



UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
ESCOLA DE INFORMÁTICA APLICADA

Utilizando Dados Conectados na Consulta de Informações de Jogos de Entretenimento

Mateus dos Santos Dorneles

Orientador

Tadeu Moreira de Classe

RIO DE JANEIRO, RJ – BRASIL
NOVEMBRO DE 2024

UTILIZANDO DADOS CONECTADOS NA CONSULTA DE INFORMAÇÕES DE
INFORMAÇÕES DE JOGOS DE ENTRETENIMENTO

MATEUS DOS SANTOS DORNELES

Projeto de Graduação apresentado à Escola de
Informática Aplicada da Universidade Federal do
Estado do Rio de Janeiro (UNIRIO) para obtenção do
título de Bacharel em Sistemas de Informação.

Aprovado por:

Documento assinado digitalmente
 Tadeu Moreira de Classe
Data: 11/11/2024 11:29:11-0300
Verifique em <https://validar.iti.gov.br>

Tadeu Moreira de Classe, D.Sc. (UNIRIO)

Documento assinado digitalmente
 PEDRO NUNO DE SOUZA MOURA
Data: 11/11/2024 11:37:51-0300
Verifique em <https://validar.iti.gov.br>

Pedro Nuno de Souza Moura, D.Sc. (UNIRIO)

Documento assinado digitalmente
 EDUARDO GOMES DE OLIVEIRA
Data: 11/11/2024 14:09:02-0300
Verifique em <https://validar.iti.gov.br>

Eduardo Gomes de Oliveira, M.Sc. (Colégio Pedro II e UNIRIO)

RIO DE JANEIRO, RJ – BRASIL.

NOVEMBRO DE 2024

Catálogo informatizada pelo(a) autor(a)

dd713u dos Santos Dorneles, Mateus
Utilizando Dados Conectados na Consulta de Informações
de Jogos de Entretenimento / Mateus dos Santos Dorneles. --
Rio de Janeiro : UNIRIO, 2024.
57

Orientador: Tadeu Moreira de Classe.
Trabalho de Conclusão de Curso (Graduação) -
Universidade Federal do Estado do Rio de Janeiro, Graduação
em Sistemas de Informação, 2024.

1. Dados Conectados. 2. Jogos de Entretenimento. 3.
Desenvolvimento de Software. I. Moreira de Classe, Tadeu,
orient. II. Título.

RESUMO

O vertiginoso crescimento da indústria de jogos eletrônicos, decorrente da sinergia entre a expansão do público-alvo, a evolução tecnológica, a proliferação de plataformas de distribuição e a natureza interdisciplinar da criação de jogos, demanda uma análise aprofundada dos dados gerados por este setor. Mediante a coleta, tratamento e análise desses dados, é possível compreender as preferências dos jogadores e, consequentemente, desenvolver experiências de jogos cada vez mais personalizadas e engajantes. No mundo atual, de forma geral orientado por dados, as organizações não medem esforços para coletar grandes quantidades de dados de fontes cada vez mais variadas. Não obstante, a ausência de uma integração confiável e eficaz de informações pode impedir a obtenção de *insights* relevantes. Neste contexto, a presente pesquisa tem como objetivo apresentar o desenvolvimento de uma ferramenta destinada a facilitar a busca, consulta e análise de dados de jogos provenientes de múltiplas fontes.

Palavras-Chave: Dados Conectados; Jogos de Entretenimento; Desenvolvimento de *Software*; *Python*; *Web Crawler*.

ABSTRACT

The dizzying growth of the electronic games industry, resulting from the synergy between the expansion of the target audience, technological evolution, the proliferation of distribution platforms and the interdisciplinary nature of game creation, demands an in-depth analysis of the data generated by this sector. By collecting, processing and analyzing this data, it is possible to understand players' preferences and, consequently, develop increasingly personalized and engaging gaming experiences. In today's increasingly data-driven world, organizations go to great lengths to collect large amounts of data from diverse sources. However, the absence of reliable and effective information integration can prevent relevant insights from being obtained. In this context, the present research aims to present the development of a tool designed to facilitate the search, consultation and analysis of game data from multiple sources.

Keywords: *Connected Data; Entertainment Games; Software Development; Python; Web Crawler.*

ÍNDICE

1. INTRODUÇÃO.....	10
1.1. Problema.....	11
1.2. Objetivo	11
1.3. Justificativa.....	12
1.4. Metodologia.....	14
1.5. Organização do Texto.....	15
2. FUNDAMENTAÇÃO TEÓRICA	16
2.1. Jogos Digitais e Video Games.....	16
2.2. Dados Conectados	17
2.3. Taxonomia de Domínio	18
2.4. <i>Web Crawlers</i>	18
3. GAMES FINDER APPLICATION	20
3.1. Taxonomia de Conceitos de Jogos	20
3.2. Arquitetura do Sistema	23
3.2.1. Requisitos e Regras de Negócio	24
3.2.2. Diagramas <i>UML</i>	25
3.2.3. Tecnologias utilizadas	28
3.3. <i>Games Crawler</i>	29
3.3.1. Metacritic Crawler.....	29
3.3.2. GOG Crawler.....	32
3.3.3. IGN Crawler	35
3.3.4. Resultados e desempenho.....	37
3.4. <i>Games Finder</i>	39
4. ILUSTRAÇÃO DE USO DO <i>GAMES FINDER APPLICATION</i>	44
5. CONSIDERAÇÕES FINAIS	49
REFERÊNCIAS BIBLIOGRÁFICAS	52

Apêndice 1. CASOS DE USO DO SISTEMA.....	55
--	----

LISTA DE FIGURAS

Figura 1 - Etapas da Metodologia.....	14
Figura 2 - Mind map de dados conectados	22
Figura 3 - Diagrama de Arquitetura do Projeto	23
Figura 4 - Diagrama de Casos de Uso do <i>Games Finder Application</i>	26
Figura 5 - Diagrama de Classes do <i>Games Finder Application</i>	27
Figura 6 - Página da lista de todos os jogos do site <i>Metacritic</i>	30
Figura 7 – Trecho de código responsável em armazenar todas as <i>URLs</i> do <i>Metacritic</i> . 31	
Figura 8 – Trecho de código que realiza a leitura dos campos da página	31
Figura 9 - Trecho do arquivo <i>JSON</i> de saída do <i>Metacritic Crawler</i>	32
Figura 10 - Página da lista de todos os jogos do site GOG	33
Figura 11 - Trecho de código responsável em armazenar todas as <i>URLs</i> do GOG	34
Figura 12 - Trecho de código que realiza a leitura dos campos da página.....	34
Figura 13 - Trecho do arquivo <i>JSON</i> de saída do GOG Crawler	35
Figura 14 - Trecho de código que verifica se a <i>URL</i> é válida e realiza a leitura dos campos da página.....	36
Figura 15 - Trecho do arquivo <i>JSON</i> de saída do <i>IGN Crawler</i>	37
Figura 16 - Quantidade de jogos coletados	38
Figura 17 - Relação de média de jogos coletados por hora	39
Figura 18 - Trecho de código de leitura dos três arquivos <i>JSON</i>	40
Figura 19 - Trecho de código responsável em unificar os três <i>dataframes</i>	41
Figura 20 - Trecho de código da página inicial da aplicação	42
Figura 21 - Trecho de código da página de detalhes de um jogo	42
Figura 22 - Trecho de código da aplicação responsável pelas operações de busca no <i>dataframe</i>	43
Figura 23 - Página Inicial	44
Figura 24 - Opções de paginação	45
Figura 25 - Opções de busca	46
Figura 26 - Opção de busca selecionada e termo de busca digitado	46
Figura 27 - Resultado de Busca.....	47
Figura 28 - Página de Detalhes do Jogo	48

LISTA DE TABELAS

Tabela 1 - Requisitos Funcionais do Sistema.....	24
Tabela 2 - Requisitos Não Funcionais do Sistema	25
Tabela 3 - Regras de Negócio.....	25

1. INTRODUÇÃO

Na atualidade, o público de jogos eletrônicos está em um crescimento constante globalmente. Mais especificamente no Brasil, estima-se que 3 a cada 4 brasileiros jogam *games* (Tadeu e Tortella, 2022). A estimativa é que até o fim do ano de 2024 o número de pessoas no mundo todo que vão jogar jogos digitais pode atingir o número de 3,4 bilhões (Newzoo, 2024). Supõe-se que o mercado de games irá gerar 187,7 bilhões de dólares até o fim do ano de 2024 (Newzoo, 2024).

Em um mundo onde há tanto valor dentro da indústria de jogos, a busca por informações relacionadas a esse mercado tende a crescer cada vez mais. Os dados referentes aos jogos assumem uma relevância significativa (Hope, 2023).

Considerando essa relevância, existem diversos *websites* que armazenam informações tanto de jogos lançados recentemente quanto de jogos que foram criados décadas atrás. Contudo, não existe uma base consolidada onde seja possível encontrar todas as informações que estão distribuídas em fontes distintas (Kurman, 2018).

Embora a dispersão de informações sobre jogos digitais na internet represente um desafio de pesquisa, iniciativas como o *MobyGames*¹ - que utiliza o método de colaboração de *crowdsourcing* - demonstram a viabilidade de centralizar e organizar tais dados. O *MobyGames*, por meio de, tem como objetivo compilar informações detalhadas sobre video games, contribuindo significativamente para a construção de um repositório abrangente e acessível sobre a temática.

Contudo, uma oportunidade que se apresenta é desenvolver uma solução que possibilite integrar informações e dados de múltiplas fontes e *websites* visando a criação de um repositório de interface unificado. Neste sentido, a presente pesquisa busca

¹ Disponível em: < <https://www.metacritic.com/> > Acesso em: 13 dez, 2024.

explorar as possibilidades oferecidas pelas tecnologias mais recentes. (Mcquillan, 2022).

1.1. Problema

Desde o advento dos jogos digitais em nossa sociedade, milhares de jogos foram desenvolvidos e publicados. As informações sobre jogos eletrônicos, abrangendo gêneros, classificação indicativa, tipologia, plataforma e desenvolvedora, entre outras, apresentam uma estrutura complexa e variável nas distintas bases de dados consultadas. A tarefa de unificar e detalhar esses dados de forma consistente representa um obstáculo significativo. (De Groat et al., 2018).

Em diversos *websites* pode-se encontrar bases de jogos, cada uma contendo características e focos distintos nos materiais disponíveis ao público. Entretanto, a tarefa de agregar informações dispersas em múltiplos *websites* em uma única interface coesa exige um processo de padronização e integração complexos (Mcquillan, 2022).

A heterogeneidade de informações, formatos e estruturas de dados nas diferentes fontes representa um desafio significativo para a integração semântica das informações. A identificação e o agrupamento de dados com significados semelhantes exigem um profundo conhecimento das relações semânticas entre os termos utilizados nas diversas fontes.

Em resumo, o problema que este trabalho aborda é: como coletar e armazenar dados em bases heterogêneas de jogos, disponibilizando-os de forma centralizada em uma única plataforma de consulta baseadas em dados conectados.

1.2. Objetivo

Este trabalho tem como **objetivo principal** a implementação de mecanismos de buscas para obtenção de informações contidas em três *websites* populares de jogos (*Metacritic*², *IGN*³ e *GOG*⁴) que armazenam diversas informações relevantes para os jogos dentro de seus catálogos. Esses mecanismos citados anteriormente também visam

² Disponível em: < <https://www.metacritic.com/> > Acesso em: 13 dez, 2024.

³ Disponível em: < <https://www.ign.com/> > Acesso em: 13 dez, 2024.

⁴ Disponível em: < <https://www.gog.com/> > Acesso em: 13 dez, 2024.

permitir que dentro dos dados gerados possa haver atualização das informações e novos jogos serem cadastrados. Com esses dados em mãos, é possível realizar uma consulta em tempo real em cada uma das fontes de dados e apresentar de uma forma objetiva todos os dados relacionados a um determinado jogo.

A partir disso, os dados devem ser organizados a partir de uma taxonomia de conceitos de jogos, desenvolvida para dar suporte ao sistema ao alinhar dados semânticos similares existentes em fontes de dados heterogêneas de jogos de entretenimento de modo a possibilitar uma consulta de informações usando dados conectados.

Com esses dados em mãos, é o sistema que possibilitará a realização de consultas em tempo real em cada uma das fontes de dados e apresentar de uma forma objetiva todos os dados relacionados a um determinado jogo, como se estivessem em uma única base de informação.

Ao acessar o sistema, o usuário terá acesso ao catálogo de milhares de jogos, podendo realizar buscar com filtros específicos, navegar entre as páginas e visualizar todos os detalhes de cada jogo de acordo com seu interesse.

Portanto, como **objetivos secundários**, espera-se:

1. Observar os dados das três fontes de informação de jogos e definir uma taxonomia de conceitos de jogos, alinhando os dados semelhantes de tais bases;
2. Desenvolver mecanismos de extração de dados das três bases de jogos estabelecidas (*crawlers*);
3. Consolidar os dados heterogêneos e conectados em uma única base de informação tendo como suporte a taxonomia de conceitos de jogos;
4. Implementar um sistema de consulta de informação de jogos, considerando a taxonomia de conceitos e a base de dados conectados.

1.3. Justificativa

Ao utilizar a ferramenta de consulta desenvolvida, torna-se possível identificar possíveis tendências e padrões nos jogos, ranquear avaliações, obter diversidade de dados distintos relacionados a um ou mais jogos, acesso a informações atualizadas, análises e

avaliações, permitindo que os jogadores e pesquisadores rastreiem a evolução dos jogos ao longo do tempo. Isso viabiliza uma análise que abrange diversas fontes de dados, tornando o espectro analisado mais vasto (Ewald et al., 2022).

Ao se encontrar uma solução que consiga condensar todas essas bases de conhecimento e conseguir êxito no resultado obtido, possibilita que a mesma arquitetura seja expandida para outras áreas de conhecimento, como filmes, séries, veículos, etc. (Saha, 2018).

A partir de uma análise do cenário relatado, abre-se um leque de oportunidades de análises para uma melhor compreensão na evolução da indústria de jogos digitais.

Como inferências que podem surgir a partir dessa análise pode-se citar:

- Engajamento do jogador;
- Aspectos técnicos;
- Sucesso de crítica.

