



Bloco 4: Checklist de Qualidade (IA Code Review)

Elevando o código sugerido ao padrão de produção.



O Copilot escreve rápido.

→ Gera boilerplate em segundos, mas falta contexto de arquitetura. ←



Você garante a qualidade.

→ Sua responsabilidade é validar segurança, lógica e estilo. ←

1 | </>
2 | segurança
3 | </>
4 | </>

1 | </>
2 | lógica
3 | </>
4 | </>

1 | </>
2 | estilo
3 | </>
4 | </>

[O Novo Papel do Desenvolvedor: De Escritor para Revisor.
Confie, mas verifique.]

Os 3 Pilares do Code Review com IA



1. Segurança e Sanitização (O Escudo)

Foco em vulnerabilidades, injeção de código e proteção de dados sensíveis.



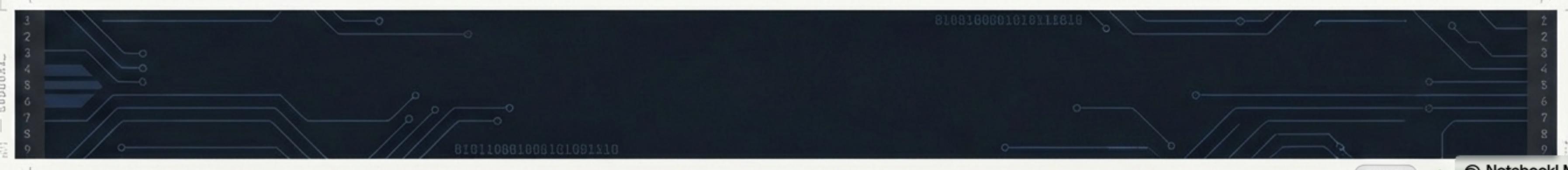
2. Lógica e Contexto (O Cérebro)

Foco em funcionalidade real, identificação de alucinações e performance.



3. Coesão e Estilo (A Forma)

Foco em legibilidade, manutenção e adequação aos padrões de projeto.



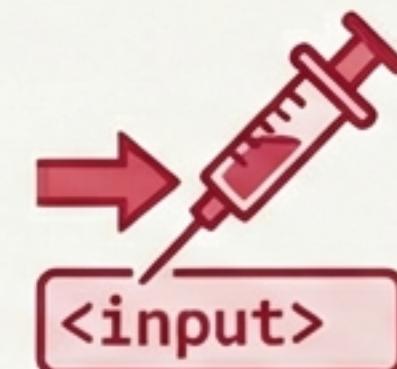
Pilar 1: Segurança e Sanitização

Identificando vulnerabilidades **invisíveis**.



Vazamento de Dados

Logs ou saídas que expõem inadvertidamente credenciais, senhas ou tokens de API.



Injeção

Falhas onde inputs de usuários são concatenados diretamente em queries SQL ou comandos de sistema sem tratamento.



Dependências

Sugestões de bibliotecas desconhecidas, desatualizadas ou inseguras.

A Lente do Auditor: Perguntas de Segurança

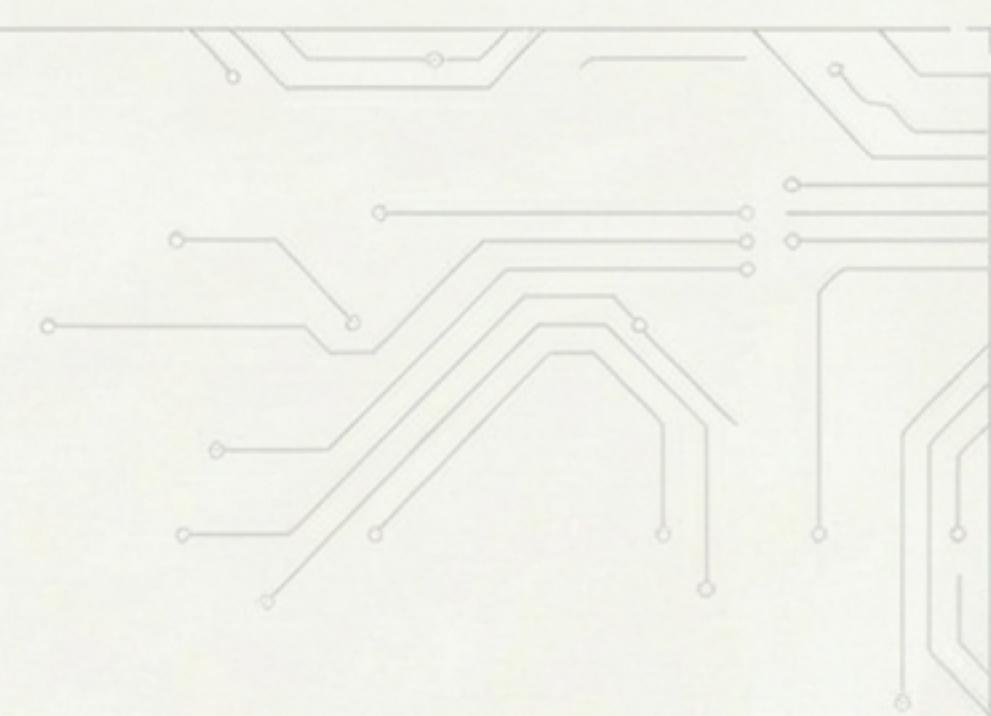
- Vazamento:** A IA sugeriu logs que expõem senhas ou tokens?
- Injeção:** Inputs de usuários estão sendo concatenados diretamente?
- Dependências:** As bibliotecas sugeridas são conhecidas e seguras?

```
console.log(userPassword)
```

