Comparação de Ferramentas para Dúvidas em Desenvolvimento de Software: ChatGPT vs Stack Overflow - Análise da Qualidade do Código

Davi S. Silva[1], **Kleyann M. Barros**[1], **Maria E. M. Miranda**[1], **Rafael D. Pereira**[1], **Samuel R. de Freitas**[1]

[1]Engenharia de Software – Pontifícia Universidade Católica de Minas Gerais (PUC MG)
302140-002 - Belo Horizonte - MG - Brasil

{davi.santos, kleyann.barros, maria.miranda.1183021}@sga.pucminas.br

{rafael.pereira.1296852, samuel.freitas}@sga.pucminas.br

Abstract. *Stack Overflow is an online QA platform for programmers, while ChatGPT is an artificial intelligence (AI) based language model designed to interact conventionally and provide a detailed answer. The purpose of this analysis is to provide an overview of the ChatGPT tool compared to the Stack Overflow platform, in order to help developers choose the best option to solve their questions related to the quality of the generated code. For this, using the API of the tools, 2.000 questions were extracted and analyzed, where readability, efficiency and maintainability were evaluated. Finally, it is not possible to state categorically that one tool is superior to the other.*

Resumo. *O Stack Overflow é uma plataforma online de perguntas e respostas para programadores, enquanto o ChatGPT é um modelo de linguagem baseado em inteligência artificial (IA) projetado para interagir convencionalmente e fornecer uma resposta detalhada. O objetivo desta análise é fornecer uma visão geral da ferramenta ChatGPT em comparação com a plataforma Stack Overflow, com o intuito de auxiliar os desenvolvedores na escolha da melhor opção para solucionar suas dúvidas relacionadas à qualidade do código gerado. Para isso, utilizando a API das ferramentas, foram extraídas e analisadas 2.000 perguntas, onde foram avaliadas a legibilidade, eficiência e manutenibilidade. Por fim, não é possível afirmar categoricamente que uma ferramenta é superior que a outra.*

1. Introdução

O **Stack Overflow**[1] é uma plataforma online de perguntas e respostas para programadores profissionais, estudantes, ou pessoas interessadas em tecnologia e desenvolvimento, com um amplo alcance e acesso aberto a todos os usuários. Devido à natureza colaborativa, os códigos oferecidos podem não seguir boas práticas de escrita, uma vez que qualquer pessoa pode contribuir, resultando em uma possível baixa qualidade do código. Por outro lado, tem-se o **ChatGPT**[2], um modelo de linguagem baseado em inteligência artificial projetado para interagir de forma convencional, treinada para seguir instruções fornecidas

[1]Disponível em: https://stackoverflow.com/. Ultimo acesso: 18 de jun. 2023
[2]https://openai.com/blog/chatgpt/

em um *prompt* e prover aos usuários uma resposta detalhada, representando assim uma fonte confiável de informações.

Neste contexto, ao comparar as duas ferramentas é importante considerar os pontos fortes e as características de cada uma delas. O Stack Overflow é uma comunidade ativa e diversa, com um grande volume de perguntas e respostas disponíveis. Os usuários têm a oportunidade de votar nas respostas de outros usuários com base em sua qualidade percebida, o que permite que as respostas mais bem votadas ganhem maior visibilidade. Ou seja, as respostas com maior número de votos são consideradas mais confiáveis e eficientes. Por sua vez, o ChatGPT, como uma inteligência artificial tem a vantagem de oferecer respostas consistentes e detalhadas, seguindo boas práticas de codificação, visto que, foi codificado para isso.

A motivação por trás deste estudo está em auxiliar os desenvolvedores na seleção da ferramenta mais adequada para solucionar dúvidas relacionadas a código e desenvolvimento. Atualmente, o Stack Overflow é amplamente reconhecido como a principal plataforma onde os desenvolvedores podem encontrar soluções para seus problemas de programação. No entanto, recentemente tem-se observado o crescimento do ChatGPT, uma ferramenta que vem ganhando forças rapidamente nessa área [Reuters 2023]. Mas por se tratar de uma nova tecnologia, ainda carece de estudos abrangentes que comprovem sua eficácia e confiabilidade em comparação ao Stack Overflow ou outras ferramentas.

Este estudo fornece uma avaliação objetiva, comparando seus recursos, facilidade de uso, relevância e qualidade das respostas fornecidas, podendo servir de apoio para auxiliar os desenvolvedores na escolha de qual ferramenta utilizar, baseando-se na qualidade do código fornecido. Desta forma, o objetivo desta análise comparativa é fornecer uma visão geral e comparativa das ferramentas ChatGPT e Stack Overflow. Pois, no cenário do desenvolvimento de software, é crucial ter acesso a recursos confiáveis que possam fornecer respostas detalhadas e precisas para questões técnicas.

Com base nesse contexto, a pesquisa proposta visa abordar as seguintes questões de pesquisa (RQ, *Research Questions*):

RQ1. Qual ferramenta oferece códigos mais legíveis? O objetivo desta RQ é determinar qual das ferramentas, Stack Overflow ou ChatGPT, oferece códigos com maior capacidade de compreensão humana.

RQ2. Qual ferramenta oferece códigos mais eficientes do ponto de vista computacional? O objetivo desta RQ é investigar qual ferramenta pode oferecer códigos mais eficientes do ponto de vista computacional. Esse questionamento surge devido à capacidade do ChatGPT ser treinado com uma abundância de dados, o que poderia permitir uma geração de códigos mais otimizados.

RQ3. Qual ferramenta oferece códigos com melhor manutenibilidade? Esta RQ visa determinar qual das duas ferramentas, oferece códigos com melhor manutenibilidade, ou seja, quais códigos fornecidos possuem maior facilidade de manutenção ao longo do tempo.

A estrutura deste estudo é organizada da seguinte forma: uma discussão sobre trabalhos relacionados é apresentada na Seção 2, destacando suas diferenças em relação ao presente estudo. Em seguida, é descrita a metodologia utilizada para a coleta e análise de

dados na Seção 3. Posteriormente, os resultados são apresentados na Seção 4 e discutidos na Seção 5, em seguida são apresentadas as ameaças a validade na Seção 6 e, por fim encerra-se com as conclusões na Seção 7.

