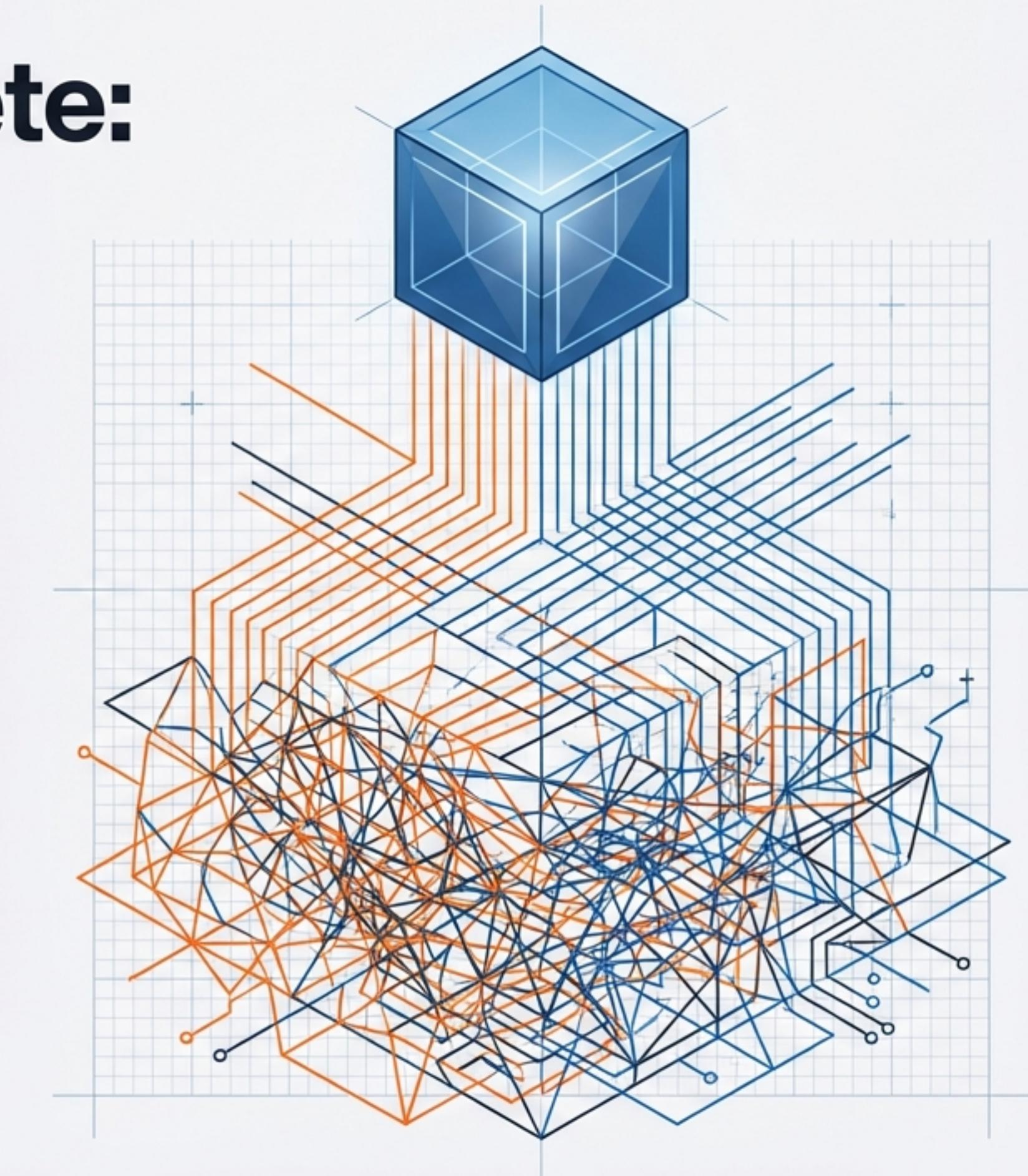


# Além do Autocomplete: O Copilot como Seu Revisor Sênior

Refatoração, Testes e  
Documentação Inteligente  
(Bloco 3)

Inter

Curso de Inteligência Artificial para Desenvolvedores



# A Realidade do Desenvolvimento de Software



Até agora, focamos na criação. Mas o dia a dia real de um programador é gasto limpando a "sujeira" e garantindo que o sistema não quebre.