ALERTA: Code generated by AI is not secure by default. Verify every input.

Pilar 2: Lógica e Contexto

O código funciona, mas **faz sentido?**



Cenários de Borda

Tratamento de situações extremas, como valores nulos, arrays vazios ou falhas de conexão.

Alucinação

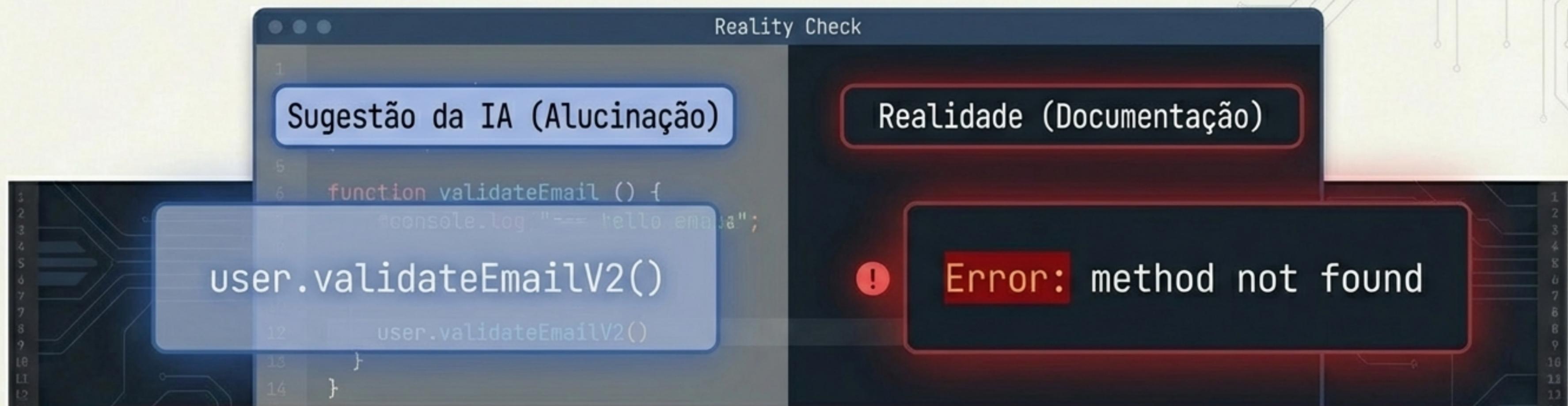
A invenção de métodos ou bibliotecas que não existem na versão específica do framework em uso.

Performance

Soluções funcionalmente corretas, mas ineficientes (ex: loops pesados em vez de estruturas otimizadas).



A Lente do Auditor: Perguntas de Lógica



- Cenários de Borda: A solução trata valores nulos e arrays vazios?
- Alucinação: O método sugerido realmente existe na minha versão do framework?
- Performance: A IA sugeriu um loop pesado que poderia ser resolvido com um Map?

Pilar 3: Coesão e Estilo

Manutenibilidade e legibilidade a longo prazo.



Nomenclatura

Evitar nomes genéricos (ex: data1, temp) e seguir o padrão do domínio.



Consistência

Respeitar a indentação, uso de chaves e estrutura do arquivo original.



Comentários

A diferença entre explicar o código e explicar o negócio.

Deep Dive: A Qualidade dos Comentários

O “Porquê” vs. O “O Quê”

O Quê (Ruim)

```
i++; // Incrementa a variável i
```

Redundante. Apenas repete o que o código já diz.