2. Trabalhos Relacionados

Para o embasamento teórico e desenvolvimento do estudo, foram utilizados estudos que abordam os temas: Inteligência Artificial, ChatGPT, Stack Overflow, qualidade de software e análise de código Java. Neste tópico, serão descritos o contexto geral dos trabalhos, analisando os seus resultados e relacionando-os com os aspectos mencionados neste artigo.

Segundo De Oliveira Silva et al. (2021) , o artigo tem como objetivo analisar as vantagens e desvantagens da utilização do modelo de Inteligência Artificial GPT-3 na programação. Nesse estudo, observou-se a utilização desse modelo para o desenvolvimento em conjunto com programadores. Destacando os benefícios que o GPT-3 oferece ao desenvolvimento de software. O estudo conclui que a alta capacidade na geração de códigos oferecida pela Inteligência Artificial não irá substituir programadores no mercado de trabalho, todavia, reestruturará organizações. Pode-se relacionar esse artigo com o objetivo deste estudo, pois ambos servirão de apoio para auxiliar os desenvolvedores na escolha da ferramenta a ser utilizada.

O processamento de linguagem natural tornou-se cada vez mais futurista na última década e pode, em breve, transformar a forma como as informações são geradas e acessadas online, conforme demonstrado no estudo de Saravanan et al. (2022) . Esse trabalho também aborda como a Inteligência Artificial na forma de um *chatbot*, pode envolver-se em dinâmicas com o usuário para responder a solicitações de maneira mais precisa. O estudo conclui que o modelo foi preciso em relação à interação com o usuário. Pode-se relacionar com o conteúdo deste artigo, a análise do código gerado pelo modelo, em que os dados coletados indicaram respostas mais legíveis que as fornecidas pelo Stack Overflow.

Na análise de código estático, o código-fonte é analisado quanto à conformidade com os padrões de codificação sem executar o programa, segundo Ashfaq et al. (2022). O estudo traz uma revisão para a seleção de uma ferramenta de análise de código estático. Concluindo que a eficiência e a qualidade na detecção de defeitos podem ser alcançadas usando ferramentas de identificação automatizada de código-fonte. Neste trabalho, utilizou-se a ferramenta Designite Java, com objetivo de análise dos códigos escritos no Stack Overflow e dos códigos gerados no ChatGPT. Através da respectiva ferramenta, possibilitou-se a verificação dos dados utilizando as métricas de qualidade de software.

Stack Overflow é uma das comunidades mais ativas para desenvolvedores compartilharem seus conhecimentos de programação. Segundo Zhang et al. (2021) , em agosto de 2017, havia 32,3 milhões de comentários associados a respostas. O estudo fornece uma compreensão sobre as atividades de comentários na plataforma. Como resultado, observou-se que a maioria dos comentários é informativo, possibilitando o incremento do resultado para uma resposta. Conforme os dados coletados nesse artigo, 23% das respostas com comentários tem um tópico mais longo do que a resposta real, indicando detalhamento da respectiva sessão. Em relação a este trabalho, observou-se que o Stack Overflow

proporciona códigos com maior índice de manutenibilidade comparado ao ChatGPT. Um dos fatores seria devido à produção dos códigos serem feitas por desenvolvedores que possuem experiência na manutenção e na produção. Embora, a interação entre os usuários em uma pergunta seja determinante para o enriquecimento da resposta.

Conforme descrito no estudo de Binanto et al. (2018) , métricas de software são necessárias para medir a qualidade do software desenvolvido e também para estimar o custo e esforço de projetos de software. O artigo utiliza métricas CK para medir qualidades por versão de um software de código aberto chamado Statcato[3]. Utilizou-se a ferramenta CKJM[4], para calcular as métricas. Como resultado, concluiu que a qualidade do software melhorou durante seu ciclo de vida, embora suas funcionalidades aumentem. Durante esse estudo, observou-se a exposição dos gráficos gerados a partir dos dados coletados, associado as métricas selecionadas. O trabalho recorreu às métricas relacionadas à interação entre as classes do sistema, servindo de embasamento na apresentação dos resultados deste artigo. Como exemplo, a construção da Seção 4 referente aos resultados obtidos.

3. Metodologia

O desenvolvimento e a análise do estudo seguem uma metodologia composta por seis etapas, que envolvem a definição do projeto, coleta de informações e análise das métricas obtidas, conforme especificado no diagrama apresentado na Figura 1, onde será detalhado nas subseções abaixo.

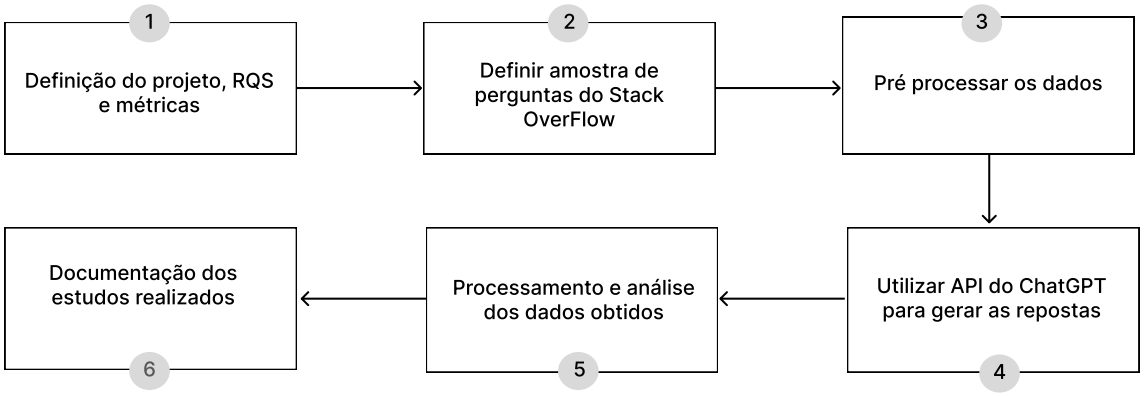

Figura 1. Processo de obtenção e análise dos dados

3.1. Etapa 1. Definição do projeto, RQS e métricas

Esta etapa é fundamental para estabelecer o escopo e os objetivos da análise comparativa entre as ferramentas Stack Overflow e ChatGPT em relação à qualidade do código gerado em suas respostas. Nessa fase, é primordial definir claramente o propósito do projeto, identificando as principais questões de pesquisa a serem abordadas. Além disso, se faz necessário definir as métricas que serão utilizadas para avaliar a qualidade do código gerado por cada ferramenta. A definição desses parâmetros é crucial para garantir que a análise seja objetiva e forneça *insights* significativos sobre as diferenças e semelhanças entre as duas ferramentas em relação à qualidade do código gerado.

