

**CENTRO UNIVERSITÁRIO UNIRUY  
WYDEN CAMPUS SALVADOR/BA**



**MEMORIAL DESCRITIVO DO PROJETO  
STEAMPOINT**

**2025  
Salvador/BA**

**CENTRO UNIVERSITÁRIO UNIRUY  
WYDEN CAMPUS SALVADOR/BA**

**MEMORIAL DESCRITIVO  
PROJETO STEAMPOINT**

Sinézio da Silva Ramos Junior – 202302375081  
Paulo Henrique Ribeiro Chaves – 202303677308  
Felipe da Silva Chimeres - 202302589472  
Bruno Santos Oliveira – 202302375138  
Gabriel Salazar Araujo Alcântara - 202302375022

Trabalho para obtenção de nota na  
disciplina de “*Tópicos de BigData em  
Python*”.

Orientador(a): Roney Malaguti

**2025  
Salvador/BA**

## Sumário

1.	Objetivo do projeto.....	4
2.	Justificativa para escolha do tema. ....	4
3.	Base de dados utilizadas: .....	4
4.	Tecnologias aplicadas. ....	5
5.	Metodologia e Etapas de Elaboração. ....	5
6.	Grupo de trabalho. ....	17
7.	Link no GitHub. ....	17
8.	Considerações Finais.....	17

## **1. Objetivo do projeto.**

O objeto deste projeto é destinado ao aprendizado sobre técnicas de análise de dados, utilizando Python como linguagem predominante, e conteúdo ministrado em sala. O projeto é focado no tratamento, análise, exploração e direcionamento dos dados fornecidos da plataforma Steam, para averiguar tendências categóricas, comparações, desenvolvimento de consumo e produção no mercado de jogos.

## **2. Justificativa para escolha do tema.**

Fundamentalmente pelo conhecimento, e vastidão dos dados que a plataforma possui, ampla gama de dados onde visualizamos a possibilidade de poder trabalhar e trazer resultados conforme proposta requerida em sala de aula. A dimensão do setor de jogos também foi um icónica, visto o crescente mercado em que o próprio Brasil ocupa o 11ª colocação global de compra e venda de jogos, com receita de 1,458 Bilhão de dólares, além do que os objetos listados possuem consideráveis itens agregadores que acreditamos serem essenciais para contribuir para o então desenvolvimento.

## **3. Base de dados utilizadas:**

Extraído do repositório commitado por NewbieIndieGameDev, constitui-se de dados em arquivos “Comma Separated Values” (.CSV), atualizados até Outubro de 2024. Dentre as características contidas, envolvem:

- a) Informações dos jogos (nome, data de lançamento, se é gratuito);
- b) Desenvolvedores, publicadoras, idiomas e gêneros;
- c) Reviews e recomendações;
- d) Categorias e tags;
- e) Imagens promocionais.

Link: [GitHub - NewbieIndieGameDev/steam-insights: Database of Steam video game data from October 2024, including game details, genres, reviews, tags, and SteamSpy insights](https://github.com/NewbieIndieGameDev/steam-insights)

## 4. Tecnologias aplicadas.

### a. Linguagem de Programação em Python.

Bibliotecas utilizadas:

- Flask: Microframework web e flexível em Python, para criar a aplicação web com a API com mais agilidade.
- Pandas: Biblioteca Python para a manipulação, e análise de dados.

### b. Gerenciador de Pacotes Poetry.

- Framework para simplificar o gerenciamento das dependências e empacotar com o projeto com melhor performance.

### c. HTML, CSS, JAVASCRIPT.

- HyperText Markup Language: Linguagem de marcação para dimensionar e estruturar o front da aplicação web.
- Cascading Style Sheets: Linguagem de estilização da aplicação web.
- JavaScript: Linguagem de programação de alto nível para implementar funcionalidades interativas a aplicação web.
  - Chart JS: Implementação de gráficos.

### d. Jupyter.

- i. Ambiente interativo para programação.

## 5. Metodologia e Etapas de Elaboração.

### Coleta de Dados:

Conforme “Item 3.”, os dados provém de arquivos .CSV, embuidos sobre jogos contidos na base de dados da Steam, até Outubro de 2024, disponibilizados no diretório do GitHub.

### Análise Exploratória:

Para melhor aproveitamento dos dados, realizamos a remoção de algumas colunas como as de video, preço, e screenshots.

```

# Tratar o arquivo promotional.csv / Remover as colunas de video e screenshots

import csv

input_file = 'DataCSV/promotional.csv'
output_file = 'newpromotional.csv'

with open(input_file, mode='r', encoding='utf-8') as infile, \
    open(output_file, mode='w', newline='', encoding='utf-8') as outfile:

    reader = csv.DictReader(infile)
    fieldnames = ['app_id', 'header_image', 'background_image']
    writer = csv.DictWriter(outfile, fieldnames=fieldnames, quoting=csv.QUOTE_ALL)

    writer.writeheader()
    for row in reader:
        new_row = {key: row.get(key, "") for key in fieldnames}
        writer.writerow(new_row)

print("Arquivo salvo como", output_file)

```

Figura 1: Remoção da coluna de video e screenshors. Fonte: treatdata.py

```

#Tratar o arquivo games.csv / Remover as colunas de preço

import re

# Caminhos dos arquivos
entrada_csv = 'DataCSV/games.csv'
saida_csv = 'DataCSV/games_limpo.csv'

# Expressão regular para remover conteúdo dentro de chaves, mantendo as chaves
regex_chaves = re.compile(r'\{.*?\}')

with open(entrada_csv, 'r', encoding='utf-8') as entrada, \
    open(saida_csv, 'w', encoding='utf-8') as saida:

    for linha in entrada:
        linha_limpa = regex_chaves.sub('{}', linha) # Substitui conteúdo das chaves por {}
        saida.write(linha_limpa)

print("Pronto! Novo CSV criado com conteúdo das chaves limpo.")

