

# Por dentro da biblioteca de usuários do Steam

## Analisando a relação entre jogos

Ettore Bruno Mlaker dos Santos  
Universidade Federal do ABC  
Santo André, Brasil  
ettore.bruno@aluno.ufabc.edu.br

Leonardo Ryo Nakagawa  
Universidade Federal do ABC  
Santo André, Brasil  
leonardo.ryo@aluno.ufabc.edu.br

Maria Fernanda Oliveira Silvestre  
Universidade Federal do ABC  
Santo André, Brasil  
oliveira.fernanda@aluno.ufabc.edu.br

Pedro de Souza Tunin  
Universidade Federal do ABC  
Santo André, Brasil  
pedro.tunin@aluno.ufabc.edu.br

**Resumo - O Steam é atualmente a maior plataforma de jogos para PC no mundo, além de ser uma grande loja, também é uma rede social, com aproximadamente 90 milhões de usuários ativos por mês. Esse projeto tem como objetivo a interpretação da relação entre os jogos, analisando as tendências gerais da comunidade e sugerindo assim um sistema de recomendação mais coerente.**

**Foi utilizada a modelagem de grafos para visualizar de uma maneira qualitativa o relacionamento entre os jogos da plataforma. As linguagens de programação Java e Python foram utilizadas para realizar a extração de dados.**

### I. INTRODUÇÃO

O Steam é uma plataforma de distribuição digital de jogos e aplicativos, lançada em 2003 pela empresa Valve Corporation. No seu início, era uma forma de publicar e atualizar os jogos da própria empresa, mas com o passar do tempo, jogos de outras empresas passaram a ser distribuídos pelo Steam. Atualmente, esta plataforma evoluiu, e é muito mais do que apenas uma distribuidora de jogos, pois possui também um sistema de amizades, o qual possibilita a conversa, por meio de texto, jogar, e realizar trocas com seus amigos, além da possibilidade de transmitir os jogos ao vivo, e diversos eventos que ocorrem em determinados períodos, como a Black Friday e o Summer Sale, que oferecem grandes descontos nos jogos.

### II. CONCEITOS BÁSICOS

Com o crescimento do mercado de jogos digitais, amplia-se da mesma forma a necessidade de aprimoramento de serviços e funcionalidades oferecidas ao usuário.

Na plataforma Steam há uma grande quantidade de jogadores ativos, onde cada jogador possui um grande número de jogos. Neste projeto buscamos analisar e quantificar esses dados, a fim

de identificar a relação entre os usuários por meio de seus jogos e também a relação existente entre esses jogos disponíveis.

Uma ferramenta imprescindível para uma plataforma dessas é o sistema de recomendação de jogos, tanto para a empresa que conseguirá vender mais os aplicativos, como para os usuários, que encontrarão jogos de seus interesses. Por isso, neste projeto procuramos uma forma diferente de recomendar as aplicações para os usuários, tendo como base os coeficientes de teoria de grafos, apresentando quais jogos mais se adequam ao perfil do jogadores.

O método atual de recomendação personalizada do Steam se baseia nos jogos das bibliotecas de seus amigos ou os jogos que estes têm interesse, jogos similares aos que o usuário tem jogado, e em conteúdos disponíveis para os jogos que o jogador já possui. Em contraste, o método proposto pelo nosso projeto seria baseado na tendência de um usuário que possui um jogo X, possuir um jogo Y também. Caso essa tendência seja alta, é altamente provável que um outro usuário que possua o jogo X também tenha interesse no jogo Y, independentemente da similaridade entre os jogos.

### III. METODOLOGIA

Para este projeto, primeiramente, foi preciso coletar uma quantidade razoável de ID's dos perfis dos usuários do Steam, sendo 1000 ID's o valor inicial. Estes dados foram obtidos pelo link <https://steamcommunity.com/groups/steamuniverse/memberslistxml/?xml=1&p=1>, de uma coleção de dados feita pela comunidade dos usuários do Steam, e utilizamos um algoritmo para isolar apenas os números dos ID's.