O Porquê (Bom)

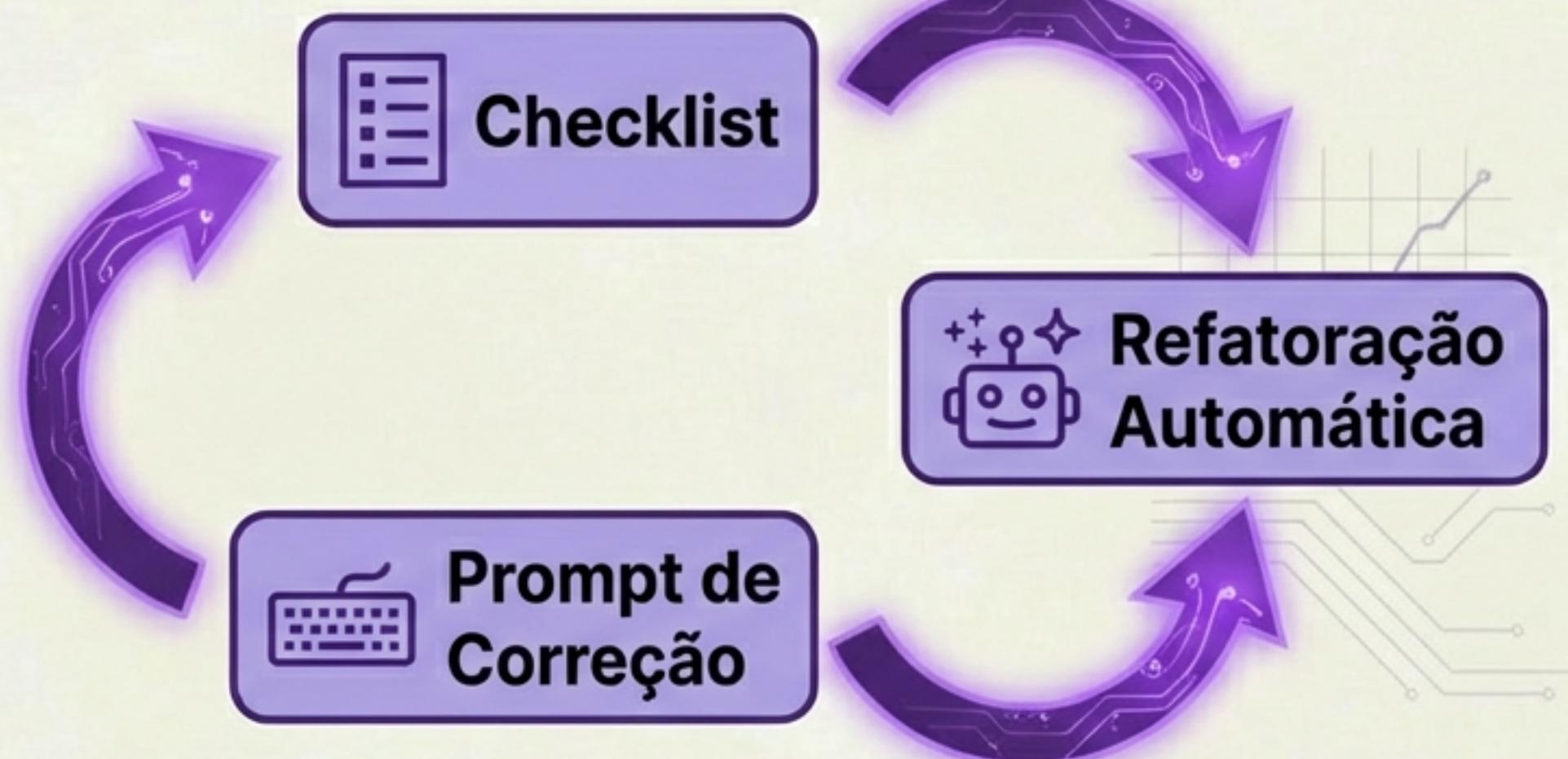
```
// Incrementa contador para garantir retry da conexão até 3 tentativas  
i++;
```

Contexto de Negócio. Explica a razão da existência do código.

Os comentários gerados explicam a regra de negócio ou apenas traduzem a sintaxe?

A IA como Solução

Usando o Copilot para corrigir o Copilot.



Uma vez identificada a falha pelo checklist, você não precisa reescrever tudo manualmente. Use a IA como ferramenta de refatoração.

“O checklist é o mapa; o prompt é o veículo.”

Atividade Prática: O Revisor Crítico

1. Abra o código:

Retome o código refatorado no Bloco 3.

