

# Análise de consumo de memória do Apple Vision Pro para renderização de mapa no Minecraft multiplayer

## *Apple Vision Pro memory consumption analysis for map rendering in Minecraft*

Allyson R E da Silva<sup>1</sup>, Jorge L C de Freitas<sup>2</sup>, Lucas G de Oliveira Silva<sup>3</sup>

Universidade Federal de Pernambuco, Recife, Brasil

Email: {<sup>1</sup>ares, <sup>2</sup>jlcf, <sup>3</sup>lgos}@cin.ufpe.br

**Resumo**—As opções de dispositivos de realidade virtual cresceram, mas o cenário ainda enfrenta falhas de mercado. A gigante Apple lança sua primeira tentativa o Apple Vision Pro, com excelentíssima qualidade de imagem, mas depende da indústria de jogos para evitar o insucesso padrão do segmento. Este artigo investiga se o consumo de memória exigido pelo processo de renderização de mapa do jogo *Minecraft* excede o limite seguro para evitar o superaquecimento no chip M2 com 16GB de RAM incluído no óculos, quando o mesmo é utilizado como servidor local para partidas multiplayer. Foram coletados 500 dados referente ao uso da memória RAM (em MB) por parte do processo específico executado em um aparelho que desempenhava a função de servidor local. Houve em primeira análise uma avaliação em relação a distribuição de dados, a fim de verificar sua normalidade, foi instituído um teste de hipótese com  $H_0$  sendo ela: distribuição é normal. Para buscar evidências foi calculado o p-valor encontrando-se 0.546978321591035, logo assumindo em um intervalo de confiança de 95% um valor maior que  $\alpha = 0.05$ , consequentemente evidenciando a inclinação positiva para a aceitação de  $H_0$ . Houve um segundo teste de hipótese, tratado com teste Z, tal foi realizado para avaliar se a média de utilização de memória por parte do processo de renderização de imagens quando o dispositivo funciona como servidor local seria considerada não otimizada ( $H_0 : \mu \geq 500MB$ ), resultando em z-observável =  $-0.530353166378$  e z-crítico =  $-1.645$  (95% de confiança), indicando que  $H_0$  foi aceita. Concluindo-se que a utilização de memória por parte do processo de renderização de mapa em partidas multiplayer não está otimizado o bastante para prevenir superaquecimentos ao dispositivo servidor.

**Palavras Chave** - distribuição normal, realidade virtual, uso de memória, renderização de imagem

**Abstract**— The options for virtual reality devices have grown, but the scenario still faces market failures. The giant Apple launches its first attempt at the Apple Vision Pro, with excellent image quality, but depends on the gaming industry to avoid the segment's standard failure. This article investigates whether the memory consumption required by the game map rendering process *Minecraft* exceeds the safe limit to avoid overheating on the M2 chip with 16GB of RAM included in the glasses, when it is used as a local server for multiplayer matches. 500 data were collected regarding the use of RAM memory (in MB) by the

specific process executed on a device that performed the function of a local server. In the first analysis, there was an evaluation regarding the data distribution, in order to verify its normality, a hypothesis test was instituted with  $H_0$  being: distribution is normal. To search for evidence, the p-value was calculated, finding 0.546978321591035, thus assuming in a 95% confidence interval a value greater than  $\alpha = 0.05$ , consequently showing the positive inclination towards the acceptance of  $H_0$ . There was a second hypothesis test, treated with Z test, this was carried out to evaluate whether the average memory usage by the image rendering process when the device works as a local server would be considered non-optimized ( $H_0 : \mu \geq 500MB$ ), resulting in z-observable =  $-0.530353166378$  and z-critical =  $-1.645$  (95% confidence), indicating that  $H_0$  was accepted. It is concluded that memory usage by the map rendering process in multiplayer matches is not optimized enough to prevent overheating of the server device.

**Keywords**—normal distribution, virtual reality, memory usage, image rendering

### I. INTRODUÇÃO

As big techs como Google, Samsung e Meta tentaram popularizar a realidade virtual, mas não conseguiram o engajamento necessário. Agora, a Apple, o legado de Steve Jobs, busca ingressar e ter sucesso nesse nicho. O Apple Vision Pro, disponível em versões de 1TB, 512GB ou 256GB, é anunciado como uma promessa para a empresa. Ele utiliza o sistema operacional visionOS e possui um chipset avançado e sofisticado, projetado especificamente para essa finalidade. Apesar do esforço, o histórico das outras organizações, mesmo com altas expectativas, deve ser considerado para contornar mais uma falha no mercado e logo sobressair-se, para isso a Apple visualiza o investimento na indústria dos jogos, a qual apenas em 2023 gerou um lucro maior que US\$ 183,9 bilhões de dólares com um crescimento constante [1], para isso deve-se buscar games os quais atraiam um público fiel e que a temática seja favorável para encantar no VR.