```

Figura 2: Remoção das colunas de preço. Fonte: treatdata.py

```
#Tratar o arquivo reviews.csv / Organizar a coluna de reviews

import csv

def corrigir_csv_v2(input_file, output_file):
    """
    Corrige as linhas quebradas em um arquivo CSV e adiciona aspas aos campos.

    Args:
        input_file (str): O caminho para o arquivo CSV de entrada.
        output_file (str): O caminho para o arquivo CSV de saída corrigido.
    """
    with open(input_file, 'r', encoding='utf-8') as infile, \
        open(output_file, 'w', newline='', encoding='utf-8') as outfile:
        reader = csv.reader(infile)
        writer = csv.writer(outfile, quoting=csv.QUOTE_ALL) # Adiciona aspas a todos os campos
        header = next(reader) # Lê o cabeçalho
        writer.writerow(header)

        linha_acumulada = []
        num_colunas_esperado = len(header)

        for row in reader:
            linha_acumulada.extend(row)
            if len(linha_acumulada) == num_colunas_esperado:
                writer.writerow(linha_acumulada)
                linha_acumulada = []
            elif len(linha_acumulada) > num_colunas_esperado:
                print(f"Aviso: Linha com mais colunas que o esperado. Ignorando o excesso: {linha_acumulada}")
                writer.writerow(linha_acumulada[:num_colunas_esperado])
                linha_acumulada = []

        # Se sobrar algo na linha_acumulada no final do arquivo
        if linha_acumulada:
            print(f"Aviso: Fim do arquivo com linha incompleta: {linha_acumulada}")
            writer.writerow(linha_acumulada)

if __name__ == "__main__":
    arquivo_entrada = 'DataCSV/reviews.csv' # Substitua pelo nome do seu arquivo de entrada
    arquivo_saida = 'review_corrigido.csv' # Nome do arquivo de saída corrigido
    corrigir_csv_v2(arquivo_entrada, arquivo_saida)
    print(f"Arquivo corrigido '{arquivo_saida}' gerado com sucesso.")
```

**Figura 3:** Reorganização da coluna reviews. Fonte *treatdata.py*

Interseccionamos as tabelas para juntar todas as colunas de informações contidas em cada uma delas com Merge/Inner afim de acomoda-los em um único CSV, chamado “games\_merged\_final.csv”, em seguida separamos o DataFrame interseccionado em DataFremes separados, conforme “**Figura 4**.”

```
import pandas as pd

# Função para realizar o inner join, salvar em CSV e contar removidos
def process_and_save(df, df_name, on='app_id'):
    """
    """
    output_filename = f'{df_name}.csv'
    df.to_csv(output_filename, index=False, encoding='utf-8')
    print(f"Arquivo CSV '{output_filename}' criado com sucesso!")

# Carregar os dados dos arquivos CSV
# Use os nomes dos seus arquivos
categories_df = pd.read_csv('DataCSV/categories.csv', sep=',')
games_limpo_df = pd.read_csv('DataCSV/games_limpo.csv', sep=',')
newpromotional_df = pd.read_csv('DataCSV/newpromotional.csv', sep=',')
review_corrigido_df = pd.read_csv('DataCSV/review_corrigido.csv', sep=',')
steamspy_insights_df = pd.read_csv('DataCSV/steamspy_insights.csv', sep=',')
tags_df = pd.read_csv('DataCSV/tags.csv', sep=',')

# Selecionando colunas
dg_filter = games_limpo_df[['app_id', 'name', 'release_date', 'is_free', 'type']]
ds_filter = steamspy_insights_df[['app_id', 'developer', 'publisher', 'languages', 'genres']]
dr_filter = review_corrigido_df[['app_id', 'review_score', 'review_score_description', 'positive', 'negative', 'total', 'metacritic_score', 'recommendations']]

# Realizar merges encadeados
df_merged = pd.merge(dg_filter, ds_filter, on='app_id', how='inner')
df_merged = pd.merge(df_merged, dr_filter, on='app_id', how='inner')
df_merged = pd.merge(df_merged, newpromotional_df, on='app_id', how='inner')
df_merged = pd.merge(df_merged, categories_df, on='app_id', how='inner')
df_merged = pd.merge(df_merged, tags_df, on='app_id', how='inner')

# Salvar o resultado final
process_and_save(df_merged, 'games_merged_final')

# Separar o DataFrame mergeado em DataFrames individuais
games_limpo_merged = df_merged[['app_id', 'name', 'release_date', 'is_free', 'type']].drop_duplicates()
steamspy_insights_merged = df_merged[['app_id', 'developer', 'publisher', 'languages', 'genres']].drop_duplicates()
review_corrigido_merged = df_merged[['app_id', 'review_score', 'review_score_description', 'positive', 'negative', 'total', 'metacritic_score', 'recommendations']].drop_duplicates()
newpromotional_merged = df_merged[['app_id'] + [col for col in df_merged.columns if 'promotional' in col]].drop_duplicates()
categories_merged = df_merged[['app_id'] + [col for col in df_merged.columns if 'category' in col]].drop_duplicates()
tags_merged = df_merged[['app_id'] + [col for col in df_merged.columns if 'tag' in col]].drop_duplicates()
```

