

# Avaliação exploratória da adoção de inteligência artificial na Manutenção e Evolução de sistemas de *software*

Aylton B. de Almeida Junior

<sup>1</sup> Belo Horizonte – MG – Brasil

almeida@aylton.dev

## 1. Introdução

O estudo sobre inteligência artificial (IA) e aprendizado de máquina não é algo recente, tendo sido tópico de discussão na sociedade desde meados do século XX. Porém, o lançamento de ferramentas como o *Chat GPT-4* e *Google Bard* para o público geral vem mostrando a capacidade que essa tecnologia tem de alterar como tarefas são executadas nas mais diversas áreas. Ambas aplicações são *chatbots* que utilizam de IA para responder perguntas simulando linguagem natural [OpenAI 2023a, Google 2023], e demonstraram a capacidade de gerar respostas plausíveis para diversos tópicos. Alguns exemplos são a escrita de textos bem estruturados como *posts* para um *blog*, e necessidades mais abstratas, como responder perguntas relacionadas à filosofia e à sociologia.

Enquanto a utilidade dessas ferramentas para o público geral ainda está sendo descoberta, a utilização de tecnologias de IA para o desenvolvimento de *software* não é uma novidade. Modelos como o *Codex* da *Open AI*, capazes de analisar linguagem natural a fim de escrever blocos de código em resposta [OpenAI 2023b], vem sendo utilizados há alguns anos na tentativa de otimizar o desenvolvimento de sistemas. Dentro desse caso de uso, esses modelos têm sido capazes de auxiliar desenvolvedores a encontrar possíveis *bugs* por meio da análise de algoritmos previamente escritos [Prenner et al. 2022]. Demonstram também, a capacidade de gerar trechos de código por meio de ferramentas como o *Github Copilot* e o *Amazon CodeWhisperer*, ambas utilizadas para gerar sugestões em um editor de texto por meio da utilização de inteligência artificial [Github 2023, Amazon 2023].

Ainda que ferramentas de IA tenham se mostrado úteis para o desenvolvimento de *software*, há muito o que se estudar quanto às diversas formas em que elas podem ser aplicadas para melhor atender desenvolvedores. Tendo isso em vista, neste projeto é proposto um trabalho na área de **Engenharia de Software**, que tem como objetivo **avaliar a capacidade de ferramentas que utilizam de inteligência artificial para aprimorar o processo de manutenção e evolução de sistemas de *software***. Dessa mesma forma, três objetivos específicos são definidos, sendo eles:

1. Verificar se existem melhorias de desempenho notáveis ao comparar desenvolvedores que utilizam ferramentas de IA durante o desenvolvimento em relação aos que não utilizam;
2. Analisar se existe algum impacto na qualidade do código resultante da utilização de ferramentas de IA;
3. Estudar possíveis formas de utilizar IAs a fim de tornar a manutenção de sistemas mais eficiente.

Este documento está organizado da seguinte forma: A Seção 2 busca apresentar o referencial teórico que servirá de fundamentação para o projeto. Na Seção 3, é abordada a metodologia proposta para realização do projeto. Por fim, a Seção 4 descreve o cronograma de atividades proposto.

## 2. Referencial Teórico

Esta seção visa a definir conceitos a serem utilizados ao longo do projeto, assim como mencionar trabalhos relacionados a fim de melhor explorar possibilidades de expansão do assunto.

O principal conceito a ser trabalhado neste projeto é o de inteligência artificial. Esse é um termo relativamente amplo, de forma que não há uma única definição para ele. Porém, uma das mais usadas é a de que IA é o estudo de como fazer computadores fazerem coisas em que pessoas são melhores [Rich 1985]. Isso é, IA é o estudo de como capacitar computadores a melhor atender a necessidades que demandam características humanas, como a capacidade de raciocinar e tomar decisões.

Tendo como base IA, outros trabalhos podem ser citados a fim de realçar as possibilidades existentes para este projeto. O primeiro deles é o estudo de Imai (2022), no qual um breve experimento é conduzido a fim de definir a capacidade que o *Github Copilot* tem de substituir a necessidade de um segundo desenvolvedor durante a prática de *pair programming*, prática essa que consiste em dois desenvolvedores programarem em uma mesma máquina. Durante esse estudo foi observado que ao utilizar o *Github Copilot* desenvolvedores acabam sendo mais produtivos, porém, a qualidade do código gerado é menor quando comparado ao código gerado por dois desenvolvedores trabalhando juntos.

Um segundo estudo interessante é o feito por Hourani et al. (2019), no qual é analisado o impacto do uso de IA para testes de *software*. Nesse estudo, é observado que a utilização de IA para testar sistemas tem o potencial de diminuir o tempo para que uma funcionalidade ou produto sejam lançados no mercado. Isso ocorre devido a capacidade que IA tem de analisar uma grande base de código a fim de encontrar possíveis *bugs* e vulnerabilidades.