O revolucionário e marcante jogo eletrônico Minecraft da empresa Mojang, atualmente pertencente à grande Microsoft,

pode ser a solução, o sandbox proporciona experiências de imersão significativa, sendo característico por tais fatores. Entretanto, no desenvolvimento da versão para o sistema operacional visionOs deve ter-se como premissa a alta possibilidade de superaquecimento do aparelho diante da combinação de dois chipsets, diante disso analisar a exigência computacional de cada processo é de extrema relevância. Dentre esses processos, um específico tem sido analisado, o qual é responsável pela projeção dos chunks do mapa na perspectiva de realidade virtual de maneira viável para a experiência do jogador, mais precisamente quando a tecnologia desempenha a função de servidor local, ou seja ponto de conexão para outros usuários através da mesma rede, pois precisa verificar-se se cumpre os requisitos de otimização, que apesar de ter como premissa o uso do chip M2, a qual é muito potente com seus 16GB de RAM, não deve-se ultrapassar os limites seguros para evitar o superaquecimento. [2]

Essa análise estatística pode fornecer aspectos pertinentes para o panorama a qual tem construído-se sobre essas tecnologias ainda não consolidadas, expondo algumas preocupações e limitações características que devem ser consideradas por desenvolvedores interessados em produzir softwares para tais plataformas, as organizações idealistas que formalizam o hardware e o sistema operacional e até mesmo os usuários, os quais em alguns anos poderão escolher entre uma gama maior de produtos e considerar atributos próprios.

Este artigo analisa o consumo de memória RAM do processo de renderização de imagem do jogo Minecraft, investigando se tal exige da memória ao ponto de exceder o limite seguro, potencialmente causando superaquecimento no chip M2 com 16GB de RAM do Apple Vision Pro. Especificamente, o estudo foca em cenários onde o aparelho é utilizado como servidor local para partidas multiplayer, logo exigindo maior desempenho, avaliando a viabilidade e os riscos associados ao uso intenso de memória nesse contexto.

Os dados foram coletados durante a execução do jogo com o servidor sendo acessado por um segundo jogador, durante aproximadamente 1 hora enquanto ocorria o processo de renderização da imagem do cenário, sendo por fim analisado estatisticamente o uso de memória pelo mesmo, a fim de avaliar o respeito ao consumo de RAM partindo desse.

Na Seção II deste artigo, a metodologia para tal investigação é apresentada, incluindo a ficha técnica do Apple Vision Pro, a realização da coleta de dados, as ferramentas utilizadas para análise e as condições dos testes de hipóteses. A análise descritiva dos dados e a visualização por meio de gráficos como boxplot, histograma e gráfico de linhas estão na Seção III. Ainda nesta seção da discussão dos resultados, foram conduzidos dois testes de hipóteses: o primeiro para verificar a normalidade da distribuição dos dados e o segundo para avaliar o consumo de memória do processo de renderização de imagem em relação o limite de segurança. A Seção IV traz as conclusões do trabalho. Por fim, a Seção V apresenta o dataset e o repositório criado para o desenvolvimento do projeto.

## II. METODOLOGIA

### A. Configurações do Apple Vision Pro

O novo aparelho da Apple traz um chip M1 que disponibiliza uma memória unificada de 16GB de RAM através de 8 núcleos para CPU, sendo 4 deles para tarefas intensivas e outros 4 para processos de menor demanda, além de outros 10 núcleos para GPU, reservados para processamento gráfico e por fim, um motor neural com 16 núcleos dedicado a IA e visão computacional (reconhecimento de imagem). Há ainda um segundo chip, o R1, para lidar com o sistema de câmeras, sensores, mapeamento 3D, microfones e entre outros componentes que exigem processamento quase instantâneo, logo possuindo 256GB de RAM. [3] [4] Apesar do alto desempenho assegurado com tal hardware há desvantagens em dispôr desses componentes sofisticados: O uso de bateria e consequentemente a necessidade de uma alta dissipação térmica [5] [2]. O produto pode ser utilizado apenas mediante o acoplamento de uma bateria externa conectada via cabo ao aparelho. O consumo exigido de energia é tão considerável que o produto oferece menos de duas horas de uso, logo sendo problemático diante das atividades que propõe-se (acompanhar eventos esportivos, filmes e jogar), assim os desenvolvedores terão que se preocupar consideravelmente com o uso dos componentes para contornar superaquecimentos.

