IA GENERATIVA: IMPACTOS NO DESENVOLVIMENTO DE SOFTWARE[1]

Guilherme Noronha da Silva[2]

Abstract: With advancement in the development of generative artificial intelligence in the year 2022, and with the rapid popularization of tools that use it such as ChatGPT. It was possible to identify a great focus on the potential that these tools can have in different areas of the market. Thus began debates about the impacts that these tools would bring. Using it, it was possible to identify productivity gains in work and studies in different areas, mainly in the area of software development. Therefore, this article aims to study the positive and negative impacts that this new technology has brought to the software development process. A bibliographical research was then carried out to become familiar with these tools and thus obtain a basis to be able to produce this article. It was thus possible to identify the various positive points such as increased efficiency for coding, automation of non-critical tasks, facilitating the production of documentation of the code among others; Some risks and negative points have been identified that can lead to pitfalls and complications when trying to take advantage of the improved efficiency and productivity offered by the tool. In the end, it was possible to conclude that these tools still cannot function perfectly without human interaction to refine the responses they generate.

Key-words: Software development. Artificial intelligence. Generative artificial intelligence.

Resumo: Com o avanço no desenvolvimento da inteligência artificial generativa no ano de 2022, e com a rápida popularização de ferramentas que a utilizam como o ChatGPT. Foi possível identificar um grande foco no potencial que essas ferramentas podem ter em diversas áreas do mercado. Assim teve inicio a debates sobre os impactos que essas ferramentas iriam trazer. Com o uso foi possível identificar ganhos de produtividades em trabalhos e estudos de diversas áreas, principalmente na área de desenvolvimento de software. Com isso esse artigo tem como objetivo estudar os impactos positivos e negativos que essa nova tecnologia veio a trazer para o processo de desenvolvimento de software. Foi então realizada uma pesquisa bibliográfica para se familiarizar com essas ferramentas e com isso obter uma base para poder produzir esse artigo Foi assim possível identificar os diversos pontos positivos como aumento na eficiência para codificação, automatização de tarefas não criticas, facilitar a produção da documentação do código entre outros; Alguns riscos e pontos negativos foram identificados que podem levar a armadilhas e complicações ao tentar tirar proveito da melhora de eficiência e produtividade oferecida pela ferramenta. No final foi possível concluir que essas ferramentas ainda não conseguem funcionar perfeitamente sem a interação humana para refinar as respostas que elas geram.

Palavras-chave: Desenvolvimento de software. Inteligência artificial. Inteligência artificial generativa.

---

1 Artigo apresentado como requisito parcial para a conclusão do curso de Sistemas de Informação, da Universidade do Sul de Santa Catarina – UNISUL, orientado pelo professor Claudio Henrique da Silva.

2 Guilherme Noronha da Silva. E-mail: guinoronha7@gmail.com.

1 Introdução

Com início no ano de 2022, a inteligência artificial teve um significativo avanço em seu desenvolvimento, principalmente devido aos lançamentos de ferramentas de inteligência artificial generativas, alcançando um público amplo fora da área de desenvolvimento e, com isso, começou a impactar diversas áreas no mercado de trabalho. Uma dessas área impactadas é a de desenvolvimento de software. Segundo Ebert e Louridas (2023, pg. 2 Tradução nossa), "Quando o ChatGPT foi finalmente lançado para um amplo público em 2022, a corrida armamentista de IA começou em uma velocidade nunca vista antes. O ChatGPT levou apenas dois meses para atingir cem milhões de usuários."

Com a alta demanda por desenvolvimento de software e com isso desenvolvedores, existe uma grande expectativa para que a IA ajude a cobrir essa demanda.

O impacto que a inteligência artificial mostra no cenário de desenvolvimento está principalmente em auxiliar os desenvolvedores tanto na automatização de tarefas, permitindo o foco em trabalhos mais críticos, ou tentando melhorar a eficiência que ele produz com sugestões e recomendações, ajudando a encontrar problemas rapidamente e escrever um código mais eficiente.

> É provável que os próximos anos tenhamos ainda mais ferramentas baseadas em IA apoiando ou até mesmo atuando sozinhas no desenvolvimento de software. Essas ferramentas continuarão a automatizar tarefas, melhorar a eficiência e ajudar os desenvolvedores a criar um software melhor. (Oliveira, 2023)

Essas ferramentas estão gerando impactos tanto positivos quanto negativos na área. Por exemplo, um impacto positivo é o aumento da eficiência no desenvolvimento de aplicações, com ferramentas que auxiliam na codificação por meio de sugestões, recomendações ou até mesmo automatizando algumas etapas como testes ou documentação, e para um exemplo de impacto negativo temos vazamento de códigos importantes e privados de empresas, esses vazamentos geralmente ocorrem devido ao mal uso das ferramentas pelos usuários. Com mais pontos positivos que negativos está sendo mais vantajoso para desenvolvedores aprenderem a utilizar essas ferramentas para aumentar a produtividade em seus trabalhos.

Assim esse artigo aborda o tema mostrando os impactos positivos e negativos do uso da inteligência artificial generativa para o desenvolvimento de software permitindo que se tenha uma ideia do estado atual da mesma e quais possíveis beneficios se pode obter do uso dela, além de mostrar algumas das ferramentas que podem ser usadas para auxiliar nesse

processo. O artigo esta dividido em quatro seções: introdução, conceitos teóricos, resultados e considerações finais.

2. Seção teórica

Nessa sessão serão definidos alguns dos conceitos que são necessários para se ter uma melhor compreensão com o tema abordado pelo artigo.

2.1 Software

Software é um conjunto de instruções interpretadas por computadores para a realização de ações. É o termo usado para referenciar as aplicações e programas de dispositivos eletrônicos. Costa (2020) comenta que “Em termos práticos, o software é uma coleção de dados ou instruções que informam a um mecanismo como trabalhar. Ele, nada mais é do que um programa que você acessa no celular, tablet, PC, ou qualquer outro dispositivo eletrônico.”

2.2 Desenvolvimento de software

Segundo Cunha (2022) "O desenvolvimento de softwares consiste na criação e implementação de soluções de informática utilizando linguagens de programação." Para Hudson (2007) trata-se de "Um processo de desenvolvimento de software pode ser visto como um conjunto de atividades organizadas, usadas para definir, desenvolver, testar e manter um software." Essas atividades são divididas em etapas que podem ser diferentes conforme a metodologia usada para o desenvolvimento. Para este trabalho utilizou-se as etapas definidas por Hudson (2007): Levantamento de requisitos; Análise de Requisitos; Projeto; Implementação; Testes e Implantação.

