



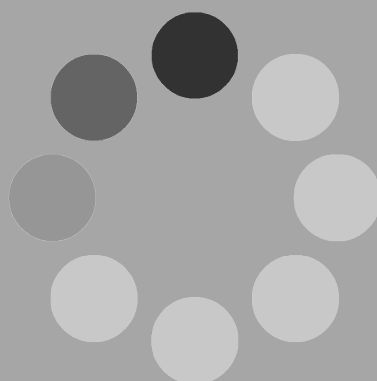
Monte um slide sobre Test-Driven Development

Monte um slide sobre Test-Driven Development

Claro! Irei montar agora mesmo!

Monte um slide sobre Test-Driven Development

Claro! Irei montar agora mesmo!



Monte um slide sobre Test-Driven Development

Claro! Irei montar agora mesmo!



erro.pptx

Claro! Irei montar agora mesmo!



erro.pptx

Ops! Ocorreu algum erro, deseja que passe as informações e você adiciona ao seu slide?

Desejo sim!

Certo, irei começar!

Dupla:
Lucas Emanuel e
Luan Bezerra



O que é TDD?

Entre as metodologias mais discutidas nos últimos anos está o **TDD (Test-Driven Development)**, uma prática que inverte a lógica tradicional: antes de escrever o código, o desenvolvedor cria os testes.

Isso garante que cada funcionalidade seja validada desde o início, aumentando a confiança no sistema e reduzindo retrabalho.

Dessa forma, o TDD é muito valorizado em ambientes ágeis e DevOps, nos quais ciclos rápidos de entrega exigem código robusto e bem testado.