### B. Coleta de dados

O jogo foi executado em um aparelho desempenhando a função de servidor local e sendo acessado por um segundo jogador. Durante aproximadamente 41 minutos e 40 segundos de execução foi coletado o uso de memória RAM a cada cinco segundos (pausa necessária para analisar o ápice da exigência evitando diminuições advindas do armazenamento de memória cache) pelo processo de renderização da imagem do cenário. O cenário não estava previamente carregado e era suficientemente grande em informações, além de ser constantemente explorado e estando nas configurações ideais. Foi observado se a experiência do usuário não era prejudicada por delays da imagem em relação ao acompanhamento do movimento, sendo por fim analisado estatisticamente o uso de memória RAM partindo deste processo em tais condições, a fim de avaliá-lo para classificá-lo como otimizado o suficiente para precaver superaquecimento ou não.

### C. Ferramentas utilizadas

A análise estatística desses dados foram manipuladas por várias ferramentas, incluindo aplicações web e instaladas, além de uma variedade de bibliotecas Python, assim trazendo medidas de dispersão, posição e outros cálculos juntamente. Para plotagem e visualização de dados foram utilizadas as bibliotecas Matplotlib e Seaborn, para problemas matemáticos usufruiu-se da biblioteca SciPy combinada com módulos das bibliotecas NumPy, essas usadas principalmente para o cálculo das medidas de centralidade e dispersão. Além dessas ferramentas na linguagem Python, na qual foi consolidada um arquivo "notebook" para melhor visualização desse uso, também houve auxílio do Minitab para plotagem de gráfico de

distribuição e StatDisk para confirmar as medidas facilmente relacionado a intervalos de confiança especificados.

#### D. Teste de hipótese I - Distribuição normal

Os dados coletados é evidenciado como uma possível distribuição normal através notoriamente do gráfico Q-Q (quantile-quantile plot), além disso em aspectos gráficos enfatiza-se o desenho poligonal obtido pelo comportamento do histograma de frequência, sendo tal similar a um sino. A função reservada da biblioteca de Python, a SciPy, *normaltest* também foi utilizada para calcular-se o p-valor comparando com um nível de significância de 0.05, sendo explorado a  $H_0$ : A distribuição é normal e  $H_a$ : A distribuição não é normal. Conforme informa a documentação da biblioteca, o algoritmo usufruiu-se principalmente da medida de curtose e assimetria, assim realizando um teste D'Agostino e Pearson para encontrar o p-valor [6]. Por fim, a comprovação ressaltase ainda através do uso da ferramenta online do StatDisk com taxas de confiança estabelecidas em 95% e 99% em um teste de Ryan-Joiner.

#### E. Teste de hipótese II - Consumo seguro de memória RAM por parte de processo de renderização

O consumo da memória RAM advindo do processo de renderização de imagem foi estudado por meio da média amostral, em um intervalo de confiança de 95%, e utilizando módulo *math* da biblioteca Scipy. O Teste Z foi utilizado, pois o número amostral é suficientemente abrangente, ou seja superior a 30, logo as medidas podem assumir proximidade significativa da população diante do fato de ser uma distribuição normal, foi comparado o valor do z-observável calculado com o z-crítico, de modo a aceitar o  $H_0$  ou rejeitá-lo, consequentemente aceitando a hipótese alternativa. A hipótese nula define-se no como: ( $H_0 : \mu \geq 500MB$ ), ou seja a média de utilização de memória por parte do processo de renderização de imagens quando o dispositivo funciona como servidor local seria considerada **não** otimizada por desrespeitar o limite seguro contra superaquecimentos (500MB), reverberase como hipótese alternativa ( $H_a$ ), justamente  $\mu < 500$ , ou seja não alcançar ou extrapolar o limite médio seguro do consumo de RAM por parte do processo.

### III. RESULTADOS E DISCUSSÃO

#### F. Análise descritiva dos dados

Os dados coletados trata a variável X, sendo o consumo de memória RAM por parte do processo de renderização, como continua, isso pois seu intervalo apresenta tal natureza diante do contexto e ao assumir valores x em um intervalo [311.691, 668.172]. As medidas de centralidade e dispersão foram calculadas a fim de estudar o comportamento dos dados e investigar hipóteses. Além disso as mesmas são úteis para os algoritmos das bibliotecas utilizadas para construção dos gráficos e medição do p-valor. As medidas foram registradas na Tabela 1. Enfatiza-se que a média adequa-se ao intervalo com nível de confiança de 99% em relação a média populacional, conforme apresentado na Figura 1, tal

sendo obtida através do StatDisk Online.