Também há espaço para que o artigo escrito por Prenner et al. (2022) seja citado. Nesse, um experimento é conduzido a fim de verificar a capacidade que o *Codex* tem de encontrar *bugs* comparado a ferramentas específicas para essa funcionalidade. Para a execução da pesquisa foram utilizadas as linguagens Python e Java, sendo que, mesmo não tendo sido treinado com esse objetivo em mente, o modelo se apresentou eficaz em encontrar e resolver bugs, demonstrando sua capacidade de adaptação a diferentes cenários.

Por fim, pode ser mencionado o trabalho de Nguyen e Nadi (2022), no qual uma avaliação é feita a fim de medir a qualidade do código gerado pelo *Github Copilot*. Nesse projeto os resultados apontam que, no geral, o código gerado tem complexidade baixa e uma assertividade média, não possuindo diferenças relevantes entre linguagens de programação. Um problema notável é que a ferramenta não se apresentou capaz de utilizar métodos auxiliares em suas sugestões, o que pode levar a funções mais complexas dependendo da regra de negócio implementada.

Tendo em vista os projetos citados, alguns estudos interessantes podem ser desenvolvidos a fim de melhor explorar o assunto. O primeiro é verificar a capacidade que

IAs tem de fazer atualização automática de dependências que possuem *breaking changes*, essas que são mudanças que quebram interfaces de comunicação entre a aplicação e sua dependência, uma dificuldade que se mostra comum no tópico de manutenção de sistemas. Outro ponto de interesse é a forma como a utilização de IAs pode acelerar o desenvolvimento de sistemas ao mesmo tempo que garante um nível de qualidade para eles, seja por meio da implementação de testes automatizados ou por meio da geração de código que segue boas práticas de programação. Para ambos os tópicos mencionados, é possível focar na utilização de ferramentas de *chatbot* como o *Chat GPT*, já que, devido a sua recente disponibilização ao público, ainda não há um estudo profundo sobre os impactos do uso dessa ferramenta.

### 3. Metodologia

O projeto proposto é classificado como exploratório, tendo como objetivo a realização de estudos e experimentos a serem seguidos a fim de melhor explorar o tema proposto. Nesta seção, são descritas as atividades para a realização do projeto, assim como as etapas necessárias para a conclusão de cada uma delas.

#### 3.1. Atividades

Tendo como referência os três objetivos específicos propostos, se faz necessário definir quais as perguntas que devem ser respondidas a fim de alcançar esses objetivos. Para isso, um estudo maior precisará ser feito com o objetivo de expandir o referencial teórico apresentado na Seção 2 para propor perguntas que auxiliem na evolução do conhecimento já existente na área. Tendo definido os questionamentos, um estudo mais voltado para métricas deverá ser conduzido a fim de definir quais serão utilizadas durante os experimentos para avaliar os resultados obtidos. Por fim, será necessário definir quais serão os experimentos a serem conduzidos, assim como a maneira pela qual eles serão realizados.

#### 3.2. Disciplinas

Ao longo do projeto, algumas disciplinas serão cursadas a fim de melhor capacitar o candidato a conduzir os experimentos propostos. Elas podem ser divididas em dois grupos, o primeiro sendo de disciplinas fundamentais e que servirão de base para o projeto. Já o segundo grupo é composto de disciplinas complementares que servirão de auxílio para a realização do trabalho proposto, mas que poderão ser substituídas por outras disciplinas conforme a necessidade.

Tendo isso em vista, a primeira disciplina do primeiro grupo é a **Projeto e Análise de Algoritmos**. Esta é a única matéria obrigatória do curso de mestrado. Ela será responsável por capacitar o aluno a analisar a complexidade de algoritmos e a propor soluções para problemas computacionais, servindo de base para os demais estudos propostos. A segunda disciplina proposta é a chamada **Engenharia de Software**. A engenharia de *software* é parte fundamental da manutenção e evolução de sistemas, de forma que seu estudo será de grande importância para melhor compreender este processo e definir experimentos congruentes com sua realidade. Por fim, também é necessário cursar a disciplina de **Inteligência Artificial**. Essa visa a abordar fundamentos da inteligência artificial e mostrar suas aplicações em problemas práticos. Ela será de grande importância

para maior entendimento de como essa tecnologia funciona, a fim de melhor definir e analisar experimentos.

Como parte do segundo grupo de disciplinas serão mencionadas apenas três, porém, como mencionado previamente, elas poderão ser alteradas para melhor anteder às necessidades do projeto. Dentre elas, a primeira é a **Análise e Modelagem de Desempenho de Sistemas de Computação**. Essa tem como objetivo apresentar técnicas de análise e modelagem de desempenho de sistemas de computação. Ela será de grande importância para a definição de métricas a serem utilizadas durante os experimentos. A segunda disciplina é a **Fundamentos teóricos da Computação**. Essa procura apresentar os fundamentos matemáticos nos quais se baseia a computação, sendo de grande importância para o estudo sobre o funcionamento de IA. Por fim, há a possibilidade de cursar a disciplina de **Tópicos em Engenharia de Software** ou **Tópicos em Inteligência Artificial**. Ambas matérias se relacionam com o tema proposto, entretanto, como elas possuem uma ementa variável, seria interessante avaliar qual das duas cursar com base no tópico discutido no semestre e na sua capacidade de melhor qualificar o candidato para execução do projeto.