**Figura 4:** Interceção entre tabelas. Fonte: *Main.py*

```

# Salvar os DataFrames separados
process_and_save(games_limpo_merged, 'games_limpo_merged')
process_and_save(steamspy_insights_merged, 'steamspy_insights_merged')
process_and_save(review_corrigido_merged, 'review_corrigido_merged')
process_and_save(newpromotional_merged, 'newpromotional_merged')
process_and_save(categories_merged, 'categories_merged')
process_and_save(tags_merged, 'tags_merged')
...

...

import pandas as pd
|
# Carrega o arquivo CSV que contém todos os dados
games_merged_final_df = pd.read_csv('DataCSV/games_merged_final.csv', sep=',')

# Carrega o arquivo CSV promocional, que contém apenas a coluna 'app_id'
promotional_df = pd.read_csv('DataCSV/promotional.csv', sep=',') # Certifique-se de que o caminho está correto

# Garante que temos apenas um 'header_image' e 'background_image' por 'app_id'
promotional_data_from_merged = games_merged_final_df[['app_id', 'header_image', 'background_image']].drop_duplicates()

# Realiza o merge para adicionar as colunas ao DataFrame promocional.
promotional_df_updated = pd.merge(promotional_df, promotional_data_from_merged, on='app_id', how='left')

# Salva o DataFrame atualizado em um novo arquivo CSV
promotional_df_updated.to_csv('promotional_atualizado.csv', index=False, encoding='utf-8')
print("Arquivo CSV 'promotional_atualizado.csv' criado com sucesso!")
...

```

*Figura 5: Save dos arquivos, e tratamento do promotional.csv. Fonte: Main.py*

Salvamos os arquivos separadamente, e após isso, tratamos o arquivo “*promotional.csv*”, por má formatação da base de dados.

## Modelagem de Dados:

Para a extração de insights a partir da base de dados utilizada no projeto, foram aplicadas técnicas de modelagem exploratória de dados com foco na visualização e análise de distribuições e padrões. A modelagem não se concentrou em algoritmos estatísticos ou de machine learning avançados, mas sim na organização, agregação e apresentação visual das informações de forma interativa e acessível ao usuário.

Os dados foram obtidos de uma API construída em Python (Flask) e consumidos no front-end via JavaScript. A extração de informações relevantes — como os gêneros e idiomas mais frequentes entre os jogos — foi realizada diretamente no cliente (navegador), utilizando processamento de arrays para agrupar e contar as ocorrências de cada categoria.

A biblioteca **Chart.js** foi utilizada para a geração de gráficos dinâmicos e interativos, permitindo identificar rapidamente tendências como os gêneros mais populares ou os idiomas mais suportados. O processo de modelagem consistiu nas seguintes etapas:

**Pré-processamento no front-end:** separação de strings, remoção de valores nulos e contagem de frequência;

**Ordenação e seleção:** os valores foram classificados por frequência e limitados às 20 categorias mais recorrentes;

**Mapeamento visual:** as categorias foram associadas a cores fixas previamente definidas, garantindo uma visualização clara e padronizada;



**Renderização gráfica:** os dados foram apresentados em gráficos de barras horizontais (Chart.js), promovendo a análise intuitiva e visual dos dados agregados.

## Visualização de Dados:

Por unanimidade do grupo de trabalho, foi decidido fomentar o presente projeto através de uma Aplicação Web, separamos as páginas com os resultados de pesquisa utilizando gráficos linha, rosca, pizza, pontos e coluna.

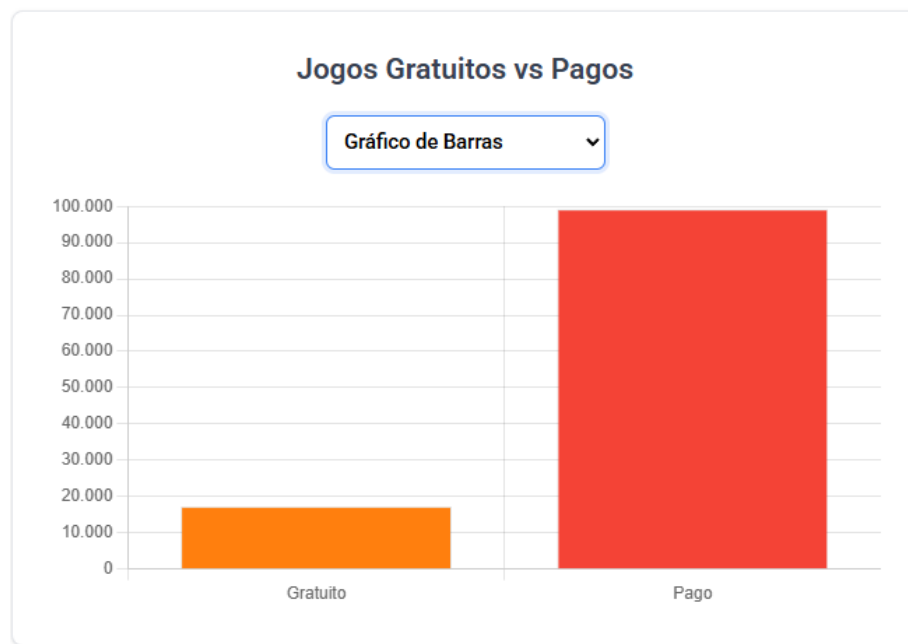


**Figura 6:** Gráfico-Coluna - gamexdemonstração. Fonte: domínio/Steam-Games.html

Gráfico da “**Figura 6**” mostra o a quantidade de jogos em demonstração em relação aos que já está disponíveis na plataforma, em números é uma proporção de **114.526 disponíveis para 1.967 em demonstração**. A imensa maioria dos conteúdos disponíveis são jogos completos, o que indica o foco comercial da plataforma,