TABELA 1. Medidas de tendência central e dispersão.

Medida	Valor
Média	498.751298
Mediana	499.3625
Desvio padrão	52.6476079082242
Coeficiente de variância	10.555883888290994%
1º Quartil	463.721
2º Quartil	499.3625
3º Quartil	532.2655
4º Quartil	668.172
IQR (Intervalo entre quartis)	68.54449999999997
Limite inferior	360.90425000000005
Limite superior	635.08224999999999
Quantia de outliers	3
Valor mínimo	311.691
Valor máximo	668.172
Curtose	0.19438793618208416
Assimetria de Fisher	-0.05198264539374111

Nível de confiança:  
0.99  
Tamanho da amostra, n:  
500  
Média da amostra:  
498.751298  
Desvio padrão da amostra, s:  
52.6476079082242  
  
Margem de erro, E = 6.08800  
  
99% de confiança que a média populacional está no seguinte intervalo:  
492.66330 <  $\mu$  < 504.83929

Fig. 1. Resultado de intervalo de confiança da média para um  $\alpha = 0.01$

Ao analisar o histograma apresentado na Figura 2, é notório observar que provavelmente a distribuição dos dados é normal, visto que os valores estão distribuídos com significativa simetria, inclusive probabilidade que é ressaltada com o valor da assimetria presente na Tabela 1 (-0,05198264539374111). Além disso, o polígono da frequência apresentando em contorno linear assemelha-se a um sino, característica comum das distribuições normais, devido a simetria.

Com a Figura 3 verifica-se que os dados apresentam uma dispersão simétrica em torno da mediana, apesar da existência de alguns outliers. É válido ainda ressaltar a relevância de observar a variação dos dados, por tratar-se de consumo de memória RAM, sendo normal haver variações, por depender da quantia de informações que está sendo processada naquele instante.

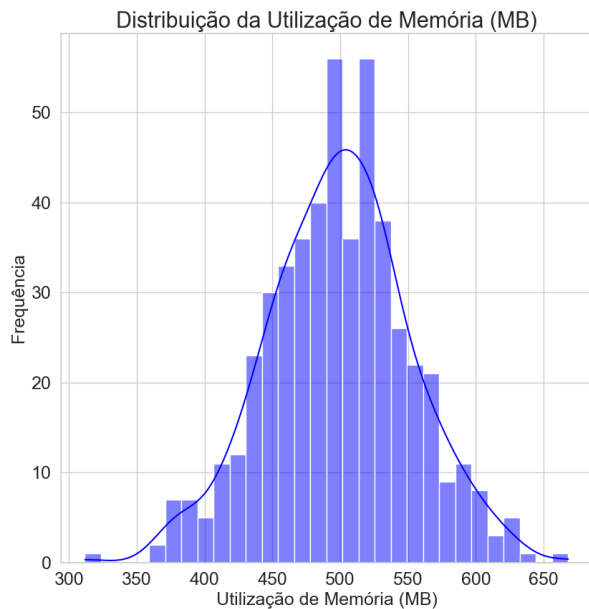


Fig. 2. Histograma - Consumo de memória (MB) x Frequência

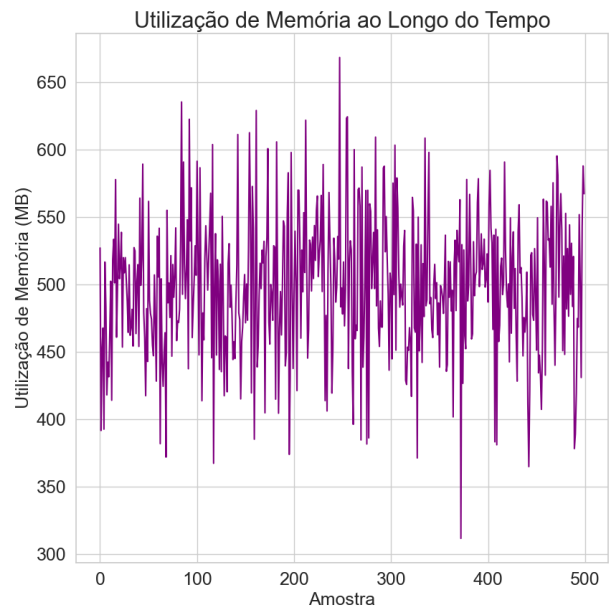


Fig. 4. Gráfico temporal do consumo de memória (MB)