Figura 1 – Fluxograma ilustrativo de um Desenvolvimento de Software

Fonte: Página Arkan System.[3]

2.2.1 Levantamento de requisitos

No que diz respeito à etapa de Levantamento de Requisitos Hudson (2007) explica que "Esta atividade tem como objetivo, compreender o problema, dando aos desenvolvedores e usuários, a mesma visão do que deve ser construído para resolução do problema." Primeira etapa do desenvolvimento onde a equipe de desenvolvimento alinha com o cliente quais são as necessidades e problemas que precisam ser solucionados para poder realizar a entrega final do software. Negligenciar essa etapa pode acabar travando o projeto em uma etapa futura ou ate o abandono do mesmo. O autor complementa:

> O Levantamento de Requisitos é a etapa mais importante, no que diz respeito ao retorno de investimentos no projeto. Vários projetos são abandonados pelo baixo levantamento de requisitos, ou seja, membros da equipe não disponibilizaram tempo suficiente para essa fase do projeto, em compreender as necessidades dos clientes em relação ao sistema a ser desenvolvido. (Hudson, 2007)

3 Disponível em: https://arkansystem.com.br/desenvolvimento-de-softwares_e_seus_processos/. Acesso em 04 dez. 2023.

2.2.2 Análise de requisitos

Durante esta etapa os analistas do projeto vão criar modelos para representar o funcionamento do software. Esses modelos vão ser usados como base para o desenvolvimento, assim eles precisam atender todas as necessidades do cliente que foram levantadas durante a etapa anterior, e com isso estabelecer as prioridades durante o desenvolvimento do software. Hudson (2007) comenta que "Esta etapa, também chamada de especificação de requisitos, é onde os desenvolvedores fazem um estudo detalhado dos dados levantados na atividade anterior. De onde são construídos modelos a fim de representar o sistema de software a ser desenvolvido." Após definir esses modelos e confirmar que eles estão conforme os requisitos levantados, pode se definir uma estratégia para seguir durante o desenvolvimento. Cunha (2022) comenta que

> Alguns modelos serão construídos para representar o software a ser desenvolvido, e a partir desses modelos serão feitas uma análise de validação e outra de verificação. A análise de validação assegura que o software está atendendo de fato as necessidades do cliente. Já a análise de verificação observa se os modelos construídos na análise cumprem os requisitos do cliente. (Cunha 2022)

2.2.3 Projeto

Nessa etapa são definidos alguns aspectos técnicos que vão ser usados para o desenvolvimento do software, como a arquitetura do sistema, linguagem de programação a ser utilizada e padrão de interface gráfica a ser usada. Nas palavras de Hudson (2007) "Nesta fase é que deve ser considerado, como o sistema funcionará internamente, para que os requisitos do cliente possam ser atendidos.".

2.2.4 Implementação

Essa fase utiliza dos dados definidos na etapa anterior para começar o desenvolvimento, iniciando a codificação do software para implementar os requisitos levantados nas etapas anteriores e obter uma aplicação que pode ser testada. Em projetos com baixa complexidade essa etapa pode vir junto da etapa anterior.

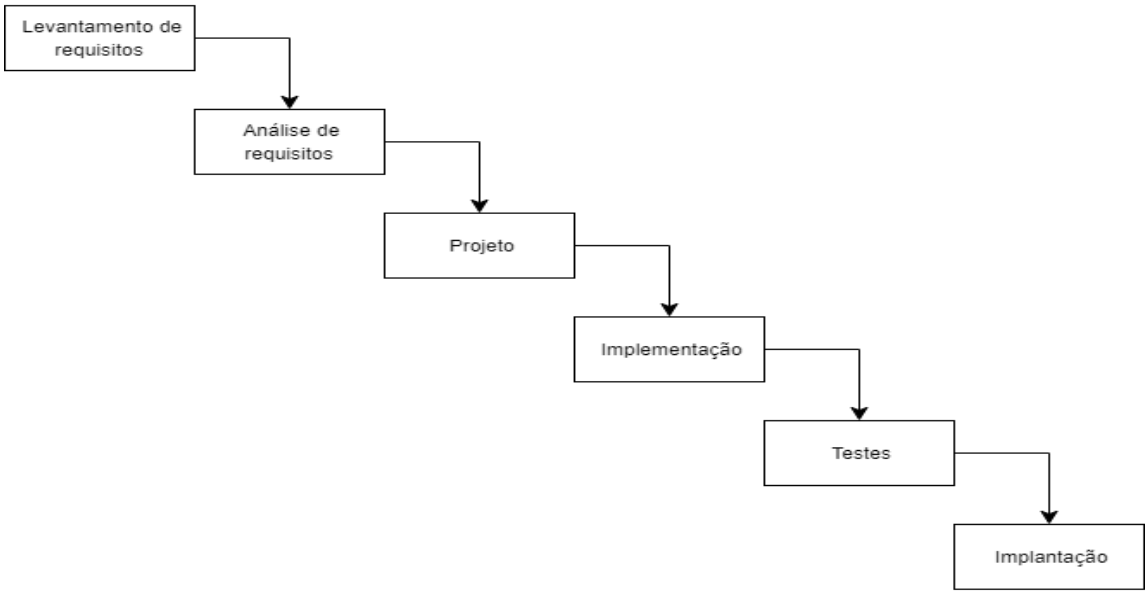
2.2.5 Testes

Etapa que é testada a implementação dos requisitos simulando como o usuário final utilizara a aplicação, para garantir que não tenham erros na aplicação. Para cada erro encontrado é repassado a equipe de desenvolvimento para poderem ser corrigidos e novos testes serem aplicados. O objetivo dessa etapa é garantir que a aplicação vai ser entregue funcional e de acordo com o desejo do cliente. Nas palavras de Cunha (2022) "Dispensar esta etapa é um grande erro, pois testar é essencial para a entrega de um software funcional e estável.".

2.2.6 Implantação

Etapa onde é repassado o software para o cliente, colocando a aplicação no servidor do cliente, e realizando o repasse de documentação e ou treinamento necessários para a o uso correto do sistema. "No entanto ainda é responsabilidade da desenvolvedora fazer com que o cliente esteja integrado e adaptado ao software que foi entregue a ele." (Cunha 2022).

Figura 2 – Fluxo das Etapas de Desenvolvimento definidas por Hudson (2007)



Fonte: Elaborada pelo autor.

2.3 Inteligencia artificial (IA)

O conceito de inteligência artificial é amplo e difícil de se definir, assim recebe diversas definições "As definições também mudaram ao longo do tempo, devido aos rápidos desenvolvimentos. As definições mais recentes falam em “imitar o comportamento humano inteligente”, o que já é uma definição muito mais forte." (Kok et al. 2009 Tradução nossa).

> ...afirmam que a inteligência artificial é a capacidade dos sistemas cibernéticos (formados por computadores, softwares, sensores e atuadores) de imitar funções cognitivas dos seres humanos, funções estas que podemos resumir na resolução de problemas por meio do aprendizado apoiado na percepção. (Russel e Norvig 2013 apud OLIVEIRA 2018, pg. 11)

Assim temos como as características básicas de uma inteligência artificial a capacidade de reconhecer padrões, aprendizado e aplicar regras logicas. Essas características permitem a IA tentar imitar a inteligência humana e com isso solucionar problemas em um tempo que humanos normalmente não conseguem. Atualmente alguns modelos de IA tentam simular as capacidades criativas também.

Diversas disciplinas realizam pesquisas nesse campo de estudos, com um avanço maior na área da computação. Recentemente com a computação moderna a inteligência artificial começou a receber mais foco e permitindo ter um grande avanço na área.