Após coletar os ID's dos jogadores, utilizamos o algoritmo da Figura 1 para coletar os jogos de cada usuário.

```
for i in range(1000):
    aux = input()
    response = requests.get('link da API')
    if len(response.text) <= 15:
        continue
    else:
        resp = json.loads(response.text)
        games_count = resp["response"]["game_count"]
        arq = open("diretório" + aux + ".txt", "w+")
        for x in range(games_count):
            game = resp["response"]["games"][x]
            if game["playtime_forever"] == 0:
                continue
            else:
                arq.write(str(game["appid"]))
                arq.write('\n')
        arq.close()
```

**Figura 1:** Algoritmo em Python 3 usado para extrair os jogos das bibliotecas de cada um dos 1000 usuários iniciais.

O algoritmo funciona da seguinte maneira: Primeiramente é salvo um .json obtido pela api <[<http://api.steampowered.com/IPlayerService/GetOwnedGames/v0001/?key=4C600DB6CDDD9A219350C6FAECBA5575&steamid=\(id\\_usuario\)>](http://api.steampowered.com/IPlayerService/GetOwnedGames/v0001/?key=4C600DB6CDDD9A219350C6FAECBA5575&steamid=(id_usuario))>, então é verificado se o perfil do usuário é privado ou não. Caso for privado, a requisição da API retorna um json vazio, portanto, não é necessário salvar o arquivo. Os jogos com tempo de jogo igual a 0 também não são considerados, pois estes não apresentam nenhuma relevância para o propósito do projeto, além de reduzir consideravelmente o tamanho da amostra de jogos, facilitando o processo. Após filtrar esses casos, os ID's dos jogos são salvos em arquivos de texto individuais para cada usuário.

Em seguida, criamos um algoritmo para montar uma matriz de adjacência de todos os ID's dos jogos, acrescentando 1 na posição para cada usuário que possua dois jogos em comum, como a Tabela 1.

Além disso, utilizamos a biblioteca iGraph do software estatístico R para extrair medidas para melhor entender o grafo.

	Jogo 1	Jogo 2	Jogo 3	Jogo 4
Jogo 1	0	40	567	8
Jogo 2	40	0	10	580
Jogo 3	567	10	0	9
Jogo 4	8	580	9	0

**Tabela 1:** Matriz de adjacência com pesos entre dois pares de jogos, representando a quantidade de vezes que esses jogos aparecem juntos nas bibliotecas dos usuários.

```
For i in range(500):
    read_arq = open("diretório" + files[cont], "r+")
    read = []
    read = read_arq.readlines()
    for x in range(0, len(read) - 1):
        readx = read[x][:-1]
        if readx in dicionario:
            indexx = dicionario.index(readx)
        else:
            dicionario.append(readx)
            indexx = dicionario.index(readx)
        for y in range(x + 1, len(read)):
            ready = read[y][:-1]
            if ready in dicionario:
                indexy = dicionario.index(ready)
            else:
                dicionario.append(ready)
                indexy = dicionario.index(ready)
            graph[indexx][indexy] = graph[indexx][indexy] + 1
            graph[indexy][indexx] = graph[indexy][indexx] + 1
    read_arq.close()
```

**Figura 2:** Algoritmo em Python 3 usado para montar a matriz de adjacência de tamanho (número total de jogos) x (número total de jogos). Algumas partes do código foram omitidas para melhorar a visualização.

#### IV. RESULTADOS

A matriz de adjacência gerada tem um tamanho de 7725 linhas por 7725 colunas (cada linha da matriz representa as ligações que um jogo específico faz com os outros, representados pelas colunas). Logo, com um total de 59.675.625 elementos.

Dado um trecho da matriz de adjacências:

ID Jogo	240	4000	6060
240	0	64	5
4000	64	0	28
6060	5	28	0

**Tabela 2:** Trecho da matriz de adjacência gerada pelo algoritmo da Figura 2.

É possível perceber que jogos mais populares, como Garry's Mod (ID 4000) e Counter-Strike: Source (ID 240), ambos com mais de 10.000.000 de unidades vendidas, aparecem mais vezes juntos, pois muitos usuários os possuem.

Já jogos menos populares, e logo com um número menor de usuários que os possuem, como Star Wars: Battlefront 2 (ID 6060), aparecem menos vezes juntos com outros jogos.

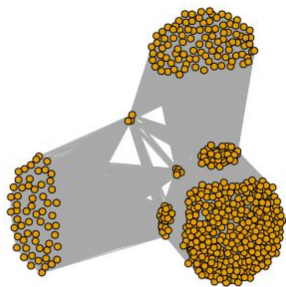
Com isso, é possível saber quais jogos mais aparecem juntos, logo sendo possível saber a preferência de um usuário por determinados jogos, dado que ele possui um determinado conjunto de jogos. Sabendo isso, é possível sugerir com mais eficiência jogos para aquele usuário.

Dado o tamanho do grafo, não foi possível o plotá-lo, porém foi possível extrair algumas medidas, como a ordem e o tamanho do grafo, correspondentes ao número de vértices e arestas.