[3]https://statcato.org/

[4]https://www.spinellis.gr/sw/ckjm/

Com esse objetivo e conforme dito anteriormente nesta pesquisa, será trabalhado as seguintes perguntas e métricas:

RQ1. Qual ferramenta oferece códigos mais legíveis?

Hipótese Nula (H0): Não há uma diferença na legibilidade das respostas oferecidas tanto pelo ChatGPT quanto pelo Stack Overflow.

Hipótese Alternativa (H1): O ChatGPT, como uma inteligência artificial, consegue gerar códigos mais legíveis devido à sua programação, em contraste com as respostas oferecidas pelo Stack Overflow, que são elaboradas por desenvolvedores reais que, em sua maioria, podem ter manias ou preferências pessoais na escrita de seus códigos.

Métricas:

1. Taxa de comentários - (Número de comentários / Total de linhas de código). Quanto maior a taxa de comentários, maior uma linguagem natural, indicando uma boa legibilidade no código.

2. Complexidade ciclomática - É avaliado a quantidade de caminhos diferentes presentes no código. Foi considerado que um menor número de caminhos indica uma melhor legibilidade. Para a realização desse cálculo, foi utilizado a ferramenta Designite-Java[5].

3. Linhas de código (LOC) - Conforme seu nome, considera-se o número de linhas dos códigos gerados nas respostas das ferramentas. Um código muito pequeno em Java pode não possuir uma boa legibilidade e para realizar este cálculo foi utilizada a ferramenta Cloc[6].

RQ2. Qual ferramenta oferece códigos mais eficientes do ponto de vista computacional?

Hipótese Nula (H0): Não há uma diferença na eficiência computacional das respostas oferecidas tanto pelo ChatGPT quanto pelo Stack Overflow.

Hipótese Alternativa (H1): O ChatGPT gera códigos mais eficientes do ponto de vista computacional, comparado com as respostas obtidas no Stack Overflow, uma vez que essa inteligência artificial pode ser treinada com uma grande quantidade de dados e, portanto, ser capaz de gerar códigos melhor otimizados.

Métricas:

1. NPATH Complexity - Esta é uma métrica semelhante à complexidade ciclomática, capaz de calcular o número total de caminhos que o código percorre em sua execução. Quanto menos caminhos encontrados, melhor é o seu desempenho. Para realizar esse cálculo, foi utilizada a ferramenta PMD[7].

2. Complexidade ciclomática - Mesma métrica descrita na RQ1.

RQ3. Qual ferramenta oferece códigos com melhor manutenibilidade?

Hipótese Nula (H0): Não há uma diferença na métrica de manutenibilidade dos

[5]https://www.designite-tools.com/designitejava/
[6]https://cloc.sourceforge.net/
[7]https://pmd.github.io/

códigos oferecidos tanto pelo ChatGPT quanto pelo Stack Overflow.

Hipótese Alternativa (H1): O Stack Overflow oferece códigos mais fáceis de serem mantidos em comparação com os códigos gerados pelo ChatGPT, uma vez que os códigos presentes no do Stack Overflow são escritos por desenvolvedores reais e, portanto, tendem a seguir boas práticas de programação.

Métricas:

1. Complexidade cognitiva - Esta é uma métrica que mede o nível de dificuldade que o desenvolvedor tem para compreender uma funcionalidade no código. Está também foi uma métrica calculada por meio da ferramenta PMD.

2. Índice de manutenibilidade - Esta métrica indica quantativamente a manutenibilidade de um sistema. Para calcular esta métrica foi utilizada a fórmula a seguir:

$$MI = 171 - 5.2 * ln(V) - 0.23 * (G) - 16.2 * ln(L)$$

Onde:

V: Linhas de código (LOC)

G: Complexidade Ciclomática

L: Tamanho Médio das Variáveis

3.2. Etapa 2. Definir amostra de perguntas do Stack Overflow

Para realizar uma análise adequada, é imprescindível ter uma quantidade suficiente de dados. Nesse sentido, o objetivo deste trabalho foi coletar 2000 perguntas do Stack Overflow seguindo os seguintes critérios:

- Perguntas filtradas pela linguagem de programação Java;
- Perguntas contendo pelo menos uma resposta aceita;
- Perguntas com pontuação maior que 1000;
- Perguntas com mais de 1000 visualizações;

Utilizando a API (*Application Programming Interface*) do Stack Overflow, as perguntas foram extraídas e filtradas com base nos critérios mencionados anteriormente. Dessa forma, foram obtidas informações como o título da pergunta, o link correspondente, o conteúdo da resposta, o link da resposta e os identificadores únicos (IDs) tanto da pergunta quanto da resposta. Esses dados foram persistidos em um arquivo CSV para posterior análise.

3.3. Etapa 3. Pré-processar os dados

Essa etapa compreende a realização das seguintes ações no arquivo CSV obtido na etapa anterior:

- Remoção de perguntas duplicadas;
- Remoção de perguntas que não estejam no idioma inglês;
- Remoção de perguntas que não possuem código em sua resposta.

3.4. Etapa 4. Utilizar API do ChatGPT para gerar as respostas

Nesta etapa, foi implementado um *script* que realiza requisições a API do ChatGPT, a qual possibilita fazer integrações com suas funcionalidades de conversação. No caso específico, o *script* utiliza os títulos das 2000 perguntas extraídas do Stack Overflow como entrada para a API, adicionando a frase *"Generate a Java code to answer the following question:"* no início de cada título. A API do ChatGPT processa essas entradas e gera respostas de código em Java correspondentes às perguntas. Em seguida, os trechos de código resultantes são adicionados ao CSV construído nas etapas anteriores.