## O Desafio:

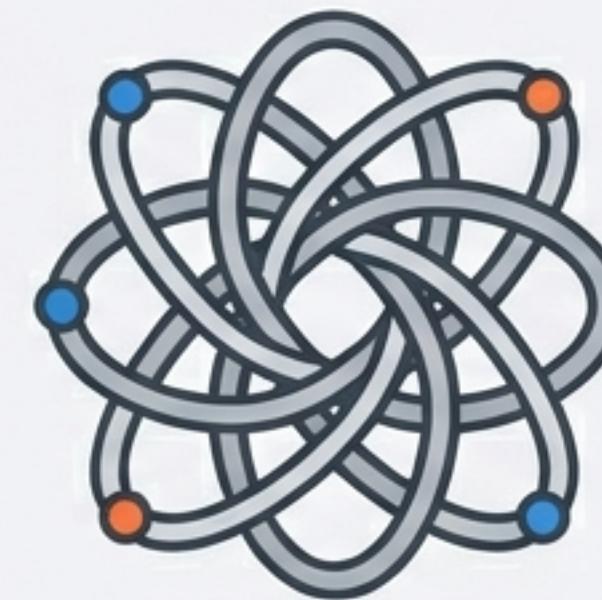
Escrever código é a parte fácil. Manter o código funcionando, limpo e legível é onde reside a verdadeira engenharia.

Vamos transformar o Copilot em um parceiro para dominar esses 80%.

# Refatoração: Limpando a Casa sem Mudar os Móveis

Melhorar a estrutura e a legibilidade sem alterar o comportamento final.

## Cenário Comum:



Funções gigantes, 'ifs' aninhados,  
lógica confusa.

## A Solução:



Refatore esta função para usar o padrão Early Return.

Simplificar lógica e eliminar aninhamentos.

Extraia a lógica de cálculo para uma função separada.

Modularizar blocos gigantes.



**Dica Pro:** Transforme funções monolíticas em blocos modulares. Se é difícil de ler, peça para a IA quebrar em pedaços menores.

Vamos transformar o Copilot em um parceiro para dominar esses 80%.

# Testes Unitários: Sua Rede de Segurança

Criar testes é um trabalho repetitivo onde a IA brilha. O objetivo não é apenas confirmar o “Caminho Feliz”, mas descobrir onde o código quebra.

/tests

(Gera a estrutura básica de testes unitários)