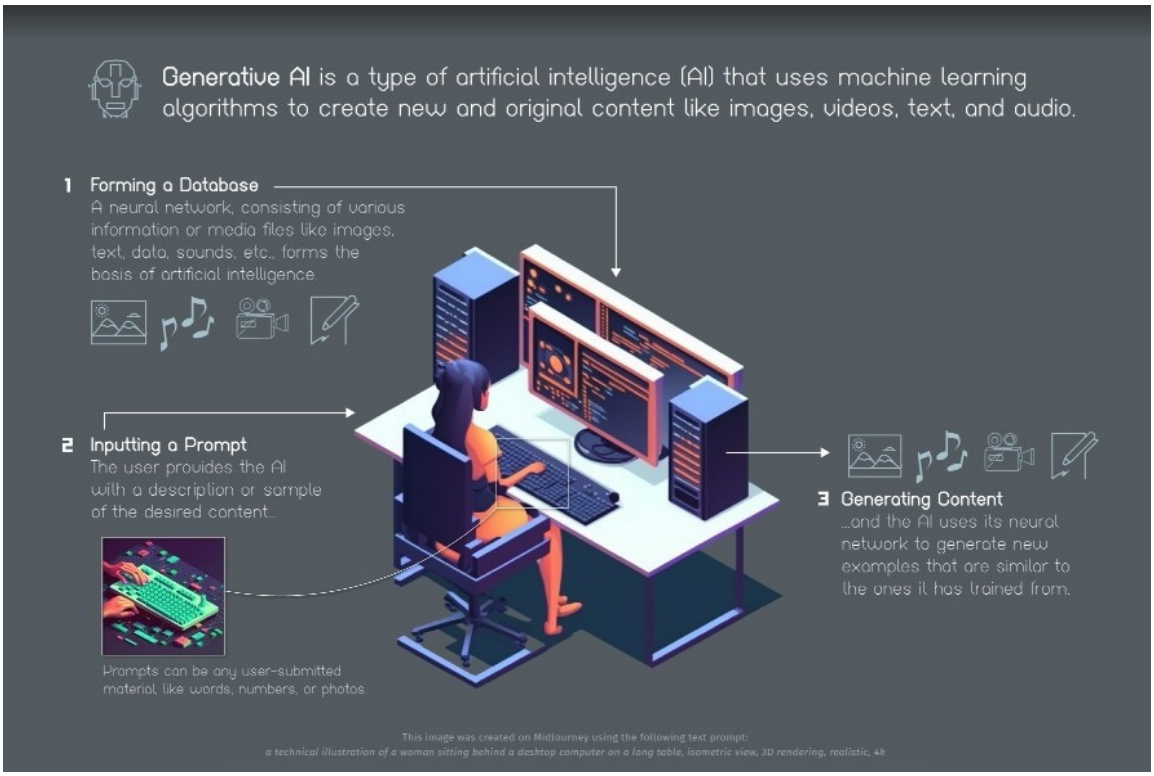
2.4 Inteligencia artificial generativa

Inteligência artificial generativa ou IA generativa é um tipo de IA utilizado para geração de novos textos, imagens ou outros formatos, utilizando como base os dados usados para o treinamento do modelo utilizado e comandos em linguagem comum, como o português e o inglês.

A citação a seguir foi criada por uma IA:

> Inteligência artificial generativa refere-se a uma categoria de algoritmos de inteligência artificial que geram novos resultados com base nos dados nos quais foram treinados. Ao contrário dos sistemas tradicionais de IA, projetados para reconhecer padrões e fazer previsões, a IA generativa cria novos conteúdos na forma de imagens, texto, áudio e muito mais. (Routley, 2023 Tradução nossa)

Figura 3 - Inteligência artificial generativa



Fonte: Adaptado de Routley 2023

Ainda sobre a IA generativa, Ebert e Louridas (2023, pg. 2 Tradução nossa) comentam que

> Como o nome sugere, a IA generativa pode sintetizar, ou gerar, as respostas às perguntas que você faz. Em vez de buscar uma resposta pré-fabricada, como fazem os mecanismos de pesquisa clássicos, ele criará uma resposta para você. A resposta é baseada em grandes quantidades de dados nos quais foi treinado, como aqueles arquivados e indexados por motores de busca. Para fornecer respostas significativas, a IA Generativa passa por treinamento adicional com base em feedback humano. (Ebert e Louridas 2023, pg. 2 Tradução nossa)

3. Resultados

Nessa Seção vai ser abordado os impactos e usos da IA Generativa para área de desenvolvimento de software. Atualmente muitos cientistas acreditam que essa tecnologia vai alterar drasticamente a área de desenvolvimento de software principalmente devido aos grandes investimentos que essa área esta recebendo.

> O impacto da IA generativa está se tornando aparente rapidamente, mas ainda está em seus primeiros dias. Apesar disso, já estamos vendo uma proliferação de aplicativos, produtos e projetos de código aberto que usam modelos de IA generativa para alcançar resultados específicos para pessoas e organizações. (GUIA... 2023)

3.1 Impactos positivos

Pelas palavras de Ebert e Louridas (2023, pg. 4 Tradução nossa) "A IA generativa tem o potencial de melhorar significativamente a produtividade do software, automatizando tarefas, aumentando a criatividade, melhorando a precisão e a eficiência e simplificando os processos de desenvolvimento."

Além do auxilio direto no desenvolvimento, a ia generativa pode ser usada como uma ferramenta para auxiliar no aprendizado de assunto da área, tanto em facilitando a pesquisa feita pelos usuários quanto explicando trechos de códigos fornecidos.

A IA generativa tem o potencial de melhorar diversos pontos do desenvolvimento de software, esses aspectos são listados da seguinte forma por Ebert e Louridas (2023, pg. 5) :

a) Melhorar a criatividade - Auxiliar os desenvolvedores com criação de novos designs para as logos ou interface de usuário. Permitindo o desenvolvedor ter uma base para ser trabalhada em cima e com isso atingir um resultado satisfatório mais rápido e pratico;

b) Resumo da documentação - Utilizando de solicitações e palavras-chaves é possível ajustar a ia para resumir informações, e com os ajustes das informações passadas é possível refinar e melhorar esse resumo. Isso permite a redução do tempo gasto nessa tarefa;

c) Solução de problemas - "Foi demonstrado que a IA generativa é capaz de exibir habilidades matemáticas e algorítmicas, embora atualmente ainda seja fraca." (Ebert e Louridas 2023, pg. 5 Tradução nossa). Desenvolvedores podem descrever o seu problema e fornecer uma direção de como esse problema deve ser solucionado e a ferramenta ira gerar um pedaço de código usável e a explicação do funcionamento do mesmo para esse problema. No entanto quando maior a complexidade do problema maior a chance de voltar com erros;

d) Desenvolvimento eficiente - Permite a automatização de etapas simples do desenvolvimento, como alguns testes e implantação. "As ferramentas de codificação de IA generativa podem ajudar a automatizar algumas das tarefas mais repetitivas, como testes" (GUIA... 2023);

e) Manutenção do legado - A ia generativa pode ser usada para gerar a documentação de códigos legados, além de possibilitar a tradução dessa

linguagem para uma linguagem mais nova. Como por exemplo Cobol para Python ou Javascript;

f) Melhorar a qualidade do software - Com a capacidade de analisar grandes volumes de dados, a ia generativa pode identificar padrões que desenvolvedores pode deixar escapar. Nas palavras de Ebert e Louridas (2023 pg.5 Tradução nossa) "Isso pode ajudar os desenvolvedores a escrever códigos mais precisos e eficientes e a identificar e corrigir bugs mais rapidamente.";

g) Melhorar a qualidade dos dados - A ia generativa responde baseado na data que ela usa para seu treinamento, com isso é possível ajustar o modelo fornecendo dados específicos e deixando o treinamento ser feito com esses dados. Isso permite ajustar o modelo a gerar respostas mais relevantes para o problema;

e) Alcançar confiança - "Os dados sintéticos desenvolvidos pela IA generativa em casos extremos e cenários críticos facilitam o desenvolvimento, teste, aprovação e homologação de sistemas automáticos, robóticos e autônomos em indústrias críticas." (Ebert e Louridas 2023, pg. 5 Tradução nossa).

