nesta função, o algoritmo realiza uma requisição HTTP na API da OpenCritic e busca todos os jogos disponíveis na plataforma.

No trecho abaixo, o script escreve o cabeçalho do arquivo csv que será salvo as avaliações. Em seguida, realiza a requisição para buscar os jogos e salva o id e o nome do jogo em um vetor. A variável *skip* ao final será utilizada para a paginação dos dados na url da requisição.

Algoritmo 16: Bsucando avaliações dos jogos

```
1  Function execute():
2      openCriticFile ← open('openCritic.csv',' w')
3      openCriticWriter ← csv.writer(openCriticFile)
4      headerOpenCriticFile ← (
5      'id',' company',' author',' rating',' comment',' date',' topCritic',' gameId'
6      )
7      openCriticWriter.writerow(headerOpenCriticFile)
8
9      response ← requests.get("https : //api.opencritic.com/api/game")
10
11     games ← json.loads(response.text)
12
13     formattedGames ← []
14     for game ∈ games do
15         formattedGame ← (
16         game[id],
17         game[name]
18         )
19         formattedGames.append(formattedGame)
20
21     skip ← 20
```

Em seguida, no algoritmo 17 com os dados já formatados, um laço de repetição percorre este vetor e utiliza o repositório do módulo OpenCritic para verificar se o jogo existe na Epic Games. Caso exista, uma nova requisição é realizada à API do OpenCritic para buscar as avaliações do jogo no qual o atributo identificador é passado por parâmetro.

Com os dados das avaliações recebidos, um novo laço de repetição é utilizado para percorrer o vetor de avaliações. Por fim, cada variável desejada é adicionada ao objeto referente à avaliação e enviado para o repositório escrever em seu respectivo arquivo csv.

Contudo, o trecho demonstrado no algoritmo 18 é um novo laço de repetição que itera enquanto existir avaliações, pois a cada requisição só é retornado 20 objetos. Portanto, a API utiliza uma paginação para não enviar todos os dados de uma vez. O restante do trecho segue os mesmos passos do algoritmo 17.

Algoritmo 17: Buscando avaliações dos jogos

```
Function execute():
    for formattedGame ∈ formattedGames do
        epicGamesId ← gameReviewRepository.findGame(formattedGame)
        openCritcGameId ← formattedGame[id]

        if epicGamesId ≠ None then
            responseReview ← requests.get("https:
              //api.opencritic.com/api/review/game/openCritcGameId")

            reviews ← json.loads(responseReview.text)
            for review ∈ reviews do
                authors ← []

                for author ∈ review[Authors] do
                    authors.append(author[name])
                score ←''
                snippet ←''
                if score ∈ review then
                    score ← review[score]
                if snippet ∈ review then
                    snippet ← review[snippet]

                formattedReview ← (
                review[_id],
                review[Outlet][name],
                ','.join(authors),
                score,
                snippet,
                review[publishedDate],
                notreview['Outlet']['isContributor']
                epicGamesId
                )

                gameReviewRepository.create(formattedReview, openCriticWriter)

            skipReview ← 20
```

Algoritmo 18: Buscando avaliações dos jogos

```
1  Function execute():
2      while len(reviews) > 0 do
3          responseReview = requests.get("https :
             //api.opencritic.com/api/review/game/openCritcGameIdskip =
             skipReview")
4
5          reviews ← json.loads(responseReview.text)
6          for review ∈ reviews do
7              authors ← []
8
9              for author ∈ review[Authors] do
10                 authors.append(author[name])
11
12             score ←″
13             snippet ←″
14
15             if score ∈ review then
16                 score ← review[score]
17             if snippet ∈ review then
18                 snippet ← review[snippet]
19
20             formattedReview ← (
21             review[_id],
22             review[Outlet][name],
23             ′,′ .join(authors),
24             score,
25             snippet,
26             review[publishedDate],
27             notreview[′Outlet′][′isContributor′]
28             epicGamesId
29             )
30
31             gameReviewRepository.create(formattedReview, openCriticWriter)
32
33             skipReview ← skipReview + 20
```

A.3.1 Repositório das avaliações

Para finalizar a parte de coleta de avaliações dos jogos, será demonstrado a seguir o repositório de avaliações. Essa classe é composta por 2 funções simples e com nomes bem intuitivos. A primeira função, como mostrado no algoritmo 19 irá apenas escrever no arquivo *open_critic.csv* os dados recebidos por parâmetros e coletados diretamente da API do OpenCritic.