#### 4. Cronograma

Nesta seção é proposto um cronograma para realização das atividades mencionadas, conforme apresentado na Tabela 1. Ele prevê um período de dois anos para realização do projeto, tendo seu início no segundo semestre de 2023 e seu fim no primeiro semestre de 2025.

**Tabela 1. Cronograma de atividades proposto.**

Tarefas	2023	2024		2025
	2º Semestre	1º Semestre	2º Semestre	1º Semestre
Projeto e Análise de Algoritmos	X			
Engenharia de Software	X			
Inteligência Artificial		X		
Análise e Modelagem de Desempenho de Sistemas de Computação		X		
Fundamentos Teóricos da Computação			X	
Tópicos em Engenharia de Software ou Inteligência Artificial			X	
Primeiro Artigo Científico		X	X	
Segundo Artigo Científico			X	X
Dissertação		X	X	X

O **2º semestre de 2023** teria como principal objetivo a realização das disciplinas **Projeto e Análise de Algoritmos** e **Engenharia de Software**. Ambas disciplinas são consideradas fundamentais para o desenvolvimento do projeto, de forma que iniciar por elas é de grande importância. Além disso, também será neste semestre que o aluno proponente terá que se familiarizar com o curso de Mestrado e com a área de pesquisa.

Durante o **1º semestre de 2024** é proposto que a disciplina **Inteligência Artificial** seja cursada. Ela é considerada fundamental para o projeto, sendo importante cursá-la logo que possível. Também é previsto que a disciplina **Análise e Modelagem de Desempenho de Sistemas de Computação** sejam cursadas, de forma a melhor capacitar o candidato a definir métricas e experimentos que sejam congruentes com a realidade do desenvolvimento de *software*. Neste mesmo semestre é esperado que sejam iniciados ambos o **primeiro artigo científico** proposto e a **dissertação** do mestrado.

A disciplina **Fundamentos Teóricos da Computação** será cursada no **2º semestre de 2024**. Também é proposto que a disciplina **Tópicos em Engenharia de Software** ou **Tópicos em Inteligência Artificial** seja cursada nesse semestre. Elas são consideradas importantes para o projeto, porém não fundamentais, de forma que podem ser cursadas em um momento posterior. Durante este semestre também é proposto que o **primeiro artigo científico** seja finalizado e que o **segundo artigo científico** iniciado, sendo definido seu tema e principais objetivos.

No **1º semestre de 2025** é proposto que o **segundo artigo científico** proposto seja finalizado e publicado. Durante este período também é previsto que a dissertação seja finalizada a fim de apresentar os resultados obtidos durante o curso e propor trabalhos futuros.

## Referências

- [Amazon 2023] Amazon (2023). Gerador de código de ai. <https://aws.amazon.com/pt/codewhisperer>. [Online, accessed: 2023-05-27].
- [Github 2023] Github (2023). Your ai pair programmer. <https://github.com/features/copilot>. [Online, accessed: 2023-05-27].
- [Google 2023] Google (2023). Meet bard. <https://bard.google.com>. [Online, accessed: 2023-05-27].
- [Hourani et al. 2019] Hourani, H., Hammad, A., and Lafi, M. (2019). The impact of artificial intelligence on software testing. In *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, pages 565–570.
- [Imai 2022] Imai, S. (2022). Is github copilot a substitute for human pair-programming? an empirical study. In *2022 IEEE/ACM 44th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, pages 319–321.
- [Nguyen and Nadi 2022] Nguyen, N. and Nadi, S. (2022). An empirical evaluation of github copilot’s code suggestions. In *2022 IEEE/ACM 19th International Conference on Mining Software Repositories (MSR)*, pages 1–5.
- [OpenAI 2023a] OpenAI (2023a). Introducing chatgpt. <https://openai.com/blog/chatgpt>. [Online, accessed: 2023-05-27].
- [OpenAI 2023b] OpenAI (2023b). Open ai codex. <https://openai.com/blog/openai-codex>. [Online, accessed: 2023-05-27].
- [Prenner et al. 2022] Prenner, J. A., Babii, H., and Robbes, R. (2022). Can openai’s codex fix bugs?: An evaluation on quixbugs. In *2022 IEEE/ACM International Workshop on Automated Program Repair (APR)*, pages 69–75.
- [Rich 1985] Rich, E. (1985). Artificial intelligence and the humanities. *Computers and the Humanities*, 19(2):117–122.