

Será que Roda? Experiências de jogabilidade em diferentes sistemas computacionais

João Lucas P. Oliveira¹, Ariel M. Rodrigues¹, Rafael H. D. Zottesso¹,
Ayslan T. Possebom¹

¹Instituto Federal do Paraná (IFPR) campus Paranavaí
Rua José Felipe Tequinha, 1400. 87703-536. Jardim das Nações.
Paranavaí – PR – Brasil.

Lucas@sergiosouza.com.br, arielmachador3@gmail.com,
rafael.zottesso@ifpr.edu.br, ayslan.possebom@ifpr.edu.br

Abstract. *The popularization of electronic games has increasingly become a way of entertainment among young people and adults. However, the general public does not have advanced technical knowledge of their hardware to verify that a certain game is compatible with their machine. In this work it is presented an alternative to this problem, the implementation of an artificial intelligence algorithm, specifically linear regression, to create a graphical sketch of user experiences. With this data, it is possible to assume a better satisfaction and experience for users when choosing a certain game and running it on their respective computers.*

Resumo. *A popularização dos jogos eletrônicos têm cada vez mais se tornado um meio de diversão entre os jovens e os adultos. Entretanto, o público em geral não possui conhecimentos técnicos avançados sobre seu hardware para verificar se determinado jogo é compatível com sua máquina. Neste trabalho é apresentada uma alternativa para este problema, a implementação de um algoritmo de inteligência artificial, em específico regressão linear, para criar um esboço gráfico das experiências dos usuários. Com estes dados, é possível pressupor uma melhor satisfação e experiência para os usuários ao escolher um determinado jogo e executá-lo em seus respectivos computadores.*

1. Introdução

Conforme o tempo passa, a tecnologia evolui, incluindo os computadores residenciais, que diversas pessoas usam como forma de lazer, na maioria das vezes jogando jogos eletrônicos. A presença dos jogos tem influenciado na cultura popular e na forma com que os dispositivos (em especial, telefones celulares) são usados [BHARAMBE et al. 2018]. Entretanto, esse avanço tecnológico deixa muitas lacunas ao consumidor final, com a criação de diversos modelos de hardware, muitos jogos eletrônicos deixam suas especificações mínimas de requerimento de forma ambígua e genérica, dificultando a aquisição de determinado jogo, visto que o usuário precisa se preocupar com o fato do jogo ser compatível com sua máquina.

Quando se é comparado dois processadores distintos, de marcas diferentes, como por exemplo, um Intel Core i7-8550u vs. AMD Ryzen 7-8550u, para a maioria dos

usuários é um grande desafio conseguir reconhecer qual é melhor, e a maioria recorre a sites como *cpubenchmark*¹, porém há 2 problemas quando falamos desse tipo de comparação: (i) nem todos os jogadores sabem ver qual seu hardware; e (ii) quando se compara em sites como esse, é comparado individualmente, sem contar placas de vídeo em sua somatória, que influenciam diretamente na capacidade de processamento de determinada máquina.

As lojas que comercializam os jogos eletrônicos como a Steam² acabam deixando informações de hardware necessárias para jogos de forma vaga. Por exemplo, se um usuário possui um processador inferior ao mínimo requerido para executar um jogo X, mas possui uma placa de vídeo superior à mínima requerida pelo mesmo, torna a escolha de aquisição de algum jogo difícil, dado que, o jogo ter uma boa performance em sua máquina depende de algo que está além de seu controle.

É de interesse deste público que não possui conhecimento avançado em informática utilizar uma aplicação que demonstre todas as configurações presentes no computador e que, ao selecionar um determinado jogo a ser instalado, esta aplicação possibilita informar ao usuário se ele poderá ter uma boa experiência ou não com base em suas configurações atuais. Desta forma, os usuários apenas comprariam os jogos em que tivessem a certeza de que eles seriam executados de forma satisfatória em seus computadores.

Quando se fala de jogos de computadores, a gama de loja de aplicativos disponível ao usuário é muito limitada, dentre as poucas que existem, é destacado a Steam [STEAM 2022]. Na atualidade, a loja conta com mais de 30 mil jogos em seu catálogo, e cerca de 30 milhões de usuários ativos no presente momento. Através do aplicativo que a Steam disponibiliza, é possível que os usuários conversem entre si, de forma privada, por grupos de voz ou mensagens, enviar arquivos, fotos, e vídeos. Os usuários também podem transmitir suas partidas ao vivo para os demais membros da comunidade e é possível que os jogadores criem modificações para diversos jogos.

