

Usos da I.A. e Machine Learning na Engenharia de Software

Ian Breda - 211023965

João Martineli - 211021296

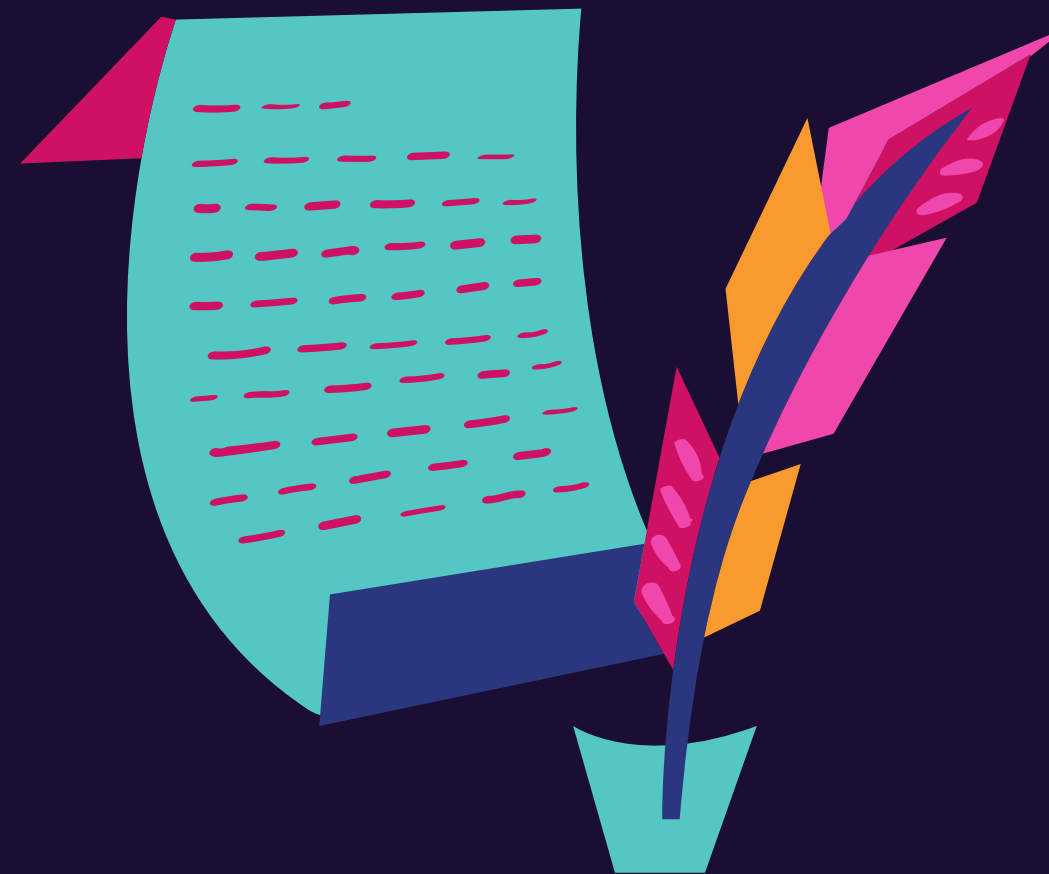
Lucas Fernandes - 211022195

Lucas Yuki - 211024678



Roteiro

- Algoritmos de aprendizado automático para análise de qualidade do código
- Uso da I.A. e M.L. para automatizar os testes de software
- IA como assistente de código
- Análise de métricas e medição de qualidade do código
- Complexidade ciclomática