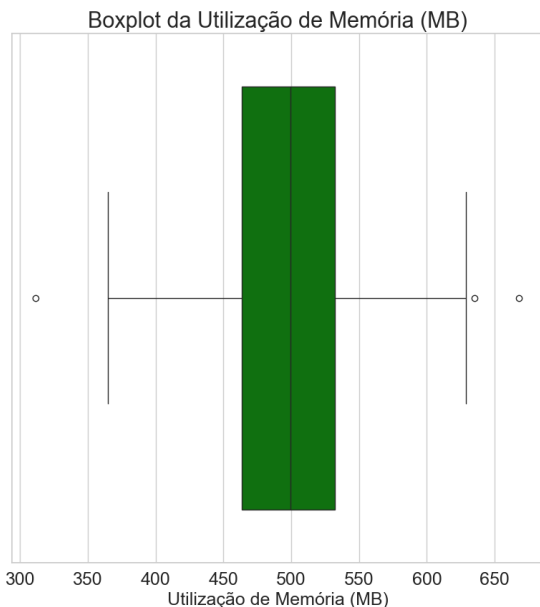


Fig. 3. Boxplot - Consumo de memória (MB)

Diante disso, apresenta-se a Figura 4 representando as variações do consumo em relação ao tempo de observação. Ressalta-se que o tempo no contexto apresenta-se como a própria sequência de amostragem por estar ordenada conforme a coleta, logo apontar a utilização de memória (MB) na amostra de número 100, trata-se de indicar o uso de memória

no tempo em que a amostra 100 foi coletada ou simplesmente o "tempo 100".

#### G. Teste de hipótese 1 - Distribuição normal

A normalidade dos dados pode ser investigada através da função *normaltest* nativa da biblioteca SciPy, a qual tem como finalidade calcular o p-valor, além disso foi realizado, a fim de observar o comportamento dos dados, a plotagem do gráfico Q-Q (quantile-quantile) como pode ser observado na Figura 5.

A Figura 5 apresenta um gráfico Q-Q, o qual compara os quantis teóricos de uma distribuição, no contexto sendo a normal, com os quantis observados, nesse caso os dados relacionados ao consumo de memória RAM por parte do processo de renderização de imagem, logo quanto maior a adesão dos pontos de relação sobre a reta de 45°, mais evidencia-se a distribuição indicada diante da similaridade. É nitido a proximidade ao observar-se majoritariamente os pontos sobre a reta referenciada, trazendo apenas outliers como exceções significativas as quais escapam da reta de distribuição normal.

O valor-p foi calculado, considerando a Hipótese Nula ( $H_0$ ): A distribuição é normal e a Hipótese Alternativa ( $H_a$ ): A distribuição não é normal. O valor-p encontrado foi de 0.5469783215910351, o qual é válido para um intervalo com confiança de 95%, por ser maior que o valor de complemento de 0.05, ou seja a significância ( $\alpha$ ). Portanto, isso significa que a amostragem tolerando apenas 5% de chance de cometer um erro do tipo I (rejeita-lá ainda que seja verdadeira), logicamente contendo 95% de assertividade, apresenta uma probabilidade de  $\approx 54.7\%$  de ao acaso obter um resultado

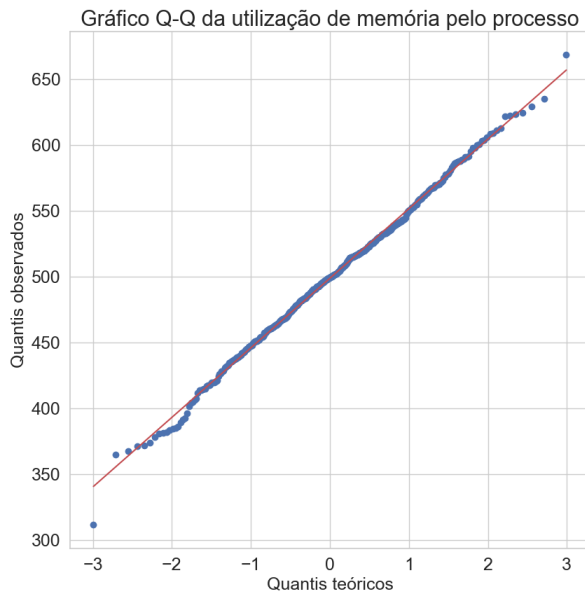


Fig. 5. Gráfico quantile-quantile

similar. Pode-se inferir que há normalidade na distribuição da base de dados, aceitando a Hipótese Nula ( $H_0$ ).

A distribuição normal evidencia-se também em primeiro momento pela simetria dos dados expostas significativamente no histograma, apresentando um formato característico em seu desenho poligonal de frequência, sendo tal similar a um sino. Por fim, as investigações solidificam-se ainda mais no uso do teste de Ryan-Joiner através da ferramenta online StatDisk, a qual oferece um teste avaliativo para conferir a normalidade com tamanha precisão, o resultado está na Figura 6.