Os pontos aqui citados são apenas possibilidades ao se interligar diferentes campos relacionados aos jogos, mas a quantidade de análises que podem ser feitas é potencializada exponencialmente à medida que cada vez mais campos que representam alguma esfera de conhecimento relacionado a jogos digitais são adicionados.

1.4. Metodologia

Figura 1 - Etapas da Metodologia



Fonte: Do autor (2024)

Este projeto teve seu desenvolvimento realizado em algumas etapas (Figura 1). Primeiramente foi realizada uma pesquisa e consulta das melhores fontes de pesquisa relacionadas a informações de jogos, ou seja, quais *websites* seriam utilizados para a obtenção das listas de jogos e das informações contidas em cada fonte. Durante esta etapa foram escolhidos *websites* relevantes e cujas bases sejam abrangentes em relação ao tamanho de seus catálogos. O número de *websites* escolhidos foi de três no total.

Uma vez compreendido o método, a próxima etapa consistiu em detalhar as funcionalidades necessárias que o software de apoio precisaria fornecer para se obter os dados da forma mais assertiva e como disponibilizar os dados resultantes de forma a fornecer ao usuário uma interface de fácil entendimento.

Por fim, através de pesquisa e estudo, foram definidas as tecnologias que seriam utilizadas. Parte desse processo incluiu a escolha da linguagem mais apropriada, qual software seria utilizado para desenvolver o código, de que forma tornar o resultado de modo que o usuário pudesse consumir os dados obtidos da maneira mais adequada e em qual formato de arquivo os dados seriam armazenados.

Portanto, neste trabalho, o foco será a análise da utilidade deste sistema de apoio no processo de obtenção de dados de jogos para consumo de informações, observando o que pode ser extraído e análises que podem ser feitas a partir das saídas.

1.5. Organização do Texto

A presente pesquisa está estruturada em capítulos e, além desta introdução, compreende os seguintes tópicos:

- **Capítulo 2:** Neste capítulo, apresentamos a base teórica que sustenta esta pesquisa, explorando conceitos fundamentais como a definição de jogos digitais e vídeo games, dados conectados, taxonomia e *web crawlers*.
- **Capítulo 3:** Este capítulo apresenta a descrição da arquitetura do sistema, taxonomia de Conceitos de Jogos, os requisitos, os diagramas de classe e de casos de uso e os casos de uso da ferramenta.
- **Capítulo 4:** Neste capítulo, é apresentada uma aplicação prática do sistema.
- **Capítulo 5:** As conclusões e considerações finais são apresentadas neste capítulo.

2. FUNDAMENTAÇÃO TEÓRICA

2.1. Jogos Digitais e Video Games

Dentro da academia, entende-se que há uma certa complexidade em se especificar o que é um jogo digital. Dentro das definições dos principais pesquisadores, existem termos que não possuem uma definição aceita de forma universal.

Entretanto, Miranda (2017) analisam as definições dos principais pesquisadores da área e propõem uma definição baseada nelas.

Desta forma, (...) o termo “jogo digital” é definido (...) como: atividade voluntária, com ou sem interesse material, com propósitos sérios ou não, composta por regras bem definidas e objetivos claros, capazes de envolver os(as) jogadores(as) na resolução de conflitos e que possui resultados variáveis e quantificáveis. Esta atividade deve ser gerenciada por software e executada em hardware. (Miranda, 2017)

Pode-se definir o termo video game como um dispositivo de jogo digital que envolve a interação com uma interface de usuário ou dispositivo de entrada (controle, *joystick*, *console*, teclado ou dispositivo de detecção de movimento) para produzir um *feedback* audiovisual em uma tela de exibição (Sputnik Brasil, 2023).

Porém, neste trabalho seguiremos a definição proposta por Xexéo et al. (Xexéo et al., 2017, p. 10):

Jogos são atividades sociais e culturais voluntárias, significativas, fortemente absorventes, não-produtivas, que se utilizam de um mundo abstrato, com efeitos negociados no mundo real, e cujo desenvolvimento e resultado final é incerto, onde um ou mais jogadores, ou equipes de jogadores, modificam interativamente e de forma quantificável o estado de um sistema artificial, possivelmente em busca de objetivos conflitantes, por meio de decisões e

ações, algumas com a capacidade de atrapalhar o adversário, sendo todo o processo regulado, orientado e limitado, por regras aceitas, e obtendo, com isso, uma recompensa psicológica, normalmente na forma de diversão, entretenimento, ou sensação de vitória sobre um adversário ou desafio. (Xexéo et al., 2017, p. 10)

Em outras palavras, neste estudo a concepção de jogos adotada é que jogos são práticas realizadas de maneira espontânea pelos jogadores, apresentando elementos sociais e culturais, envolvendo desafios físicos ou mentais que culminam em uma experiência integradora, provocando igualmente a possibilidade de os participantes não perceberem a passagem do tempo enquanto estão imersos na atividade.

2.2. Dados Conectados

Atualmente, vastas quantidades de informações são disponibilizadas na internet em uma variedade de formatos (*pdf*, *xls*, *csv*, *jpg*, entre outros). No entanto, apesar da abundância de dados, em sua maioria, essas informações são interpretadas por seres humanos. Nesse contexto, os Dados Interconectados possibilitam que dados provenientes de diversas fontes sejam processados por máquinas, proporcionando um contexto mais rico de compreensão em relação a um conjunto específico de dados (Isotani e Bittencourt, 2015).

Além disso, a divulgação de dados interconectados com características abertas facilita o processamento automático de grandes volumes de dados e a realização de inferências sobre eles. Isso se deve ao fato de que os dados abertos, por sua natureza, possuem atributos que incentivam seu amplo uso e reutilização para diversas finalidades (Isotani e Bittencourt, 2015).

De acordo com Heath e Bizer (2022), para uma melhor compressão do conceito e valor dos dados conectados, é importante considerar os mecanismos que são utilizados atualmente para compartilhamento e reutilização de dados na web. Além disso, eles frisam a importância de analisar a maturidade em que os dados estão estruturados. Quanto mais regular e bem definida for a estrutura de dados, com mais facilidade as

peças podem criar soluções para que ela seja processada de forma confiável para reutilização.

2.3. Taxonomia de Domínio

Conforme abordado por Vital e Café (2011) tem-se que em um mundo onde a produção de informações cresce cada vez mais, os sistemas de representação e organização de conhecimento são considerados processos de suma importância. Esses sistemas visam representar, recuperar e interligar informações tendo em vista a necessidade específica de cada ambiente.

As taxonomias tiveram sua origem na área da Biologia, porém seu conceito se expandiu para outros meios. Tratando especificamente de sua utilização em ambientes digitais, elas estão atreladas às possíveis maneiras em que a organização de informação pode ser automatizada (Vital e Café, 2011).

No âmbito da Gestão do Conhecimento, as taxonomias são definidas como "elementos estruturantes, estratégicos e centrais para negócios baseados em informação e conhecimento [...] para classificar e facilitar o acesso à informação" (Terra et al., 2005, p. 1).

A taxonomia classifica a organização de dados desde os níveis mais amplos até os mais detalhados, empregando a relação hierárquica ou a relação de categoria-subcategoria entre os termos. Segundo Adams (2000, p. 2) as "[...] taxonomias organizam e criam significados para os relacionamentos entre coisas ou ideias".

2.4. Web Crawlers

Em uma tradução livre *web crawler* significa rastreador web, que é um programa de computador cuja função é navegar automaticamente pela internet, visitando páginas da *web* e coletando informações (Cloudflare, s.d.). Essa abordagem é significativa, uma vez que a análise contribui para identificar padrões e embasar decisões com maior probabilidade de sucesso (Simplilearn, 2023).

Entende-se que, em muitas situações, realizar essa extração de forma manual é viável. No entanto, ao lidar com um grande volume de dados dispersos em diversos *websites* ou em várias seções de um mesmo site, a extração manual torna-se impraticável. Nessas circunstâncias, é imperativo automatizar o processo de coleta. Para isso, os *bots* são utilizados, os quais são capazes de recuperar imagens, arquivos, tabelas e percorrer todo o conteúdo do site. Contudo, é essencial incorporar expressões regulares no *script* para que o agente automatizado consiga identificar, no código fonte, o padrão desejado, permitindo assim a extração dos dados relevantes do site (Crawlbase, 2022).

O conceito de *web crawler* está intrinsicamente ligado à *web scraping* em projetos que envolvem extração de dados. O *webscraping* é um termo utilizado para definir o ato da coleta automática de dados a partir de um *website* utilizando uma ferramenta de *software*. Note-se que as informações obtidas podem ser armazenadas de diversos modos (banco de dados, arquivos, etc.) (Souza, 2022).

A diferença entre os dois termos consiste em: no *web scraping* ocorre a extração de dados de um ou mais *websites*. Enquanto o *web crawling* trata de encontrar ou descobrir *URLs* ou links na *web*. Para se alcançar o objetivo do projeto, os dois conceitos serão utilizados (Zyte, s.d.).

3. GAMES FINDER APPLICATION

A *Games Finder Application* é uma aplicação desenvolvida em *Python* que foi criada com o intuito de coletar dados de jogos de múltiplas fontes e fornecer ao usuário essas informações ao usuário de forma centralizada.

3.1. Taxonomia de Conceitos de Jogos

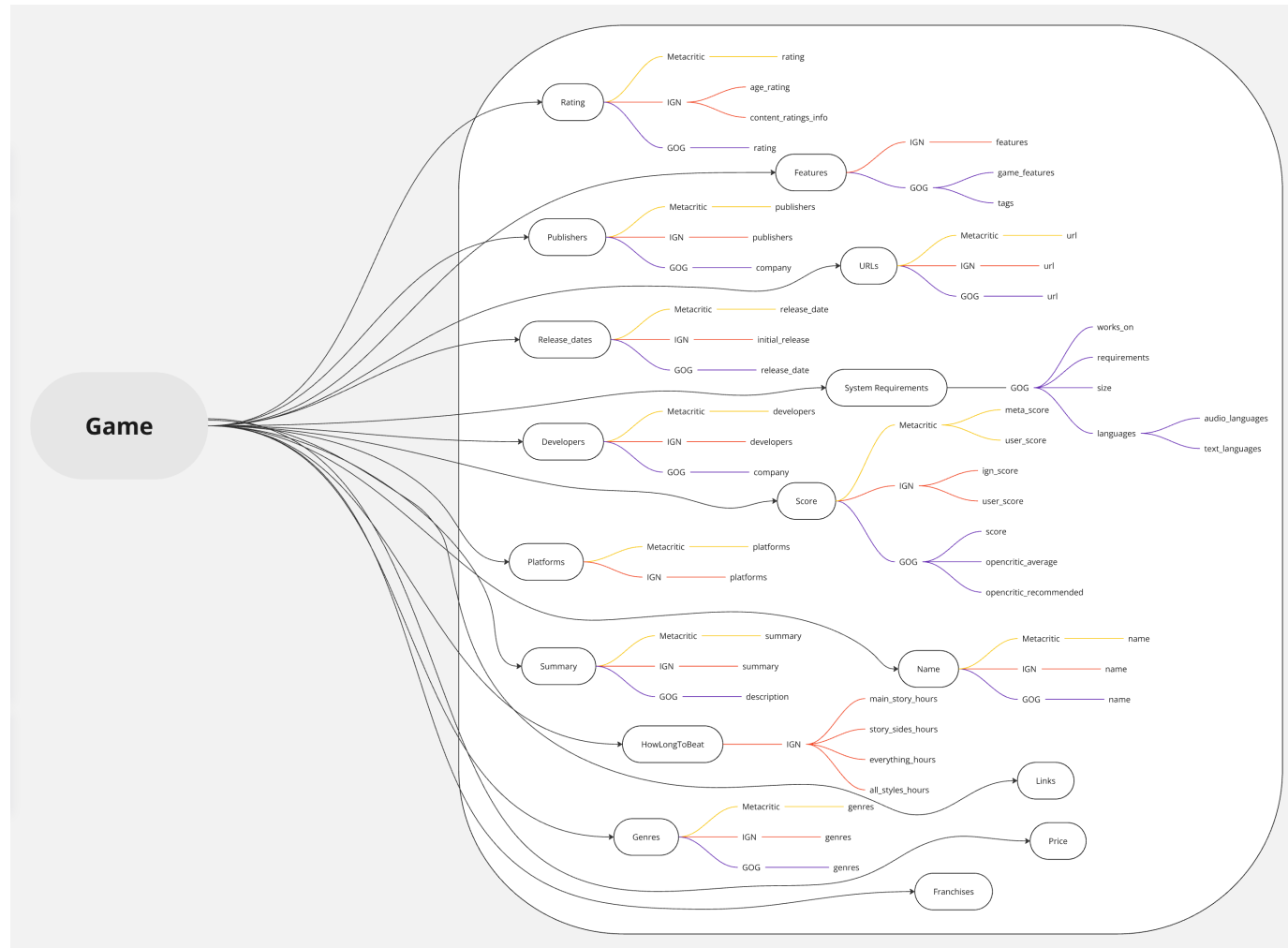
No contexto dos jogos, além da taxonomia ser um sistema de classificação, ela também busca organizar, categorizar e hierarquizar os diversos elementos que compõem um jogo, desde as mecânicas mais básicas até os elementos narrativos mais complexos.

A análise dos termos utilizados nos três sites revelou a presença de um vocabulário comum relacionado a alguns elementos como gênero, plataforma e mecânicas de jogo. Por exemplo, todos os sites em questão empregam termos como "ação", "aventura", "RPG" e "FPS" para classificar os jogos. Além disso, termos como "PC", "Xbox" e "PlayStation" são frequentes, evidenciando a relevância das plataformas de console. Em relação às mecânicas de jogo, termos como "multiplayer", "open world" e "loot boxes" são amplamente utilizados, demonstrando a valorização dessas características pelos jogadores.

A análise do conteúdo das três bases, utilizando um conjunto de categorias pré-definidas, permitiu a construção de um mapa conceitual que demonstra a interrelação entre os conceitos e a existência de temas em comum. Assim, foi realizado o casamento de campos de fontes distintas e que possuem o mesmo valor semântico. A categorização aqui proposta resulta de uma interpretação individual e reflexiva dos dados, sendo influenciada pela experiência e perspectiva do autor. A Figura 2 apresenta o resultado

desta análise, que consiste em uma representação gráfica de todas as informações que um jogo pode conter, indicando a fonte de cada um dos campos.