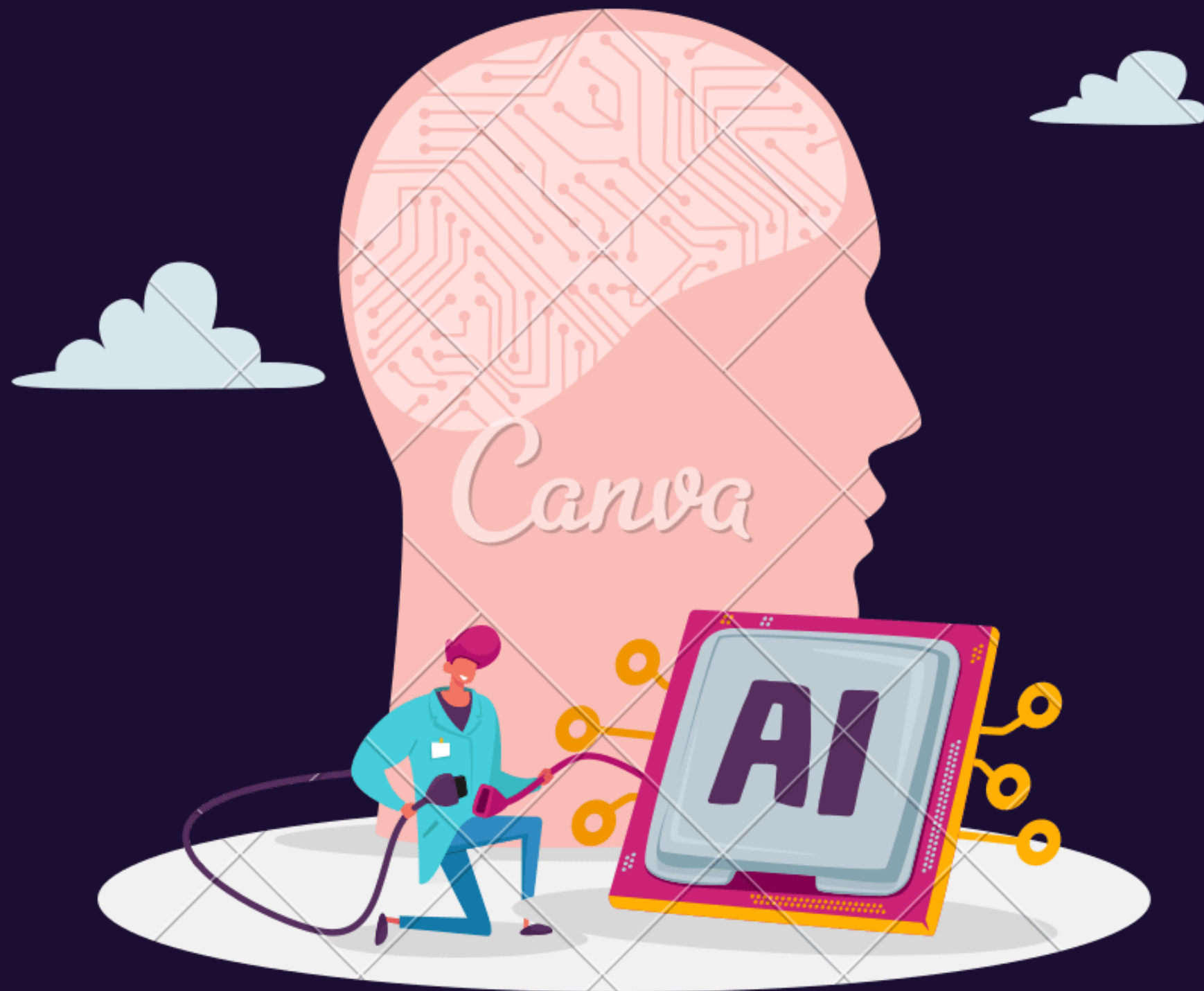


Algoritmos de
aprendizado
automático
para análise
de qualidade
do código.



Uso de técnicas de machine learning para:

- Avaliar a qualidade do código
- Detectar bugs e outros problemas
- Sugerir melhorias



Suas diversas aplicações



1. Avaliar a qualidade do código

Como a IA faz isso?

Aprendizado com parâmetros
de "BOM" e "RUIM"

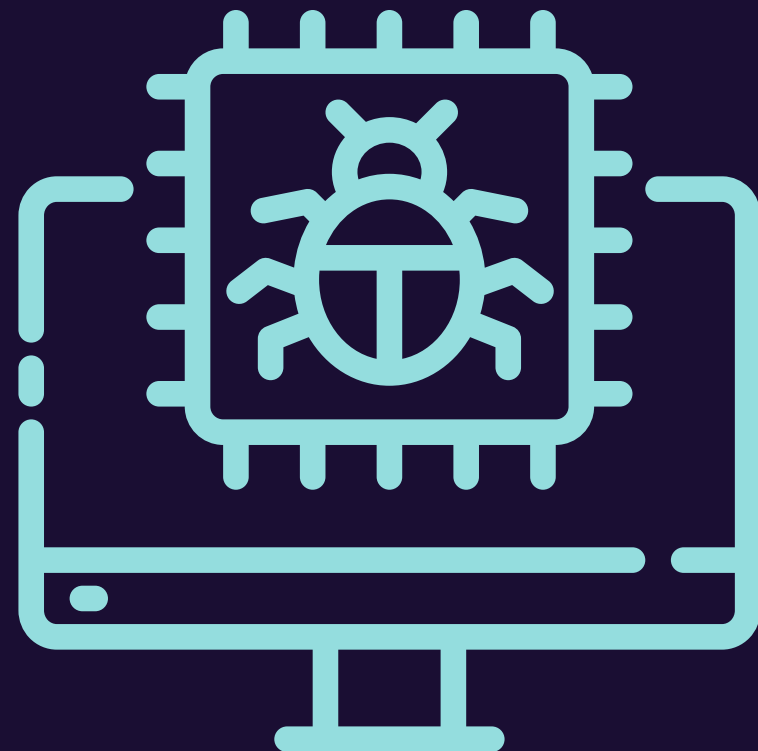
Analisa métricas:
Como número de erros,
cobertura de teste e
complexidade ciclomática