De acordo com [LIN et al. 2019], a Steam é uma plataforma de distribuição de jogos digitais. Formada por três principais componentes:

1. **Steam Store:** a maioria dos jogos são pagos, onde este processo de compra e pagamento é realizado diretamente pelo aplicativo da Steam ou por meio de cartões de presente;
2. **Steam Community:** os usuários possuem acesso a uma rede social formada por lista de amigos, podendo manter comunicação com estes usuários e participar de canais exclusivos sobre determinados jogos, e trocar produtos entre as contas;
3. **Steam Client:** os usuários, ao fazerem login no Steam, possuem acesso aos seus jogos, podendo atualizá-los quando necessário e baixar modificações.

Neste sentido, a principal questão motivadora do desenvolvimento deste projeto surge: “Como saber se um determinado computador será capaz de executar um jogo disponível na loja da Steam?”. Diante desta pergunta de pesquisa, este projeto visa desenvolver um software que seja capaz de detectar as configurações de hardware e software do computador utilizado por um usuário *gamer* e, a partir da seleção de um determinado jogo, este software deve ser capaz de apresentar ao usuário se o jogo poderá ou não ser

¹Disponível em <https://www.cpubenchmark.net/>.

²Disponível em <https://store.steampowered.com/>.

executado naquele computador, ou seja, o sistema passa contribuir a área de sistemas de recomendação de jogos digitais [KHOLODYLO and STRAUSS 2019].

Este trabalho está organizado da seguinte maneira. A seção 2 trata da fundamentação teórica para o desenvolvimento da aplicação. A seção 3 apresenta o desenvolvimento da aplicação. A seção 4 trata dos resultados obtidos pela aplicação. Por fim, as conclusões e os trabalhos futuros são apresentados na seção 5.

2. Fundamentação Teórica

Inicialmente foi realizado um levantamento das linguagens de programação que melhor atendiam às necessidades que o projeto demandava, com pesquisa e tentativas de implementação. Com base em conhecimentos prévios, as linguagens Java, C e Python foram testadas, sendo que a linguagem Python foi a escolhida, visto a sua versatilidade e simplicidade de implementação, bem como suas bibliotecas são de fácil uso e interpretação. Essa seção trata sobre as tecnologias e as bibliotecas/frameworks utilizadas para desenvolvimento do algoritmo.

2.1. Linguagem Python

Um dos fatores que influenciam diretamente na escolha desta linguagem é que Python é do tipo *scripting*, que é mais simples que linguagens compiladas como a Java. A linguagem Python também é muito utilizada no mercado de trabalho, esses fatos somados à facilidade de encontrar boas bibliotecas fez com que essa fosse a melhor escolha.

2.2. Escolha do ambiente de Desenvolvimento

Na sessão anterior foi apresentado que Python é do tipo *scripting*, isso significa que não é necessário um programa para “compilar” a linguagem de alto nível para a máquina. Porém ainda sim é necessário usar um interpretador que realizará a tradução dos códigos para a linguagem de baixo nível. Os interpretadores de Python também são comumente chamados de IDLE (*Integrated Development and Learning Environment* ou Ambiente Integrado de Desenvolvimento e Aprendizado). Para o desenvolvimento desta aplicação foi escolhido o Visual Studio Code dado a gratuidade e também a sua gama de extensões que facilitam a programação de sistemas computacionais.

2.3. Bibliotecas e Frameworks

Bibliotecas são um conjunto de subprogramas utilizados no desenvolvimento de softwares. Elas contêm códigos que auxiliam na hora do desenvolvimento. Para a progressão do software é indispensável o uso das mesmas. Um dos principais desafios encontrados era captar as informações de hardware que o usuário possuía, onde até o próprio sistema operacional não mostrava de maneira acessível. Para obtenção das características de hardware do usuário, as seguintes bibliotecas foram utilizadas:

- **Psutil:** psutil ou *process and system utilities*, é uma biblioteca que monitora o sistema mas também consegue dados preciosos do processador do usuário, como as cores, e a frequência, que são indispensáveis para a comparação quando se trata de potência, e também foi captado a quantidade de memória RAM [PSUTIL 2022];
- **Cpuinfo:** é uma biblioteca assim como o Psutil que consegue dados do processador do usuário, informações como a frequência para auxiliar na comparação com processadores de outras marcas [CPUINFO 2022];

- **Gpuinfo:** Essa biblioteca, por sua vez, traz informações e monitoramento da placa de vídeo, que possui menos modelos que o processador. Com ela conseguimos identificar o modelo da placa de vídeo para efeitos de comparação [GPUINFO 2022];
- **Sklearn:** Um framework que implementa um algoritmo que usa inteligência artificial. O algoritmo utilizado é o da regressão linear, que tenta prever o valor de uma variável dependente a partir de um conjunto de variáveis independentes numéricas [SKLEARN 2022].

