

at_S13_A1_SL06_backend

Roteiro de Atividade Prática

Nome: _____ Turma: _____

Título da atividade: Práticas de desenvolvimento orientado a testes (TDD) em programação *back-end*

Contexto

No desenvolvimento de software, é comum que erros apareçam em códigos complexos ou ao realizar alterações em funcionalidades já implementadas. O *Test-Driven Development* (TDD) é uma prática que ajuda a reduzir esses problemas ao priorizar a escrita de testes antes da implementação de código. O TDD segue um ciclo conhecido como *Red, Green, Refactor*, que orienta o desenvolvedor a escrever testes, implementar o código e refatorar para garantir a qualidade.

Imagine que você está desenvolvendo uma API para um sistema de gerenciamento de tarefas. Esse sistema permite criar, visualizar e excluir tarefas, mas precisa ser confiável para evitar bugs, especialmente ao manipular dados críticos dos usuários.

Objetivo

Compreender o ciclo do TDD (*Red, Green, Refactor*), aprender a escrever testes unitários para funcionalidades básicas e entender os benefícios do TDD para garantir qualidade e confiabilidade no código.

Enunciado

Seu desafio é seguir o ciclo de TDD para implementar uma função que cria novas tarefas no sistema. A função deve permitir a criação de tarefas com um

título e uma descrição, garantindo que não sejam criadas tarefas sem título. Você deve escrever testes para essa funcionalidade antes de implementar o código, seguindo o ciclo de TDD para orientar o desenvolvimento.

Roteiro de atividade

1. Ciclo de TDD: **Red, Green, Refactor**

- o Comece pelo **Red**: escreva um teste para a função de criação de tarefas, verificando se ela lança um erro ao tentar criar uma tarefa sem título.
- o Em seguida, vá para o **Green**: implemente o código mínimo necessário para passar o teste.
- o Por fim, **Refactor**: avalie o código para verificar se há melhorias possíveis sem alterar a funcionalidade. Isso pode incluir simplificação de lógica ou melhorias de legibilidade.

2. Escrevendo testes unitários

- o Crie testes adicionais para a função de criação de tarefas. Por exemplo:
 - Verifique se uma tarefa com título e descrição é criada corretamente.
 - Confirme que uma tarefa não é criada se o título estiver ausente.
- o Execute os testes e analise os resultados para garantir que o código atende a todos os requisitos.

3. Benefícios do TDD

- o Reflita sobre o processo e descreva os benefícios do TDD que você observou ao seguir o ciclo. Considere aspectos como a detecção precoce de erros e a confiança no código implementado.

Perguntas para conclusão da atividade:

1. Qual é a importância de seguir o ciclo *Red, Green, Refactor* no TDD?
2. Quais são os principais aspectos a considerar ao escrever testes unitários?
3. Quais benefícios você observou ao usar TDD na implementação da função de criação de tarefas?
4. Como o TDD pode contribuir para a manutenção e evolução do sistema no longo prazo?