3.5. Etapa 5. Processamento e análise dos dados obtidos

O *script* apresenta uma sequência de etapas que visam realizar análises de métricas de código-fonte usando diferentes ferramentas. A configuração das ferramentas utilizadas no estudo envolveu várias etapas. Inicialmente, foi definido o diretório de entrada para os arquivos $.java$ e um diretório de saída separado para os resultados da ferramenta Designite. O formato de saída escolhido foi o CSV. O mesmo método foi utilizado para a ferramenta Cloc.

No caso da ferramenta PMD, por ser tratar de uma ferramenta de $report$ de violações de padrões, configurou-se um arquivo XML de relatório para emitir alerta quando as métricas $npath$, complexidade cognitiva e nome de variáveis ultrapassassem o limite de 1 e definiu-se o diretório de entrada, o diretório de saída e o formato de saída como CSV. Desse modo, no arquivo de saída do PMD encontraram-se os valores das métricas e os nomes de todas as variáveis que foram calculadas por meio de uma função no *script* $python$.

Após a configuração, as ferramentas foram executadas para cada pergunta. Foi criado um diretório específico para armazenar o código-fonte e os resultados. Primeiramente, em cada pergunta, os códigos gerados pelo ChatGPT foram extraídos e salvos em um arquivo específico com extensão $.java$. Em seguida, as ferramentas de análise Designite, Cloc e PMD foram executadas em ordem nesse arquivo. Após a análise, o arquivo $.java$ foi excluído para evitar conflitos com o código da próxima pergunta. Esse processo também foi repetido para as respostas obtidas do Stack Overflow.

Os resultados das ferramentas foram analisados e as métricas relevantes foram coletadas. Para o Designite, a métrica de interesse foi a complexidade ciclomática. Para o Cloc, foram extraídas as métricas de linhas de código e número de comentários. O PMD forneceu métricas como complexidade de caminhos, complexidade cognitiva e tamanho médio dos nomes das variáveis.

Com base nos resultados, calculou-se a densidade de comentários e o índice de manutenibilidade. As métricas extraídas foram adicionadas às perguntas correspondentes. Em seguida, os arquivos $.java$ e os arquivos de saída das ferramentas foram excluídos para evitar conflitos com as próximas perguntas. Esse processo foi repetido para as respostas do Stack Overflow.

Após remover o corpo das respostas do *dataframe*, foram descartadas as perguntas nas quais as ferramentas não puderam ser executadas, garantindo apenas dados completos para análise. Por fim, os dados foram salvos em um novo arquivo CSV, contendo as métricas de código-fonte obtidas para cada pergunta. Essa documentação e a disponibilidade das métricas auxiliam na referência futura e utilização pela comunidade interessada.

O *script* adotado consiste em uma abordagem estruturada, na qual o código-fonte de perguntas é extraído do ChatGPT e do Stack Overflow. Em seguida, são aplicadas ferramentas de análise, seguidas pela coleta e cálculo de métricas. Por fim, os resultados são armazenados para posterior análise.

O processo de análise envolve a geração de gráficos comparativos, como ECDF *(Empirical Cumulative Distribution Function)*, histograma de distribuição, *boxplot* com gráfico de violino e *boxplot* sem outliers, para cada métrica extraída. Além disso, são calculados os quartis, os valores máximos e mínimos, bem como a contagem de vezes em que cada ferramenta apresentou um valor superior. Também é realizado um teste estatístico conhecido como teste de Wilcoxon (R de Wilcoxon), com um nível de significância de 5%, para comparar as diferenças entre as amostras obtidas do ChatGPT e do Stack Overflow.

3.6. Etapa 6. Documentação dos estudos realizados

Após a conclusão das pesquisas e análises, a etapa final consiste no registro minucioso e na documentação abrangente de todo o trabalho realizado. Esse processo é essencial para assegurar a rastreabilidade e a validação dos resultados obtidos neste estudo, tornando-os facilmente acessíveis para referência futura e para a comunidade interessada. Dessa forma, a pesquisa fica devidamente documentada, disponível e pronta para consulta e uso posterior, garantindo a transparência e a utilidade duradoura do estudo.

4. Resultados

A fim de abordar as perguntas de pesquisa estabelecidas no início deste estudo, realizou-se uma análise dos dados coletados. Para facilitar a visualização desses dados, foram utilizadas ferramentas de geração de gráficos. A seguir, serão apresentadas as respostas para cada pergunta de pesquisa, acompanhadas das respectivas análises.

RQ1. Qual ferramenta oferece códigos mais legíveis?

A fim de responder à pergunta, a Figura 2 mostra a comparação entre a distribuição do LOC do ChatGPT e do Stack Overflow, onde pode-se observar que 80% dos dados tanto do ChatGPT quanto do Stack Overflow estão abaixo de 20 LOC. Tendo o Stack Overflow um LOC próximo a 0 em aproximadamente 40% da amostra e mantendo um LOC mais baixo que o ChatGPT em quase 90% dos dados. O teste de *Wilcoxon* executado resultou num valor de $0,00$, como este valor é menor que o valor de significância estipulado de 5%, pode-se afirmar que há uma diferença estatística entre o LOC das ferramentas.

Na Figura 3, pode-se observar a variação do LOC nas duas ferramentas analisadas. Para o LOC do ChatGPT tem o primeiro quartil em $9,00$, o segundo quartil em $13,00$ e o terceiro quartil em $19,00$. Já o Stack Overflow, é possível observar que o primeiro quartil está em $1,00$, o segundo em $6,00$ e o terceiro em $14,00$.

O total das linhas de código do ChatGPT é 22.481, enquanto do Stack Overflow é 17.174. Dos 1.459 dados coletados o ChatGPT apresentou um valor de LOC maior que o do Stack Overflow em 1015 das perguntas, em seguida o Stack Overflow apresentou um LOC maior em 388 perguntas, e igual em 56 perguntas.