Algoritmo 19: Repositório das avaliações dos jogos

Parameters : formattedReview, openCriticWriter

```
1 Function create():
2     openCriticWriter.writerow(formattedReview)
```

Por fim, a segunda função é chamada também nos algoritmos da seção anterior. Seu objetivo é basicamente retornar o atributo identificador do jogo solicitado que está salvo na tabela *games*, como demonstrado no algoritmo 20.

Algoritmo 20: Repositório das avaliações dos jogos

Parameters : formattedGame

```
1 Function findGame():
2     csvFile ← open('games.csv')
3     csvReader ← csv.DictReader(csvFile)
4
5     for row ∈ csvReader do
6         if formattedGame['name'] ∈ row['name'] then
7             returnrow['id']
```

A.4 Módulo do Twitter

Para finalizar a descrição dos scripts será apresentado o último módulo do algoritmo, o módulo de busca de dados no Twitter. Como demonstrado na Figura 16 e no algoritmo 1, ao escolher a opção 3 serão feitas as buscas de dados na API do Twitter.

A função demonstrada no algoritmo 21 recebe como parâmetro a variável *headers*. Esta variável contém as chaves de autenticação necessárias para acessar a API do Twitter, como explicado anteriormente. No início da função, um arquivo denominado *twitterAccounts.csv* é preparado para receber os dados que serão requisitados.

Em seguida, é preciso acessar a tabela *social_networks* para buscar todos os nomes de usuário de jogos que possuem uma conta no Twitter. Dessa forma, será possível utilizar o endpoint disponível na API para buscar os dados dessa conta através do nome de usuário.

Como não é possível buscar mais de 100 contas de uma só vez na requisição da

API, é preciso utilizar uma paginação para buscar de 100 em 100 contas. Sendo, após buscar todos os nomes de usuário, o script salva os 100 primeiros nomes do array em uma variável e inicia uma laço de repetição em cada posição desse novo array de 100 nomes de usuário. Ao encerrar todos estas 100 posições, o vetor é preenchido novamente com os próximos 100 valores. Contudo, o script segue dessa forma até buscar os dados de todos os nomes de usuário recuperados anteriormente.

Algoritmo 21: Buscando dados da conta no Twitter

```
   Parameters: headers
1  Function execute():
2      twitteAccountsFile ← open('twitterAccounts.csv','w')
3      twitterAccountsWriter ← csv.writer(twitterAccountsFile)
4      headerTwitterAccountsFile ← ('id','name','username','bio','location',
5      'website','joinDate','following','followers','fkGameId')
6      twitterAccountsWriter.writerow(headerTwitterAccountsFile)
7      usernames ← socialNetworksRepository.getAllUsernames()
8      if len(usernames) > 0 and len(usernames) <= 100 then
9          paginationUsernames ← usernames
10         usernames ← []
11     else
12         paginationUsernames ← usernames[: 100]
13         delusernames[: 100]
14
15     while len(paginationUsernames) > 0 do
16         response ← requests.get("https :
            //api.twitter.com/2/users/by?paginationUsernames", headers =
            headers)
17         if 'data' ∈ json.loads(response.text) then
18             twitterAccounts ← json.loads(response.text)['data']
19             twitterAccountsRepository.create(twitterAccounts,
20             twitterAccountsWriter)
21             twitterAccountIds ← []
22             twitterAccountUsernames ← []
23
24             for twitterAccount ∈ twitterAccounts do
25                 if twitterAccount['protected'] = False then
26                     twitterAccountIds.append(twitterAccount['id'])
27                     twitterAccountUsernames.append(twitterAccount['username'])
28             createTweetService.execute(twitterAccountIds, headers)
29             if len(usernames) > 0 and len(usernames) < 100 then
30                 paginationUsernames ← usernames
31                 usernames ← []
32             else
33                 paginationUsernames ← usernames[: 100]
34                 delusernames[: 100]
```