Figura 2 - Mind map de dados conectados



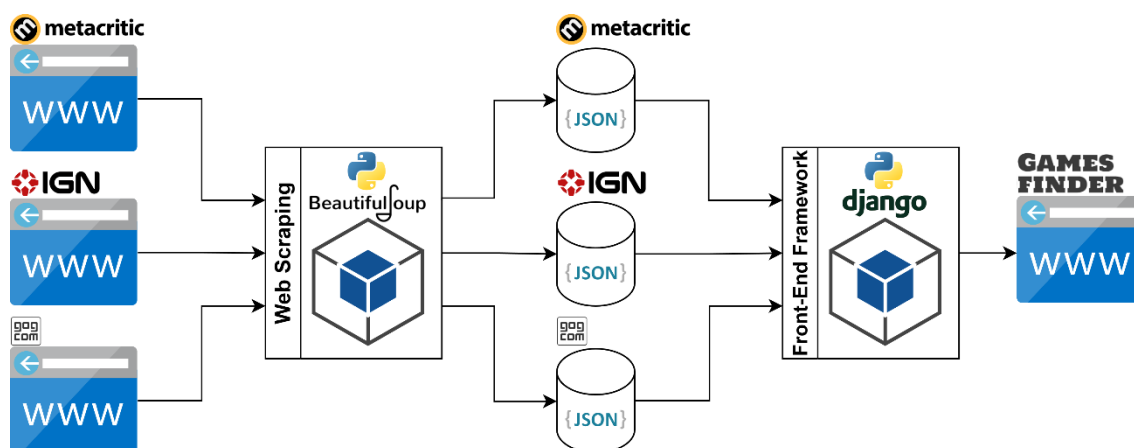
Fonte: Do Autor (2024)

Um aspecto particularmente interessante a ser observado é a utilização de diferentes nomes que definem os campos de dados para representar uma mesma ideia ou conceito, porém em grande parte dos casos há uma padronização da nomenclatura utilizada nos campos para expressar um aspecto semelhante. Essa padronização facilita a comparação de dados entre diferentes fontes de jogos. A utilização de um vocabulário comum na indústria de jogos impacta positivamente na experiência do usuário, pois reduz o esforço adicional para a adaptação à terminologia de conteúdo relacionado à descrição de dados de jogos.

3.2. Arquitetura do Sistema

Visando facilitar o intercâmbio de informações sobre jogos provenientes de múltiplas plataformas digitais, optou-se por elaborar um método sistemático de extração de dados de *websites* especializados. A fim de viabilizar essa integração, foi concebida uma arquitetura que permita a coleta e o tratamento dos dados desejados (Figura 3).

Figura 3 - Diagrama de Arquitetura do Projeto



Fonte: Do autor (2024)

O sistema encontra-se estruturado em dois módulos distintos. O primeiro módulo é dedicado à coleta de dados, acessando individualmente cada fonte de dados configurada. Os dados obtidos são armazenados em arquivos específicos, resultando em um conjunto de três arquivos. O segundo módulo tem como função a consolidação dos dados, realizando a leitura dos arquivos gerados anteriormente. Após a integração das informações, os dados dos jogos são apresentados ao usuário para análise e consulta.

Nesta seção, são apresentados os requisitos do sistema e as tecnologias utilizadas, compreendendo os requisitos funcionais, não funcionais, regras de negócio e diagramas.

3.2.1. Requisitos e Regras de Negócio

A seguir encontra-se uma lista de requisitos funcionais do *Games Finder Application* (Tabela 1).

Tabela 1 - Requisitos Funcionais do Sistema

Requisitos Funcionais	
Requisito	Descrição
RF1	O sistema deve permitir que o usuário visualize todos os jogos disponíveis no catálogo
RF2	O sistema deve permitir que o usuário alterne a página atual
RF3	O sistema deve permitir que o usuário selecione um filtro de busca
RF4	O sistema deve permitir que o usuário realize a busca
RF5	O sistema deve permitir que o usuário acesse a página de dados conectados de um jogo de sua escolha
RF6	O sistema deve permitir que o usuário visualize os dados conectados de um determinado jogo em uma página única

Fonte: Do autor (2024)

A seguir encontra-se uma lista de requisitos não funcionais do *Games Finder Application* (Tabela 2).

Tabela 2 - Requisitos Não Funcionais do Sistema

Requisitos Não Funcionais	
Requisito	Descrição
RNF1	O sistema deve ser executado em uma página <i>web</i>
RNF2	O sistema deve ser desenvolvido em linguagem <i>Python</i>
RNF3	O <i>frontend</i> deve ser desenvolvido utilizando o framework <i>Django</i>

Fonte: Do autor (2024)

A seguir está a lista de regras de negócio do *Games Finder Application* (Tabela 3).

Tabela 3 - Regras de Negócio

Regras de Negócio	
Requisito	Descrição
RN1	O usuário só poderá selecionar um filtro para cada busca

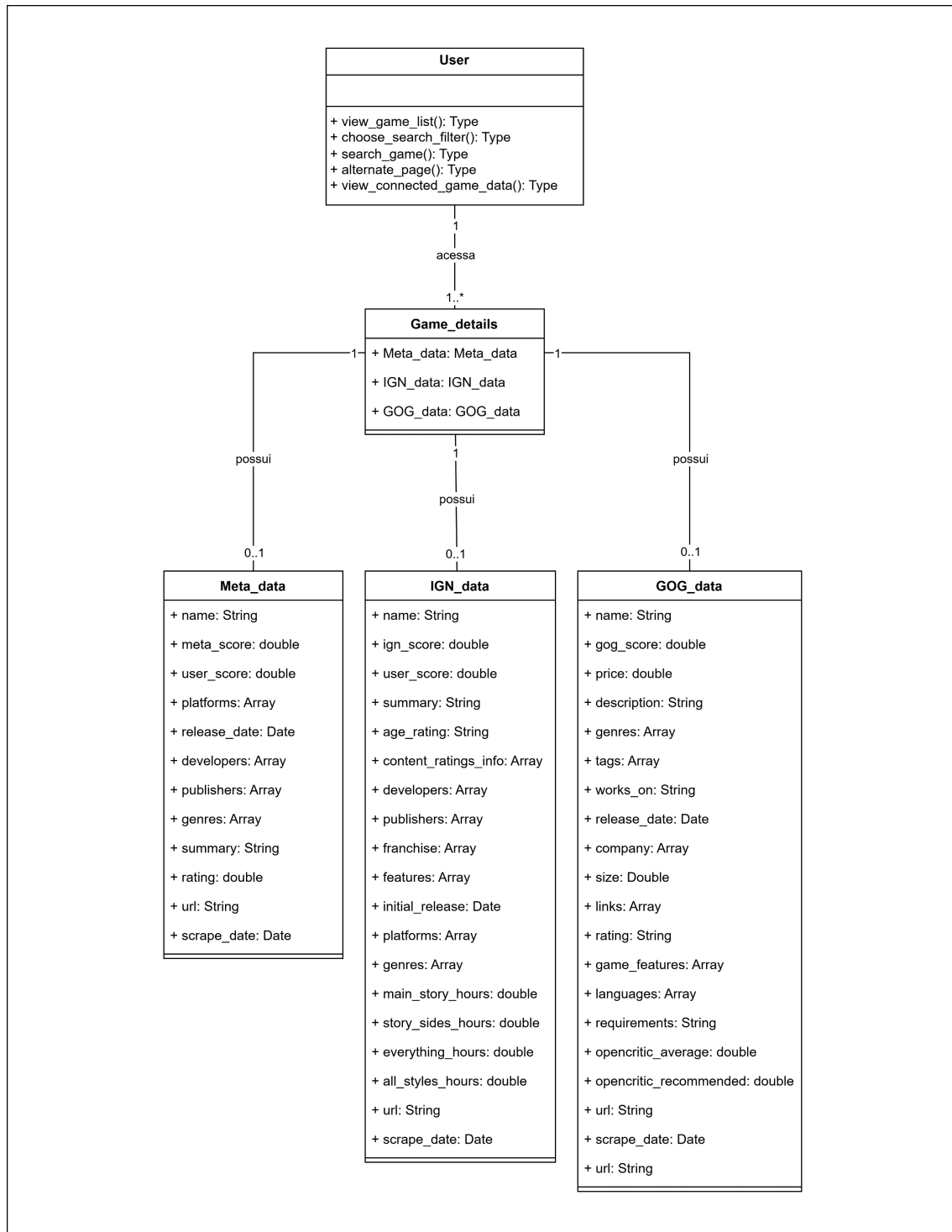
Fonte: Do autor (2024)

3.2.2. Diagramas *UML*

Segue o diagrama dos casos de uso do sistema *Games Finder Application* (Figura 4).

Mais detalhes sobre cada caso de uso podem ser consultados no Apêndice 1.

Figura 5 - Diagrama de Classes do *Games Finder Application*



Fonte: Do Autor (2024)

3.2.3. Tecnologias utilizadas

A construção do sistema foi desenvolvida utilizando a linguagem de programação *Python* (versão 3.11), devido a sua popularidade e a sua curva de aprendizado menor comparando-se com outras linguagens. Juntamente à linguagem optou-se pela utilização da biblioteca *BeautifulSoup*⁵ (versão 4.12.3), que é uma ferramenta específica para o “*scraping*” (entende-se como “*scraping*” o ato de extrair dados de um determinado documento).

Em paralelo foi utilizada a biblioteca de software *Pandas*⁶ (versão 2.2.2) do *Python*, que tem como finalidade facilitar a manipulação e análise de dados.

Para as aplicações de extração de dados, o tipo de arquivo de saída escolhido foi o *JSON*. A escolha desse formato se deve a sua popularidade, e também pelo mesmo ser leve e de fácil leitura.

A fim de disponibilizar o resultado com os arquivos gerados de uma forma clara e objetiva foi escolhida a utilização de um *framework* voltado para a *web* cujo o nome é *Django*⁷ (versão 5.0.6), também desenvolvido utilizando a linguagem *Python*.

No que tange o desenvolvimento, foi utilizado o *software PyCharm*⁸, que fornece um ambiente de desenvolvimento integrado usado especificamente para programar em *Python*. O programa permite fazer o *debug* de processos, possui integração com o *Git* além de possuir outras facilidades na linguagem utilizada.

Ainda que o código tenha sido desenvolvido de forma individual, foi utilizado o sistema de controle de versionamento *Git*⁹ para um melhor controle de versão, facilidade no desenvolvimento utilizando máquinas diferentes e *backup* de todo o código à medida em que ele foi desenvolvido (o *link* para o projeto encontra-se disponível no rodapé da página atual).

⁵ Disponível em: < <https://pypi.org/project/beautifulsoup4/> > Acesso em: 13 dez, 2024.

⁶ Disponível em: < <https://pandas.pydata.org/> > Acesso em: 13 dez, 2024.

⁷ Disponível em: < <https://www.djangoproject.com/> > Acesso em: 13 dez, 2024.

⁸ Disponível em: < <https://www.jetbrains.com/pycharm/> > Acesso em: 13 dez, 2024.

⁹ Disponível em: < <https://github.com/teusup/games-crawler.git/> > Acesso em: 13 dez, 2024.

O site *draw.io*¹⁰ foi utilizado para modelar a maioria dos diagramas utilizados neste projeto. O software *online miro*¹¹ foi utilizado para a construção do *mind map* (Figura 2).

As configurações da máquina onde o projeto foi desenvolvido e executado foram as seguintes:

- Memória *RAM*: 32GB (2x16GB), DDR4, 3200mhz;
- Placa de Vídeo: *TUF GAMING GeForce RTX™ 3070*;
- Processador: *AMD Ryzen™ 5 5600*.

3.3. *Games Crawler*

Neste módulo é realizado de forma manual a execução dos códigos de *scraping* para cada uma das fontes escolhidas. As diferentes fontes utilizadas nesse projeto necessitaram de um código *scraping* personalizado, visto que cada fonte utiliza tecnologias próprias na construção de seus respectivos sites.

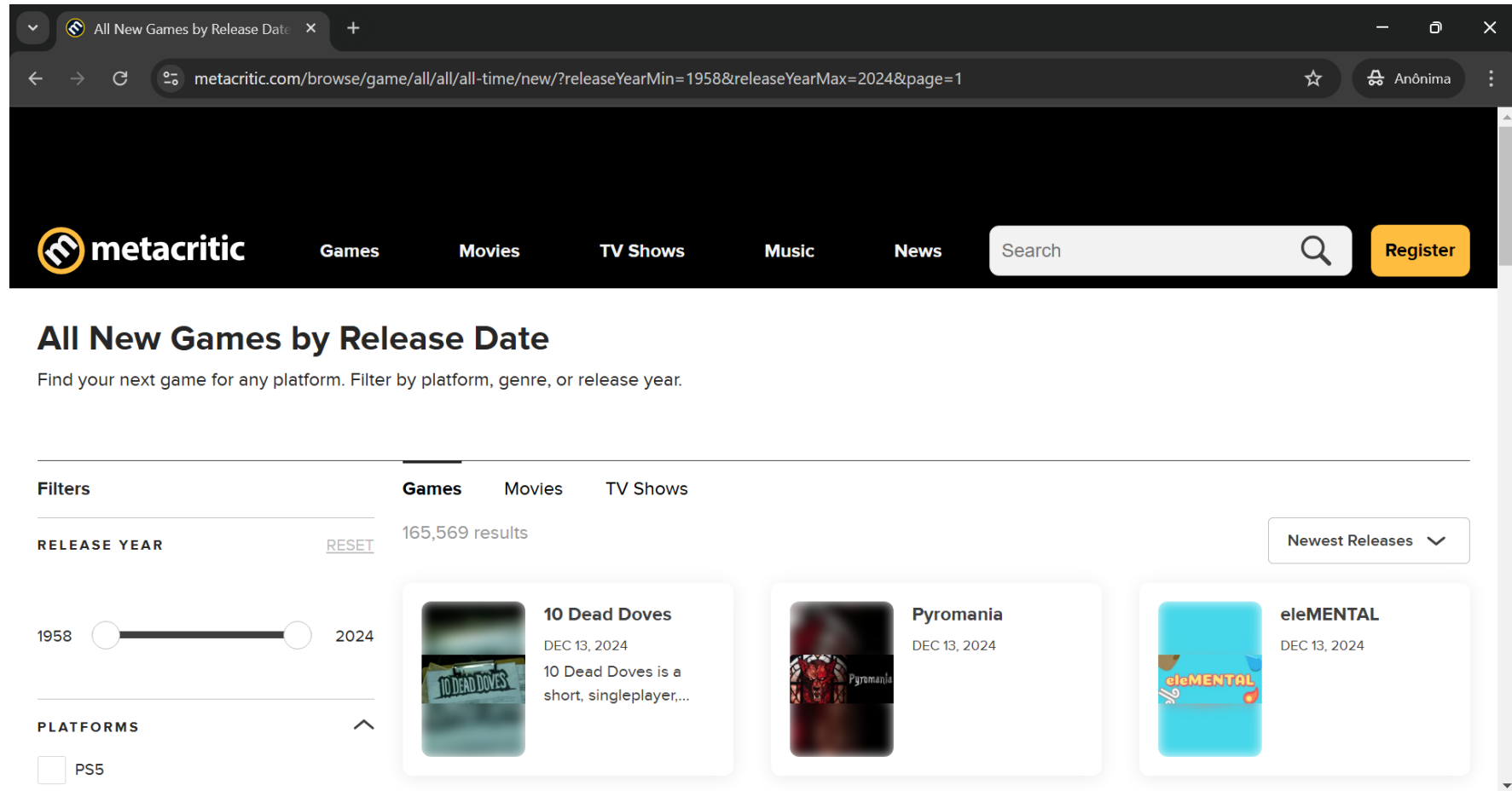
Este módulo pode ser executado por meio de uma *IDE* ou via console em um sistema operacional que possua o ambiente *Python* instalado.