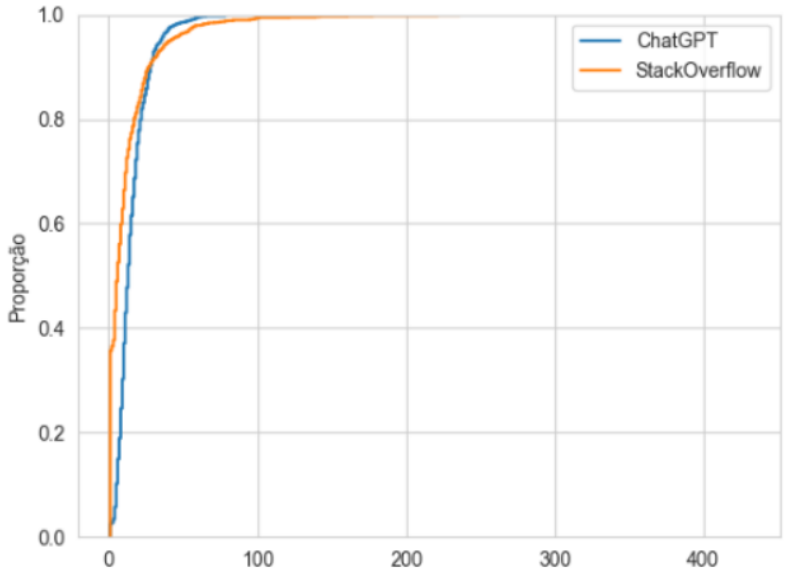

Figura 2. Comparação do ECDF (Empirical Cumulative Distribution Function) do LOC

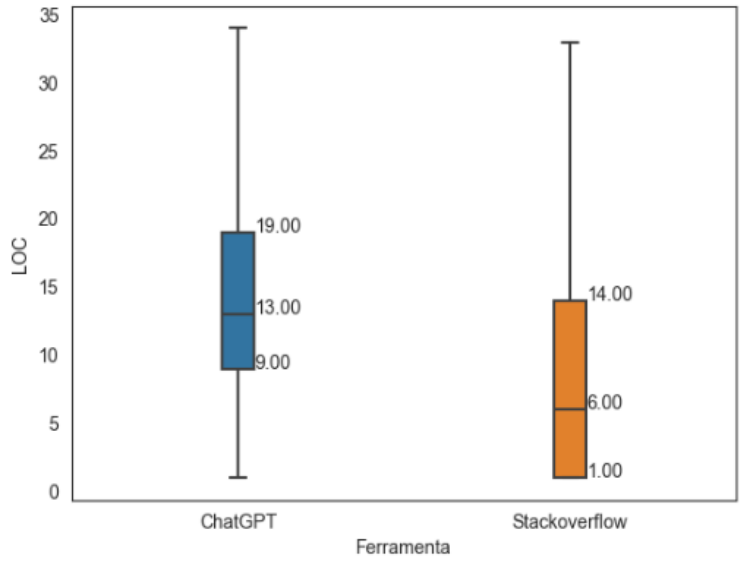

Figura 3. *Boxplot* de comparação entre o LOC do ChatGPT e Stack Overflow

A Figura 4 mostra a distribuição e a proporção da densidade de comentários para cada ferramenta. Pode-se notar que em 60% dos dados de ambas ferramentas o valor é de $0,0$ na densidade de comentários. Porém, observa-se que o Stack Overflow mantém o valor de $0,0$ até 80% da amostra enquanto o ChatGPT, possui cerca de 40% dos valores entre $0,1$ e $0,5$.

O ChatGPT varia a densidade de comentários de 0 a 3.0, enquanto o Stack Overflow varia de 0 a $3,15$. Os quartis do ChatGPT são $q1 = 0$, $q2 = 0$ e $q3 = 0,17$, enquanto os do Stack Overflow são $q1 = 0$, $q2 = 0$ e $q3 = 0$. O total da densidade de comentários do ChatGPT é $153,14$, enquanto do Stack Overflow é $64,95$. Dos $1,459$ dados coletados o ChatGPT apresentou um valor de densidade de comentários maior que o do Stack Over-

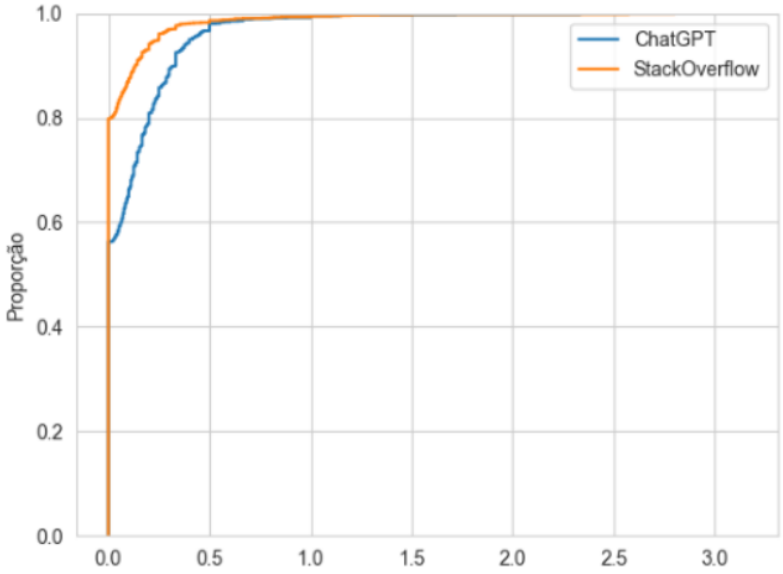

Figura 4. Comparação do ECDF da densidade de comentários

flow em 578 perguntas. Já o Stack Overflow apresentou uma densidade de comentários maior em 203 perguntas, e a densidade de comentários foi igual em 678 perguntas. O teste estatístico executado resultou num valor p de $0,000$ que está abaixo do valor de significância estabelecido, portanto se conclui que há uma diferença entre os resultados do ChatGPT em relação ao Stack Overflow.

Na Figura 5 o ChatGPT varia a Complexidade Ciclomática de 1 a 26, enquanto o Stack Overflow varia de 1 a 112. Os quartis do ChatGPT são $q1 = 1$, $q2 = 1$ e $q3 = 2$, enquanto os do Stack Overflow são $q1 = 1$, $q2 = 1$ e $q3 = 1$. O total da Complexidade Ciclomática do ChatGPT é 2.981, enquanto do Stack Overflow é 2.796. Dos 1.459 dados coletados o ChatGPT apresentou um valor de complexidade ciclomática maior ao Stack Overflow em 451 perguntas, cerca de 30%. Já o Stack Overflow apresentou uma complexidade ciclomática maior em 202 perguntas, aproximadamente 14%, e a complexidade ciclomática foi igual em 806 perguntas que representa 55% do total. O valor p obtido do teste de *Wilcoxon* é de $0,000$, que está abaixo do valor de significância estabelecido, logo, há uma diferença entre os resultados do ChatGPT em relação ao Stack Overflow.