2. Auditoria Manual:

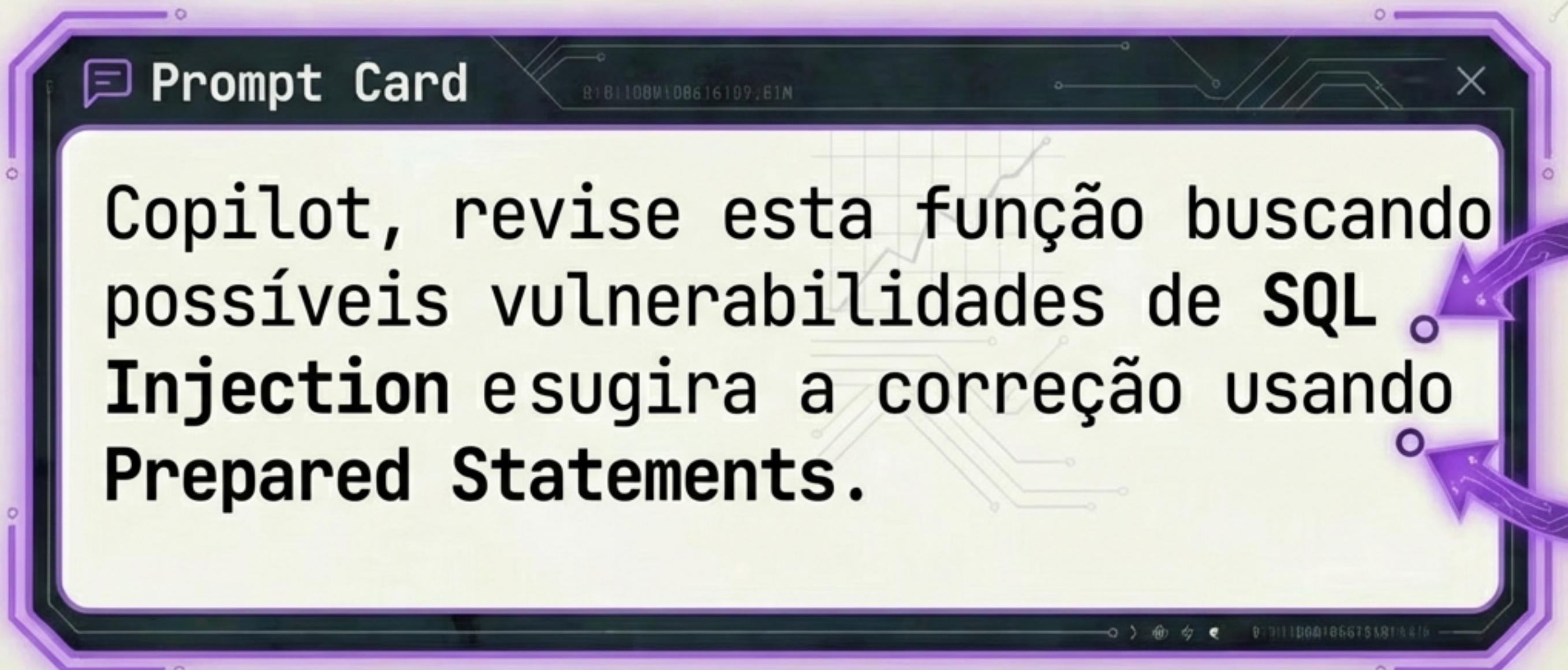
Passe o Checklist de Qualidade item por item (Segurança, Lógica, Estilo).

3. Correção Assistida: Identifique os pontos falhos e instrua a IA a corrigir.



O Prompt de Correção

Transformando o checklist em comando.



Validação Final

Comparem o código antes e depois da revisão humana guiada.

Prompt Card

Antes

- Código inseguro?
- Comentários óbvios?
- Performance duvidosa?

Depois

- Sanitização aplicada.
- Contexto de negócio claro.
- Estruturas otimizadas.

O resultado final deve ser indistinguível de um código escrito por um engenheiro sênior.

Resumo: Checklist de Qualidade

SEGURANÇA

- Sem vazamento de credenciais.
- Inputs sanitizados (sem injeção).
- Dependências seguras.

LÓGICA

- Cenários de borda tratados.
- Métodos validados (sem alucinação).
- Estruturas de dados performáticas.

ESTILO

- Nomenclatura semântica.
- Padrão visual consistente.
- Comentários de "porquê", não "o quê".

Qualidade é um Hábito



O **checklist** garante que a velocidade da IA não comprometa o rigor da engenharia.

Seu valor não está apenas em gerar código, mas em garantir que ele é seguro, eficiente e manutenível.