3.3.1. **Metacritic Crawler**

O *website Metacritic* disponibiliza uma página que contém o catálogo de todos os jogos cadastrados em seu site, com opções de filtros e paginação (Figura 6). A partir do link desta página, o código *crawler* do *Metacritic* realiza a leitura, página a página, e armazena as *URLs* encontradas em uma lista (Figura 7).

¹⁰ É um *software* de diagramação online que permite criar fluxogramas, diagramas, *UML*. Disponível em: < <https://www.drawio.com/> > Acesso em: 13 dez, 2024.

¹¹ É uma plataforma online de colaboração visual que permite a gestão de projetos e o desenvolvimento de produtos. Disponível em: < <https://miro.com/> > Acesso em: 13 dez, 2024.

Figura 6 - Página da lista de todos os jogos do site *Metacritic*

Fonte: Página do website *Metacritic*¹²

¹² Disponível em: < <https://www.metacritic.com/browse/game/all/all/all-time/new/?releaseYearMin=1958&releaseYearMax=2024&page=1> > Acesso em: 13 dez, 2024.

Figura 7 – Trecho de código responsável em armazenar todas as *URLs* do *Metacritic*

```
get_all_urls_from_metacritic_pages.py x
53 for i in tqdm.tqdm(range(1, len(pages)), ncols=100, colour="blue", desc="Coleta de páginas"):
54
55     # Make a get request
56     games = get('https://www.metacritic.com/browse/game/all/all/all-time/new?releaseYearMin=1910&releaseYearMax=2024'
57               + '&page=' + pages[i],
58               headers=headers)
59
60     # Fazendo novas tentativas de request se html vier vazio
61     while len(games.text) == 0:
62         games = get('https://www.metacritic.com/browse/game/all/all/all-time/new?releaseYearMin=1910&releaseYearMax'
63               + '=2024&page=' + pages[i],
64               headers=headers)
65
66     # Show a warning if a non 200 status code is returned
67     if games.status_code != 200:
68         warn('Request: {}; Status code: {}'.format(*args, requests, games.status_code))
69
70     # Break the loop if the requests exceed 10
71     if requests > 10:
72         warn('Number of requests was greater than expected.')
73         break
74
75     # Parse the game response content into the beautiful soup object
76     game_soup = BeautifulSoup(games.text, features='html.parser')
```

Fonte: Do autor (2024)

Assim que a lista com todas as *URLs* é concluída, o código *crawler* do *Metacritic* executa a leitura dos dados, *URL* a *URL*, e armazena os valores encontrados na página dentro de um *dataframe Pandas* (Figura 8).

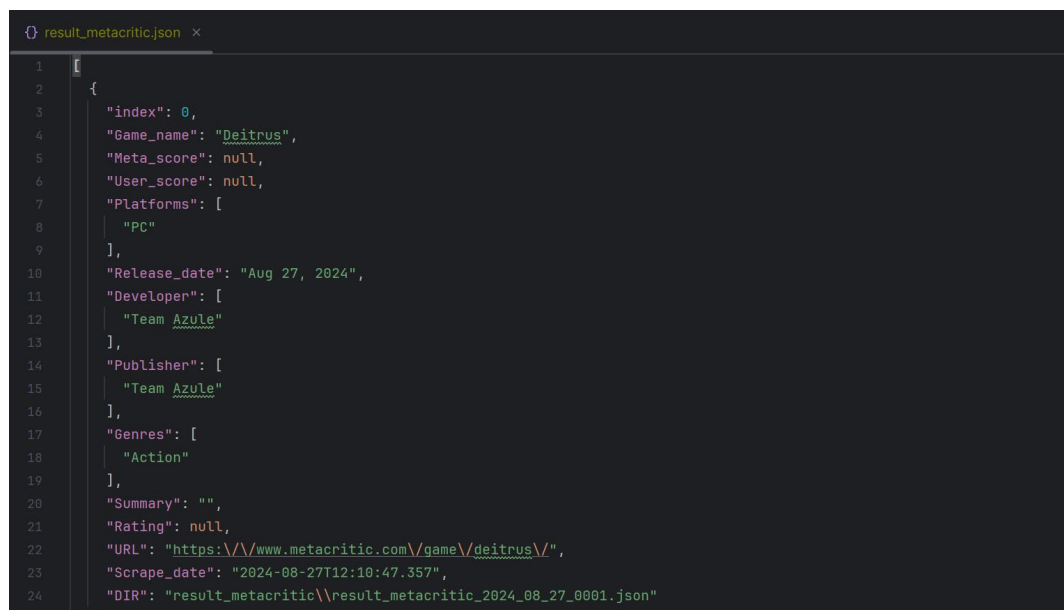
Figura 8 – Trecho de código que realiza a leitura dos campos da página

```
metacritic_crawler.py x
145 for j in tqdm.tqdm(range(len(all_urls_list)), ncols=100, colour="green", desc="Coleta de dados individuais"):
146
147     webpage = get(all_urls_list[j], headers=headers)
148
149     # Fazendo requisição até retornar o html
150     while len(webpage.text) == 0:
151         webpage = get(all_urls_list[j], headers=headers)
152
153     bs = BeautifulSoup(webpage.text, features='html.parser')
154
155     gn = ""
156     try:
157         game_name = bs.find(name='div',
158                           attrs={'class': 'c-productHero_title g-inner-spacing-bottom-medium g-outer-spacing-top-medium'})
159         gn = game_name.text.replace(_old: "\n", _new: "").strip()
160     except Exception as e:
161         gn = np.nan
162     names.append(gn)
163
164     gms = []
165     game_meta_score = bs.find_all(class_='c-productScoreInfo_scoreNumber u-float-right')
166     try:
167         gms = define_meta_or_user(game_meta_score, score_type=0) if game_meta_score is not None else np.nan
168     except Exception as e:
```

Fonte: Do autor (2024)

Após os dados de todas as *URLs* serem coletados, o sistema converte o *dataframe Pandas* e armazena todas as informações em um arquivo *JSON* (Figura 9).

Figura 9 - Trecho do arquivo *JSON* de saída do *Metacritic Crawler*



```
1 [
2   {
3     "index": 0,
4     "Game_name": "Deitrus",
5     "Meta_score": null,
6     "User_score": null,
7     "Platforms": [
8       "PC"
9     ],
10    "Release_date": "Aug 27, 2024",
11    "Developer": [
12      "Team Azure"
13    ],
14    "Publisher": [
15      "Team Azure"
16    ],
17    "Genres": [
18      "Action"
19    ],
20    "Summary": "",
21    "Rating": null,
22    "URL": "https://www.metacritic.com/game/deitrus/",
23    "Scrape_date": "2024-08-27T12:10:47.357",
24    "DIR": "result_metacritic\\result_metacritic_2024_08_27_0001.json"
25  ]
26 ]
```

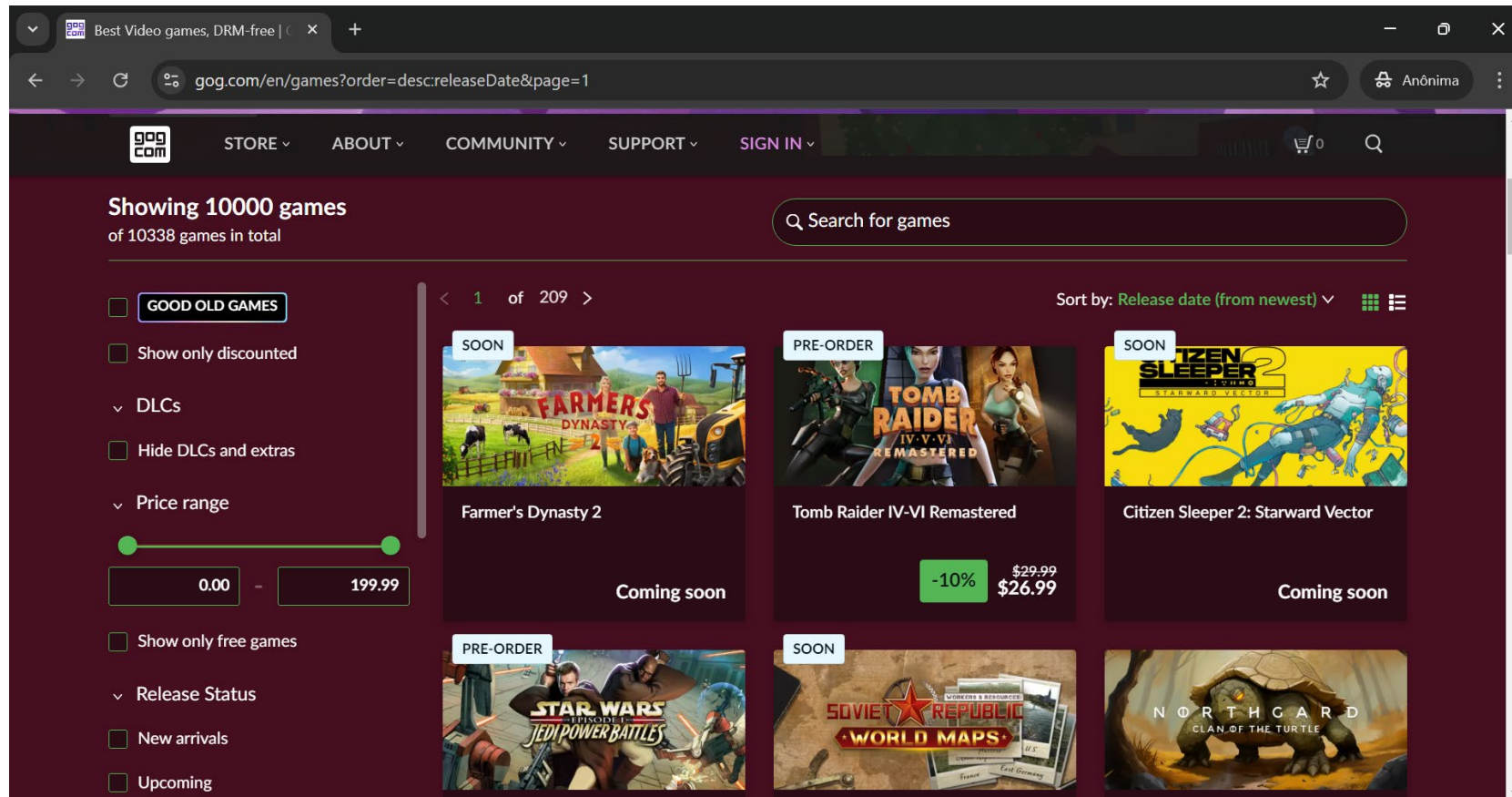
Fonte: Do autor (2024)

3.3.2. GOG Crawler

Assim como o *website* do *Metacritic*, o *website* *GOG* também disponibiliza uma página contendo todos os jogos cadastrados em seu catálogo (Figura 10) . O *crawler* do *GOG* segue um fluxo semelhante ao *crawler* do *Metacritic*, conforme demonstrado nas Figuras 11 e 12.

O formato do arquivo gerado de saída também é um *JSON* (Figura 13).

Figura 10 - Página da lista de todos os jogos do site GOG

Fonte: Página do website GOG¹³

¹³ Disponível em: < <https://www.gog.com/en/games?order=desc:releaseDate&page=1> > Acesso em: 13 dez, 2024.