Analise semântica

Compreender o significado
semântico , a fim de avaliar
a qualidade do código e
identificar problemas.

2. Detectar bugs e outros problemas ⚠

Deteção de anomalias: Algoritmos de aprendizado não supervisionado podem ser usados para detectar padrões e anomalias no código.



- **Deteção de código duplicado**
- **Identificação de código complexo ou de baixa qualidade**
- **Deteção de problemas de segurança.**



3. Sugerir melhorias

Algoritmos de aprendizado por reforço:

Otimizar o processo, sugerindo melhorias no código e na estrutura.

**Sugestão de refatoração;
Identificação de código a ser reutilizado;**

A identificação de problemas de performance;



**Análise de padrões:
Algoritmos de
aprendizado
automático analisam
padrões no código,
identificando áreas
que podem ser
melhoradas e
sugerindo melhorias
específicas.**





Gerar código: Os algoritmos de aprendizado automático podem ser treinados para gerar código a partir de uma descrição de uma tarefa ou problema, e ser usado para sugerir melhorias no código existente ou escrever novo código.

Uso da I.A. e M.L. para automatizar os testes de software

Existem 3 áreas principais para sua aplicação:

1. Criação de testes
2. Análise de testes
3. Manutenção de testes

Os testes de software são uma parte essencial do desenvolvimento do software, e a inteligência artificial vem para transformá-los.



Criação de testes de software

Usam-se frameworks que interagem com o software, controlados por scripts. A criação e teste de novos scripts pode levar horas.

A Inteligência Artificial pode reduzir o tempo de criação dos novos scripts para minutos, o automatizando enquanto ele é executado manualmente pelo testador.



Análise de testes de software

O script do teste é feito para testar partes específicas em mente. É possível deixar de testar partes do sistema ou ocorrerem falsos positivos.

A I.A. pode fazer uso da visão computacional e do M.L. para garantir que mais partes sejam testadas e diminuir o número de falsos positivos.



Manutenção de testes de software

Atualizações do sistema levam a mudanças na interface, e por consequência, o mal funcionamento dos testes, o que requer manutenção.

Aplicar um modelo de M.L. no sistema pode aliviar a demanda de manutenção dos testes, uma vez que a I.A. pode identificar mudanças feitas na interface e alterar os scripts de teste de acordo.