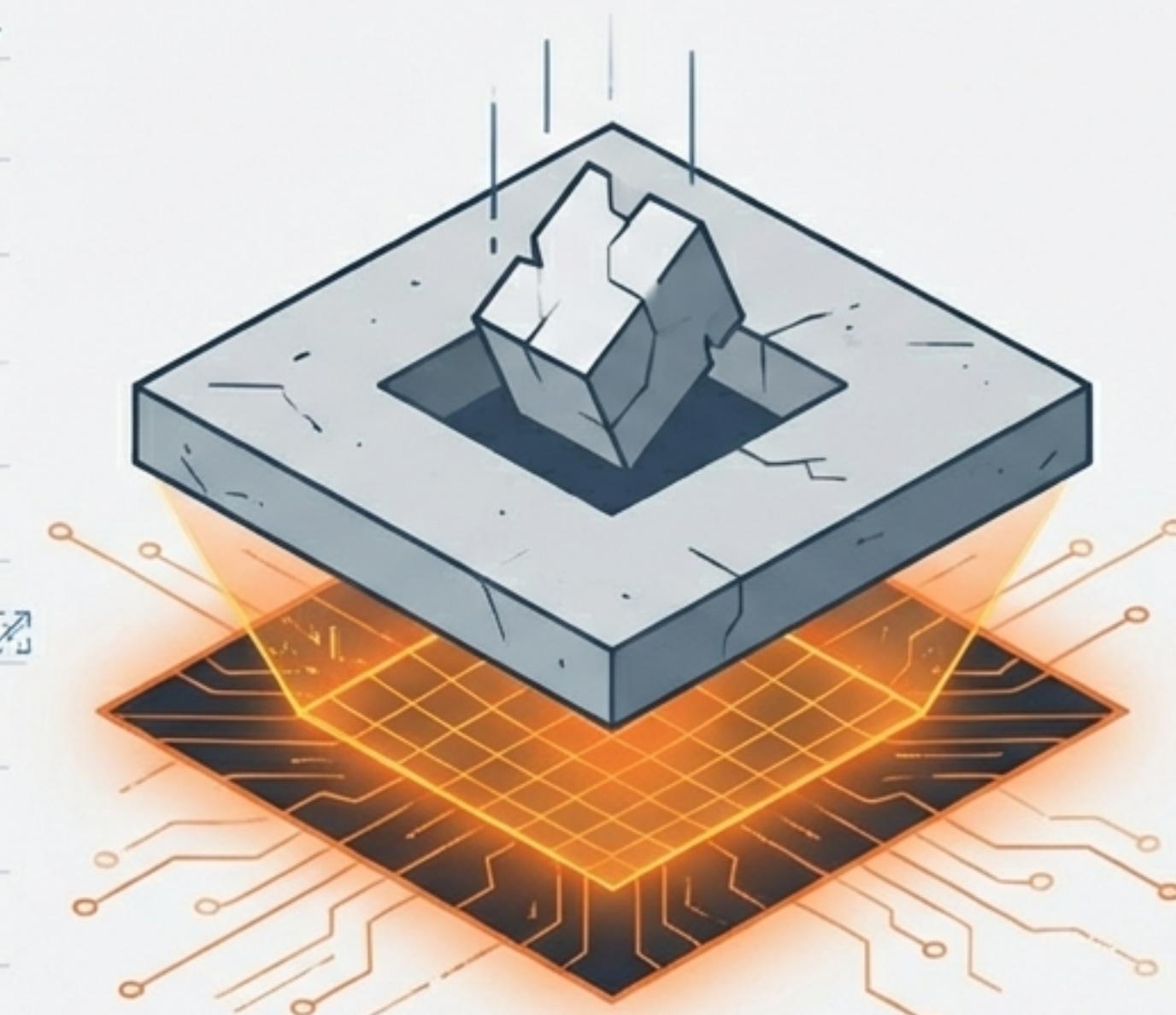
## Estratégia Sênior (Edge Cases)

Não peça apenas um teste simples. Desafie a IA:

Crie testes para esta função, incluindo casos com valor nulo, vazio e números negativos.



O **Copilot** ajuda a cobrir os erros que você esqueceria na pressa do dia a dia.



Vamos transformar o **Copilot** em um parceiro para **dominar** esses **80%**.