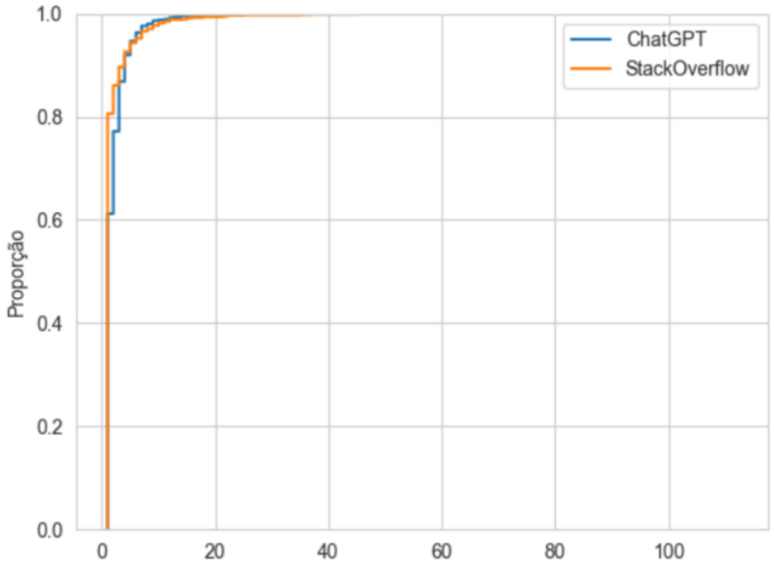

Figura 5. Comparação do ECDF da complexidade ciclomática

RQ2. Qual ferramenta oferece códigos mais eficientes do ponto de vista computacional?

Para responder RQ2 e validar a hipótese levantada, a Figura 6 mostra o ECDF comparativo da complexidade de caminhos entre o ChatGPT e o Stack Overflow. Onde é possível perceber que para as duas ferramentas analisadas os valores se concentram, em cerca de 80% da amostragem, entre $1,0$ e $2,0$. Tendo o ChatGPT um valor mais alto que o Stack Overflow em aproximadamente 30% dos dados. Ainda é possível observar a distribuição da complexidade ciclomática, tendo uma concentração alta dos dados do ChatGPT e do Stack Overflow abaixo de 5.0. Os quartis do ChatGPT são $q1 = 1$, $q2 = 1$ e $q3 = 2$, enquanto os do Stack Overflow são $q1 = 1$, $q2 = 1$ e $q3 = 1$. O total da Complexidade de caminhos do ChatGPT é 3.060, enquanto do Stack Overflow é 2.096. Dos 1.459 dados coletados o ChatGPT apresentou um valor de complexidade de caminhos maior ao Stack Overflow em 416 perguntas. Já o Stack Overflow apresentou uma complexidade de caminhos maior em 79 perguntas, e a complexidade de caminhos foi igual em 964 perguntas.

Tanto a complexidade ciclomática quanto a complexidade de caminhos mostraram o mesmo comportamento, ambas com as medianas iguais em $1,0$. O ChatGPT e o Stack Overflow apresentam uma concentração da complexidade ciclomática com mais de 60% das observações em $1,0$ e próximo de 80% dos dados da complexidade de caminhos também a $1,0$. O teste estatístico executado resultou num valor p de $0,000$, como esse valor é menor que o *alpha* estabelecido, rejeita-se a hipótese nula de que não há diferença sistemática entre as duas amostras.

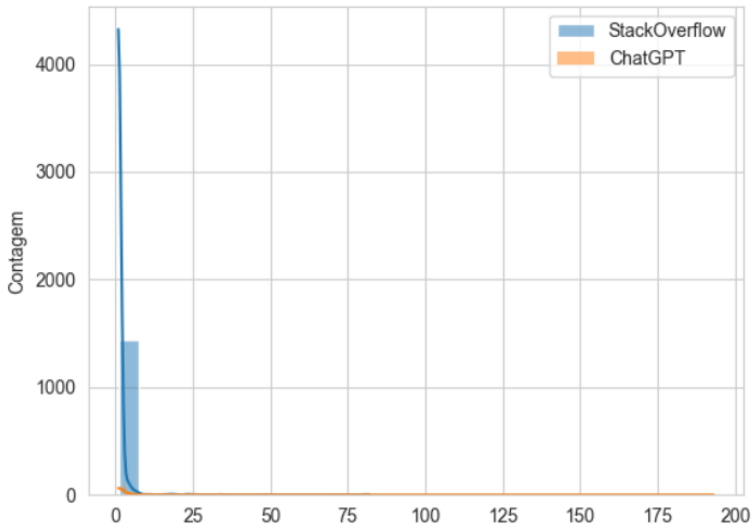

Figura 6. Comparativo do ECDF da Complexidade de Caminhos

RQ3. Qual ferramenta oferece códigos com melhor manutenibilidade?

A Figura 7 mostra o ECDF da complexidade cognitiva, onde pode-se observar que para ambas ferramentas cerca de 70% dos dados estão abaixo de $1,0$. As duas ferramentas apresentaram uma distribuição muito semelhante, porém o ChatGPT mostrou um maior valor numa parcela de aproximadamente 30% da amostra.