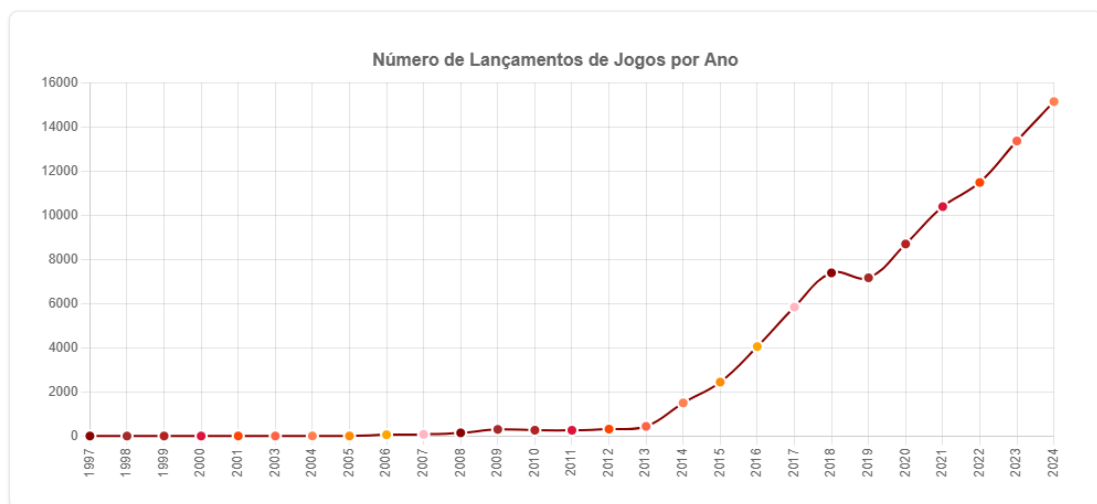
A baixa quantidade de demos pode dificultar a experiência do usuário que deseja testar antes de comprar.

Isso também reflete uma mudança na indústria: antigamente era comum lançar demos; hoje, vídeos, análises e o sistema de reembolso do Steam substituem esse modelo.



**Figura 7:** Gráfico Coluna - comparativo gratuito x pago. Fonte: domínio/Steam-Games.html

Este gráfico compara a quantidade de jogos gratuitos que chegam a aproximadamente 17mil títulos, representando 20% do montante, mesmo com o crescimento de modelos Free-To-Play, ainda dão minoria, enquanto 95mil títulos correspondem a 80% dos jogos contidos na plataforma, reforçando o modelo tradicional de monetização por compra direta.



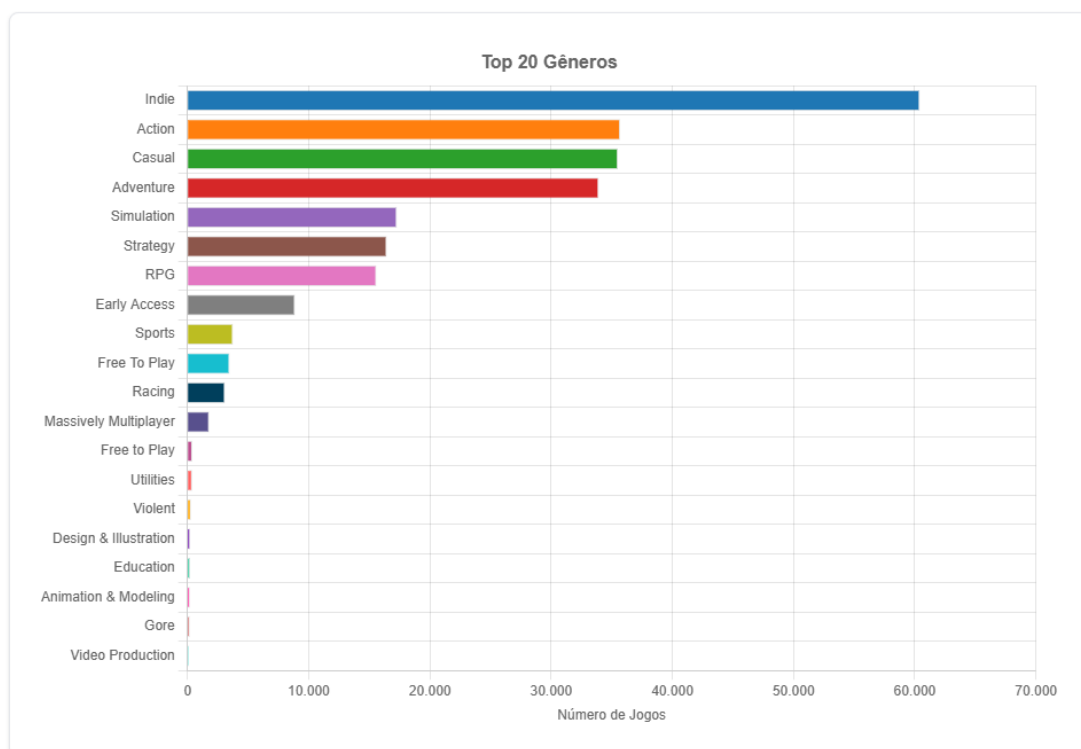
**Figura 9:** Gráfico de Linha – Lançamento de Jogos por Ano. Fonte: domínio/Steam-Games.html

Este gráfico mostra a **evolução do número de lançamentos de jogos por ano na Steam**, desde **1997 até 2024**, revelando tendências marcantes da indústria de jogos digitais.