Na perspectiva de Pothukuchi et al. (2023, pg 3, b289 Tradução nossa)

> Espera-se que a introdução da capacidade de um sistema de IA no desenvolvimento de software não apenas ajude os programadores a realizar suas tarefas com eficiência e rapidez, mas também ajude as organizações a reduzir a força de trabalho, pois os funcionários podem realizar mais tarefas no mesmo período de tempo. (Pothukuchi et al. 2023, pg 3, b289)

Além das etapas que envolvem teste, codificação e implantação mencionadas anteriormente, é possível aplicar a ia generativa na área de planejamento e levantamento de requisitos como foi visto por Pothukuchi et al. (2023, pg 2, b288 Tradução nossa), "A fase de planejamento pode mudar consideravelmente. A IA generativa pode analisar dados históricos de projetos Agile anteriores, identificar padrões e gerar estimativas mais precisas para tarefas e sprints futuros." e para a fase de levantamento de requisitos a IA pode ajudar a compreender o desejo do cliente e ajudar a definir o escopo e requisitos necessários para o projeto, evitando possíveis ambiguidades e erros nos requisitos.

3.2 Impactos Negativos

Nas palavras de Ebert e Louridas (2023, pg 1 Tradução nossa) "Como observamos muitas vezes na história recente da TI, a percepção da corrida do ouro faz com que as pessoas fechem os olhos diante de riscos óbvios."

Nesse tópico serão apresentados os riscos, limitações e problemas que chegaram com essa nova tecnologia. Como a possibilidade de cortes na força de trabalho, vazamento de dados privados, direitos autorais, informações erradas e códigos maliciosos. Assim precisamos levar em consideração essas limitações e achar formas de mitigar esses problemas para atingir uma melhor confiabilidade no uso dessas ferramentas.

> ...os tipos de problemas que os trabalhadores enfrentam em ambientes do mundo real são provavelmente mais amplos e menos previsíveis. Isto levanta preocupações sobre se as ferramentas de IA serão capazes de fornecer assistência precisa em todas as circunstâncias e – talvez mais importante – sobre se os trabalhadores serão capazes de distinguir os casos em que as ferramentas de IA são eficazes daqueles em que não o são. (Brynjolfsson et al. 2023, pg 8 Tradução nossa)

Brynjolfsson et al. (2023, pg 8 Tradução nossa) mencionam que "Foi demonstrado que o ChatGPT produz informações falsas ou enganosas de maneiras imprevisíveis, gerando preocupação sobre sua capacidade de ser confiável em situações de alto risco." Com essa informação mais as limitações levantadas por Pothukuchi et al. (2023), temos o problema da geração do conteúdo estar errada em caso de um treinamento feito com dados errados, ou ate mesmo ter um resultado incompleto em casos de problemas com alta complexidade, assim a confiança das respostas geradas acaba sendo impactada de negativamente e mantem a necessidade de uma intervenção por parte de um usuário humano para tirar melhor proveito do resultado. Complementando com Ebert e Louridas (2023) mencionam que usuários usando a ia generativa como assistente precisam ter a capacidade de identificar ou verificar se o resultado é correto, ou terão complicações durante o desenvolvimento.

Ebert et al. (2023) afirma que durante o uso dessas ferramentas os usuários precisam estar sempre cientes de casos tem implicação em segurança e privacidade. Com a possibilidade dessas ferramentas analisarem o código do usuário, e possívelmente que utilizem o mesmo como treinamento ou deixe algum rastros na ferramenta, para códigos privados isso seria um problema possibilitando o vazamento do mesmo.

Outro risco que Ebert et al. (2023) menciona esta na segurança oferecida por códigos gerados, as ferramentas podem ser treinadas para inserir código malicioso no resultado. Esses fragmentos de código podem introduzirem maneiras para acessar, manipular ou fornecer dados, com isso violando a segurança do software produzido já que permitiria o acesso a dados importantes por indivíduos que não deveriam ter acesso aos mesmos.

3.3 Ferramentas

Nesse tópico serão apresentadas as ferramentas ChatGPT e GitHub Copilot, duas das mais populares ferramentas que utilizam a IA generativa, em ambas é possível identificar os pontos positivos e negativos apresentados no tópico anterior. ChatGPT sendo a ferramenta mais popular atualmente para IA generativa sendo utilizada por um público dentro e fora da área de tecnologia. O Copilot tem um público mais seleto por ser uma ferramenta especifica para auxiliar os desenvolvedores, principalmente durante a etapa de codificação. Ebert e Louridas (2023, pg 3 Tradução nossa) possuem o seguinte comentário em relação as ferramentas "Estão disponíveis diversas plataformas de software generativo, que permitem transformar instruções simples em código de computador."