Figura 11 - Trecho de código responsável em armazenar todas as URLs do GOG

```
get_all_urls_from_gog_pages.py x
45 for i in tqdm.tqdm(range(len(pages)), ncols=100, colour="blue", desc="Coleta de páginas"):
46     # for i in tqdm.tqdm(range(1, get_total_pages()), ncols=100, colour="blue", desc="Coleta de páginas"):
47
48     # Make a get request
49     games = get('https://www.gog.com/en/games?order=asc:storeReleaseDate&page=' + pages[i],
50               headers=headers)
51
52     # Fazendo novas tentativas de request se html vier vazio
53     while len(games.text) == 0:
54         games = get('https://www.gog.com/en/games?order=asc:storeReleaseDate&page=' + pages[i],
55                   headers=headers)
56
57     # Show a warning if a non 200 status code is returned
58     if games.status_code != 200:
59         warn('Request: {}; Status code: {}'.format(*args, requests, games.status_code))
60
61     # Break the loop if the requests exceed 10
62     if requests > 10:
63         warn('Number of requests was greater than expected.')
64         break
65
66     # Parse the game response content into the beautiful soup object
67     game_soup = BeautifulSoup(games.text, features='html.parser')
68
```

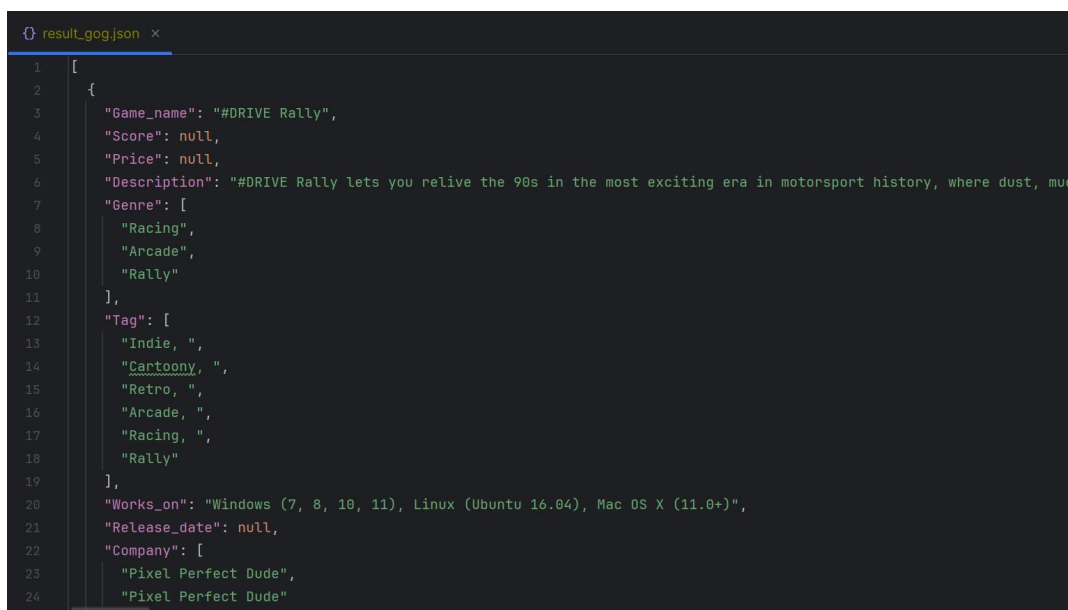
Fonte: Do autor (2024)

Figura 12 - Trecho de código que realiza a leitura dos campos da página

```
gog_crawler.py x
48 for j in tqdm.tqdm(range(len(all_urls_list)), ncols=100, colour="green", desc="Coleta de dados individuais"):
49
50     webpage = get(all_urls_list[j], headers=headers)
51
52     # Fazendo requisição até retornar o html
53     while len(webpage.text) == 0:
54         webpage = get(all_urls_list[j], headers=headers)
55
56     bs = BeautifulSoup(webpage.text, features='html.parser')
57
58     gn = ""
59     try:
60         gn = bs.find('h1').text.replace(_old: "\n", _new: "").strip()
61     except Exception as e:
62         gn = np.nan
63     name_list.append(gn)
64
65     gs = ""
66     try:
67         gs = bs.find(name='div', class_='rating productcard-rating__score').text.split('/')[0]
68     except Exception as e:
69         gs = np.nan
70     score_list.append(gs)
71
```

Fonte: Do autor (2024)

Figura 13 - Trecho do arquivo *JSON* de saída do GOG Crawler



```

1  [
2  {
3    "Game_name": "#DRIVE Rally",
4    "Score": null,
5    "Price": null,
6    "Description": "#DRIVE Rally lets you relive the 90s in the most exciting era in motorsport history, where dust, mud
7    "Genre": [
8      "Racing",
9      "Arcade",
10     "Rally"
11   ],
12   "Tag": [
13     "Indie, ",
14     "Cartoony, ",
15     "Retro, ",
16     "Arcade, ",
17     "Racing, ",
18     "Rally"
19   ],
20   "Works_on": "Windows (7, 8, 10, 11), Linux (Ubuntu 16.04), Mac OS X (11.0+)",
21   "Release_date": null,
22   "Company": [
23     "Pixel Perfect Dude",
24     "Pixel Perfect Dude"

```

Fonte: Do autor (2024)

3.3.3. IGN Crawler

O *website* *IGN* foi a única fonte utilizada que não forneceu em seu site uma relação de todos os jogos cadastrados em sua base em uma única página. Para contornar essa situação, antes de executar o seu código *scraping*, foi identificada a necessidade de realizar a execução do *scraping* do site *Metacritic*. Assim que o código *crawler* do site *Metacritic* finaliza a sua execução, é gerada uma lista de todos os jogos que foram encontrados em sua base.

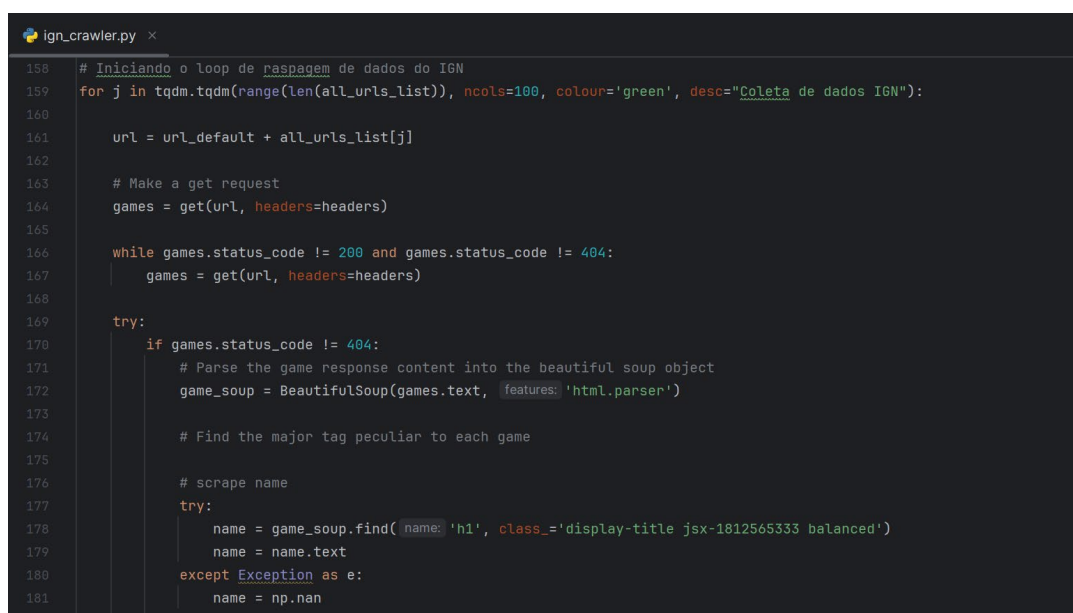
A partir dessa lista é realizada uma leitura de todo o catálogo encontrado no *website* do *Metacritic*, e o sistema verifica se o jogo também existe no *website* *IGN* através de uma chave única, que nada mais é que a parte final que compõe a *URL* do jogo em questão. Por exemplo, a *URL* dos detalhes do jogo *Portal 2* no *Metacritic* é "<https://www.metacritic.com/game/portal-2>".

O sistema então lê essa *URL* armazena tudo o que está a direita da última barra ("/") da *URL* e acrescenta essa informação na *URL* base da *IGN* (<https://www.ign.com/games/>). Após essa operação, como resultado temos a seguinte *URL* construída: "<https://www.ign.com/games/portal-2>". O sistema não prevê o caso de

um mesmo jogo estar descrito de forma diferente nos dois *sites*. Por exemplo, “*unravel-two*” e “*unravel-2*”.

Logo após, o sistema verifica se a *URL* recém construída é válida ou não. Caso a *URL* seja válida, o sistema executa o *scraping* da página e em caso negativo o sistema pula para a *URL* seguinte da lista do *Metacritic* (Figura 14).

Figura 14 - Trecho de código que verifica se a *URL* é válida e realiza a leitura dos campos da página

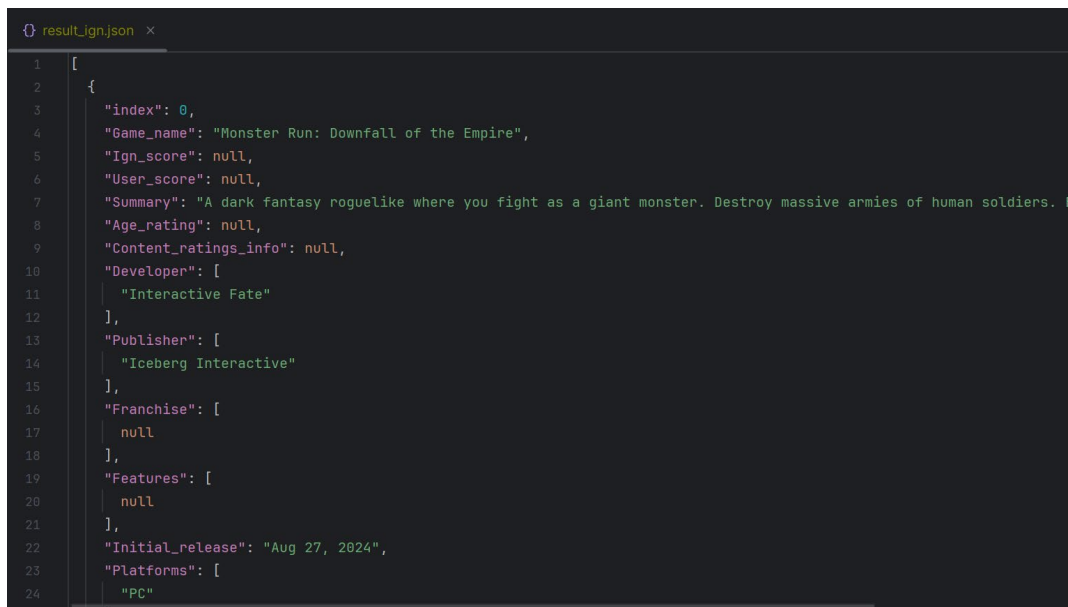


```
158 # Iniciando o loop de raspagem de dados do IGN
159 for j in tqdm.tqdm(range(len(all_urls_list)), ncols=100, colour='green', desc="Coleta de dados IGN"):
160
161     url = url_default + all_urls_list[j]
162
163     # Make a get request
164     games = get(url, headers=headers)
165
166     while games.status_code != 200 and games.status_code != 404:
167         games = get(url, headers=headers)
168
169     try:
170         if games.status_code != 404:
171             # Parse the game response content into the beautiful soup object
172             game_soup = BeautifulSoup(games.text, features='html.parser')
173
174             # Find the major tag peculiar to each game
175
176             # scrape name
177             try:
178                 name = game_soup.find(name='h1', class_='display-title jsx-1812565333 balanced')
179                 name = name.text
180             except Exception as e:
181                 name = np.nan
```

Fonte: Do autor (2024)

Assim como nos demais casos, o formato de arquivo gerado no fim da execução também é um *JSON* (Figura 15).

Figura 15 - Trecho do arquivo *JSON* de saída do *IGN Crawler*



```
1 [
2   {
3     "index": 0,
4     "Game_name": "Monster Run: Downfall of the Empire",
5     "Ign_score": null,
6     "User_score": null,
7     "Summary": "A dark fantasy roguelike where you fight as a giant monster. Destroy massive armies of human soldiers. E",
8     "Age_rating": null,
9     "Content_ratings_info": null,
10    "Developer": [
11      "Interactive Fate"
12    ],
13    "Publisher": [
14      "Iceberg Interactive"
15    ],
16    "Franchise": [
17      null
18    ],
19    "Features": [
20      null
21    ],
22    "Initial_release": "Aug 27, 2024",
23    "Platforms": [
24      "pc"
```

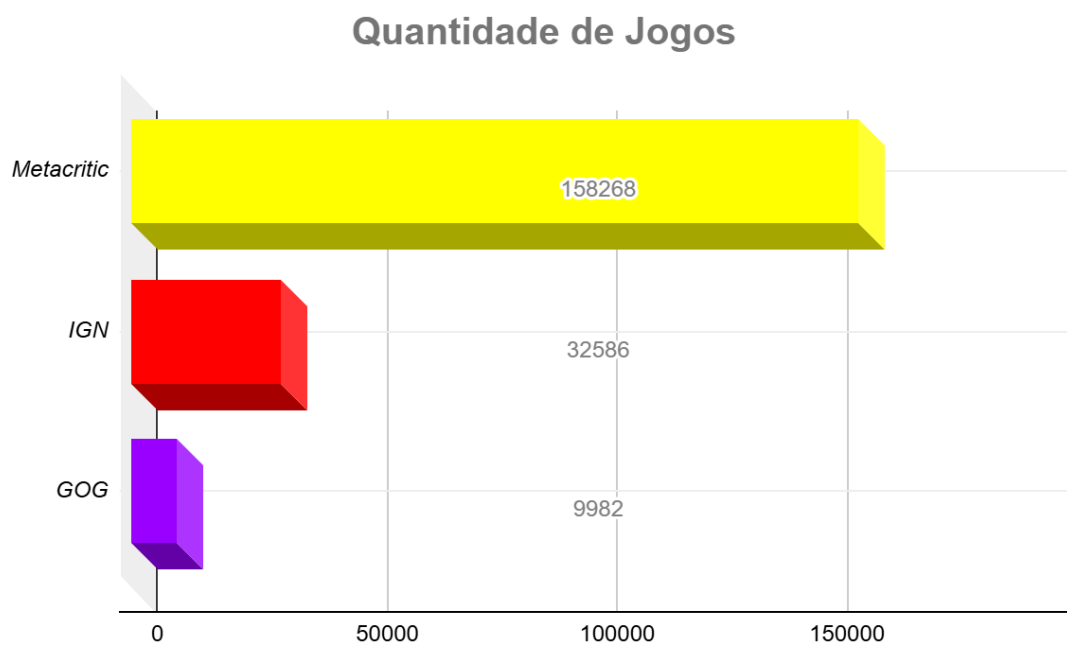
Fonte: Do autor (2024)

3.3.4. Resultados e desempenho

Para a execução dos três *crawlers*, obteve-se a seguinte quantidade de jogos e tempo de execução:

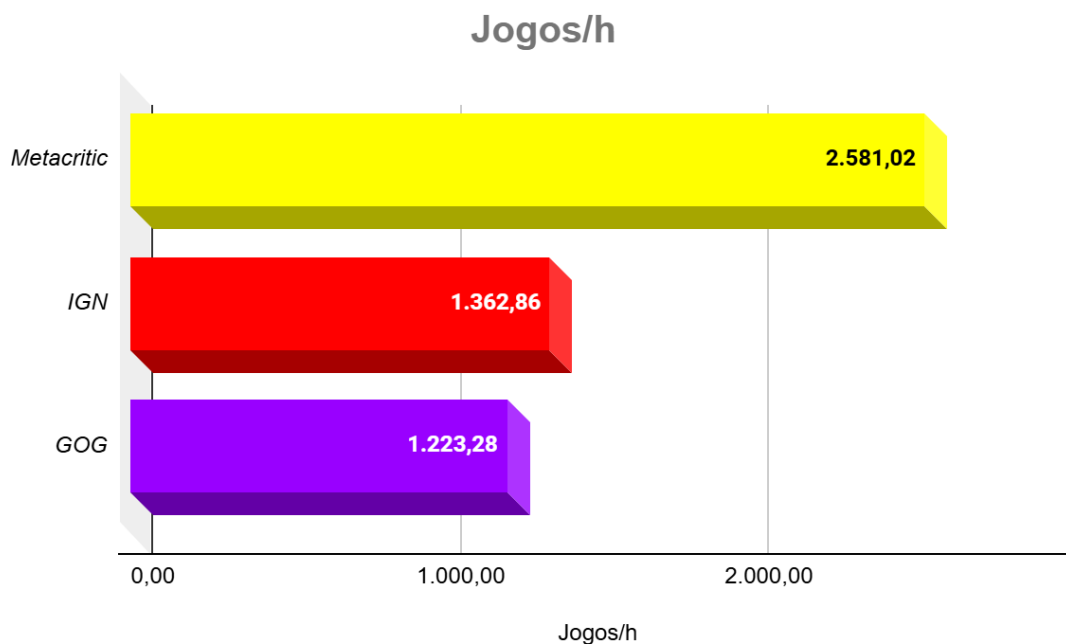
- *Metacritic Crawler*: 158268 jogos em 61,32 horas;
- *IGN Crawler*: 32586 jogos em 23,91 horas;
- *GOG Crawler*: 9982 jogos em 8,16 horas.