- **1997 - 2013**, o número de lançamentos anuais **foi** baixo e estável — sempre abaixo de 1.000 jogos por ano.
- A partir de **2014**, há um aumento acentuado, com crescimento exponencial até 2024.
- **2016 - 2018**: crescimento rápido, indicando o início da “explosão de lançamentos”.
- **2020**: leve queda (possivelmente ligada à pandemia e atrasos na produção).
- **2021 em diante**: retomada forte e contínua no crescimento.
- **2024**: projeção ou dado atual com recorde de mais de 15.000 jogos lançados no ano.

#### As possíveis explicações para aumento estão:

1. Facilidade de publicação com ferramentas como Unity, Unreal e Godot.
2. Programas da Steam como o Steam Direct (desde 2017), que facilitou o acesso de desenvolvedores independentes à plataforma.
3. Crescimento do mercado indie e globalização da indústria.
4. Popularização do desenvolvimento de jogos por indivíduos e pequenos estúdios.

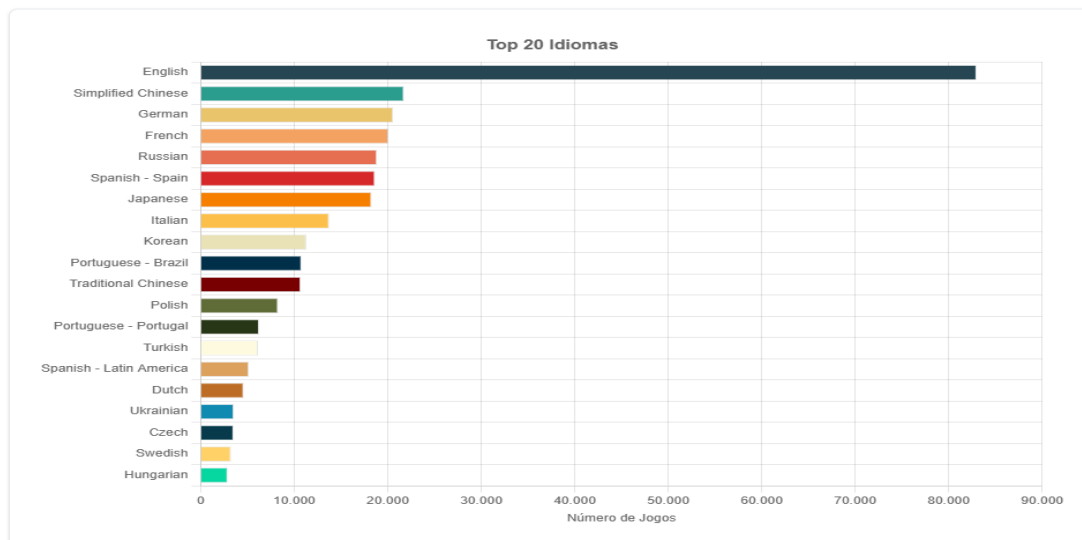


**Figura 10:** Gráfico Barras – Top 20 gêneros. Fonte: domínio/Games-Insights.html.

Com o Gráfico Barras demonstrado na “**Figura 7**”, nos mostra os 20 gêneros mais comuns de jogos na Steam, com base no número de títulos cadastrados por gênero. Ele traz insights importantes sobre o perfil da produção e preferências dentro da plataforma. **Indie** é o gênero mais popular, com larga vantagem (mais de 60 mil jogos). Isso evidencia a grande presença de desenvolvedores independentes na plataforma, aproveitando a acessibilidade da Steam para publicar seus títulos.

Action, Casual e Adventure vêm logo em seguida, com entre 30 mil e 40 mil jogos cada. Esses gêneros são amplos e populares, geralmente oferecendo experiências dinâmicas e acessíveis a um público variado, é altamente acessível a desenvolvedores.

Simulation, Strategy e RPG também têm presença significativa, mostrando o interesse por jogos mais complexos e voltados a públicos que buscam desafios estratégicos ou narrativas elaboradas. Investir em nichos pouco saturados (como gêneros educacionais ou híbridos criativos) pode ser uma forma de se destacar.