3. Desenvolvimento

Após recolher os dados de hardware do usuário, foi utilizado um banco de dados do *cpubenchmark.net*, onde havia um ranking com todos os processadores existentes, e também de todas as placas de vídeo. Por meio destes dados, foi possível comparar marcas distintas de hardwares. Com isso, o grande problema se dava ao fato do processador ser inferior, porém a placa de vídeo superior ou vice-versa. Para isso, foi desenvolvido um cálculo básico onde se a placa de vídeo do usuário juntamente com o seu processador for um total de 300 pontos abaixo da soma da placa de vídeo com o processador recomendado ele ainda sim rodaria com os gráficos considerando um no mínimo de hardware requerido. Com a implementação do algoritmo para captar as informações do usuário e atribuir um valor de benchmark, foi possível introduzir a biblioteca sklearn para traçar o gráfico que mostra se a máquina é capaz de executar o jogo ou não.

Inicialmente, algumas bibliotecas devem ser importadas para que os dados do computador possam ser carregados, tais como apresentados em:

```
from distutils.log import error
from doctest import testfile
from os import sep
from pyexpat import model
import string
import pandas as pd
from IPython.display import display
import cpuinfo
```

Para a obtenção dos dados do computador, o código se torna simples. No exemplo abaixo, é possível obter modelo do processador do computador:

```
nomeProcessador = cpuinfo.get_cpu_info()['brand_raw']
```

A base de dados do site *cpubenchmark* foi convertida para um banco de dados MySQL para que consultas na linguagem SQL pudessem ser executadas. Como exemplo, o código abaixo permite obter a frequência do computador conforme configurações recomendadas para a execução do jogo *Counter Strike Global Offensive*.

```
SOCS = "7.0.0.0"
processadorCs = "2400"
corescpuCs = "2"
memoriaCs = 2
discorigidoCs = (15)
```

A partir destes dados, pode-se plotar os gráficos comparativos, utilizando os dados dos jogos (requisitos mínimos, recomendados e máximos) e os dados do computador do usuário para a tomada de decisão e a recomendação da experiência do usuário.

4. Resultados

Após a execução da aplicação, pode-se observar como as conclusões podem ser obtidas por meio da Figura 1. Nela, três pontos são identificados no gráfico, representando as configurações mínima, recomendada e máxima para um jogo. Por meio destes três valores, o algoritmo de regressão linear consegue traçar uma reta indicando a evolução no desempenho do jogo. Na parte superior da figura encontra-se o ranking obtido com o computador do usuário. Observa-se que este ponto está acima da reta da regressão, indicando que o jogo poderá ser jogado de forma satisfatória naquele hardware.

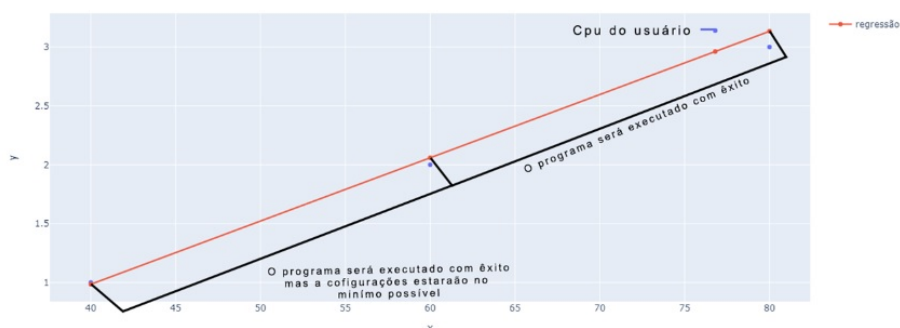


Figura 1. CPU do usuário inserido no gráfico

A Figura 2 apresenta a mesma estrutura apresentada na figura acima. Entretanto, ela lida com dados da GPU, ou seja, da placa gráfica utilizada no computador do jogador. Na figura, observa-se os dados da GPU mínimas, recomendadas e máximas, além da linha de tendência. O ponto superior indica a GPU do usuário. Como ele está acima da linha de tendência, temos a informação de que aquele jogo também poderá ser executado nesta máquina com esta GPU.