Ordem	7.725
Tamanho	5.828.055

**Tabela 3:** Medidas de ordem e tamanho do grafo completo.

Para dar uma ideia de como seria a forma final do grafo, diminuimos muito o tamanho dele (em vez de 7.725 jogos e 500 usuários, usaremos 550 jogos e 3 usuários).



**Figura 3:** Plot do grafo com 550 vértices (jogos), extraído do perfil de 3 usuários.

É possível perceber que mesmo diminuindo o tamanho do grafo não faz sentido plotá-lo, pois o mesmo possui muitos vértices e

arestas, não sendo possível extrair nenhuma informação útil do mesmo.

Já com as métricas de redes complexas, é possível extrair muitas informações e entender bem melhor o grafo e seus relacionamentos, mesmo sem vê-lo. As medidas foram retiradas utilizando a linguagem R, com auxílio da biblioteca iGraph. A Tabela 4 apresenta duas delas.

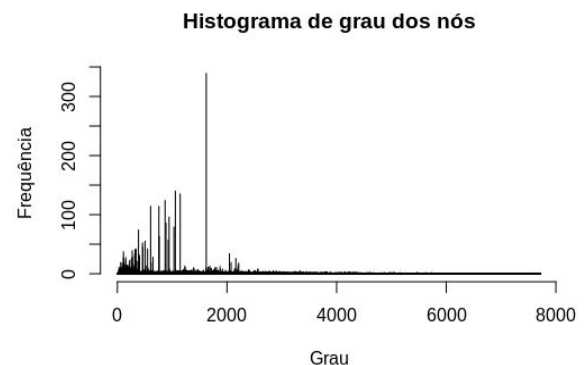
Clustering (Transitividade)	0.5442262
Diâmetro	4

**Tabela 4:** Métricas de redes complexas do grafo.

O clustering, ou transitividade, indica a presença de triângulos na rede. Ou seja, conjuntos de três vértices conectados uns aos outros. Para melhor entender, considere a analogia com uma rede social. Pode-se dizer que se A é amigo de B e C é amigo de C, existem grandes chances de A e C também serem amigos.

Já o diâmetro é o maior dos menores caminhos, ou seja, a partir de cada vértice do grafo é calculado o menor caminho até cada um dos outros vértices. O diâmetro é o maior valor encontrado dentre todos os menores caminhos.

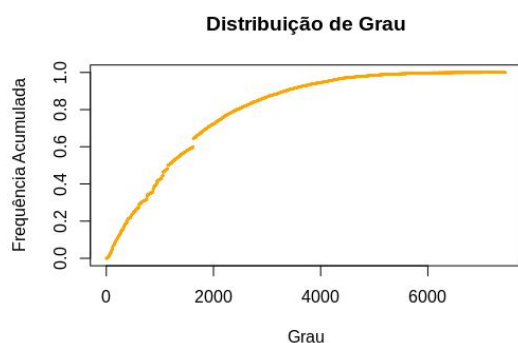
Com isso, é possível afirmar que o grafo é altamente conectado entre si, pois apresenta um alto valor de transitividade e um diâmetro baixo. O diâmetro baixo indica que o grafo tem características de uma rede de mundo pequeno (small world). Em um rede de mundo pequeno, grande parte das conexões são estabelecidas entre vértices mais próximos. Isso acontece até mesmo no grafo da Figura 3, ainda que não represente o grafo total.



**Figura 4:** Histograma da frequência de grau dos vértices.

No histograma da Figura 4, é possível perceber que no grafo muitos vértices possuem um grau baixo, enquanto que poucos vértices possuem um grau alto. Esse comportamento é típico de uma lei de potência. Uma lei de potência, ou Power Law, é usada para descrever duas variáveis onde uma varia como potência da outra.

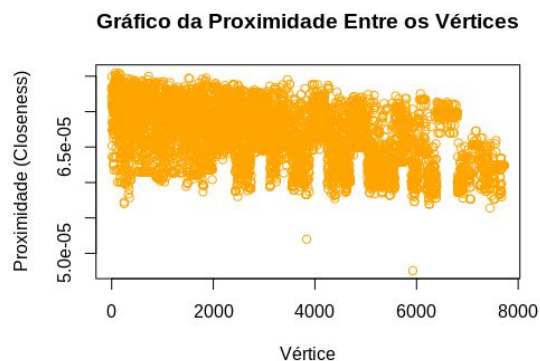
Esse comportamento também pode ser observado na Figura 5. No gráfico é possível perceber que a frequência acumulada cresce muito no começo (onde vértices com grau menores estão), indicando que há muitos vértices com baixo grau, e acaba por se estabilizar no final (onde vértices de maior grau estão), indicando que há poucos vértices com alto grau.