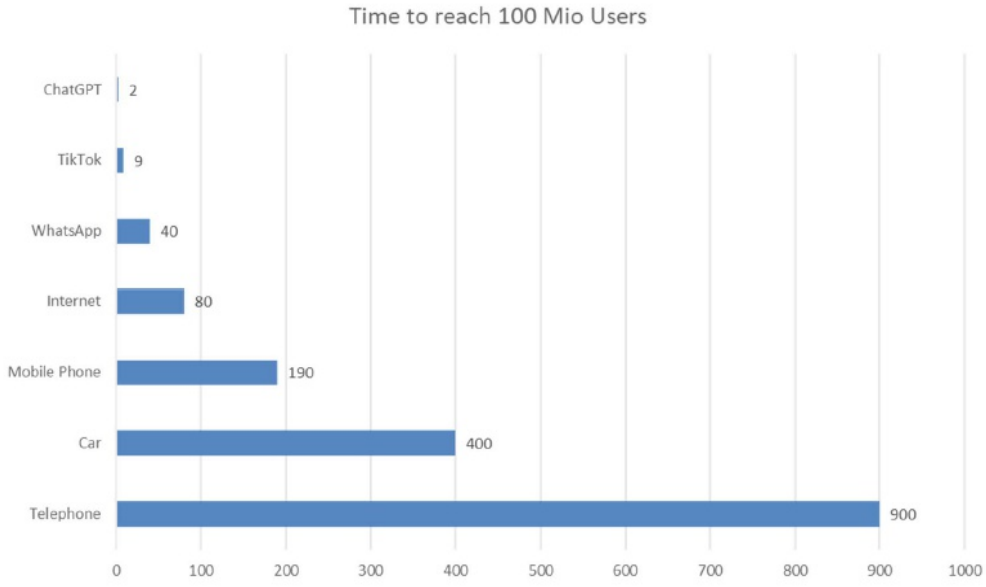
3.3.1 ChatGPT

Atualmente a ferramenta mais famosa, utiliza o Transformadores generativos pré-treinados (GPT) desenvolvido pela OpenAI como modelo para gerar seus textos. Utilizando as versões 3.5 e 4 do GPT, GPT-3.5 e GPT-4. Diferente da API[4] que é usada para o modelo GPT padrão o ChatGPT mantem a instancia que foi usada para inserir a pergunta aberta permitindo refinar a resposta gerada, como se fosse um conversa real. Por ser um modelo pré-treinado a versão gratuita do mesmo só possui dados ate o ano de 2021 e utiliza a GPT-3.5.

Popularizado por ser um chatbot que utiliza linguagem comum, assim permitindo que vários usuários sem grande conhecimento de tecnologia da informação utilizem o ChatGPT de maneira fácil e pratica. Ebert e Louridas (2023, pg. 1 Tradução nossa) mencionam que "O ChatGPT levou apenas dois meses para atingir cem milhões de usuários."

4 *Application Programming Interface*, ou Interface de Programação de Aplicação, em português.

Figura 4 - Tempo ate chegar em 100 milhões de usuarios

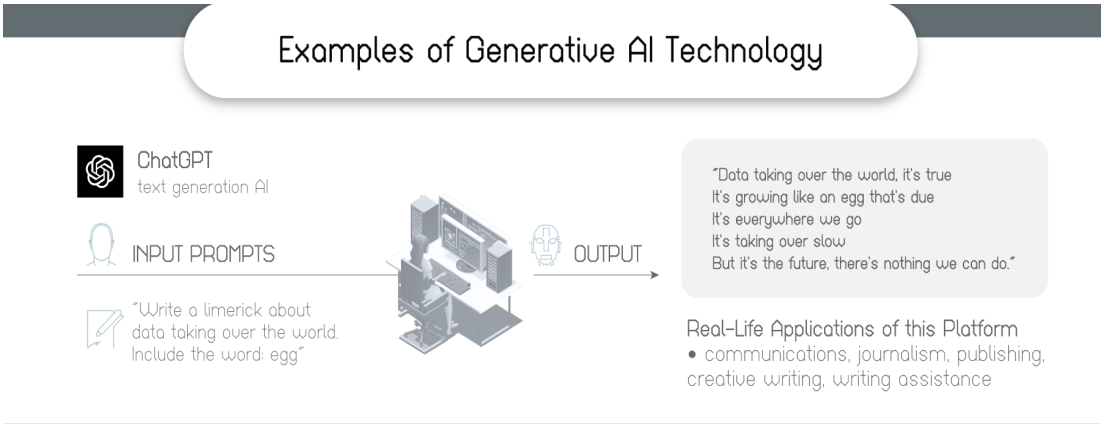


Fonte: (Ebert e Louridas 2023, pg. 2).

Pode ser usada no desenvolvimento de software para facilitar a pesquisa de duvidas ou erros em códigos, facilitando a chegar a uma solução, mas é necessário dados precisos para gerar a resposta, ou que o resultado seja refinado até chegar na solução.

Ebert at al. (2023) também menciona o uso para completar código, ou ate para compreender o código inserido.

Figura 5 - ChatGPT

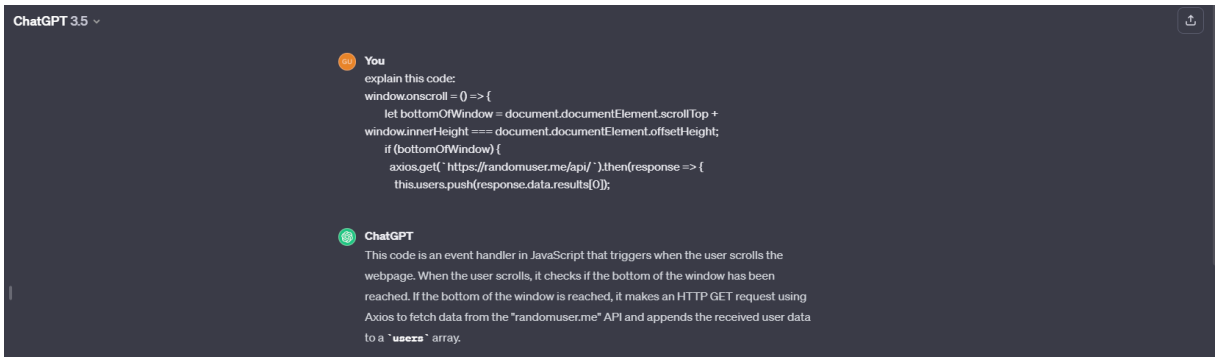


Fonte: Adaptado de Routley 2023

3.3.1.1 Uso do ChatGPT para auxiliar em entender um código

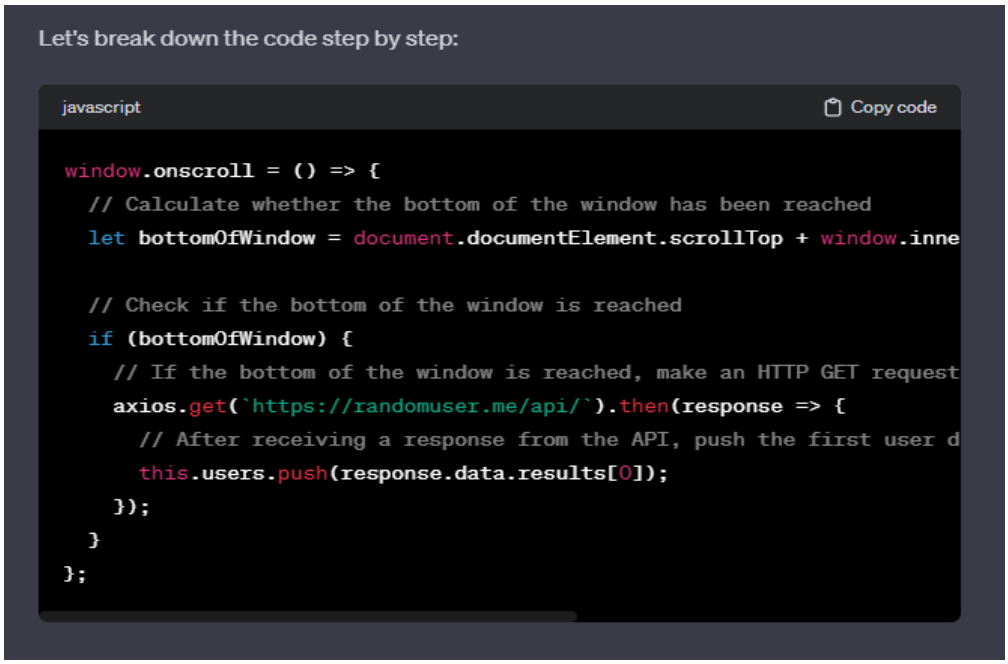
Nas figuras a seguir apresentarei o uso do ChatGPT para entender um código de Scroll Infinito obtido no site da Digital Ocean, no tópico de Como implementar um Scroll Infinito com VUE.js (BERNING, 2021). Na Figura 6 é feito a pergunta para explicar o pedaço de código, na mesma figura também contem a primeira parte da explicação, com as Figuras 7 e 8 contendo a parte mais detalhada da resposta do ChatGPT.

