

O CHECKLIST DE CODE REVIEW QUE FAZ O TECH LEAD TE RESPEITAR.

22 itens em 4 blocos: antes de abrir o PR, ao revisar, ao comentar e ao aprovar. No fim, o template de descrição.

ANTES DE ABRIR O PR

- ☐ **01** Título claro: o que muda, em uma linha (ex.: "fix: evita cobrança duplicada no webhook").
- ☐ **02** Descrição com: problema, o que mudou, como testar e riscos conhecidos.
- ☐ **03** Diff pequeno: menos de ~300 linhas, ou quebrado em commits lógicos.
- ☐ **04** Sem misturar feature, bugfix e refatoração no mesmo PR.
- ☐ **05** Você já revisou o próprio diff e removeu debug, comentário morto e arquivo gerado.
- ☐ **06** Testes novos ou atualizados cobrem o bug e o caso de borda principal.

PR fácil de revisar: contexto claro, diff pequeno, teste do caso de borda e comentários que apontam risco — não ego.

AO REVISAR O DIFF

- ☐ **07** Correção: a mudança resolve o problema descrito?

- ☐ **08** Casos de borda: entrada vazia, nulo, timeout, concorrência, permissão.

- ☐ **09** Nomes: variáveis, funções e arquivos dizem o que fazem?

- ☐ **10** Complexidade: dá para entender em uma leitura? Há caminho mais simples?

- ☐ **11** Segurança: auth, validação de input, dados sensíveis, SQL/command injection.

- ☐ **12** Performance: N+1, loop desnecessário, alocação grande, cache errado.

- ☐ **13** Legibilidade: o próximo dev vai entender daqui a 6 meses?

AO COMENTAR

- ☐ **14** Comentário acionável: o que está errado e o que mudaria (ou uma pergunta).

- ☐ **15** Pergunte em vez de mandar: "O que acontece se X for nulo?" > "Muda isso."

- ☐ **16** Separe bloqueante de sugestão (prefixos: issue / nit / question / suggestion).

- ☐ **17** Critique o código, não a pessoa. Zero review de ego ("eu faria diferente").

- ☐ **18** Aponte o risco: bug, segurança, performance ou manutenção — não só estilo.

AO APROVAR

- ☐ **19** Você leu o diff de verdade. "LGTM" em 30 segundos não conta.

- ☐ **20** Pedidos bloqueantes resolvidos ou com issue aberta e acordo explícito.

- ☐ **21** Teste do bug / caso de borda está no PR ou há motivo documentado para não estar.

- ☐ **22** Descrição e commits ainda batem com o que ficou no final.

TEMPLATE DE DESCRIÇÃO DO PR

Problema

[O que estava errado ou o que faltava, em 1–2 linhas.]

O que mudou

- [Mudança 1]
- [Mudança 2]

Como testar

1. [Passo]
2. [Passo]

Resultado esperado: [...]

Riscos

- [Risco conhecido / migração / feature flag / rollback]

PARA IR ALÉM · SEGUE @LUCASADVICE

Guia de code review do Google (eng-practices, em inglês): google.github.io/eng-practices/review/

Guia para quem abre o PR (Google eng-practices, em inglês): google.github.io/eng-practices/review/developer/

Conventional Comments: prefixos para comentários acionáveis: conventionalcomments.org/

Sobre pull request reviews no GitHub (docs oficiais, em inglês): docs.github.com/en/pull-requests/collaborating-with-pull-requests/reviewing-changes-in-pull-requests/about-pull-request-reviews