Abaixo, no algoritmo 22 há um *Else* caso a tentativa de requisição falhe e não encontre os dados. Dessa forma, os 100 nomes de usuários são substituídos por novos caso esteja faltando mais algum, senão só remove os nomes de usuário. Por fim, o arquivo *twitterAccountsFile* é fechado.

Algoritmo 22: Buscando dados da conta no Twitter

```
   Parameters: headers
1  Function execute():
2      else
3          if len(usernames) > 0 and len(usernames) < 100 then
4              paginationUsernames ← usernames
5              usernames ← []
6          else
7              paginationUsernames ← usernames[: 100]
8              delusernames[: 100]
9
10     twitterAccountsFile.close()
```

A.4.1 Repositório de Contas no Twitter

Nota-se que o algoritmo 21 chama outras duas funções passando alguns parâmetros. A primeira função chamada (linha 19) está no repositório, no qual irá escrever de fato os dados coletados no arquivo. A segunda função (linha 28) chama o *Service* dos Tweets para requisitados os dados dos mesmos.

Nesta subseção será descrito a função de criar no repositório. Sendo assim, a função recebe como parâmetro um array com os dados das contas no twitter coletados na requisição (*twitterAccounts*) e a variável responsável por escrever os dados no arquivo (*twitterAccountsWriter*).

Já no início da função um laço de repetição é iniciado para percorrer o vetor com os dados recebidos. Em seguida, é preciso buscar o atributo identificador do jogo na tabela *socialNetworks* para salvar também na tabela *twitterAccounts*. Como o link do twitter pode ser retornado da loja da Epic Games em vários formatados, foi necessários realizar algumas validações para buscar de forma assertiva este atributo através do nome de usuário do jogo.

Caso encontre o atributo identificador do jogo, um novo objeto é preparado para enfim escrever os dados recebidos no arquivo corretamente, como demonstrado no algoritmo 23.

Algoritmo 23: Salvando dados da conta do Twitter

```
Parameters: twitterAccounts, twitterAccountsWriter
Function create():
    for twitterAccount ∈ twitterAccounts do
        httpUrl ←′ http : //twitter.com/′ + twitterAccount[′username′].lower()
        httpsUrl ←′ https : //twitter.com/′ + twitterAccount[′username′].lower()
        httpsUrlwww ←′ http :
         //www.twitter.com/′ + twitterAccount[′username′].lower()
        httpUrlwww ←′ https :
         //www.twitter.com/′ + twitterAccount[′username′].lower()

        gameId ← None
        csvFile ← open(′socialNetworks.csv′,′ r′)
        csvReader ← csv.DictReader(csvFile)
        for row ∈ csvReader do
            if row[′url′].lower() = httpUrl or
            row[′url′].lower() = httpsUrl or
            row[′url′].lower() = httpsUrlwww or
            row[′url′].lower() = httpUrlwww
             then
                gameId ← row[′fkGameId′]

        if gameId ! = None then
            location ←″
            if ′location′ ∈ twitterAccount then
                location ← twitterAccount[′location′]

            values ← (
            twitterAccount[′id′],
            twitterAccount[′name′],
            twitterAccount[′username′],
            twitterAccount[′description′],
            location,
            twitterAccount[′url′],
            twitterAccount[′createdAt′],
            twitterAccount[′publicMetrics′][′followingCount′],
            twitterAccount[′publicMetrics′][′followersCount′],
            gameId
            ) twitterAccountsWriter.writerow(values)
    csvFile.close()
```

A.4.2 Buscando dados dos tweets

Como foi dito anteriormente, o algoritmo 21 chama outras duas funções. Nesta subseção será explicado como funciona a coleta dos tweets das contas que já foram salvas.