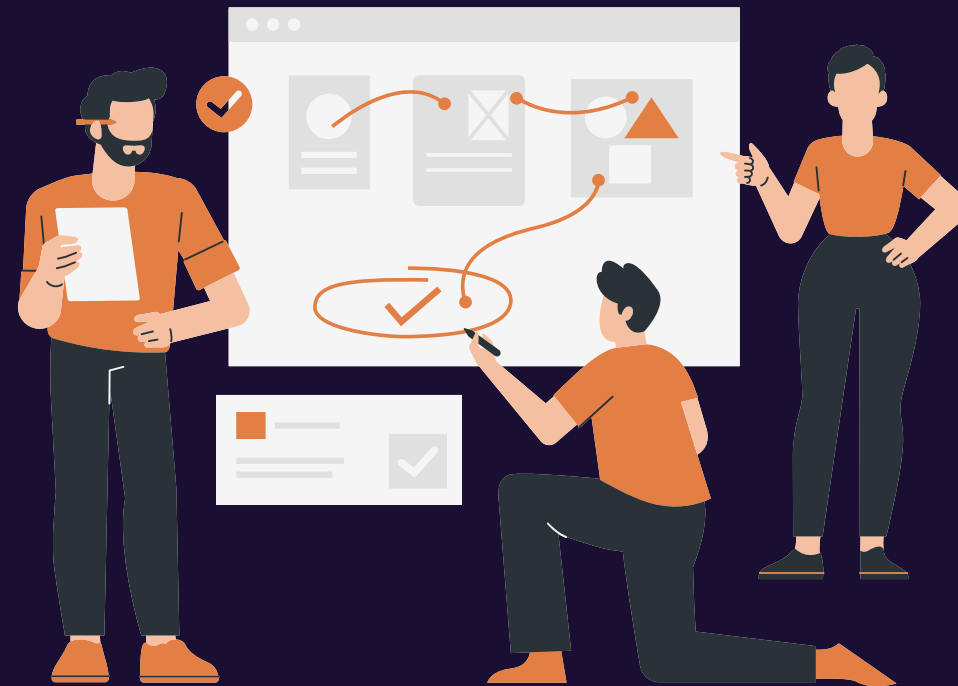
IA como assistente de código

Plugins de preenchimento automático de código que permitem aos desenvolvedores que escrevam códigos com mais rapidez e precisão



1. Prototipagem rápida

Escrita de código por pessoas menos especializadas utilizando linguagem natural ou interfaces visuais

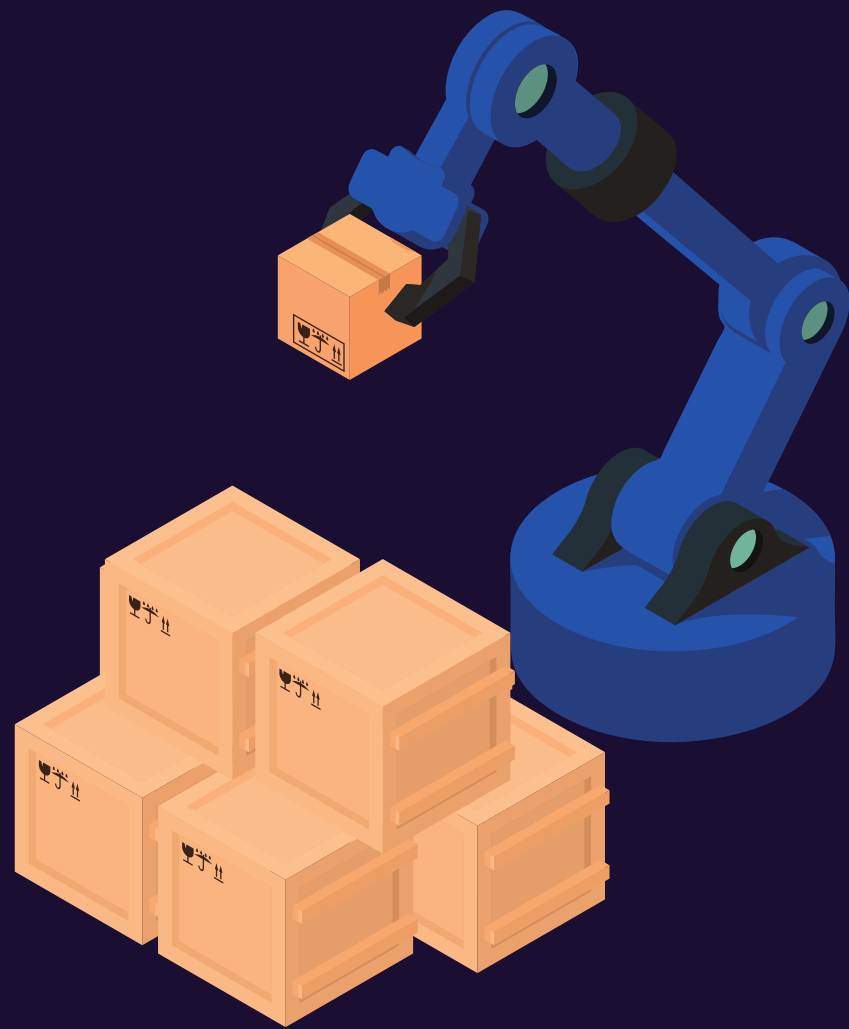


2. Programação inteligente

Ajuda e recomendações em tempo real: documentos, exemplos de código e práticas recomendadas

3. Análise automática e tratamento de erros

Correção de erros automática durante a escrita do software



4. Refatoração automática de código

Revisão de software já escrito para melhorar fatores como interpretabilidade e desempenho

5. Estimativas precisas

Previsões melhores de tempo e esforço necessário
com base em projetos anteriores



6. Tomada de decisões estratégicas

Analisar desempenho de aplicativos já existentes para ajudar na
decisão de que projetos valem
a pena ser iniciados

Análise de métricas e medição de qualidade do código



As métricas:

- Funcionalidade
- Velocidade e facilidade de uso
- Responsividade
- Escalabilidade
- Manutenção



Qual a importância disso?