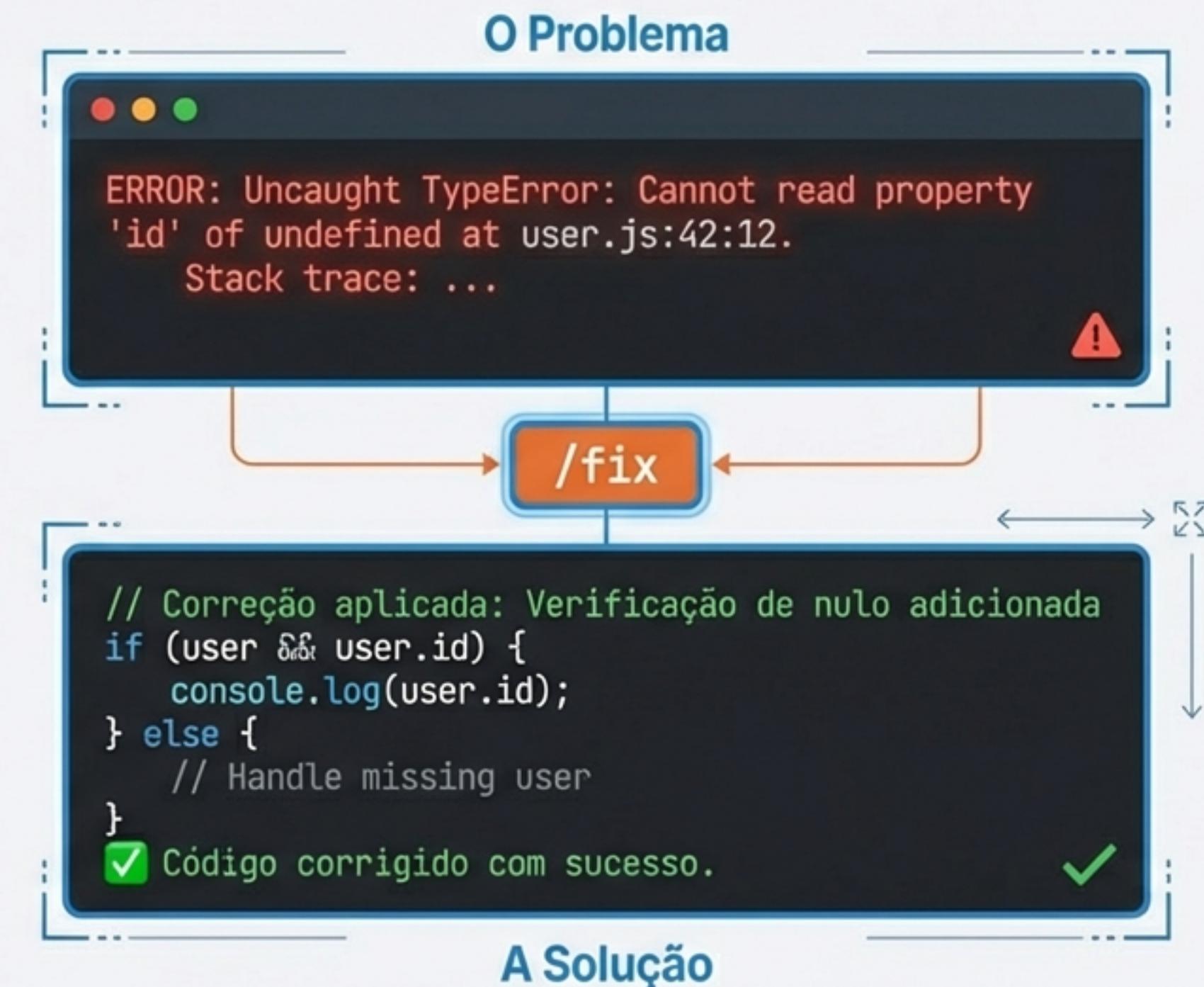
# O Comando /fix: Quando o Console Fica Vermelho

## Workflow de Correção Rápida:

1. Copie o erro do console.
2. Selecione o código problemático.
3. Digite **/fix** no chat.

## Por que usar?

O **/fix** analisa o contexto do seu código e a mensagem de erro simultaneamente. Uma solução explicada que poupa o tempo de pesquisa no Stack Overflow.



Vamos transformar o **Copilot** em um parceiro para dominar esses **80%**.

# O Olho do Sênior: Análise Estática e Performance

Use a IA como um consultor antes da entrega final. Pergunte: "O que eu esqueci?"