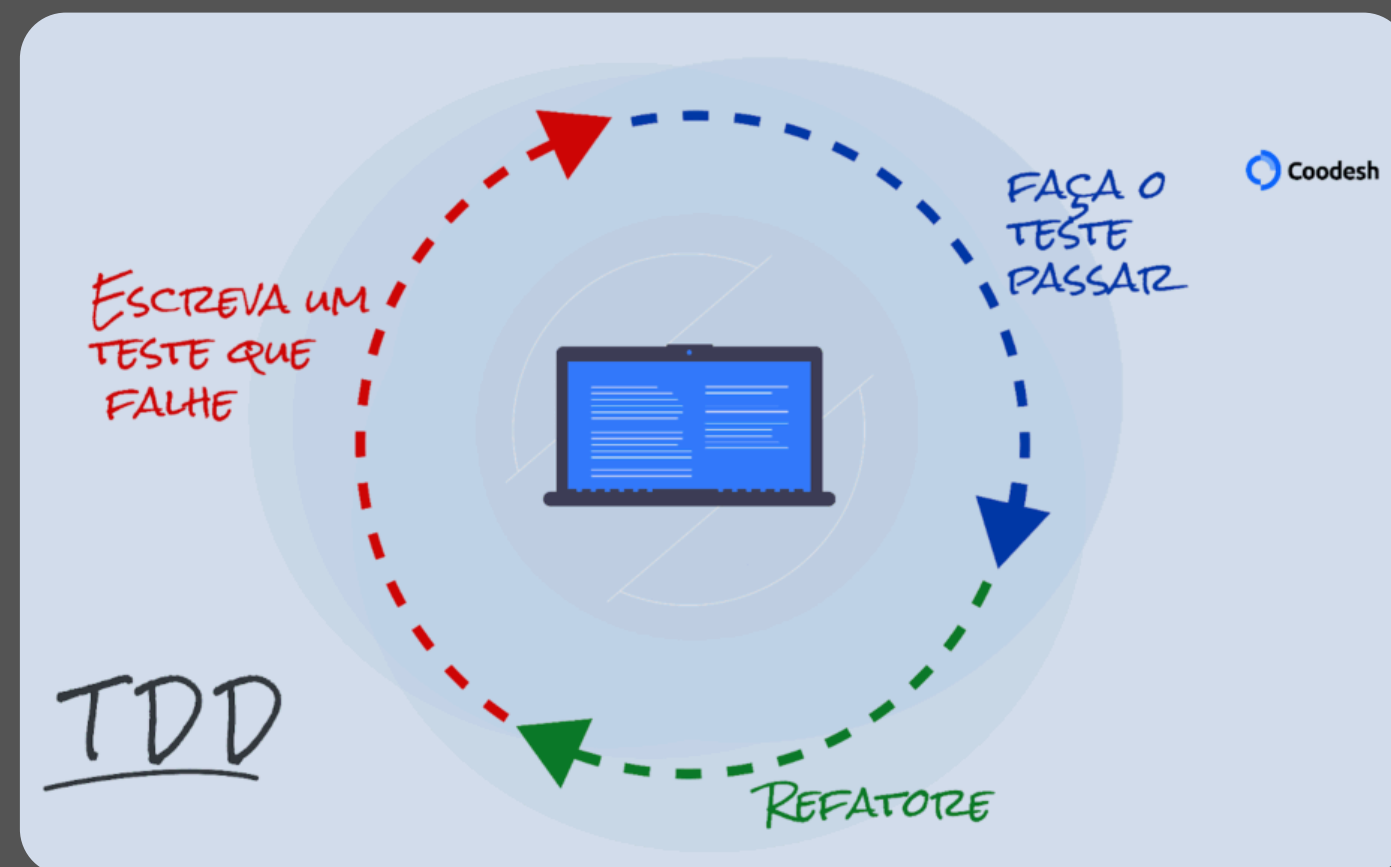
Objetivos

- **Orientar o design do software** – escrever testes antes ajuda a criar funções menores, mais modulares e coesas.
- **Fornecer feedback rápido** – qualquer alteração pode ser validada em segundos.
- **Melhorar a documentação viva** – os testes passam a servir como especificação funcional, descrevendo como o sistema deve se comportar.

Ciclo básico do TDD

O TDD é construído sobre um ciclo repetitivo e simples, mas extremamente poderoso: **Red, Green e Refactor.**

Mais do que uma técnica, trata-se de um guia que organiza o fluxo de pensamento do programador, estimulando disciplina e clareza.



Benefícios de usar TDD

- Redução de falhas;
- Maior capacidade de refatorar sem medo;
- Diminuição do gasto na construção do software;
- Design mais limpo e escalável.

Escolha a melhor resposta!

Descrição

Cada funcionalidade nasce com um teste, o que facilita detectar falhas logo no desenvolvimento.

Base de testes garante que mudanças não quebrem funcionalidades já existentes.

TDD incentiva funções pequenas, testáveis e com responsabilidades claras.

Resposta 1

Impactos

Menos bugs em produção, menor custo de correção, maior confiança da equipe e stakeholders.

Equipe pode melhorar código sem medo, aplicando boas práticas e mantendo o sistema robusto.

Design mais organizado, fácil manutenção e escalabilidade do sistema.

Resposta 2

Pronto! Essas foram todas as informações que consegui!

Deseja mais alguma coisa? 😎

- Quer que as referências dentro das normas da ABNT?
- Quer informações mais aprofundadas?
- Deseja que o slide seja em tópicos?

Faça as referências por favor!

Claro! Aqui está!

Referências

NOGUEIRA, Alexandre. TDD: o que é Test-Driven Development e como aplicar na prática. HostGator Brasil, 16 out. 2025. Disponível em: https://www.hostgator.com.br/blog/tdd/?gad_source=1&gad_campaignid=22134387764&gclid=Cj0KCQiAnJHMBhDAARIsABr7b85Ndyz25HiCdUwC4Yxle-HeYemqUo53jG8p4ssU_dIOEFULJGzMWJQaAqypEALw_wcB. Acesso em: 05 fev. 2026.



Obrigado chat e a todos que viram a essa apresentação!