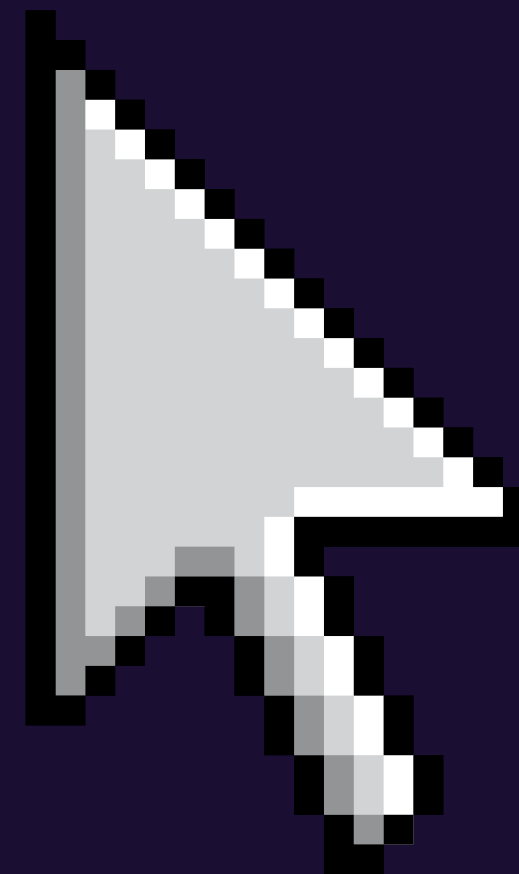
Usando essas métricas, podemos definir se um software vai atender nossas expectativas, se estamos desperdiçando dinheiro ou se ele possui uma tendência para imprevistos.

Complexidade ciclomática

Todo desenvolvedor possui uma noção do quanto um código escrito por ele está bem escrito ou não.

Contudo, num projeto grande, onde muitas mãos são utilizadas, isso foge ao controle.

A complexidade ciclomática é uma medida da complexidade estrutural de um código.



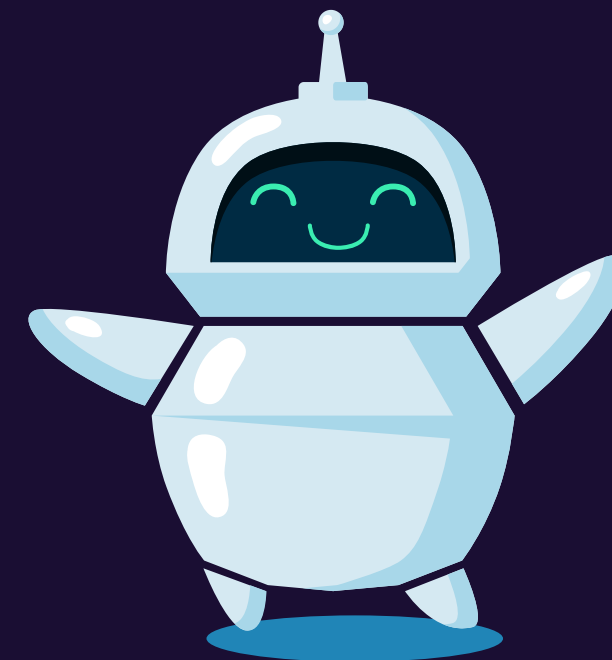
Complexidade ciclomática

A complexidade ciclomática é calculada contando o número de condições e loops. Quanto maior o valor da complexidade ciclomática, maior a complexidade estrutural do código, implicando que o código é difícil de entender, manter e testar, e pode ser mais propenso a bugs e falhas.

