

Para saber mais: Resumo do TDD

TDD (Test Driven Development ou em português Desenvolvimento guiado por testes) é uma das práticas de desenvolvimento de software sugeridas por diversas metodologias ágeis, como [XP](https://pt.wikipedia.org/wiki/Programa%C3%A7%C3%A3o_extrema) (https://pt.wikipedia.org/wiki/Programa%C3%A7%C3%A3o_extrema).

A ideia é fazer com que o desenvolvedor escreva testes automatizados de maneira constante ao longo do desenvolvimento. Mas, diferentemente do que estamos acostumados, TDD sugere que o desenvolvedor **escreva o teste antes mesmo da implementação**.

Essa simples inversão no ciclo traz diversos benefícios para o projeto. Baterias de testes tendem a ser maiores, cobrindo mais casos, e garantindo uma maior qualidade externa. A prática nos ajuda a escrever um software melhor, com mais qualidade, e um código melhor, mais fácil de ser mantido e evoluído.

Esses dois pontos são importantíssimos em qualquer software, e TDD nos ajuda a alcançá-los. Toda prática que ajuda a aumentar a qualidade do software produzido deve ser estudada.

Um ponto que é sempre levantado em qualquer discussão sobre testes manuais versus testes automatizados é **produtividade**.

O argumento mais comum é o de que agora a equipe de desenvolvimento gastará tempo escrevendo código de teste; antes ela só gastava tempo escrevendo código de produção. Portanto, essa equipe será menos produtiva.

A resposta para essa pergunta é: o que é produtividade? Se produtividade for medida através do número de linhas de código de produção escritos por dia, talvez o desenvolvedor seja sim menos produtivo.

Agora, se produtividade for a quantidade de linhas de código de produção sem defeitos escritos por dia, provavelmente o desenvolvedor será mais produtivo ao usar testes automatizados.

[..] Aniche, Maurício. **Test Driven Development**: Teste e Design no Mundo Real. [Casa do código](https://www.casadocodigo.com.br/products/livro-tdd) (<https://www.casadocodigo.com.br/products/livro-tdd>).