Teste Ryan-Joiner	
Estatística de teste, Rp:	0.99874
valor crítico para nível de significância de 0,05:	0.99700
valor crítico para nível de significância de 0,01:	0.99600
Sem evidências para rejeitar normalidade com 0.05 de significância;	
Sem evidências para rejeitar normalidade com 0.01 de significância;	
Possíveis valores discrepantes:	
Número de valores de dados abaixo do Q1 em mais que 1,5 IQR:	1
Número de valores de dados abaixo do Q3 em mais que 1,5 IQR:	3

Fig. 6. Teste de Ryan-Joiner pelo StatDisk com  $\alpha = 0.01$  e  $\alpha = 0.05$

Observa-se que há poucos valores discrepantes e há confirmação de uma aprovação em um intervalo com 99% de confiança, mas há tanta notoriedade para a hipótese que como foi apontando pela função do SciPy no intervalo mais restrito com 95% de confiança já é evidenciado que encaixa-se a hipótese nula como verdadeira.

## H. Teste de hipótese II - Consumo seguro de memória RAM por parte de processo de renderização

Foi realizado um teste de hipótese para investigar se consumo médio da memória RAM (MB) por parte do processo de renderização de imagem CPU respeita o limite seguro, que é 500MB, para um intervalo de 95% de confiança. Dessa forma, foi considerada a Hipótese Nula ( $H_0$ ) sendo  $\mu \geq 500$  e a Hipótese Alternativa ( $H_a$ ):  $\mu < 500$ , busca-se quando trata-se de otimização ter evidência para rejeitar  $H_0$  e aceitar a  $H_a$ . Para esse estudo estatístico foi realizado um Teste Z, pois o número de amostras é superior a 30 e a distribuição é normal.

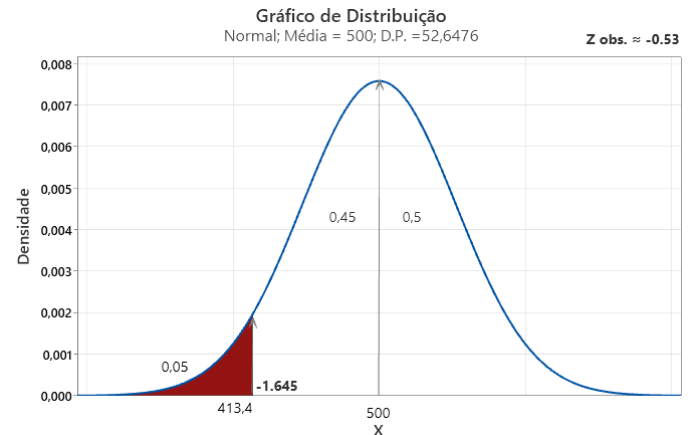


Fig. 7. Gráfico de distribuição com probabilidade de um teste unilateral à esquerda produzido no Minitab

A formulação de hipótese baseou-se em trazer o intuito de rejeição para  $H_0$ , ou seja:

$$\begin{cases} H_0 : \mu \geq 500 \\ H_a : \mu < 500 \end{cases}$$

Assim, o aspecto de otimização dá-se ao rejeitar  $H_0$  e consequentemente aceitar a hipótese alternativa, a qual apresenta tal cenário. Analisando o operador lógico, sendo  $>$ , da hipótese alternativa identifica-se para o Teste Z a necessidade de um gráfico de distribuição com a probabilidade de uma área de significância ( $\alpha$ ) exposta na cauda esquerda e unilateral como na Figura 7 produzida no Minitab. A área restrita, a região crítica na esquerda é delimitado pelo valores que apresentam-se na distribuição em uma frequência abaixo de 5% (nesse caso esses valores estão abaixo de 413,4), pelo fato da confiança tratada ser de 95%, ou seja a área em vermelho em rejeição equivale ao 0,05, justamente sendo o complementar da taxa de confiança, sendo encontrando em tal região valores que escapam da confiabilidade consolidada, possíveis apontamento para erros.

Ela está delimitada pelo z-crítico que é encontrado através da tabela normal (A tabela normal utilizada está referenciada [7]) através da observação do valor que equivale a área entre a parte antecessora da região crítica até a marcação do meio do gráfico, logo sendo 0,45, buscando tal valor

na tabela encontra-se um z-crítico de -1.645, valor também reafirmado pelo StatDisk Online em sua ferramenta de teste de hipótese [8]. Com o z-crítico delimitando a área restrita, deve-se calcular o z-observável a fim de que analise-se sua possível posição no gráfico. O z-observável pode ser obtido também facilmente pelo StatDisk (Figura 8), ressaltando o fato de que no cálculo o desvio padrão amostral pode ser utilizado como uma aproximação suficiente do populacional devido ao número de amostras e a distribuição ser normal, além disso a medida também foi verificada em nível de confiança pela ferramenta.