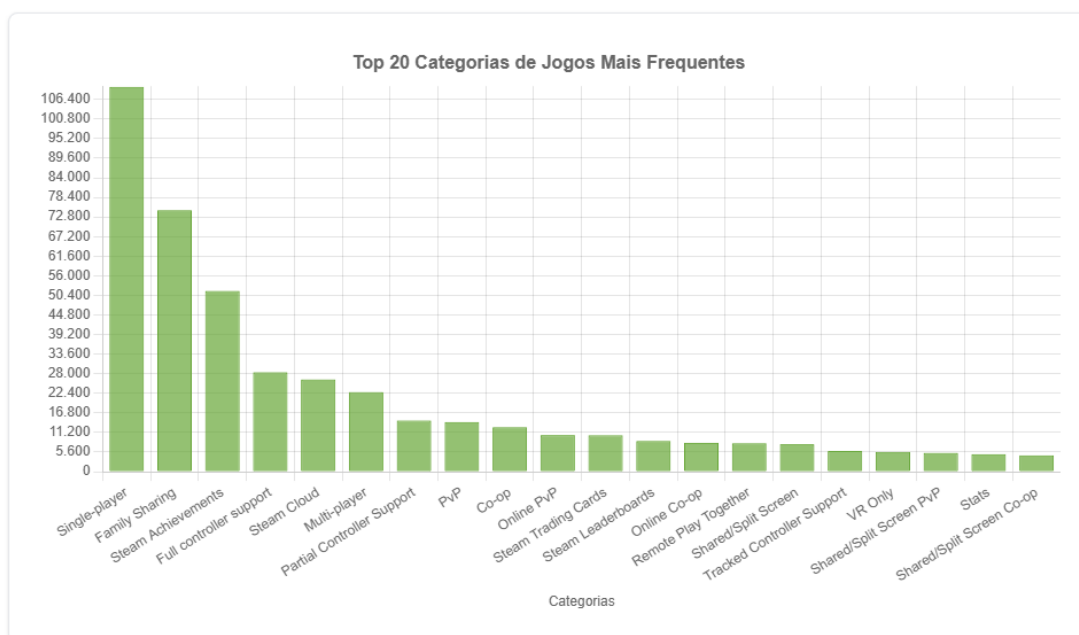


**Figura 11:** Gráfico Barras – Top 20 Idiomas. Fonte: domínio/Games-Insights.html.

Conforme ilustrado na “**Figura 8**”, baseado no número de títulos que oferecem suporte a cada idioma (interface, áudio ou legendas). Ele revela informações relevantes sobre **acessibilidade linguística** e **foco de mercado** da plataforma.

O inglês está presente em cerca de 90 mil jogos, consolidando-se como o idioma-padrão da Steam. Isso é esperado, dado que a maior parte dos desenvolvedores e usuários da plataforma está em regiões onde o inglês é amplamente compreendido ou usado como segundo idioma. Chinês Simplificado e Chinês Tradicional estão entre os idiomas mais presentes, com destaque para o Simplificado (mais de 30 mil jogos). Isso aponta para a importância crescente do mercado chinês, o maior do mundo em número de jogadores.

Idiomas como alemão, francês, russo, espanhol e italiano aparecem com grande presença. Essa diversidade sugere que a Steam busca atender à grande base de usuários da Europa Ocidental e Oriental, com localização frequente de jogos para esses idiomas. O Português - Brasil aparece entre os 10 primeiros, com suporte em aproximadamente 15 mil jogos, indica a relevância do mercado brasileiro dentro da Steam, sendo o principal idioma da América Latina no gráfico.



**Figura 12:** Gráfico Coluna – Top 20 Categorias mais Frequentes. Fonte: domínio/Categories.html

Na “Figura 9”, podemos observar as 20 categorias mais frequentes de jogos na Steam, baseando-se no número de títulos que pertencem a cada categoria. Ele oferece uma boa visão sobre os recursos e modos de jogo mais comuns na plataforma.

A categoria Single-player lidera com folga, com mais de 106 mil jogos. Isso mostra que a maior parte dos jogos da Steam é voltada para experiências solo, o que pode ser reflexo da produção independente e de menor escala, onde desenvolver modos multiplayer é mais custoso.

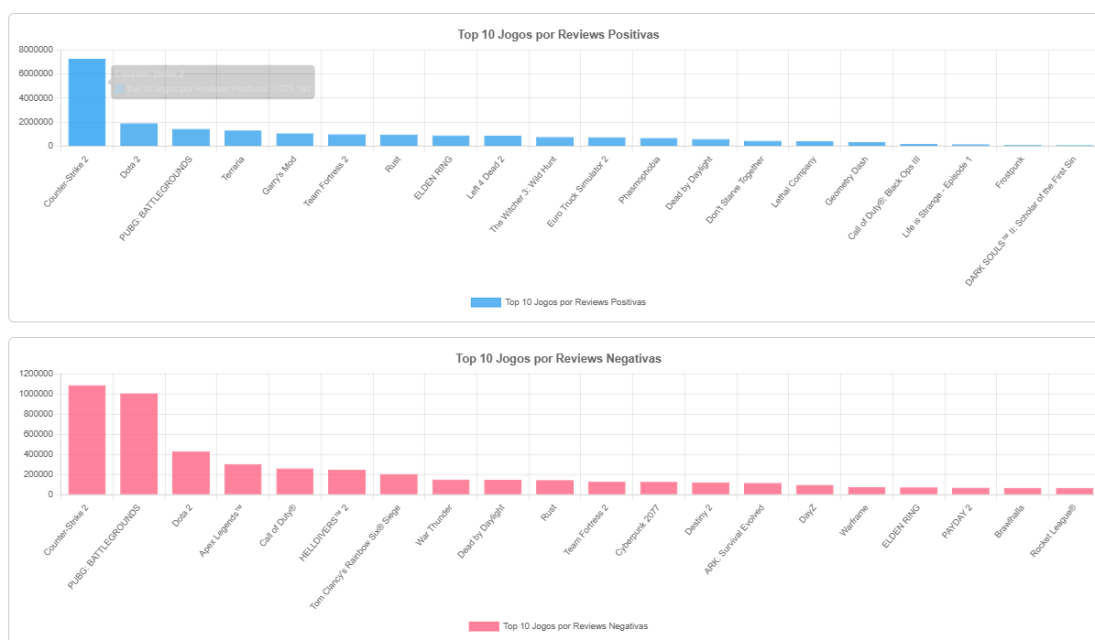
Outras categorias como VR Only (realidade virtual) ou Remote Co-Op aparecem com menor frequência, mas ainda têm representatividade. Stats, Shared/Split Screen e Play Together mostram a diversidade de experiências oferecidas, inclusive para jogos locais.