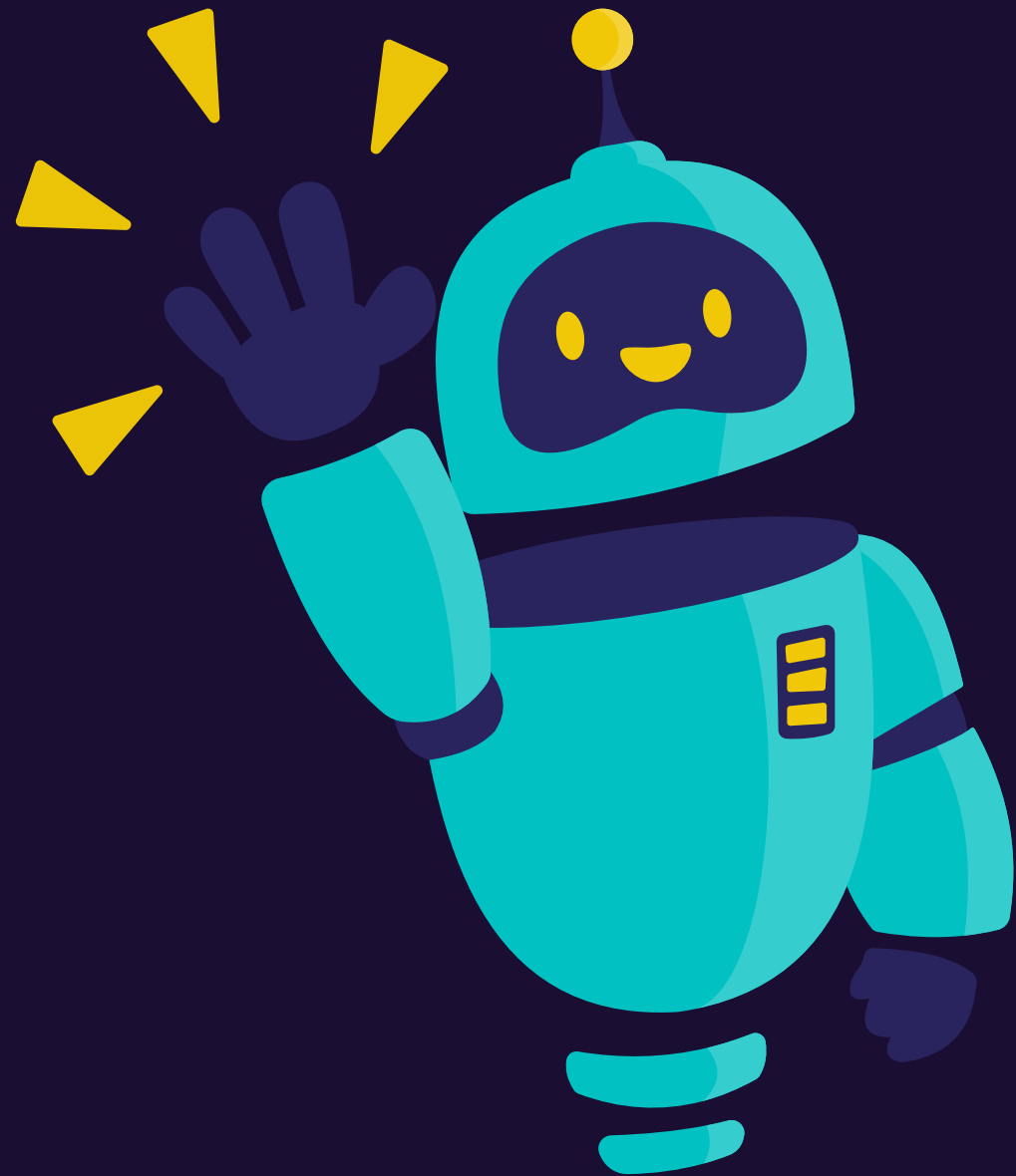


Como a I.A. pode nos ajudar?

Pode automatizar a análise e medição de métricas e complexidade do código, economizando tempo e esforço humano. Além disso, a IA pode identificar padrões e tendências difíceis de detectar manualmente, o que pode ajudar a identificar problemas mais rapidamente e com maior precisão.



OBRIGADO PELA ATENÇÃO!



Fontes utilizadas

<https://www.supero.com.br/blog/desenvolvimento-de-sofware-como-a-inteligencia-artificial-pode-acelerar/>

<https://www.testingcompany.com.br/blog/como-a-inteligencia-artificial-pode-ser-aplicada-nos-testes-de-software>

<https://uds.com.br/blog/qualidade-de-software/#:~:text=Entre%20as%20m%C3%A9tricas%20de%20qualidade,quando%20j%C3%A1%20%C3%A9%20tarde%20demais.>

<https://www.devmedia.com.br/analise-de-codigo-metricas-de-qualidade-e-estimativas/26882>