## Segurança

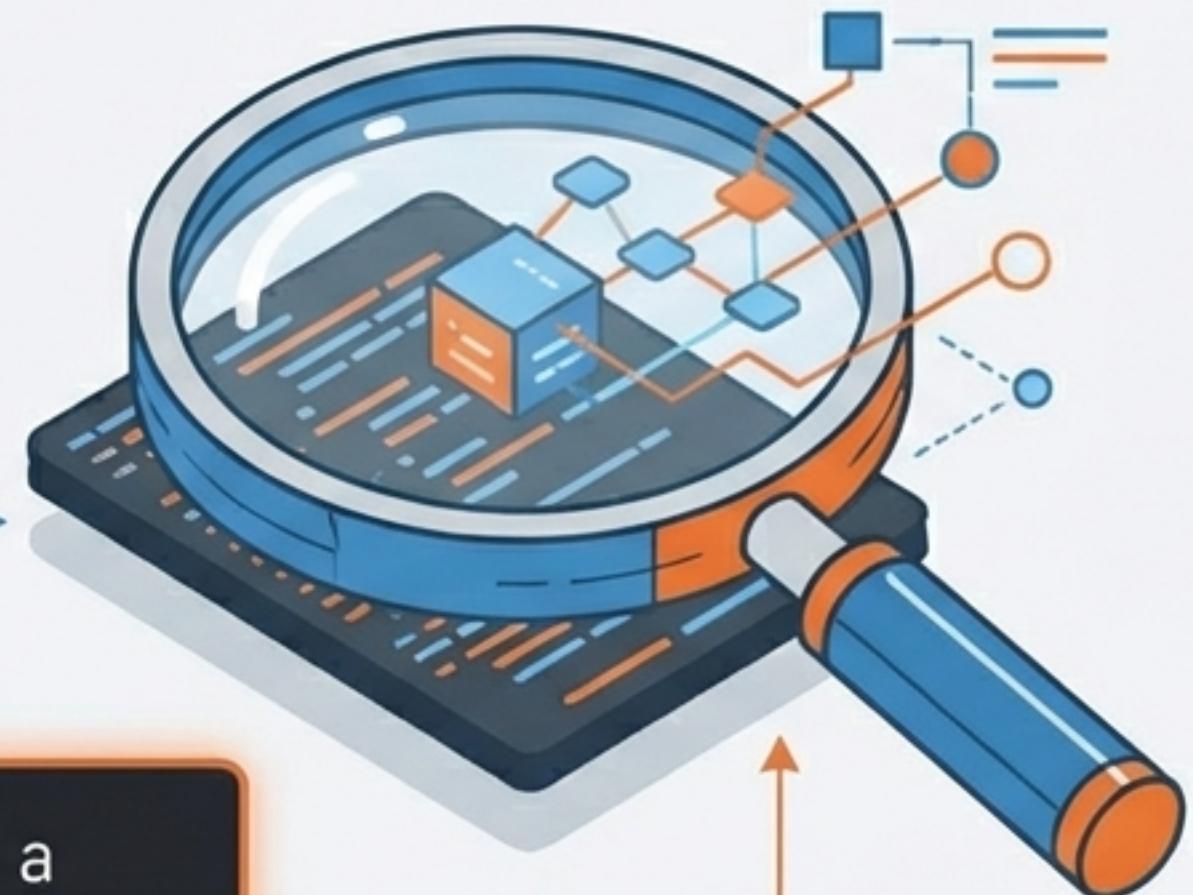
Existe algum problema de segurança neste código?

Identifica vulnerabilidades invisíveis.

## Performance

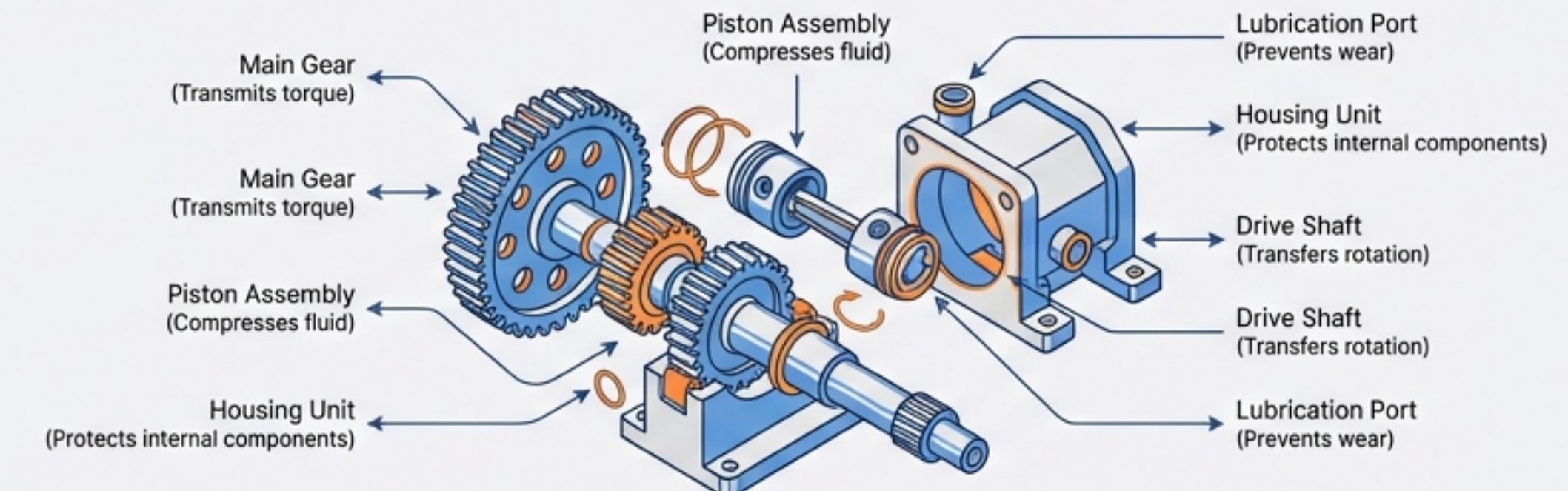
Como posso diminuir a Complexidade Big O desta função?

