

Recuperando a suíte de testes com IA.

• \$ bundle exec rspec spec/

• diagnosticar & mapear falhas

• agrupar erros por contexto

• quebrar em tasks no ClickUp

• estabilizar pipeline · CI verde

O PANO DE FUNDO

IA acelera muito a capacidade de desenvolvimento — mas isso aumenta ainda mais a importância de ter **testes confiáveis** e um **CI** saudável.

Sem isso, a velocidade rapidamente vira dívida técnica. O objetivo nunca foi só “deixar o CI verde” — era recuperar confiança na pipeline para o time voltar a evoluir o sistema com segurança.



01

Entender o tamanho do problema.

Antes sair corrigindo os testes, rodei a suite localmente só para mapear o terreno. Nada de fix aleatório.

o que eu queria saber:

- ▶ quantidade total de falhas
- ▶ tipos de erro recorrentes
- ▶ padrões repetidos entre specs
- ▶ problemas de setup do ambiente
- ▶ testes flaky / instáveis

02

Usando IA pra organizar o caos.

- 01 pegar a saida completa dos erros
- 02 enviar pro Claude paraa análise
- 03 classificar por criticidade
- 04 agrupar por contexto / problema
- 05 gerar plano de ação incremental

agrupamentos que apareceram:

■ factories

■ autenticação

■ setup do ambiente

■ transaction / db cleaner

■ testes desatualizados

■ mocks inconsistentes

■ flaky tests

■ request specs

03

Quebrar em tarefas via MCP + ClickUp.

Em vez de encarar uma lista gigante de erros, virou backlog priorizado por contexto. Processo previsível.

backlog gerado

✓	priorizar problemas mais criticos	P0
✓	corrigir falhas de autenticação	P0
✓	estabilizar model specs	CRÍTICA
✓	revisar request specs	P1
✓	corrigir setup do banco	P1
✓	ajustar pipeline do GitHub Actions	P2



// ONDE A IA ACELERA

A IA destrava velocidade.

- ▶ identificar padrões repetidos
- ▶ localizar arquivos relacionados
- ▶ sugerir possíveis causas
- ▶ acelerar refactors
- ▶ revisar testes antigos

// ONDE O HUMANO VALIDA

Mas o que **fazia** **sentido** era decisão minha.

- ▶ reescrever testes defasados
- ▶ remover acoplamentos
- ▶ corrigir problemas estruturais
- ▶ revisar factories
- ▶ ajustar isolamento e setup

“Nem tudo era erro no teste. Muita coisa era problema de arquitetura acumulado.”



PASSO 05

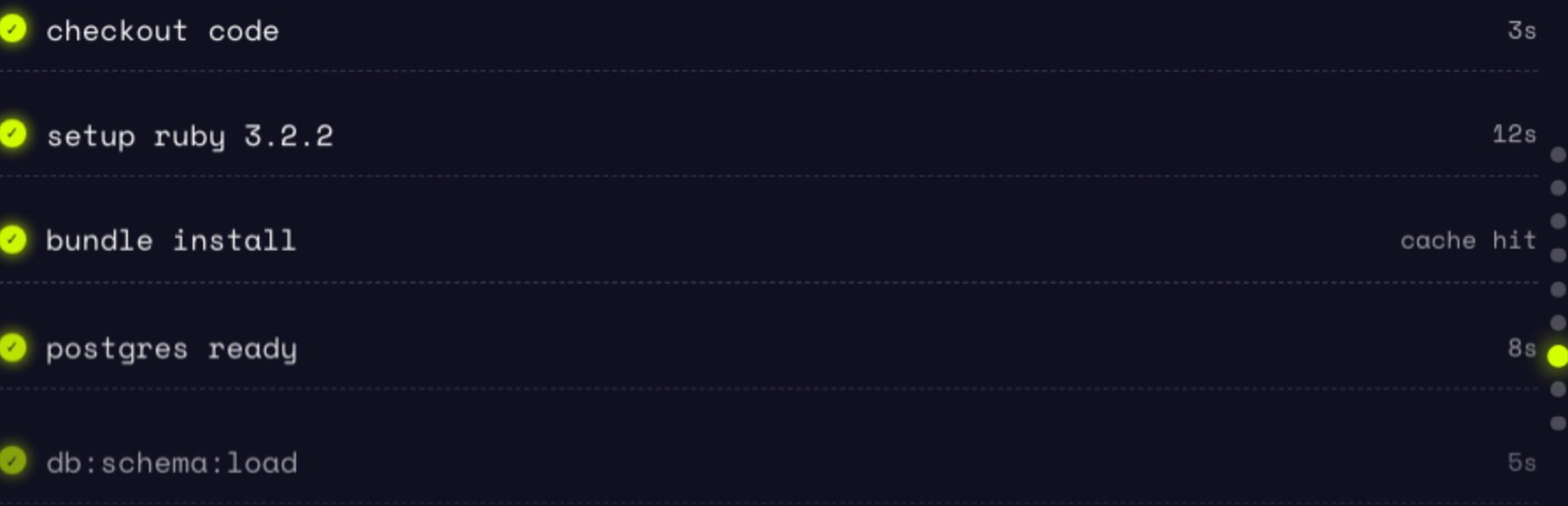
Estabilizando o GitHub Actions.

Com as specs locais saudáveis, o foco virou o pipeline. Objetivo: previsível, rápido, confiável. CI que o time pode olhar e confiar.

```
# .github/workflows/ci.yml
name: CI
on:
  push:
    branches: [ '*' ]
  pull_request:
    types: [ opened, synchronize, reopened ]

jobs:
  rspec:
    runs-on: ubuntu-latest
    services: { postgres: postgres:13 }
    steps:
      - run: bundle exec rspec spec/
```

build #1234 · main



O PRINCIPAL APRENDIZADO

Foi recuperar confiança.

Com IA acelerando geração de código e mudanças cada vez mais rápidas, testes deixam de ser apenas uma prática de qualidade.

Eles viram **infraestrutura crítica** pra permitir evolução segura do sistema.

// FIM · OBRIGADO

**Velocidade
sem
confiança vira
dívida técnica.**

Dayan Freitas · Dayan Freitas · 2026