Sendo assim, a função recebe dois parâmetros: *twitterAccountIds* e *headers*. Como os nomes já dizem, a primeira representa os IDs das contas do twitter que serão utilizados para requisitar os dados na API. Já o headers é o cabeçalho que contém as chaves de autenticação necessárias para acesso à API.

No início do algoritmo 24, o arquivo que será utilizado para escrever os dados é preparado para ser utilizado, como também é escrito o cabeçalho do arquivo com os nomes dos dados que serão coletados. Neste endpoint também é necessário fazer a paginação dos dados, pois o máximo de tweets que são retornados é 100. Com isso, um laço de repetição é iniciado percorrendo os 100 primeiros IDs recebidos por parâmetro. Em seguida, uma condicional verifica se a requisição retornou os dados esperados, em casos positivo, um outro laço de repetição é iniciado para percorrer todos os tweets retornados na requisição.

Algoritmo 24: Buscando dados dos Tweets

```
Parameters: twitterAccountIds, twitterAccountsWriter
Function execute():
    tweetsAccountsFile ← open('tweets.csv','w')
    tweetsAccountsWriter ← csv.writer(tweetsAccountsFile)
    headerTweetsAccountsFile ← (
    'id','text','urlMedia','quantityLikes','quantityRetweets','quantityQuotes',
    'quantityReplys','timestamp','twitterAccountId')
    tweetsAccountsWriter.writerow(headerTweetsAccountsFile)

    for twitterAccountId ∈ twitterAccountIds do
        url ← "https://api.twitter.com/2/users/twitterAccountId/tweets"
        response ← requests.get(url, headers = headers)

        if 'data' ∈ json.loads(response.text) then
            tweets ← json.loads(response.text)['data']
            medias ← []
            haveMedias ← False
            if 'includes' ∈ json.loads(response.text) then
                if 'media' ∈ json.loads(response.text)['includes'] then
                    medias ← json.loads(response.text)['includes']['media']
                    haveMedias ← True
            for tweet ∈ tweets do
                publicMetrics ← tweet['publicMetrics']
                formattedTweet ← ('id' : tweet['id'],'text' : tweet['text'],
                'url_media' : (),'quantity_likes' : publicMetrics['like_count'],
                'quantityRetweets' :
                 publicMetrics['retweetCount'],'quantity_quotes' :
                 publicMetrics['quoteCount'],
                'quantityReplys' : publicMetrics['replyCount'],'timestamp' :
                 tweet['createdAt'],
                'in_reply_to_user_id' : tweet['in_reply_to_user_id'],'twitterAccountId' :
                 twitterAccountId,
                )
```

Ainda dentro do laço de repetição para finalizar a primeira parte, os dados são separados em um objeto e novas verificações são realizados para saber se alguns outros dados extras estão contidos no tweet. Esses dados são adicionados ao objeto e a função

chama pelo repositório de Tweets para escrever os dados no arquivo. Essa primeira parte é demonstrada no algoritmo 25.

Algoritmo 25: Buscando dados dos Tweets

Parameters : twitterAccountIds, twitterAccountsWriter

```
1  Function execute():
2      if 'attachments' ∈ tweet then
3          if 'mediaKeys' ∈ tweet['attachments'] then
4              mediaKeys ← tweet['attachments']['mediaKeys']
5              if haveMedias = True then
6                  for media ∈ medias do
7                      for mediaKey ∈ mediaKeys do
8                          if media['mediaKey'] =
                            mediaKey and 'previewImageUrl' ∈ media then
9                              formattedTweet['urlMedia']+ =
                                (media['previewImageUrl'], )
10
11     if len(formattedTweet['urlMedia']) > 0 then
12         formattedTweet['urlMedia'] ←','.join(formattedTweet['urlMedia'])
13     else
14         formattedTweet['urlMedia'] ←''
15     tweetRepository.create(formattedTweet, tweetsAccountsWriter)
```

Para a segunda parte do algoritmo, ainda dentro do primeiro laço de repetição do algoritmo é verificado se existe um *next_token* que, basicamente, significa que ainda existe uma próxima página de dados para ser retornados. Os passos seguidos no algoritmo 26 e algoritmo 27 são praticamente os mesmos dos algoritmo 24 e algoritmo 25, respectivamente. A diferença está na adição do *next_token* para coleta dos próximos dados.