Otimização para processamento de grandes volumes de dados.



Vamos transformar o Copilot em um parceiro para dominar esses 80%.

# Documentação com Propósito (Javadoc & Docstrings)



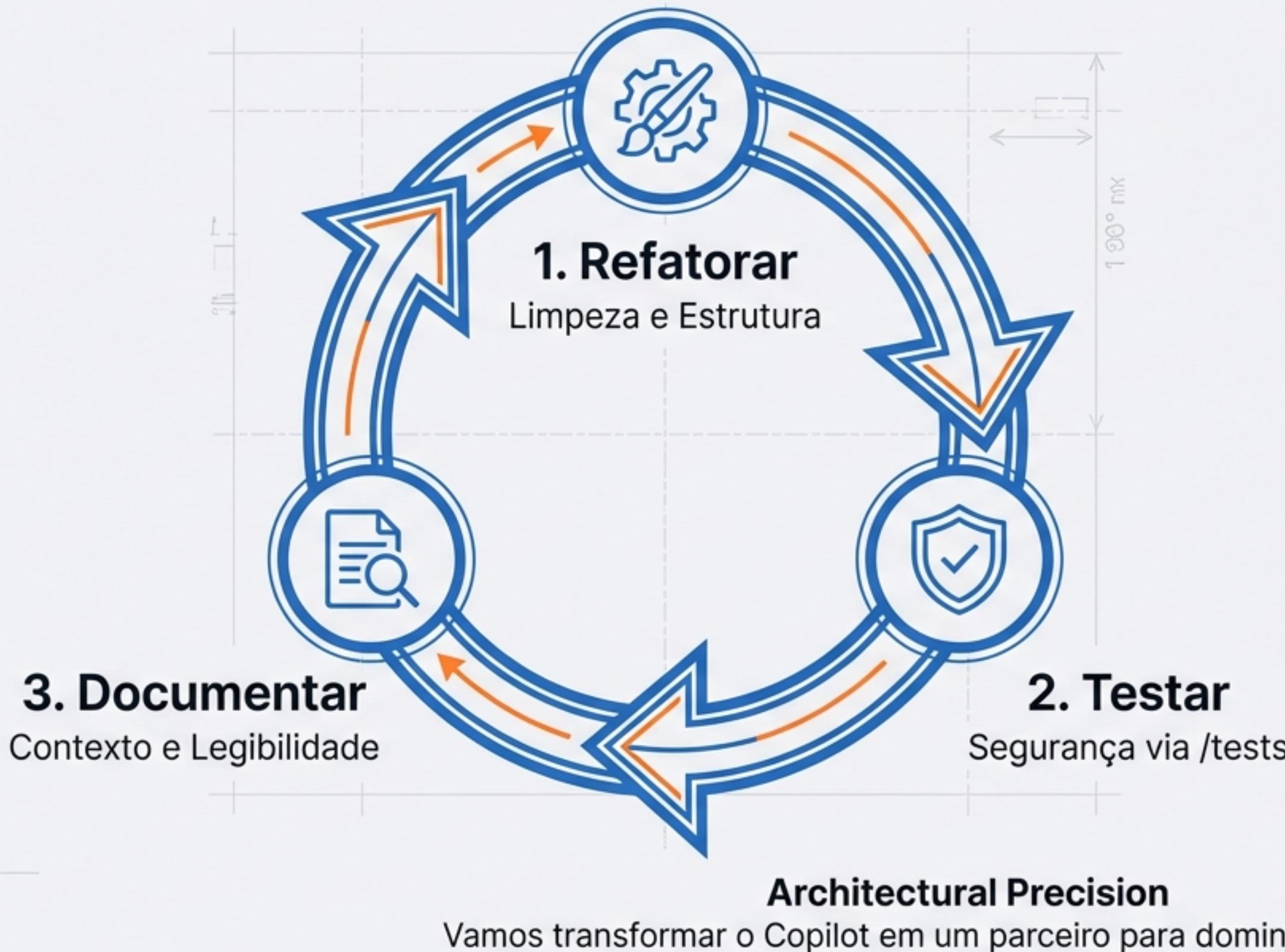
Documentar não é apenas dizer o que o código faz; é explicar o contexto e o 'porquê'.