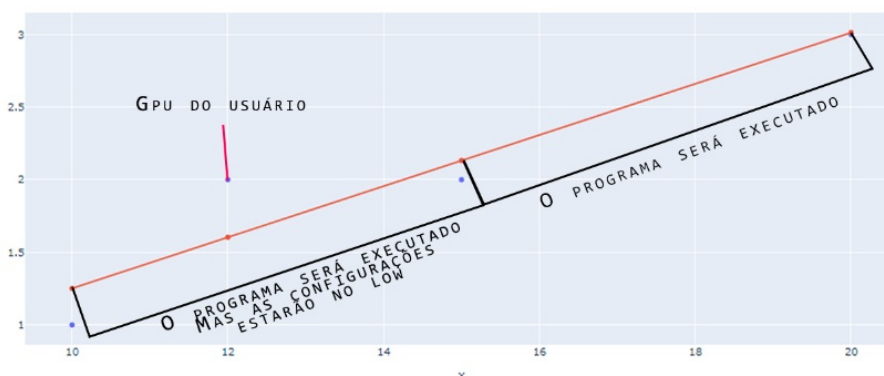


Figura 2. GPU do usuário inserido no gráfico

As figuras acima mostram como a CPU e GPU do usuário se encontram no gráfico mostrando que quanto mais longe do eixo x e y, melhor será a experiência do usuário. A Figura 3 apresenta uma prévia da aplicação desenvolvida. Nela, podemos observar que

os dados do computador do usuário e os dados de um determinado jogo (*Counter Strike Global Offensive*) foram carregados. É por meio destes dados que os gráficos comparativos são gerados e que o usuário poderá observar se sua máquina será compatível ou não com determinado jogo.

Comparando as Configurações...	
Requisitos Mínimos Cs	Configuração do seu Pc
Memoria Ram: 2 GB	Memoria Ram: 8 GB
Cores do Cpu: 2	Cores do Cpu: 4
Frequencia do Cpu: 2400 Ghz	Cpu: 3.4 Ghz
Versão do SO: 7.0.0.0	Versão do SO: 10.0.19041
Espaço em Disco: 15 GB	Espaço em Disco: 223 GB
Qualidade Gráfica: O programa será executado com êxito	

Figura 3. Software desenvolvido exibindo o resultado para o jogo CS GO

5. Conclusões e Trabalhos Futuros

Conforme demonstrado nas seções anteriores, por meio da linguagem Python e das bibliotecas, foi possível obter os dados atuais do hardware do usuário. Em comparação com a base de dados do site cpubenchmark, foi possível extrair informações sobre as recomendações mínimas exigidas pelos jogos de computadores. Por meio da geração dos gráficos, pode-se observar sobre a execução ou não do jogo no computador do usuário e ter uma ideia sobre a questão da usabilidade, caso a pontuação atingida seja inferior que a recomendada.

Como trabalhos futuros, pode-se citar o desenvolvimento de um website que seja capaz de executar esta aplicação de forma online, sem a necessidade de instalação de softwares no computador localmente. Outra questão a ser trabalhada futuramente diz respeito à identificação de cálculos de frames por segundo (fps mínimo, fps máximo, porcentagem de perda de fps, média de fps por um determinado período de tempo). Com esta informação, os usuários poderão decidir se devem ou não investir no jogo desejado, com base nas experiências de jogos de outros jogadores com configurações equivalentes.

Referências

- BHARAMBE, P., SHARMA, S. K., PATIL, A., and JAISWAL, I. (2018). Comparative study of different action fps and puzzle games. *International Journal of Advanced Research in Computer Science*, 9(2).
- CPUINFO (2022). Get cpu info with pure python 2 & 3. <https://pypi.org/project/py-cpuinfo/>. Acessado em: 2022-10-10.
- GPUINFO (2022). A quick access to nvidia gpu information. <https://pypi.org/project/gpuinfo/>. Acessado em: 2022-10-10.
- KHOLODYLO, M. and STRAUSS, C. (2019). What independent game developers expect from recommender systems: A qualitative study. In *Proceedings of the 21st International Conference on Information Integration and Web-based Applications & Services*, pages 563–570.

- LIN, D., BEZEMER, C.-P., ZOU, Y., and HASSAN, A. E. (2019). An empirical study of game reviews on the steam platform. *Empirical Software Engineering*, 24(1):170–207.
- PSUTIL (2022). Cross-platform lib for process and system monitoring in python. <https://pypi.org/project/psutil/>. Acessado em: 2022-10-10.
- SKLEARN (2022). Scikit-learn: machine learning in python. <https://scikit-learn.org/stable/>. Acessado em: 2022-10-10.
- STEAM (2022). Bem-vindo(a) ao steam. <https://store.steampowered.com/>. Acessado em: 2022-10-10.