Significância: 0.05  
Média reivindicada: 500  
Desvio Padrão: 52.6476079082242

Hipótese Alternativa:  
 $\mu < \mu(hip)$

Teste Z  
Z observável: -0.53035  
Z crítico: -1.64485

Fig. 8. Resultados do teste de hipótese no StatDisk

Recorrendo aos cálculos sem uso de ferramentas, deve atentar-se que assumirá tal expressão:

$$z = \frac{\bar{x} - \mu(hip)}{\frac{s}{\sqrt{n}}}$$

Sendo  $\mu(hip) = 500$ . Com tais cálculos foi encontrado o valor z-observável = -0.530353166378 e z-crítico = -1.64485, sendo o nível de 95% de confiança estabelecido, o que indica que a  $H_0$  deve ser aceita, pois o valor z-observável, sendo maior que o z-crítico estaria no gráfico fora da área delimitado pelo mesmo e equivalente à 0,05, distanciando-se dessa significância, essa chance de cometer-se um erro tipo I. Assim, a  $H_0$  sendo aceito, a hipótese alternativa a qual descreve um cenário de possível otimização do processo investigado deverá ser rejeitada, sendo possivelmente  $\mu \geq 500$ . Logo, para um nível de 95% de confiança, a média do uso de memória por parte do processo de renderização de imagem do jogo Minecraft inclina-se para estar acima ou igual à 500MB, conforme as evidências apresentadas que apontam a forte possibilidade de estar acima desse limite seguro ou alcança-lo, podendo portanto ser considerada não otimizada. De tal modo que de acordo com o StatDisk, aponta-se o valor

de média não incluindo 500 apenas a partir do intervalo de confiança de 40% (Figura 9):

Alternative Hypothesis:  
 $\mu < \mu(hyp)$

Teste Z  
Z observável: -0.53035  
Z-crítico: -0.52440  
P-valor: 0.29793

Intervalo de confiança de 40%:  
497.51661 <  $\mu$  < 499.98598

Fig. 9. Resultados do teste de hipótese no StatDisk com significância diferente

Esse comportamento ressalta-se diante das medidas observadas na Tabela 1, pois aproximam-se significativamente de 500 podendo indicar inclinação para tal e para valores superiores quando o número de amostras fosse maior ou até mesmo caso trata-se de um censo populacional.

Com os valores denotados em Z-observável/teste e Z-crítico é possível através das bibliotecas em Python gerar o seguinte gráfico esclarecedor da Figura 10.

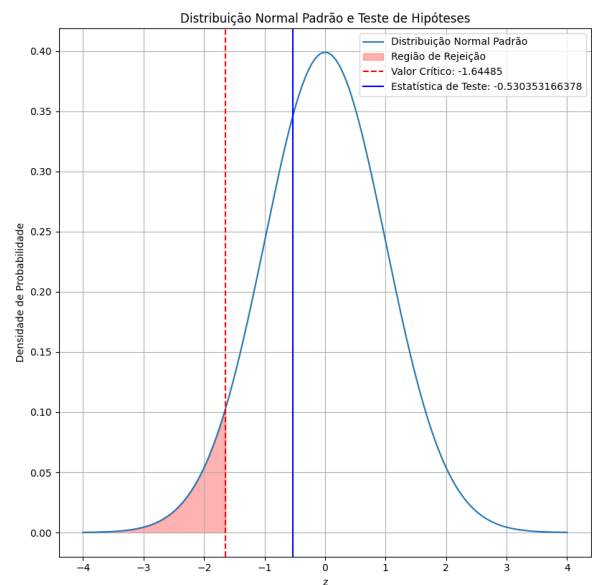


Fig. 10. Gráfico de distribuição com probabilidade de um teste unilateral à esquerda produzido pelo Matplotlib com Numpy e Scipy

#### IV. CONCLUSÃO

Portanto, é possível concluir que o uso médio de memória RAM por parte do processo de renderização de imagem exigido para geração do mapa no jogo Minecraft executado no Apple Vision Pro enquanto o mesmo é utilizado como servidor local para partida multiplayer possivelmente é não otimizado o bastante para que haja respeito ao limite seguro de 500MB a fim de evitar superaquecimento diante dos componentes do chipset. Na Figura 11 é possível observar integralmente o procedimento necessário para alcançar as conclusões apontadas.