## Técnica de Ouro:

Gere uma documentação técnica explicando os parâmetros e o que acontece se algo der errado.

*Documentação é um presente que você deixa para o seu "eu do futuro" ou para o colega que assumirá o código em seis meses.*

# O Fluxo de Trabalho Moderno: O Ciclo de Melhoria

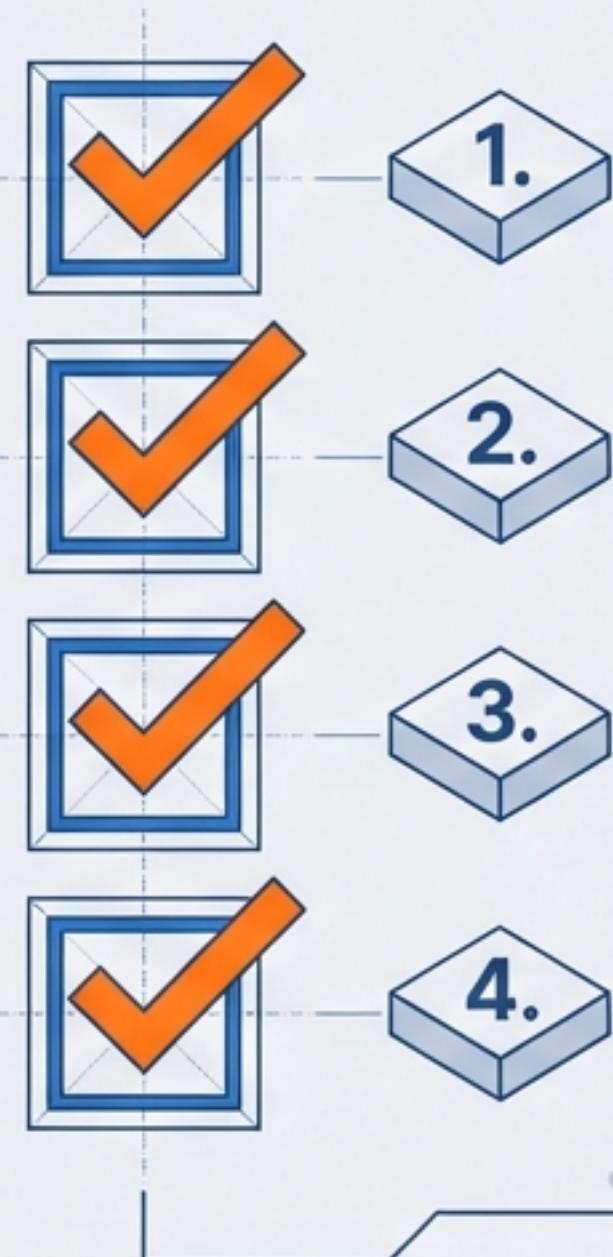


O que antes levava horas de trabalho manual, agora é um ciclo de 5 minutos.

Resultado: Limpamos a sujeira, garantimos a estabilidade e explicamos a lógica. O código sai pronto para produção.

# Laboratório Prático: O Faxineiro de Código

**Desafio:** Você recebeu um arquivo com lógica confusa e nomes de variáveis ruins. Sua missão é profissionalizá-lo.



**Explicação:** Peça ao Copilot para explicar o que o código atual faz.

**Refatoração:** Reescreva o código para torná-lo legível (Clean Code).

**Testes:** Gere pelo menos 2 testes (um de sucesso e um de erro/borda).

**Documentação:** Gere a documentação final explicando a nova estrutura.



# O Próximo Nível



- Com essas ferramentas, você deixa de ser apenas um 'codificador' para se tornar um arquiteto de soluções.
- A IA não substitui o programador, mas o programador que usa IA para refatorar, testar e documentar substitui aquele que apenas escreve linhas de código.