**Figura 5:** Gráfico de frequência acumulada de grau dos vértices.

O grafo estudado demonstrou que seus vértices estão bem conectados uns com os outros. Um indicador disso é o coeficiente usado para medir centralidade, o closeness. Esse coeficiente mede a menor distância média entre um vértice e os outros. Quanto maior o seu valor, maior é a distância média para os outros vértices. É possível observar na Figura 6 que a maioria dos vértices possuem aproximadamente a mesma faixa de valores para o closeness, e que esses valores estão na ordem de  $10^{-5}$ , ou seja, a distância média entre um vértice e os outros é muito pequena, indicando assim uma rede em que os vértices estão muito perto uns dos outros.

Ainda, foi possível perceber que o grafo, além de possuir características de uma rede de mundo pequeno, também possui características de um rede sem escala (scale free). Um dos principais aspectos dessas redes são os hubs. Hubs são vértices que possuem um alto grau de conexões com outros vértices e que servem de caminho para chegar em outras regiões do grafo.



**Figura 6:** Gráfico da proximidade entre os vértices

Na nossa rede, os hubs são jogos muito populares, ou seja, jogos que muitos usuários possuem. Esses jogos estão mais frequentemente na biblioteca dos usuários, logo aparecendo mais vezes com outros jogos. Alguns exemplos de hubs são Grand Theft Auto V (com mais de 10.000.000 de usuários) e Counter Strike: Global Offensive (com mais de 50.000.000 de usuários). No nosso grafo, o vértice com maior grau é justamente o Counter Strike: Global Offensive, confirmando as estatísticas do próprio site do Steam, no qual aparece em segundo lugar de jogos mais jogados.

Com esse conjunto de dados e métricas foi possível entender o funcionamento do grafo e seus relacionamentos.

## V. DISCUSSÃO

Com o progresso dos estudos acerca do grafo elaborado, foi constatado que um grafo não necessariamente segue um modelo específico, podendo transitar entre os atributos de diferentes tipos de rede já conhecidas e estudadas, representando assim suas particularidades e a dificuldade de se determinar a natureza de uma rede diante das possibilidades.

O trabalho aqui proposto trata de um tema da contemporaneidade, seu estudo pode apresentar grande valor para pessoas e organizações relacionadas ao mundo digital, mais especificamente ao mundo dos *games*, pois conhecendo o comportamento e tendências dessa área é permitido a esses indivíduos um melhor planejamento quanto aos próximos passos a serem seguidos.

Com base nesses resultados é possível ter uma prévia de um sistema de recomendação. Um sistema de recomendações mais lucrativo para a

empresa, por exemplo, seria sugerir os jogos mais centrais (hubs), pois se este jogador adquire este jogo, a empresa poderá sugerir uma enorme quantidade de outros jogos relacionados a ele, objetivando um grande lucro futuramente. Outro sistema de recomendações que traria mais benefícios para o usuário, seria indicar jogos que são mais compatíveis com o perfil do usuário. Com base em outros usuários, por exemplo, 9 jogadores possuem os mesmos 5 jogos, mas outro jogador só possui 4 desses jogos, então o sistema recomendaria este jogo que está faltando, assim seria uma recomendação que possivelmente mais adequa-se ao usuário.

## VI. CONCLUSÕES

Utilizando os conceitos, propriedades e métricas dos grafos foi possível obter uma análise quantitativa do grafo desenvolvido, o valor de seus coeficientes, como o coeficiente de clusterização, que é o responsável por indicar o agrupamento do grafo, e o diâmetro, que representa o maior dos menores caminhos entre os vértices.

Em decorrência dos dados adquiridos, o grafo resultante pode ser classificado como uma rede complexa, que apresenta simultaneamente características relacionadas a redes de mundo pequeno, conhecidas popularmente como small-world, e também atributos associados a redes sem escala, como a distribuição dos graus de vértices que seguem uma lei de potência.

A identificação exata do tipo de rede só pode ser feita através de muito estudo sobre o comportamento apresentado pelo grafo, não necessariamente chegando a uma conclusão que se limite a apenas um tipo.

## VII. REFERÊNCIAS

THE IGRAPH CORE TEAM. **R igraph manual pages**. [S. l.], 2015. Disponível em: <https://igraph.org/r/doc/>. Acesso em: 15 jul. 2019.

STEAM Database. [S. l.]. Disponível em: <https://steamdb.info>. Acesso em: 18 jul. 2019.

OGNYANO, Katherine. **Network analysis with R and igraph: NetSci X Tutorial**. 2016. Disponível em: <https://kateto.net/networks-r-igraph>. Acesso em: 20 jul. 2019.