Algoritmo 26: Buscando dados dos Tweets

Parameters : twitterAccountIds, twitterAccountsWriter

```
1  Function execute():
2      if 'meta' ∈ json.loads(response.text) then
3          while 'next_token' ∈ json.loads(response.text)['meta'] do
4              nextToken ← json.loads(response.text)['meta']['next_token']
5              url ← ("https://api.twitter.com/2/users/twitterAccountId?pagination_token=
                 next_token")
6              response ← requests.get(url, headers = headers)
7
8              if 'data' ∈ json.loads(response.text) then
9                  tweets ← json.loads(response.text)['data']
10                 medias ← []
11                 haveMedias ← False
12
13                 if 'includes' ∈ json.loads(response.text) then
14                     if 'media' ∈ json.loads(response.text)['includes'] then
15                         medias ← json.loads(response.text)['includes']['media']
16                         haveMedias ← True
17
18                 for tweet ∈ tweets do
19                     publicMetrics ← tweet['publicMetrics']
20
21                     formattedTweet ← ('id' : tweet['id'],'text' : tweet['text'],
22                     'urlMedia' : (),'quantityLikes' : publicMetrics['likeCount'],
23                     'quantityRetweets' :
                         publicMetrics['retweetCount'], quantityQuotes' :
                         publicMetrics['quoteCount'],
24                     'quantityReplys' : publicMetrics['replyCount'],'timestamp' :
                         tweet['createdAt'],
25                     'in_reply_to_user_id' : tweet['in_reply_to_user_id'],'twitterAccountId' :
                         twitterAccountId,
26                     )
```

Algoritmo 27: Buscando dados dos Tweets

```
   Parameters : twitterAccountIds, twitterAccountsWriter
1  Function execute():
2      if 'attachments' ∈ tweet then
3          if 'mediaKeys' ∈ tweet['attachments'] then
4              mediaKeys ← tweet['attachments']['mediaKeys']
5              if haveMedias = True then
6                  for media ∈ medias do
7                      for mediaKey ∈ mediaKeys do
8                          if media['mediaKey'] =
                            mediaKey and 'previewImageUrl' ∈ media then
9                              formattedTweet['urlMedia']+ =
                                (media['previewImageUrl'],)
10
11     if len(formattedTweet['urlMedia']) > 0 then
12         formattedTweet['urlMedia'] ←','.join(formattedTweet['urlMedia'])
13     else
14         formattedTweet['urlMedia'] ←''
15     tweetRepository.create(formattedTweet, tweetsAccountsWriter)
```

A.4.3 Repositório dos Tweets

Para concluir a descrição dos scripts, nesta subseção será demonstrado a última funcionalidade do código. Ela segue o padrão do restante dos repositórios, sendo assim, esta função irá basicamente receber os dados do tweet e a variável responsável por escrever os dados no arquivo *tweets.csv*. Dessa forma, o algoritmo 28 recebe os dados que são enviados pelo algoritmo da subseção anterior, adiciona em um objeto formatado e o escreve no arquivo.

Algoritmo 28: Salvando dados da conta do Twitter

Parameters : tweet, tweetsAccountsWriter

```
1  Function create():
2      values ← (
3      tweet['id'],
4      tweet['text'],
5      tweet['urlMedia'],
6      tweet['quantityLikes'],
7      tweet['quantityRetweets'],
8      tweet['quantityQuotes'],
9      tweet['quantityReplys'],
10     tweet['timestamp'],
11     tweet['in_reply_to_user_id'],
12     tweet['twitterAccount_id']
13     )
14
15     tweetsAccountsWriter.writerow(values)
```

Índice