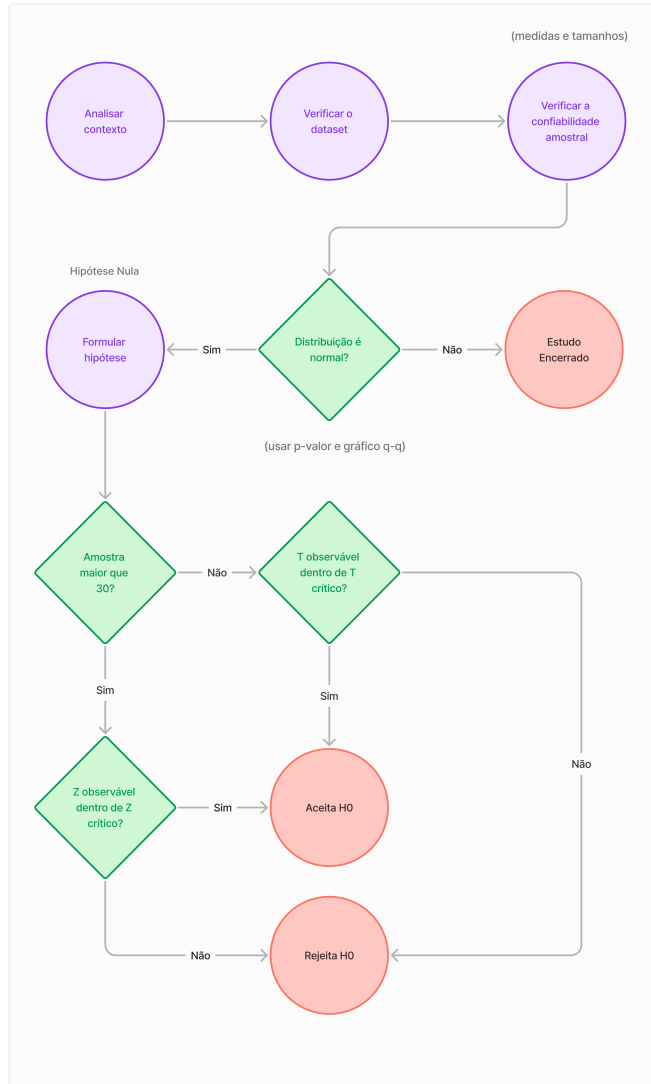


Fig. 11. Fluxo relacionado a análise dos dados e formulação de hipóteses.

#### V. ANEXOS

O material do projeto pode ser acessado no repositório: [github.com/AllysonRyanE/Estatistica-e-Probabilidade-Computacional](https://github.com/AllysonRyanE/Estatistica-e-Probabilidade-Computacional)

#### REFERÊNCIAS

- [1] Newzoom, "Global games market 2023," Disponível em: <https://newzoo.com/resources/trend-reports/newzoo-global-games-market-report-2023-free-version>. Acesso em: 15 de julho de 2024, 2024.
- [2] TudoCelular, "Apple vision pro: headset traz chip r1 que processa entrada de 12 câmeras," Disponível em: <https://www.tudocelular.com/mercado/noticias/n207210/apple-vision-pro-headset-traz-chip-r1-que-processa-entrada-de-12-cameras.html>. Acesso em: 15 de julho de 2024, 2023.
- [3] Apple, "Apple vision pro," Disponível em: <https://www.apple.com/apple-vision-pro/specs/>. Acesso em: 14 de julho de 2024, 2023.
- [4] MundoConectado, "Vision pro possui uma ram personalizada de baixa latência com o chip apple r1," Disponível em: <https://www.mundoconectado.com.br/vision-pro-possui-uma-ram-personalizada-de-baixa-latencia-com-o-chip-apple-r1>. Acesso em: 14 de julho de 2024, 2023.
- [5] CanalTech, "Apple vision pro - ficha técnica," Disponível em: <https://canaltech.com.br/produto/apple/vision-pro/>. Acesso em: 14 de julho de 2024, 2024.
- [6] SciPy, "Biblioteca scipy," Disponível em: <https://scipy.org/>. Acesso em: 15 de julho de 2024, 2024.
- [7] Conrad E. Pinheiro, Professor Guru, "Tabela normal," Disponível em: <https://www.professorguru.com.br/estatistica/tabela-normal-padrao.html>. Acesso em: 16 de julho de 2024, 2024.
- [8] StatDisk, "Statdisk," Disponível em: <https://www.statdisk.com/>. Acesso em: 14 de julho de 2024, 2024.