Multiplayer, PvP, Online PvP e Co-Op estão presentes com frequência, embora bem abaixo de Single-player. Há uma quantidade significativa de jogos que oferecem interação entre jogadores, o que aponta para uma base de títulos com foco social e competitivo.

Steam Cloud e Steam Trading Cards mostram o quanto as funcionalidades de armazenamento na nuvem e colecionáveis são comuns e incentivadas. Steam Leaderboards (placares) também aparecem, refletindo jogos que oferecem ranking entre usuários.

Family Sharing (compartilhamento entre contas familiares) e Steam Achievements (conquistas da Steam) são as próximas mais frequentes. Isso destaca o ecossistema da Steam, que valoriza funcionalidades sociais e de engajamento na plataforma, incentivando a jogabilidade contínua e recompensas ao usuário.

As categorias Full Controller Support e Partial Controller Support aparecem com força. Isso indica que muitos jogos são compatíveis com controles (joysticks), essencial para quem joga em TV ou com setup similar a consoles.



**Figura 13:** Gráfico Coluna – Top 10 Jogos por Reviews. Fonte: domínio//Reviews.html

Esse conjunto de gráficos apresenta os Top 10 jogos da Steam com maior número de reviews positivos e negativos, oferecendo um panorama claro sobre os títulos mais comentados (tanto positivamente quanto negativamente) pela comunidade.

Counter-Strike 2 é disparado o jogo com mais avaliações positivas, ultrapassando 7 milhões de reviews. Outros títulos com altíssima aprovação incluem Dota 2, PUBG: BATTLEGROUNDS e Terraria, Garry's Mod e Team Fortress 2, ambos com longa vida útil e comunidades ativas. Jogos como Rust, ELDEN RING e The Witcher 3 reforçam que tanto títulos multiplayer quanto narrativos de alta qualidade são bem recebidos. A maioria dos jogos é multiplayer ou com forte suporte da comunidade. Alguns títulos são gratuitos ou baratos, o que facilita grande volume de jogadores e reviews.

Counter-Strike 2 e PUBG: BATTLEGROUNDS também lideram em reviews negativas, o que indica altíssimo volume de jogadores, mais chances de insatisfação; Problemas de performance, atualizações impopulares ou bugs recorrentes.

Outros nomes com muitas críticas são, Dota 2, Apex Legends, Call of Duty, Rust e Cyberpunk 2077. Muitos são jogos populares com histórico de polêmicas ou decisões impopulares dos desenvolvedores, títulos como HELLDIVERS 2, Rainbow Six Siege, Destiny 2, e ARK também aparecem com destaque nas críticas. Muitos desses jogos são serviços contínuos (live service) e sofrem com atualizações mal recebidas. Alguns, como Cyberpunk 2077, tiveram lançamentos problemáticos que ainda repercutem.



**Figura 14:** Gráfico Pizza – Distribuição de Jogos por Review. Fonte: domínio//Reviews.html

Este gráfico de pizza mostra a distribuição dos jogos com base em suas notas de avaliação ("Review Score"), e **revela uma** tendência significativa de ausência de nota atribuída. "NaN" (ausente), representa quase metade do gráfico, indicando que muitos jogos não têm review score registrado. Isso pode ocorrer por falta de reviews suficientes, jogos recém-lançados, jogos descontinuados ou pouco populares.

Notas mais frequentes entre os jogos com score:

- Nota 7, 6 e 5 têm participações expressivas.
  - Nota 7: Jogos com boa aceitação.
  - Nota 6: Mediana — pode indicar desempenho ok ou divisivo.
  - Nota 5: Jogos com recepção mista.
- Nota 8, 9, 2 e 1 aparecem em menor quantidade, indicando que extremos (excelente ou muito ruim) são menos comuns.



**Figura 15:** Gráfico Coluna – Top 10 Jogos por Recomendações. Fonte: domínio//Reviews.htm

Counter-Strike: Global Offensive (CS:GO) é o jogo com o maior número de recomendações com proximadamente 4.500.000, destacando-se fortemente em relação aos demais, **PUBG: BATTLEGROUNDS**, Terraria, Garry's Mod, Rust, The Witcher 3: Wild Hunt, ELDEN RING, Entre outros títulos populares como *Cyberpunk 2077*, *HELLO NEIGHBOR*, *Left 4 Dead 2* e *Apex Legends*.

## Conclusão, Resultados e Recomendações:

A análise dos dados da Steam revela um cenário em constante expansão, repleto de oportunidades e desafios para desenvolvedores de jogos. Com base nos gráficos apresentados, é possível extrair conclusões relevantes sobre o comportamento do mercado e propor caminhos promissores para quem deseja se destacar na indústria.

O primeiro dado notável é o crescimento exponencial no número de lançamentos ao longo dos anos. Desde 2014, a Steam passou a registrar um aumento significativo de novos jogos, atingindo mais de 15 mil lançamentos em 2024. Esse crescimento reflete a popularização de ferramentas acessíveis de desenvolvimento, como Unity e Unreal Engine, além da facilidade de publicação por meio de plataformas digitais. No entanto, essa expansão também indica uma maior concorrência e a necessidade de diferenciação por parte dos desenvolvedores.