Figura 6 - Pergunta sobre o código ao ChatGPT e primeira parte da resposta



Fonte: Elaborada pelo autor.

Figura 7 - Segunda parte da explicação



Fonte: Elaborada pelo autor.

Figura 8 - Terceira parte da explicação

Explanation:

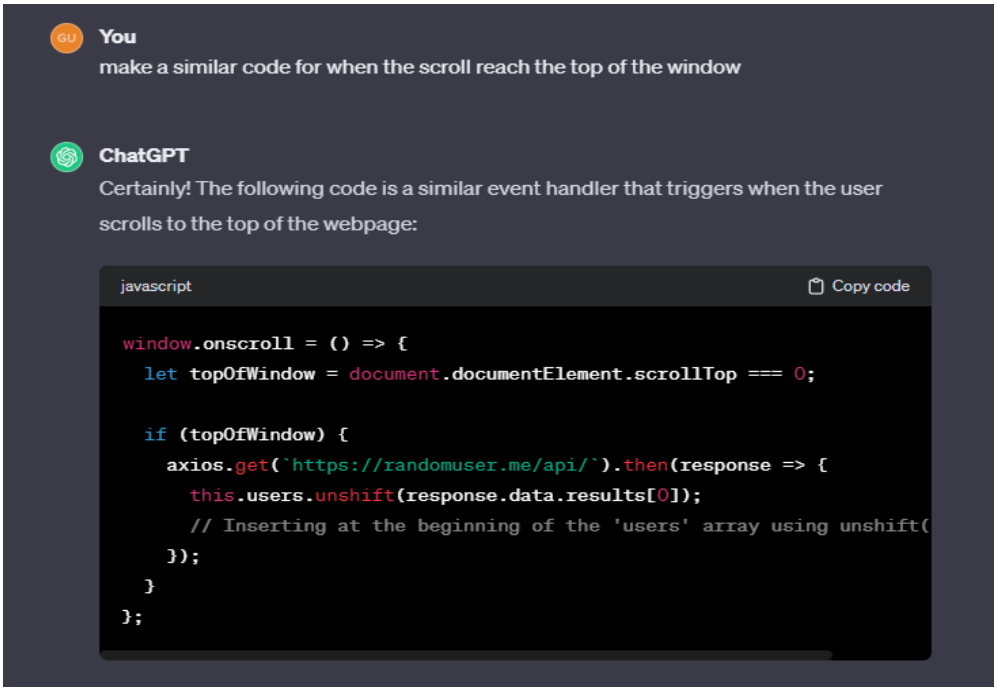
1. `window.onscroll` is an event handler that listens for the scroll event on the webpage.
2. Inside the event handler, `bottomOfWindow` is calculated using the `document.documentElement` properties:
   - `document.documentElement.scrollTop` retrieves the number of pixels that the document's content is scrolled vertically from the top.
   - `window.innerHeight` retrieves the height of the browser window's content area.
   - `document.documentElement.offsetHeight` retrieves the total height of the entire document including the scrollable content.
3. The condition `document.documentElement.scrollTop + window.innerHeight === document.documentElement.offsetHeight` checks if the user has scrolled to the very bottom of the webpage.
4. When the bottom of the window is reached (`bottomOfWindow` is `true`), an Axios GET request is made to `https://randomuser.me/api/` to fetch random user data.
5. Upon receiving a response from the API (`then()` function), the code accesses the first user's data in `response.data.results[0]` and appends it to the `users` array using `this.users.push()`.

This code essentially implements an infinite scroll feature where, upon reaching the bottom of the webpage, it dynamically loads more user data from the "randomuser.me" API and adds it to the existing `users` array.

Fonte: Elaborada pelo autor.

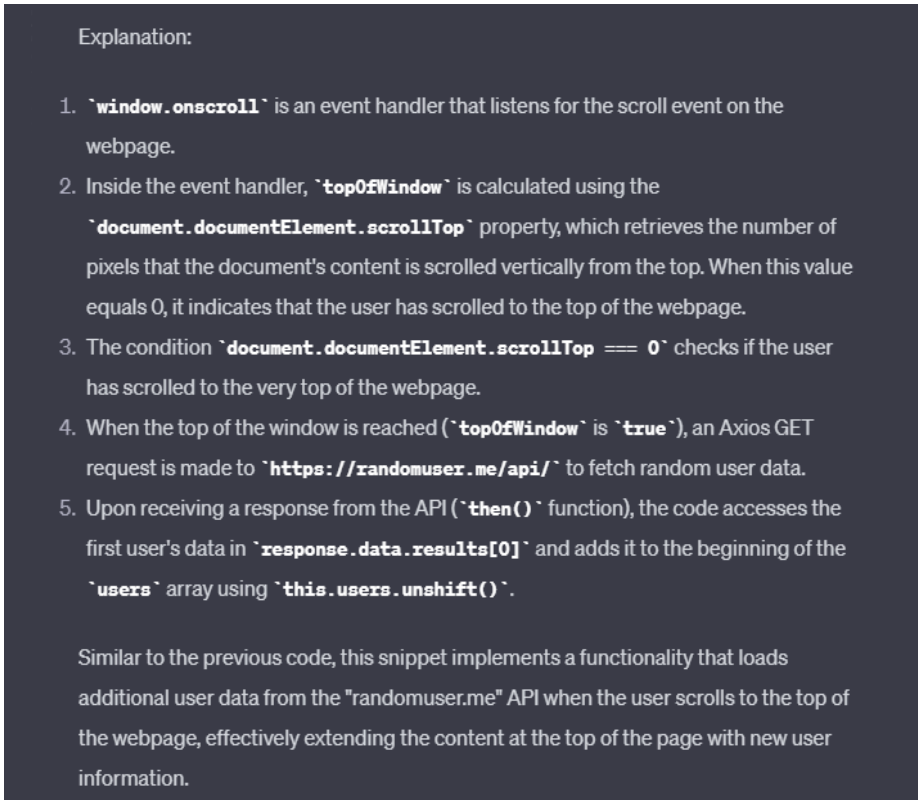
Também é possível pedir para o ChatGPT criar um pedaço de código para uso, para simular isso, nas Figuras 9 e 10 foi pedido para ele criar um código similar ao anterior mas que funciona ao dar scroll para cima ao inves de para baixo como no código anterior.

Figura 9 - Pedido e resposta para a geração de um código para Scroll Infinito



Fonte: Elaborada pelo autor.