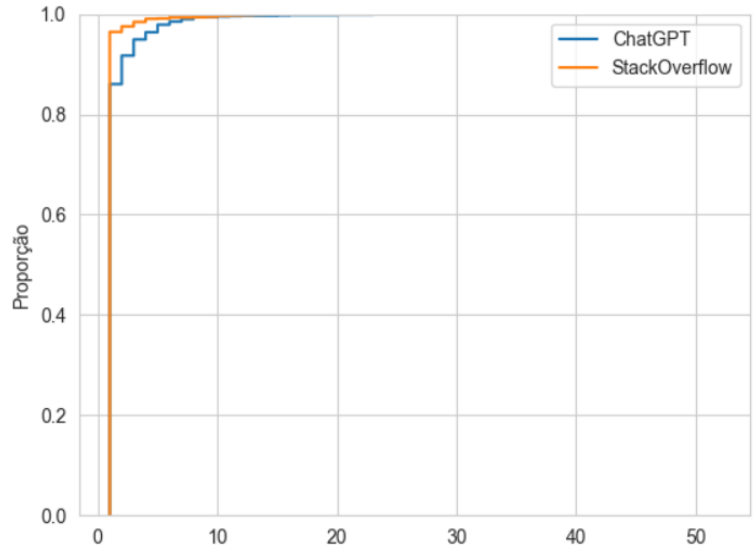

Figura 7. Comparativo do ECDF da Complexidade Cognitiva

As Figuras 8 e 9 mostram a distribuição e a variação dos quartis do índice de manutenibilidade. Nos dois gráficos pode-se observar que o índice de manutenibilidade do ChatGPT encontra-se concentrada em duas faixas, uma em seu primeiro quartil, 127.54, e outra em seu terceiro quartil, $157,43$. O Stack Overflow apresenta uma concentração acima do seu primeiro quartil em $154,24$ e próxima à 170.

Ao observar a complexidade cognitiva de ambos os objetos de estudo, Stack Over-

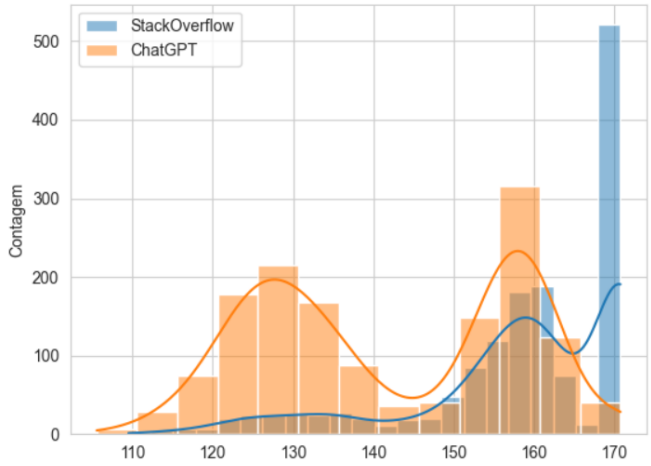

Figura 8. Distribuição do Índice de Manutenibilidade

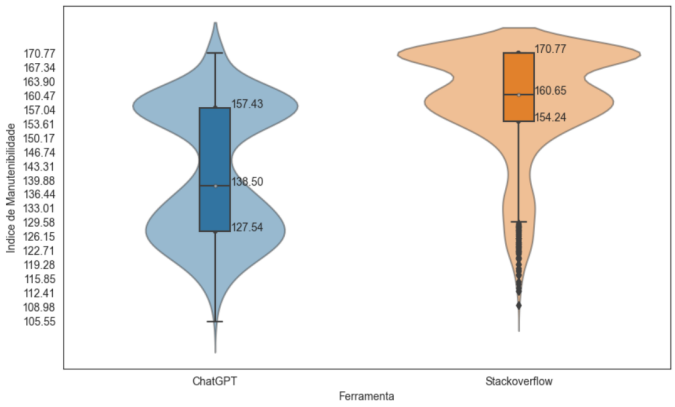

Figura 9. Índice de Manutenibilidade

flow e ChatGPT, pode-se notar que ambas se mantiveram baixas e com uma distribuição muito próxima a 1. Tendo apresentado uma diferença, com um valor maior do ChatGPT, em apenas 13% da amostragem.

Já ao observar o índice de manutenibilidade, percebe-se uma diferença considerável entre os quartis e a distribuição dos dados. No ChatGPT tem valores mais baixos de todos os quartis em relação ao Stack Overflow, e a distribuição encontra-se majoritariamente entre: aproximadamente 120 a 140 e 150 a 160. O Stack Overflow, porém, apresentou valores de quartis mais altos que o ChatGPT e teve a maioria dos dados entre 150 e 170, tendo os valores de *outliers* na mesma faixa que uma parte considerável dos dados do ChatGPT. O teste estatístico de *Wilcoxon* executado resultou em um valor $p = 0.000$, demonstrando uma diferença estatisticamente significativa entre as duas amostras.

5. Discussão

Os valores apresentados na Seção 4 demonstram, um valor de proximidade entre as ferramentas analisadas para as duas primeiras perguntas. Ao investigar a RQ1, pode-se notar que, apesar da baixa diferença entre o LOC, a taxa de comentários e a complexidade ciclomática, os testes estatísticos executados mostraram haver uma diferença entre esses

valores, tendo o ChatGPT maiores valores de LOC, densidade de comentários e de complexidade ciclomática. Altos valores linhas de código e densidade de comentários indicam uma boa legibilidade de código, porém a complexidade ciclomática alta pode indicar uma dificuldade em se ler e compreender o código escrito. Nesse contexto, com base em duas das três métricas utilizadas para medir a legibilidade, pode-se rejeitar a hipótese nula de que não há diferença entre as duas ferramentas e aceitar a hipótese alternativa de que o ChatGPT apresenta códigos mais legíveis em suas respostas.

Em relação à RQ2 pode-se notar um resultado parecido com a RQ1, pois ambas tratam de valores com pouca diferença numérica entre as ferramentas, entretanto com um valor que indica diferença no teste estatístico. Foi observado, nos resultados, que nos dois tipos de complexidades analisadas para as perguntas, o ChatGPT apresentou um valor maior, porém, ambas com valores baixos. Dessa forma, é possível afirmar que ambas ferramentas fornecem códigos de alta eficiência computacional, e este valor baixo pode estar mais atrelado ao pequeno tamanho dos trechos de códigos obtidos do que com a capacidade de se conseguir códigos computacionalmente eficientes nessas ferramentas. Com isso, como os testes estatísticos apontaram haver uma diferença entre os resultados do Stack Overflow e do ChatGPT, rejeita-se a hipótese nula e rejeita-se também a hipótese alternativa levantada, pois o ChatGPT mostrou ter menor eficiência computacional. Mesmo sendo uma diferença baixa nos dados observados, esta diferença pode se propagar considerando o uso de mais de um único trecho de código dessas ferramentas por um desenvolvedor numa situação real de desenvolvimento.