Ao analisar a distribuição dos jogos por review score, observa-se que uma parcela expressiva dos títulos possui notas baixas ou sequer foi avaliada (classificados como NaN). Isso pode apontar para uma grande quantidade de jogos com baixa qualidade, pouca visibilidade ou ambos. Muitos títulos acabam não alcançando um público relevante o suficiente para gerar avaliações, o que reforça a importância de estratégias de marketing e engajamento com a comunidade.

Outro aspecto relevante é a comparação entre jogos gratuitos e pagos. Apesar do crescimento do modelo free-to-play, os jogos pagos ainda representam a maioria esmagadora na Steam, com mais de 90 mil títulos pagos contra cerca de 17 mil gratuitos. Isso mostra que o modelo tradicional de venda ainda é predominante. No entanto, jogos gratuitos podem ser uma excelente porta de entrada para novos estúdios, permitindo a construção de uma base de usuários antes da monetização, especialmente quando combinados com mecânicas como itens cosméticos, DLCs ou passes de batalha.

No que diz respeito aos tipos de conteúdo, a grande maioria dos títulos disponíveis são jogos completos ("game"), enquanto demos representam apenas uma fração do catálogo. Isso evidencia que os jogadores preferem experiências completas, mas também levanta a possibilidade de que as demos sejam subutilizadas como ferramentas de promoção e testes antes do lançamento oficial de um jogo.

Analizando os jogos mais recomendados da plataforma, destacam-se títulos como *Counter-Strike: Global Offensive*, *PUBG*, *Terraria*, *Rust*, *The Witcher 3* e *ELDEN RING*. Esses jogos compartilham características como multiplayer ativo, mundos abertos, personalização, e/ou narrativas profundas. O sucesso desses títulos mostra que os jogadores valorizam experiências imersivas, rejogabilidade e a possibilidade de interação social dentro dos jogos.

Com base nesses dados, é possível traçar algumas recomendações para desenvolvedores:

1. **Priorize a qualidade e inovação:** Em um mercado saturado, apenas jogos com bom desempenho técnico e diferenciais claros conseguem atrair atenção. Invista em jogabilidade polida, identidade visual marcante e propostas criativas.



2. **Construa e envolva uma comunidade ativa:** Jogos com comunidades engajadas tendem a ter maior longevidade. Modos multiplayer, suporte a mods, fóruns e canais de feedback são ferramentas poderosas para fidelizar jogadores.
3. **Reavalie o uso de demos:** Embora pouco exploradas, as versões demonstrativas podem ser úteis para testar a recepção de um jogo e gerar expectativa pré-lançamento. Além disso, aumentam a confiança do consumidor antes da compra.
4. **Considere modelos gratuitos com monetização inteligente:** O modelo freemium ainda é subaproveitado por muitos estúdios. Ao oferecer um jogo gratuito com conteúdo pagos não-obrigatórios, é possível alcançar uma base ampla de jogadores e gerar receita com base na personalização e conveniência.
5. **Invista em marketing desde o início:** O grande número de jogos sem avaliação demonstra a importância da visibilidade. Promova o projeto nas redes sociais, participe de comunidades de desenvolvimento, divulgue devlogs e, se possível, envolva influenciadores e streamers para ampliar o alcance.

Em suma, o mercado de jogos na Steam oferece um terreno fértil para desenvolvedores com boas ideias e execução sólida, mas exige preparo, planejamento e foco na experiência do usuário. Aqueles que souberem combinar criatividade, estratégia e engajamento com o público terão maiores chances de sucesso nesse ecossistema altamente competitivo.

## 6. Grupo de trabalho.

Nome	Responsabilidade
Sinézio Junior	Front End - Aplicação Web.
Paulo Henrique	Front End - Aplicação Web.
Felipe Silva	Desenvolvimento Intelectual e Documental.
Gabriel Salazar	Back End – Tratamento, Exploração Processamento de Dados.
Bruno Oliveira	Back-End – Desenvolvimento da Arquitetura, Algoritmos

## 7. Link no GitHub.

Para o repositório do Front-End: [GitHub-SteamPoint-Front](#)

Para o repositório do Back-End: [GitHub-SteamPoint](#)

## 8. Considerações Finais.

Este projeto demonstrou como a análise de dados pode ser uma ferramenta essencial para compreender o comportamento do mercado de jogos digitais, especialmente em plataformas como a Steam. Ao explorar dados sobre review scores, tipos de jogos, número de lançamentos por ano, modelos de monetização e recomendações da comunidade, foi possível extrair informações valiosas sobre tendências, preferências dos jogadores e oportunidades de negócio.

A visualização dos dados facilitou a identificação de padrões e gargalos do setor, como a grande quantidade de jogos sem avaliações, a predominância de jogos pagos e o forte impacto da comunidade sobre o sucesso de determinados títulos. Esses insights não apenas revelam a dinâmica atual da plataforma, como também orientam decisões estratégicas para desenvolvedores que desejam alcançar relevância e sustentabilidade em um mercado altamente competitivo.

Como conclusão, reforça-se que a criação de um jogo vai além do desenvolvimento técnico, envolve planejamento de mercado, escuta ativa da comunidade, estratégias de visibilidade e compreensão contínua das demandas do público. Projetos que integram dados desde a fase inicial de concepção possuem maior capacidade de adaptação, diferenciação e sucesso comercial.

Assim, este trabalho não apenas alcança seu objetivo de análise, mas também propõe um olhar mais crítico e fundamentado sobre o processo de desenvolvimento de jogos, incentivando uma cultura baseada em dados e tomada de decisões inteligentes.