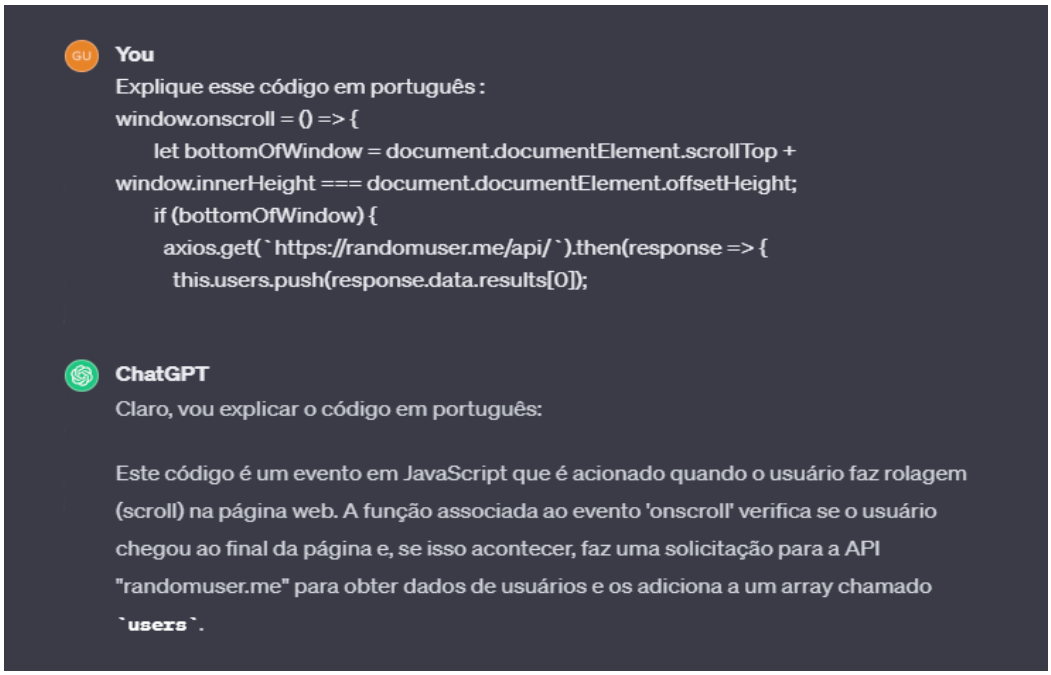
Figura 10 - Segunda parte com a explicação do código



Fonte: Elaborada pelo autor.

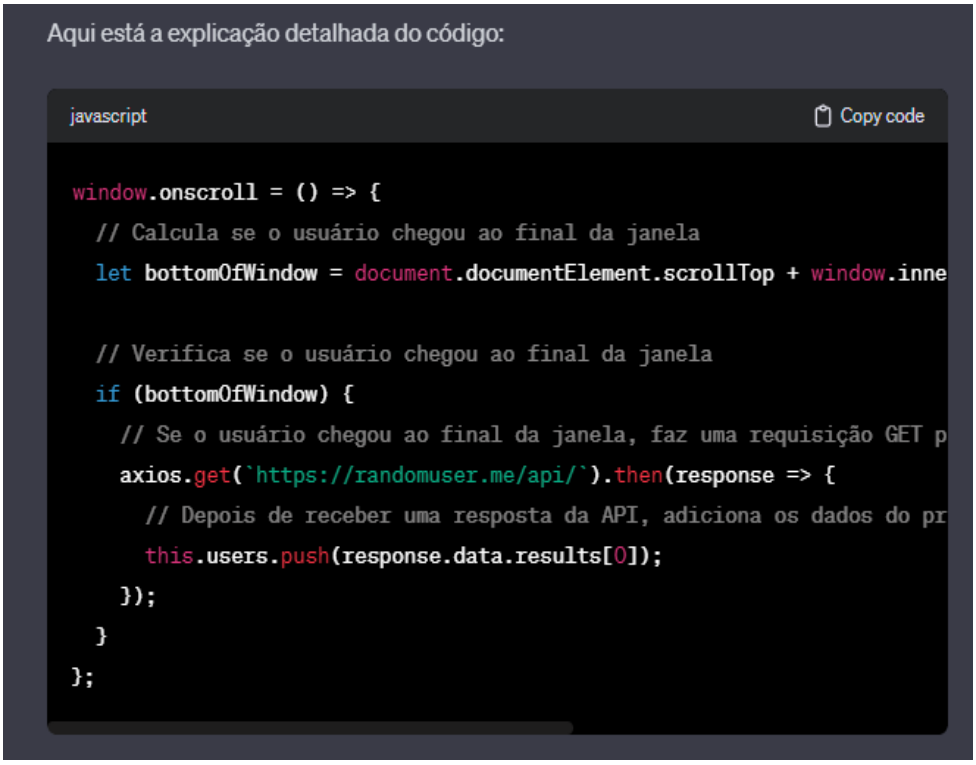
Também é possível realizar o pedido para que a explicação venha em português, utilizando do mesmo exemplo de Scroll Infinito das Figuras 6,7 e 8, foi pedido para o ChatGPT explicar o mesmo trecho de código, só que explicando em português, apresentado nas figuras 11, 12 e 13 a seguir.

Figura 11 - Pedido para explicar o código em português



Fonte: Elaborada pelo autor.

Figura 12 - Segunda parte da explicação em português.



Aqui está a explicação detalhada do código:

javascript Copy code

```
window.onscroll = () => {
  // Calcula se o usuário chegou ao final da janela
  let bottomOfWindow = document.documentElement.scrollTop + window.inne

  // Verifica se o usuário chegou ao final da janela
  if (bottomOfWindow) {
    // Se o usuário chegou ao final da janela, faz uma requisição GET p
    axios.get(`https://randomuser.me/api/`).then(response => {
      // Depois de receber uma resposta da API, adiciona os dados do pr
      this.users.push(response.data.results[0]);
    });
  }
};
```

Fonte: Elaborada pelo autor.

Figura 13 - Terceira parte da explicação em português.

Explicação:

1. `window.onscroll` é um evento que é ativado quando o usuário rola a página.
2. Dentro dessa função de evento, `bottomOfWindow` é calculado usando as propriedades de `document.documentElement`:
   - `document.documentElement.scrollTop` obtém o número de pixels que o conteúdo do documento está deslocado verticalmente a partir do topo.
   - `window.innerHeight` obtém a altura da área visível do conteúdo na janela do navegador.
   - `document.documentElement.offsetHeight` obtém a altura total do documento, incluindo o conteúdo que pode ser rolado.
3. A condição `document.documentElement.scrollTop + window.innerHeight === document.documentElement.offsetHeight` verifica se o usuário chegou ao final da página.
4. Quando o final da janela é alcançado (`bottomOfWindow` é `true`), é feita uma solicitação GET para `https://randomuser.me/api/` usando o Axios para obter dados aleatórios de usuários.
5. Após receber uma resposta da API (`then()`), o código acessa os dados do primeiro usuário em `response.data.results[0]` e os adiciona ao array `users` usando `this.users.push()`.

Em resumo, esse código implementa uma funcionalidade de rolagem infinita que carrega mais dados de usuários da API "randomuser.me" e os adiciona ao array `users` quando o usuário chega ao final da página.

