

O PASSO A PASSO PARA ENTENDER UM CÓDIGO QUE VOCÊ NUNCA VIU.

7 passos, nesta ordem, para sair do “não entendo nada” para “sei onde mexer” em 1 dia. Com comandos e as perguntas certas para o time.

1 · RODE ANTES DE LER

- ☐ Suba o projeto e use como usuário: cadastre, compre, cancele, erre de propósito.
- ☐ Anote as 3 telas ou rotas mais usadas. São elas que você vai estudar primeiro.

2 · LEIA O README E AS DEPENDÊNCIAS

- ☐ README, CONTRIBUTING e a pasta docs, se existirem.
- ☐ package.json, requirements.txt, pom.xml ou go.mod: scripts (npm run dev, npm test) e bibliotecas principais.
- ☐ A árvore de pastas: tree -L 2 -I node_modules (ou ls nas pastas principais).

3 · ACHE A PORTA DE ENTRADA

- ☐ Procure onde a requisição chega: arquivo de rotas, controllers, main, index, app.
- ☐ Busque pela URL ou pelo texto da tela: `grep -rn "/checkout" src` (ou `rg "/checkout"`).
- ☐ No front, busque pelo texto do botão que você clicou.

4 · SIGA UM FLUXO INTEIRO

- ☐ Escolha uma tela e siga do clique até o banco: rota → controller → serviço → query.
- ☐ Use “Ir para definição” e “Ver referências” do editor em cada função.
- ☐ Anote o caminho em uma lista. Não tente entender as funções que ficam fora dele.

5 · USE O DEBUGGER

- ☐ Coloque um breakpoint na entrada do fluxo e execute a tela de novo.
- ☐ Avance linha a linha e veja os valores reais de cada variável.
- ☐ Sem debugger? Um log temporário em cada etapa resolve. Apague antes do PR.

6 · LEIA OS TESTES E O HISTÓRICO

- ☐ Os testes mostram o comportamento esperado: rode os do fluxo que você está estudando.
- ☐ `git log -p --follow caminho/do/arquivo` mostra cada mudança do arquivo.
- ☐ `git blame` na linha estranha leva ao PR que explica a decisão.

7 · DESENHE E CONFIRME

- ☐ Desenhe o fluxo em 5 caixas (papel, Excalidraw ou um bloco de texto).
- ☐ Mostre para alguém do time e pergunte: "Entendi que o pedido passa por A, B e C. Está certo?"
- ☐ Guarde o desenho no README ou na wiki. O próximo dev novo vai agradecer.

PERGUNTAS CERTAS PARA O TIME

"Qual é o fluxo mais importante para eu entender primeiro?"

"Tem alguma parte do código que todo mundo evita mexer? Por quê?"

"Onde ficam as regras de negócio principais?"

"Como eu rodo os testes e como sei se quebrei algo?"

"Quem é a melhor pessoa para perguntar sobre o banco e sobre o deploy?"

Rode → README e dependências → porta de entrada → um fluxo inteiro → debugger → testes e histórico → desenho confirmado.

Segue @lucasadvice