A Figura 16 apresenta os dados citados em forma de gráfico, para uma melhor visualização.

Figura 16 - Quantidade de jogos coletados

Fonte: Do autor (2024)

No que se refere ao desempenho das aplicações, o gráfico ilustra a taxa média de processamento de jogos por hora. (Figura 17)

Figura 17 - Relação de média de jogos coletados por hora

Fonte: Do autor (2024)

3.4. *Games Finder*

Este módulo do projeto é onde os dados extraídos são unificados e centralizados. O sistema executa as instruções necessárias para realizar a consulta nas três bases distintas e compila as informações de jogos em uma única instância.

Esta consulta se dá pela leitura dos três arquivos *JSON* gerados a partir da execução do módulo *crawler*. O sistema converte cada um dos arquivos *JSON* em *dataframes* distintos (Figura 18).

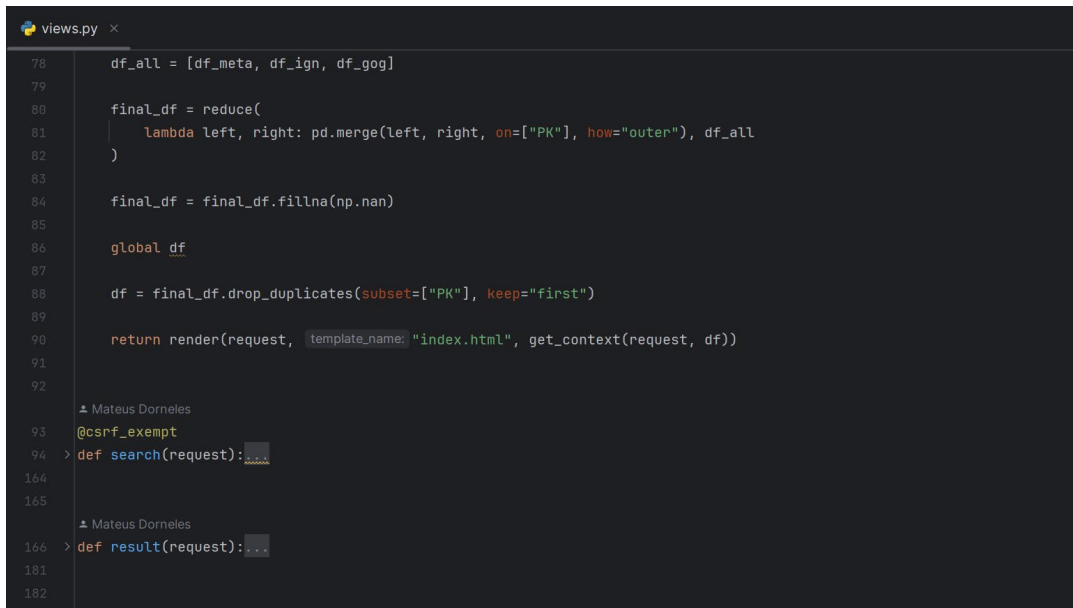
Figura 18 - Trecho de código de leitura dos três arquivos *JSON*

```
views.py x
19 def get_list(request):
20     with open("app/dados/result_metacritic.json", "r") as f:
21         search_meta = f.read()
22
23     df_meta = pd.read_json(search_meta)
24
25     df_meta = df_meta.drop_duplicates(subset=["URL"], keep="last")
26
27     del df_meta["index"]
28
29     for k, v in df_meta["URL"].items():
30         pk_meta.append(v.split("/")[-2])
31
32     df_meta["PK"] = pk_meta
33     pk_meta.clear()
34
35     shift_first_column(df_meta)
36
37     with open("app/dados/result_ign.json", "r") as f:
38         search_ign = f.read()
39
40     df_ign = pd.read_json(search_ign)
41
42     del df_ign["index"]
```

Fonte: Do autor (2024)

A próxima etapa consiste na criação de um *dataframe* unificado, para que os três *dataframes* gerados anteriormente sejam combinados, e os registros referentes aos jogos em comum entre as bases sejam normalizados, resultando em um único conjunto de dados onde cada jogo possui uma representação única (Figura 19). Na leitura de cada um dos arquivos é gerado um novo *dataframe* e nele é criada uma nova coluna (“PK”), que é composta pela última parte da *URL* de cada jogo (ver linhas 37 a 40 da Figura 18, como exemplo). Nas linhas 80 a 82 da Figura 19, é onde é realizada a união de todos os jogos que possuem o mesmo valor na coluna “PK”.

Figura 19 - Trecho de código responsável em unificar os três dataframes



```
views.py x
78 df_all = [df_meta, df_ign, df_gog]
79
80 final_df = reduce(
81     lambda left, right: pd.merge(left, right, on=["PK"], how="outer"), df_all
82 )
83
84 final_df = final_df.fillna(np.nan)
85
86 global df
87
88 df = final_df.drop_duplicates(subset=["PK"], keep="first")
89
90 return render(request, template_name="index.html", get_context(request, df))
91
92
93 @csrf_exempt
94 > def search(request):...
164
165
166 > def result(request):...
181
182
```

Fonte: Do autor (2024)

Este novo *dataframe* fica armazenado em uma variável global na aplicação e é utilizado nas operações em que o sistema de interface realiza a apresentação de dados de jogos ao usuário.

As colunas deste *dataframe* são organizadas e unificadas em tela de acordo com a taxonomia definida na Figura 2. Na página inicial, somente alguns campos são apresentados ao usuário (Figura 20).

Figura 20 - Trecho de código da página inicial da aplicação

```
<> index.html x
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <meta name="referrer" content="no-referrer">
6     <title>Games finder</title>
7     <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">
8     {% load static %}
9     <link rel="stylesheet" type="text/css" href="{% static 'style.css' %}">
10  </head>
11  <body>
12    <div class="container mt-5">
13      <form class="form-inline my-2 my-lg-0" action="{% url 'search' %}" method="post" role="form">
14        <div class="input-group mb-3">
15          <div class="input-group-prepend">
16            <span class="input-group-text" id="basic-addon1">Search Options:</span>
17          </div>
18          <label for="search-type"></label>
19          <select class="form-control mr-sm-2" id="search-type" name="search-type">
20            <option value="s-all-columns">All columns</option>
21            <option value="s-game-name">Game name</option>
22            <option value="s-genre">Genre</option>
23            <option value="s-developer-publisher">Developer/Publisher</option>
24            <option value="s-release-year" type="number">Release year</option>
```

Fonte: Do autor (2024)

Quando um jogo é escolhido para ser acessado, o *dataframe* lê todo o conteúdo da sua respectiva linha e apresenta ao usuário seguindo a taxonomia da Figura 2 (Figura 21).

Figura 21 - Trecho de código da página de detalhes de um jogo

```
<> game_details.html x
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <title>
6       {% if game_details.Game_name_x == game_details.Game_name_x %}
7       {{ game_details.Game_name_x }}
8       {% elif game_details.Game_name_y == game_details.Game_name_y %}
9       {{ game_details.Game_name_y }}
10      {% elif game_details.Game_name == game_details.Game_name %}
11      {{ game_details.Game_name }}
12      {% endif %}
13      - Games finder
14    </title>
15    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">
16    {% load static %}
17    <link rel="stylesheet" type="text/css" href="{% static 'style.css' %}">
18    <link href="https://maxcdn.bootstrapcdn.com/font-awesome/4.3.0/css/font-awesome.min.css" rel="stylesheet">
19  </head>
20  <body>
21    <div class="container">
22      <div>
23        {% if game_details.Game_name_x == game_details.Game_name_x %}
24        <h1 class="display-4"><span>{{ game_details.Game_name_x }}</span></h1>
```

Fonte: Do autor (2024)

A partir do momento em que o usuário realiza uma busca, o sistema identifica qual operação foi escolhida e filtra o *dataframe* de acordo com a escolha (Figura 22).

Figura 22 - Trecho de código da aplicação responsável pelas operações de busca no *dataframe*



```
views.py x
def search(request):
    regex = request.POST.get("search").lower()
    regex_type = request.POST.get("search-type")

    dataframe = pd.DataFrame()

    match regex_type:
        case "s-all-columns":
            textlikes = df.select_dtypes(include=[object, "string"])
            dataframe = df[
                textlikes.apply(
                    lambda column: column.str.contains(
                        regex, regex=True, case=False, na=False
                    )
                ).any(axis=1)
            ]
        case "s-game-name":
            dataframe = df[
                (df["Game_name_x"].str.lower().str.contains(regex))
                | (df["Game_name_y"].str.lower().str.contains(regex))
                | (df["Game_name"].str.lower().str.contains(regex))
            ]
        case "s-genre":
```

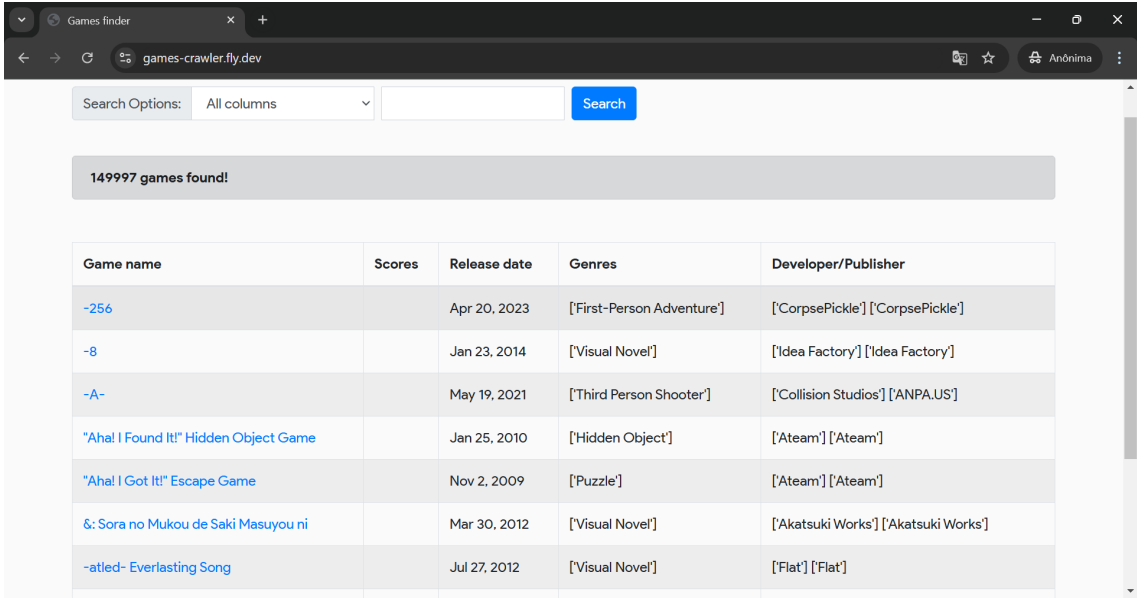
Fonte: Do autor (2024)

4. ILUSTRAÇÃO DE USO DO *GAMES FINDER APPLICATION*

Neste capítulo, serão demonstradas as telas do *Games Finder Application*, explicando suas funcionalidades e como utilizá-las.

Na primeira tela (Figura 23), que é visualizada quando o usuário acessa o sistema, contém uma lista com todos os jogos separados por páginas contendo dez jogos em cada uma delas (a quantidade de jogos por página pode ser configurada no código da aplicação). Juntamente com as páginas, é possível visualizar a quantidade total de jogos obtida ao combinar as três bases utilizadas (nesse cenário, quando um jogo é encontrado em mais de um site, ele somente é contado como um nesse número total).