Fonte: Elaborada pelo autor.

3.3.2 GitHub Copilot

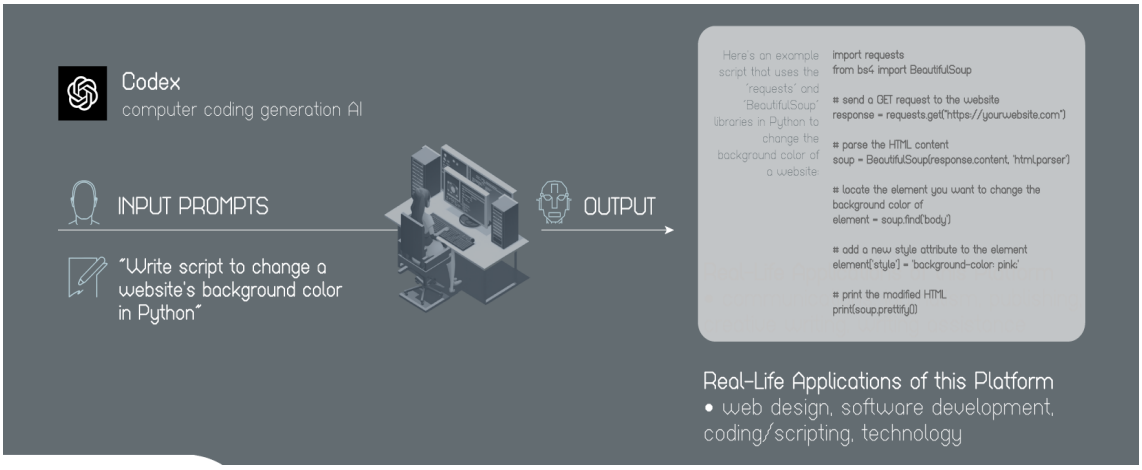
Segundo Ebert at al. (2023, pg 3) o copilot é acessível por diversas ferramentas de desenvolvimentos, como o ambiente de desenvolvimento integrado (IDE) Visual Studio. Utiliza do Codex desenvolvido pela OpenAI, uma nova versão esta em desenvolvimento utilizando o GPT-4 no lugar do Codex, o nome dessa versão é Copilot X .

Pode completar código utilizando um reconhecimento de padrões, funcionando como um assistente, também permite gerar explicações em linguagem comum do código produzido ou selecionado. O uso do Copilot é melhor em linguagens que possuem um maior numero de repositórios públicos no github devido ser treinado com os códigos desses repositórios, como por exemplo Javascript.

Diferente do ChatGPT com sua versão gratuita, durante a produção desse artigo, o Copilot não possui uma versão gratuita, contando apenas com um período para testes gratuito.

> "Peng et al. (2023b) recrutam engenheiros de software para uma tarefa de codificação específica (escrever um servidor HTTP em JavaScript) e mostra que aqueles que têm acesso ao GitHub Copilot concluem esta tarefa duas vezes mais rápido" (Peng et al. 2023b apud Brynjolfsson et al. 2023, pg 4 Tradução nossa)

Figura 14 - Codex



Fonte: Adaptado de Routley 2023

4. Considerações Finais

Como conclusão é possível notar que ainda estamos iniciando no uso e desenvolvimento da inteligência artificial generativa. Apesar do ganho de eficiência durante o desenvolvimento a ideia de uma total automatização para suprir a demanda da área ainda não é possível devido a os resultados incorretos gerados pelas ferramentas. Assim ainda é necessário que a ia generativa seja usada como um assistente no processo de desenvolvimento de software, e que o desenvolvedor tenha um olhar critico para mitigar os possíveis problemas que o uso incorreto ou um resultado errado possam causar. Também é necessário tomar cuidado com o uso extensivo dessas ferramentas para não ser gerado uma dependência das mesmas por parte do desenvolvedor impactando em sua habilidades no desenvolvimento de software. Com isso é possível identificar que o medo de alguns desenvolvedores de serem substituídos ainda está longe de acontecer, mas com o avanço apesar de vagas na área tendam a diminuir com a melhoria das ferramentas, novas vagas que trabalham diretamente no uso, como engenheiro de prompt, e na manutenção da ia generativas vão aparecer.

Sugestões para trabalhos futuros na área podem ser feitos para os avanços que estão para vir nos próximos anos, ou aprofundar a pesquisas nos modelos generativos, ou nas atualizações das ferramentas que utilizam a ia generativa, como o chatgpt usando o gpt-5 e copilot x.

Referências

BERNING, Dave.**How To Implement an Infinite Scroll with Vue.js**. 2021. **Disponível em:** https://www.digitalocean.com/community/tutorials/vuejs-implementing-infinite-scroll. Acesso em: 16 dez. 2023.

BRYNJOLFSSON, E.; LI, D.; RAYMOND, L. R; ***Generative AI at work**, 2023,* (No. w31161). National Bureau of Economic Research.

COSTA, Matheus Bigogno. **O que é software? Entenda o significado. Disponível em:** <https://canaltech.com.br/software/o-que-e-software/>. Acesso em: 18 nov. 2023.

CUNHA, Fernando. **Desenvolvimento de softwares: tudo o que você precisa saber. Disponível em:** <https://www.mestresdaweb.com.br/tecnologias/desenvolvimento-de-softwares-tudo-o-que-voce-precisa-saber>. Acesso em: 18 nov. 2023.

GUIA Completo Sobre Inteligência Artificial Generativa. Disponível em: <https://blog.dsacademy.com.br/guia-completo-sobre-inteligencia-artificial-generativa/>. Acesso em: 18 nov. 2023.

EBERT, C.; LOURIDAS, P; **Generative AI for software practitioners.** *IEEE Software*, 2023, *40*(4), 30-38.

HUDSON; **Atividades básicas ao processo de desenvolvimento de Software. Disponível em:** <https://www.devmedia.com.br/atividades-basicas-ao-processo-de-desenvolvimento-de-software/5413?=undefined>. Acesso em: 18 nov. 2023.

KOK, J. N.; BOERS, E. J.; KOSTERS, W. A.; VAN DER PUTTEN, P.; POEL, M. **Artificial intelligence: definition, trends, techniques, and cases.** *Artificial intelligence,* 2009, 1, 270-299.

OLIVEIRA, Ruy Flávio de. **Inteligência Artificial.** Editora e Distribuidora Educacional S.A. Londrina. 2018.

OLIVEIRA, Christiano. **IA e os Impactos no Desenvolvimento de Software. Disponível em:** <https://pt.linkedin.com/pulse/ia-e-os-impactos-desenvolvimento-de-software-christiano-de-oliveira>. Acesso em: 18 nov. 2023.

POTHUKUCHI, A. S.; KOTA, L. V.; MALLIKARJUNARADHYA, V. **Impact of Generative AI on the Software Development Lifecycle (SDLC).** *International Journal of Creative Research Thoughts*11, no. 8 (2023).

ROUTLEY, Nick. **Infographic: Generative AI Explained by AI. Disponível em:** <https://www.visualcapitalist.com/generative-ai-explained-by-ai/>. Acesso em: 18 nov. 2023.