Já nos resultados da RQ3 pode-se observar uma diferença significativa entre o Stack Overflow e o ChatGPT, onde nas duas ferramentas foi possível observar um valor baixo de complexidade cognitiva na maioria das respostas, porém com o Stack Overflow apresentando um valor mais baixo que o ChatGPT em uma pequena quantidade dos casos. No índice de manutenibilidade observa-se um alto valor em ambas ferramentas. Esses valores baixos de complexidade cognitiva e altos valores de complexidade cognitiva indicam que ambas ferramentas proporcionam códigos muito manuteníveis. O que pode-se notar é um valor muito diferente do índice de manutenibilidade das ferramentas, o Stack Overflow mostrou além de valores de medidas centrais mais altos uma distribuição mais concentrada acima de 150. Com isto rejeita-se a hipótese nula e aceita a hipótese alternativa levantada, de que o Stack Overflow proporciona códigos de maior manutenibilidade. Isso pode ocorrer pelo fato dos códigos encontrados no Stack Overflow serem produzidos por desenvolvedores que possuem uma experiência na produção e manutenção de códigos, além de seguirem boas práticas de programação.

6. Ameaças à validade

Durante a realização do experimento, foram identificadas várias ameaças que poderiam impactar na validade da conclusão em relação às hipóteses levantadas. Para caracterizar as ameaças encontradas, as mesmas foram divididas em quatro categorias. São elas: ameaças à construção, ameaças à conclusão, ameaças internas e ameaças externas.

Com relação à construção do experimento, foi identificada a possibilidade da divergência entre a compreensão das perguntas propostas por parte da IA que constitui o ChatGPT. Ainda que em todos os casos da amostra fosse possível analisar a qualidade de código gerada por ambas as soluções, não necessariamente os códigos gerados pela

inteligência artificial responderiam com sucesso às perguntas propostas, o que impactaria diretamente na validade das métricas extraídas.

Quanto à conclusão do experimento, foi levantada uma possível ameaça relacionada às respostas geradas pelas ferramentas utilizadas neste estudo. No ChatGPT, os códigos gerados apresentaram 13 linhas de mediana. Já no Stack OverFlow, 6 linhas. Considerando as métricas de qualidade utilizadas para a análise, infere-se que possivelmente os códigos gerados não possuem um tamanho em linhas de código adequado para a aplicação das mesmas.

Como ameaça interna, existe a possibilidade de que as ferramentas escolhidas para extração de métricas de qualidade não funcionem de maneira correta. Neste cenário, os dados obtidos a partir do uso de ambas as ferramentas não permitiriam a tomada de uma conclusão assertiva com relação aos atributos de qualidade do software gerado pelas ferramentas utilizadas.

Por fim, foi constatado o risco da impossibilidade de generalização das conclusões obtidas por meio da análise da amostra estudada. Pelo fato de terem sido utilizadas duas mil perguntas como espaço amostral, existe a possibilidade de que os dados obtidos não reflitam a realidade do universo observável relacionado ao objeto de estudo.

7. Conclusão

O presente estudo visou realizar uma análise abrangente e comparativa entre a ferramenta ChatGPT e a plataforma Stack Overflow, com o propósito de oferecer orientações aos desenvolvedores na seleção da solução mais adequada para suas dúvidas e na avaliação da qualidade do código gerado. Para tanto, foram extraídas um total de 2.000 perguntas na linguagem de programação Java por meio da API do Stack Overflow. Em seguida, utilizando a API do ChatGPT, foram realizadas solicitações à referida ferramenta.

Os resultados obtidos revelaram diferenças sutis entre as duas ferramentas avaliadas. Em relação à legibilidade do código, constatou-se que o ChatGPT apresenta códigos mais compreensíveis. Quanto à eficiência do código, ambas as ferramentas demonstraram fornecer soluções de alta eficácia computacional. No entanto, observou-se uma diferença significativa em relação à manutenibilidade, onde o Stack Overflow se destacou ao oferecer códigos mais facilmente mantidos ao longo do tempo. É importante ressaltar que, devido à amostra estudada não representar fielmente o universo observável, conforme mencionado nas considerações sobre a validade, não é possível afirmar categoricamente que o ChatGPT é superior ao Stack Overflow, ou vice-versa.

Como trabalhos futuros, recomenda-se ampliar o escopo da pesquisa para incluir outras linguagens de programação, ferramentas e/ou plataformas relevantes, como o Reddit e o GitHub Copilot, a fim de realizar uma comparação abrangente com o ChatGPT. Essa expansão permitiria uma análise mais completa, no sentido de preencher as lacunas abertas nas ameças à validade.

Referências

Ashfaq, Q., Khan, R., and Farooq, S. (2019). A comparative analysis of static code analysis tools that check java code adherence to java coding standards. In *2019 2nd International Conference on Communication, Computing and Digital systems (C-CODE)*, pages 98–103.

Binanto, I., Warnars, H. L. H. S., Gaol, F. L., Abdurachman, E., and Soewito, B. (2018). Measuring the quality of various version an object-oriented software utilizing ck metrics. In *2018 International Conference on Information and Communications Technology (ICOIACT)*, pages 41–44.

de Oliveira Silva, J. V., Pacheco, G. O., and Pugliesi, J. B. (2021). O modelo de inteligência artificial gpt-3 na programação e suas vantagens e desvantagens no desenvolvimento junto ao programador. *Revista Eletrônica de Computação Aplicada*, 2(1).

Reuters (2023). O chatgpt, o popular chatbot da openai, atingiu 100 milhões de usuários ativos mensais em janeiro. Acessado em: 12 de junho de 2023.

Saravanan, S. and Sudha, K. (2022). Gpt-3 powered system for content generation and transformation. In *2022 Fifth International Conference on Computational Intelligence and Communication Technologies (CCICT)*, pages 514–519.

Zhang, H., Wang, S., Chen, T.-H., and Hassan, A. E. (2021). Reading answers on stack overflow: Not enough! *IEEE Transactions on Software Engineering*, 47(11):2520–2533.