Figura 23 - Página Inicial



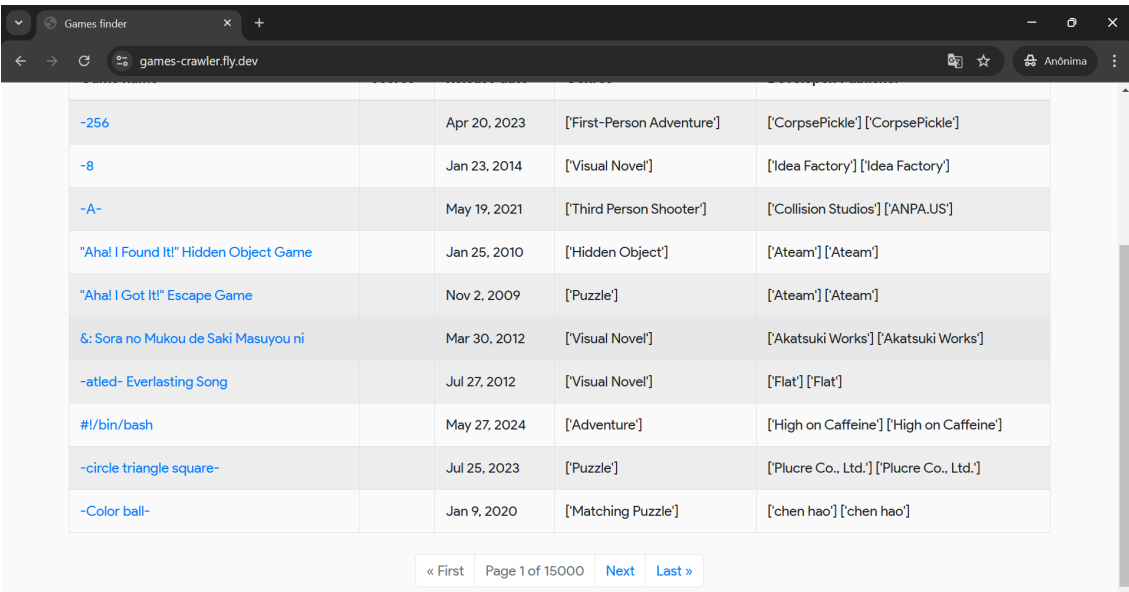
The screenshot shows a web browser window with the address bar displaying 'games-crawler.fly.dev'. The page has a search bar with 'Search Options: All columns' and a 'Search' button. Below the search bar, a grey banner states '149997 games found!'. A table displays game results with the following columns: Game name, Scores, Release date, Genres, and Developer/Publisher.

Game name	Scores	Release date	Genres	Developer/Publisher
-256		Apr 20, 2023	['First-Person Adventure']	['CorpsePickle'] ['CorpsePickle']
-8		Jan 23, 2014	['Visual Novel']	['Idea Factory'] ['Idea Factory']
-A-		May 19, 2021	['Third Person Shooter']	['Collision Studios'] ['ANPA.US']
"Aha! I Found It!" Hidden Object Game		Jan 25, 2010	['Hidden Object']	['Ateam'] ['Ateam']
"Aha! I Got It!" Escape Game		Nov 2, 2009	['Puzzle']	['Ateam'] ['Ateam']
&: Sora no Mukou de Saki Masuyou ni		Mar 30, 2012	['Visual Novel']	['Akatsuki Works'] ['Akatsuki Works']
-atled- Everlasting Song		Jul 27, 2012	['Visual Novel']	['Flat'] ['Flat']

Fonte: Do Autor (2024)

Além da visualização do número total de páginas e em qual página a aplicação se encontra, também é possível navegar nas páginas escolhendo as opções “*First*”, “*Previous*”, “*Next*” e “*Last*” (as opções apresentadas em tela variam de acordo com número de página atual). As opções de navegação entre as páginas encontram-se no fim de cada página (Figura 24).

Figura 24 - Opções de paginação



The screenshot shows a web browser window with the address bar displaying 'games-crawler.fly.dev'. The page contains a table with 5 columns. The first column lists game titles, some with blue links. The second column is empty. The third column shows dates. The fourth and fifth columns show developer names in brackets. At the bottom of the table, there are pagination controls: '« First', 'Page 1 of 15000', 'Next', and 'Last »'.

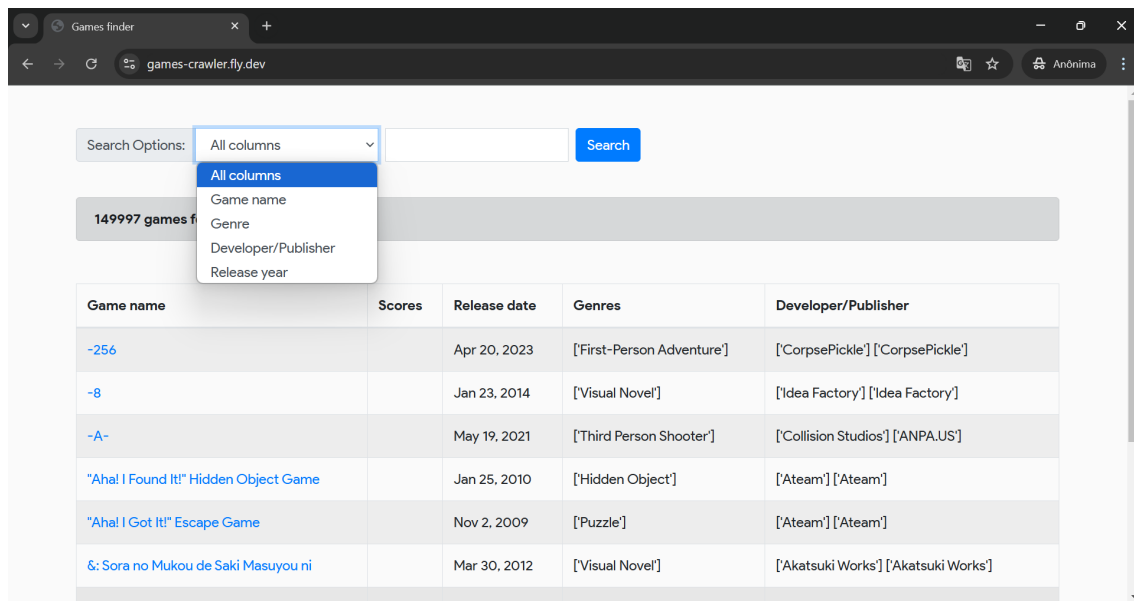
-256		Apr 20, 2023	[First-Person Adventure]	[CorpsePickle] [CorpsePickle]
-8		Jan 23, 2014	[Visual Novel]	[Idea Factory] [Idea Factory]
-A-		May 19, 2021	[Third Person Shooter]	[Collision Studios] [ANPA.US]
"Aha! I Found It!" Hidden Object Game		Jan 25, 2010	[Hidden Object]	[Ateam] [Ateam]
"Aha! I Got It!" Escape Game		Nov 2, 2009	[Puzzle]	[Ateam] [Ateam]
6: Sora no Mukou de Saki Masuyou ni		Mar 30, 2012	[Visual Novel]	[Akatsuki Works] [Akatsuki Works]
-atled- Everlasting Song		Jul 27, 2012	[Visual Novel]	[Flat] [Flat]
#!/bin/bash		May 27, 2024	[Adventure]	[High on Caffeine] [High on Caffeine]
-circle triangle square-		Jul 25, 2023	[Puzzle]	[Plucre Co., Ltd.] [Plucre Co., Ltd.]
-Color ball-		Jan 9, 2020	[Matching Puzzle]	[chen hao] [chen hao]

« First Page 1 of 15000 Next Last »

Fonte: Do Autor (2024)

Complementando, nessa mesma tela é possível visualizar um campo de seleção de opção de pesquisa e um campo onde é inserido o texto a ser buscado. A opção de pesquisa *default* é a “*All columns*”, onde a busca do termo que será inserido é realizada em cada coluna do *dataframe*. Ao clicar no *dropdown* de seleção, é apresentado em tela as opções disponíveis de pesquisa. (Figura 25)

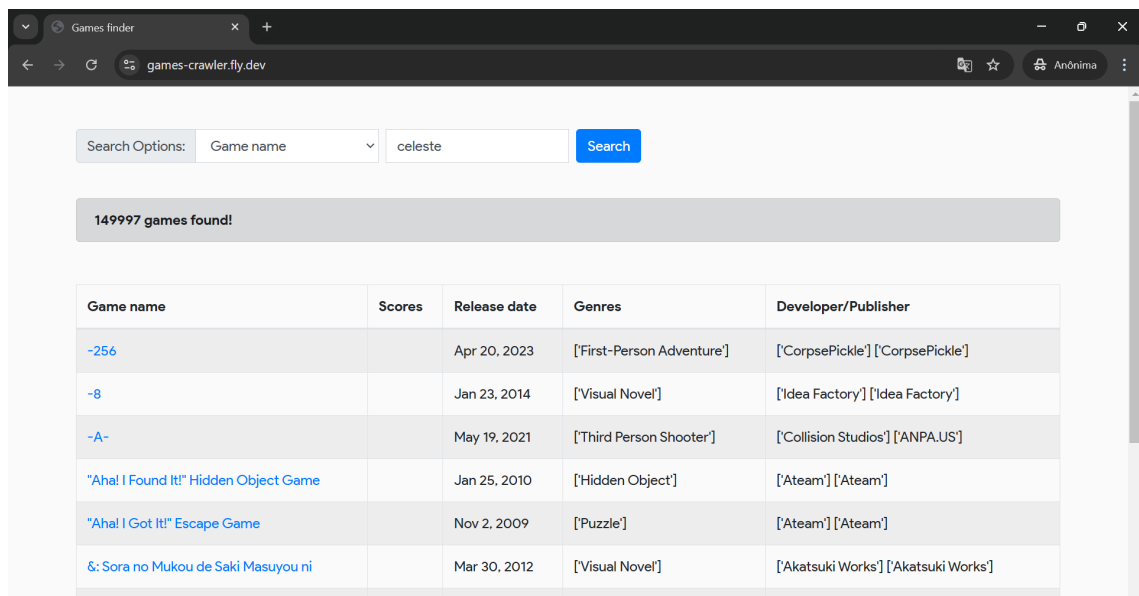
Figura 25 - Opções de busca



Fonte: Do Autor (2024)

Após o usuário realizar a seleção da opção de busca, ele então digita o termo de busca desejado (Figura 26) e aperta no botão "Search" para realizar a busca.

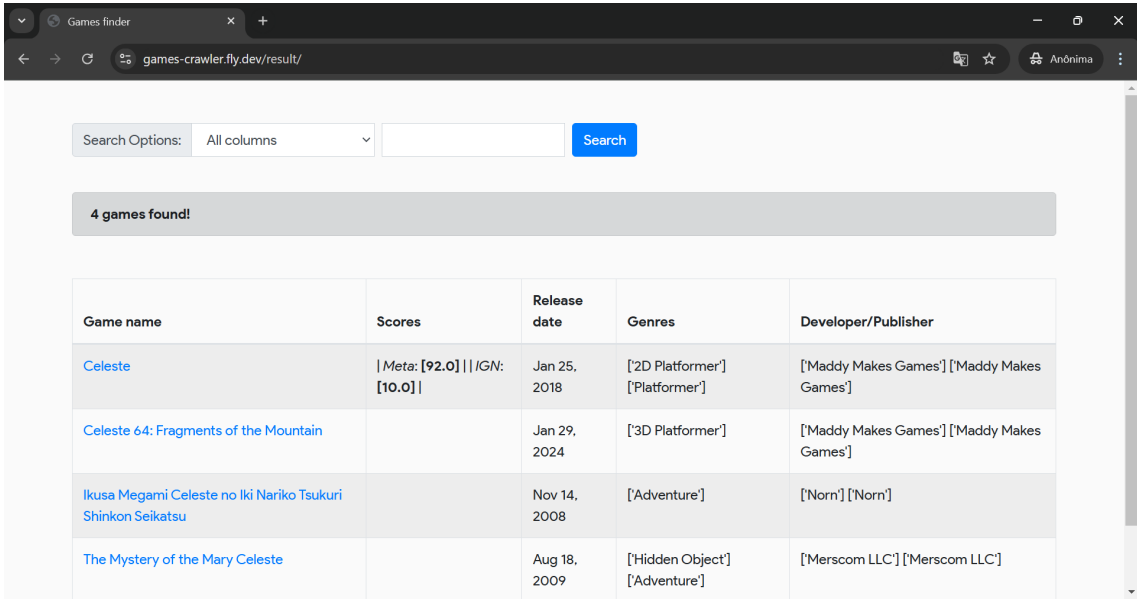
Figura 26 - Opção de busca selecionada e termo de busca digitado



Fonte: Do Autor (2024)

Em seguida são apresentados em tela os resultados de acordo com a pesquisa realizada (Figura 27), seguindo o mesmo padrão de paginação da tela da página inicial.

Figura 27 - Resultado de Busca



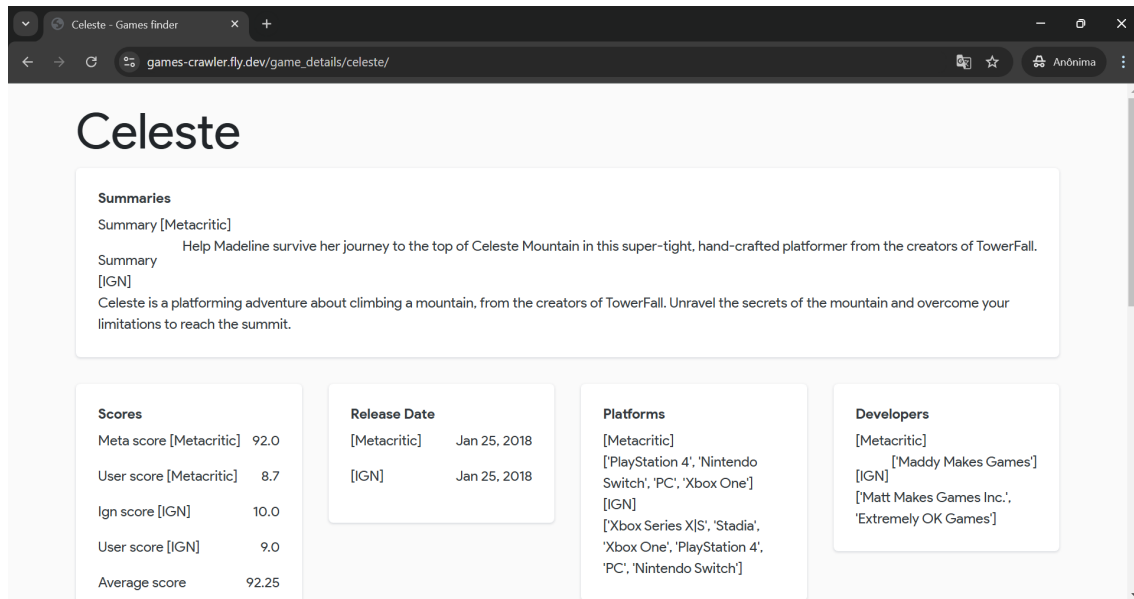
The screenshot shows a web browser window with the address bar displaying 'games-crawler.fly.dev/result/'. The page has a search bar at the top with 'Search Options: All columns' and a 'Search' button. Below the search bar, a message states '4 games found!'. A table displays the search results with the following columns: Game name, Scores, Release date, Genres, and Developer/Publisher.

Game name	Scores	Release date	Genres	Developer/Publisher
Celeste	Meta: [92.0] IG: [10.0]	Jan 25, 2018	[2D Platformer] [Platformer]	[Maddy Makes Games] [Maddy Makes Games]
Celeste 64: Fragments of the Mountain		Jan 29, 2024	[3D Platformer]	[Maddy Makes Games] [Maddy Makes Games]
Ikusa Megami Celeste no Iki Nariko Tsukuri Shinkon Seikatsu		Nov 14, 2008	[Adventure]	[Norn] [Norn]
The Mystery of the Mary Celeste		Aug 18, 2009	[Hidden Object] [Adventure]	[Merscom LLC] [Merscom LLC]

Fonte: Do Autor (2024)

Tanto na tela inicial quanto na tela de resultado de busca, é possível acessar a página de detalhes de um jogo clicando em uma das instâncias da coluna “*Game name*”.

Ao clicar na instância do jogo, o sistema redireciona a página atual para a página de detalhes do jogo escolhido (Figura 28).

Figura 28 - Página de Detalhes do Jogo

Fonte: Do Autor (2024)

Nesta tela o usuário é capaz de visualizar de forma centralizada as informações do jogo baseadas em múltiplas fontes, com a indicação da respectiva fonte da informação. Ao utilizar a aplicação web, o usuário pode consultar uma lista com todos os jogos disponíveis catalogados, consultar todas as informações encontradas de cada jogo e realizar pesquisas de acordo com o seu interesse.

5. CONSIDERAÇÕES FINAIS

Neste trabalho, foi desenvolvido e apresentado a ferramenta *Games Finder Application*, que é composta por dois módulos: o módulo do sistema responsável por coletar as informações das fontes selecionadas e o módulo onde o sistema reúne os dados de diferentes fontes e apresenta ao usuário de forma agrupada e organizada.

A construção dessa aplicação permitiu a implementação de mecanismos de buscas que auxiliam a geração de informações de jogos de forma centralizada. O usuário, ao acessar o sistema, é capaz de acessar todos os dados de múltiplas fontes de um mesmo jogo.

Durante o desenvolvimento do projeto, foi possível realizar a análise da taxonomia dos dados dos três *websites* ao modelar os campos dos sites, de modo a encontrar campos que possuíssem o mesmo valor semântico nas três bases. Obteve-se sucesso na criação de *crawlers* de extração de dados de todas as três fontes escolhidas, ao aplicar um desenvolvimento personalizado de acordo com as características da estrutura de páginas de cada fonte.

Visando também atingir a proposta deste projeto, foi possível a implementação de uma aplicação *web* onde foi desenvolvida uma solução que consolida os dados heterogêneos e conectados em uma única base de informação tendo como suporte a taxonomia de conceitos de jogos. O sistema disponibiliza de forma visual ao usuário todos os campos agrupados e apresentados de acordo com seu valor semântico.

Através da implementação de funcionalidades dentro do sistema, foi possível disponibilizar ao usuário consultar livremente e de forma consolidada os dados de jogos oriundos das bases coletadas, realizar pesquisas específicas e visualizar os dados conectados dos jogos contidos no catálogo.

Consequentemente, o *Games Finder Application* é uma ferramenta capaz de ler e armazenar dados de diferentes *websites* e condensar essas informações em uma ferramenta de visualização simplificada. A ferramenta é capaz de contribuir para objeto de estudo relacionados a semântica de jogos, permitindo análises para possíveis inferências.

No entanto, é importante ressaltar que a pesquisa apresenta algumas limitações, como o fato de o sistema não ter sido testado e avaliado em um estudo empírico, a fim de analisar a usabilidade e utilidade da ferramenta.

Outra limitação encontrada foi que, durante a execução do *Crawler* do *Metacritic*, o *website* subiu em produção uma grande reestruturação de sua interface e nomes de campos utilizados em seus arquivos HTML. Além da falha na execução da aplicação, isso resultou na necessidade de alteração de grande parte do código de seu *Crawler* para atender as novas mudanças de estrutura do código *HTML*.

Uma possível melhoria para trabalhos futuros seria o desenvolvimento de um código de forma que ocorra uma otimização em relação ao tempo em que a aplicação de "*scraping*" leva até que os dados sejam coletados. Uma vez que, ao se somar o tempo total que foi despendido no processo de coleta de dados das três fontes de pesquisa citadas ao longo do texto, tem-se um número de 93,39 horas. Como sugestões de conceitos de melhoria na implementação, que podem ser aplicadas em um trabalho futuro e que podem aumentar a eficiência da execução do código, pode-se citar: *multithreading*, *multitasking*, *multiprocessing*, *concurrency*, *parallel processing* e *hyper processing*.

Outro modo de se acrescentar valor a este projeto é o de incluir novas fontes a serem adicionadas na base consolidada. Além disso, também é possível automatizar a execução dos processos dos "*scraping*" em rotinas, para que os dados dos jogos estejam sempre atualizados de forma automática e que à medida que novos jogos são lançados, sejam incluídos na biblioteca de jogos.

Além do que já foi citado, também se abre uma oportunidade de implementar mecanismos de busca específicos para a coleta de dados relacionados à jogos *mobile*, que é um mercado que tem crescido cada vez mais o seu faturamento e adesão de grande parte do público que consomem jogos digitais.

Em conclusão, o propósito deste trabalho, que era construir um sistema de apoio para consulta de dados conectados, contendo informações de múltiplas fontes, foi atingido.

REFERÊNCIAS BIBLIOGRÁFICAS

Adams, K. C. Immersed in structure: the meaning an function of taxonomies. 2000.

Advantages Of Web Scraping vs Manual Work. Crawlbase, 2022. Disponível em: <<https://crawlbase.com/blog/advantages-of-web-scraping-vs-manual-work/>> Acesso em: 13 dez. 2024.

Cloudflare. O que é um web crawler? | Como funcionam os web spiders. Cloudflare, s.d. Disponível em: <<https://www.cloudflare.com/pt-br/learning/bots/what-is-a-web-crawler/>> Acesso em: 13 dez. 2024.

De Groat, G. et al. Best Practices for Cataloging Video Games. 2018. Disponível em: <<https://cornerstone.lib.mnsu.edu/cgi/viewcontent.cgi?article=1017&context=olac-publications>> Acesso em: 13 dez. 2024.

Ewald, H. et al. Searching two or more databases decreased the risk of missing relevant studies: a metaresearch study. 2022. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0895435622001445>> Acesso em: 13 dez. 2024.

Heath, Tom; Bizer, Christian. Linked data: Evolving the web into a global data space. Springer Nature, 2022.

Hope, Diana. The Role of Big Data Analytics in Gaming. SmartData Collective, 2023. Disponível em: <<https://www.smartdatacollective.com/role-of-big-data-analytics-gaming/>> Acesso em: 13 dez. 2024.

Isotani, S.; Bittencourt, I.. Dados Abertos Conectados: Em Busca da web do Conhecimento. Novatec Editora, 2015. Disponível em: <https://www.researchgate.net/publication/282218981_Dados_Abertos_Conectados_em_Busca_da_Web_do_Conhecimento> Acesso em: 13 dez. 2024.

Kritz, J., Costa, L., Monclar, M., Garrot, R., Classe, T. Azevedo, V.; "O que são jogos". Relatório técnico LUDES 2017, PESC 752, 2017.

Kurman, T.. RAWG is officially the largest video game database in the world. Medium, 2018. Disponível em: <<https://medium.com/rawg/rawg-gets-bigger-and-more-accessible-cc0254ac84bd>> Acesso em: 13 dez. 2024.

Mcquillan, R.. How to Integrate Multiple Databases. Budibase, 2022. Disponível em: <<https://budibase.com/blog/data/how-to-integrate-multiple-databases/>> Acesso em 13 dez. 2024.

Miranda, Frederico S.. Jogo Digital: definição do termo. In: Anais do SBGames 2017: XVII Simpósio Brasileiro de Jogos e Entretenimento Digital. 2017. p. 296-299. Disponível em: <<https://www.sbgames.org/sbgames2017/papers/ArtesDesignShort/173500.pdf>> Acesso em: 13 dez. 2024.

Newzoo's Global Games Market Report 2024. NEWZOO, 2024. Disponível em: <<https://newzoo.com/resources/blog/global-games-market-revenue-estimates-and-forecasts-in-2024>> Acesso em: 13 dez. 2024.

Os videogames russos podem entrar no mercado do BRICS?. Sputnik Brasil, 2023. Disponível em: <<https://sputniknewsbr.com.br/20230820/os-videogames-russos-podem-entrar-no-mercado-do-brics-29977335.html>> Acesso em: 13 dez. 2024

Saha, Srishti. How Web scraping and Big Data Analytics can be used to impact the Media and Entertainment industry. Datahut, 2018. Disponível em: <<https://www.blog.datahut.co/post/how-web-scraping-and-big-data-analytics-can-be-used-to-impact-the-media-and-entertainment-industry>> Atualizado em: 12 fev. 2021. Acesso em: 13 dez. 2024.

Souza, Lucas Fiamoncini de. Web scraping: coleta de dados automatizada. Aquarela Analytics, 2022. Disponível em: <<https://aquare.la/web-scraping-coleta-de-dados-automatizada/>> Acesso em: 13 dez. 2024.

Tadeu, V.; Tortella, T.. Público gamer cresce e 3 em cada 4 brasileiros consomem jogos eletrônicos. CNN Brasil, 2022. Disponível em: <<https://www.cnnbrasil.com.br/tecnologia/publico-gamer-cresce-e-3-em-cada-4-brasileiros-consomem-jogos-eletronicos/>> Acesso em: 13 dez. 2024.

Terra, J. C. C. et al. Taxonomia: elemento fundamental para a gestão do conhecimento. 2005. Disponível em: <<http://pessoal.utfpr.edu.br/mansano/arquivos/taxonomia.pdf>> Acesso em: 13 dez. 2024.

Vital, L.; Café, L.. Ontologias e taxonomias: diferenças. Disponível em: <<https://www.scielo.br/j/pci/a/VxhxGfW6zfHpfTswwPM5wXP>> Acesso em: 13 dez. 2024.

Web crawling vs web scraping. Zyte, s.d. Disponível em: <<https://www.zyte.com/learn/difference-between-web-scraping-and-web-crawling/>> Acesso em: 13 dez. 2024.

What Is Web Scraping?. Simplilearn, 2023. Disponível em: <https://www.simplilearn.com/what-is-web-scraping-article#what_is_web_scraping> Acesso em: 13 dez. 2024.

Xexéo, G., Carmo, A., Acioli, A., Taucei, B., Dipolitto, C. Mangeli, E., Kritz, J., Costa, L., Monclar, M., Garrot, R., Classe, T. Azevedo, V.; "O que são jogos". Relatório técnico LUDS 2017, PESC 752, 2017.

Apêndice 1. CASOS DE USO DO SISTEMA

Visualizar Lista de Jogos

Descrição: Usuário acessar o sistema e visualizar lista de todos os jogos

Atores: Usuário

Pré-condições: Nenhuma

Fluxo Principal:

1. Usuário acessa a página do site
2. Sistema disponibiliza a página inicial com a lista de todos os jogos

Fluxo Alternativo: -

Fluxo de Exceção: -

Pós-condições: Usuário visualiza a lista de todos os jogos

Regras de Negócio: -

Requisitos:

- RF1: O sistema deve permitir que o usuário visualize todos os jogos disponíveis no catálogo

Escolher Filtro da Busca

Descrição: Usuário define campo de seleção de opção de pesquisa

Atores: Usuário

Pré-condições: Usuário deve estar na página inicial

Fluxo Principal:

1. Usuário seleciona o *dropdown* à direita da caixa de texto “Search Options:”
2. Sistema apresenta as opções de busca
3. Usuário seleciona uma opção de busca desejada

Fluxo Alternativo: -

Fluxo de Exceção: -

Pós-condições: Usuário escolhe um campo de seleção de opção de pesquisa

Regras de Negócio:

- RN1: O usuário só poderá selecionar um filtro para cada busca

Requisitos:

- RF3: O sistema deve permitir que o usuário selecione um filtro de busca

Realizar Busca

Descrição: Usuário realiza busca de acordo com opção de pesquisa e texto a sua escolha

Atores: Usuário

Pré-condições: Usuário deve estar na página inicial, campo de seleção de opção de pesquisa definido

Fluxo Principal:

1. Usuário insere o termo de pesquisa
2. Usuário seleciona o botão “Search”

Fluxo Alternativo: -

Fluxo de Exceção: -

Pós-condições: Usuário visualiza o resultado da pesquisa realizada

Regras de Negócio: -

Requisitos:

- RF4: O sistema deve permitir que o usuário realize a busca

Visualizar Dados Conectados do Jogo

Descrição: Usuário acessa a página contendo os dados conectados de um jogo

Atores: Usuário

Pré-condições: Usuário deve estar na página inicial ou na página de resultado de busca

Fluxo Principal:

1. Usuário seleciona uma das instâncias da coluna “Game name”
2. Sistema redireciona para página de detalhes do jogo selecionado

Fluxo Alternativo: -

Fluxo de Exceção: -

Pós-condições: Usuário visualiza os dados conectados do jogo selecionada em uma nova tela

Regras de Negócio: -

Requisitos:

- RF5: O sistema deve permitir que o usuário acesse a página de dados conectados de um jogo de sua escolha
- RF6: O sistema deve permitir que o usuário visualize os dados conectados de um determinado jogo em uma página única

Alternar Página

Descrição: Usuário altera página onde o sistema se encontra

Atores: Usuário

Pré-condições: Usuário deve estar na página inicial ou na página de resultado de busca

Fluxo Principal:

1. Usuário escolhe uma das opções de alteração de página (“First”, “Previous”, “Next” ou “Last”)
2. Sistema carrega a página correspondente à escolha do usuário

Fluxo Alternativo: -

Fluxo de Exceção: -

Pós-condições: Usuário acessa a página desejada

Regras de Negócio: -

Requisitos:

- RF2: O sistema deve permitir que o usuário alterne a página atual