**06 - Faça como eu fiz: Test-Driven Development**

Chegou a hora de você executar o que foi visto na aula!

Nessa aula nós aprendemos o que é ***Test-Driven Development*** ou **Desenvolvimento Guiado por Testes**, vimos o que é cada etapa do ciclo do TDD (**Construção de Testes**, **Implementação de Código** e **Refatoração**) e ajudamos a Dominique a criar uma nova funcionalidade do zero no projeto dela através da utilização dessa metodologia.

E aí? Já colocou a mão na massa?

Chegou a hora de você desenvolver usando o método do TDD!

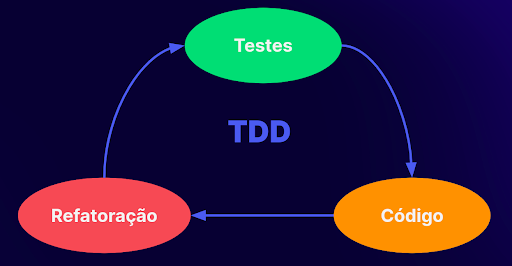
Caso tenha dúvidas, confira o andamento do seu projeto clicando na **Opinião do Instrutor**.

VER OPINIÃO DO INSTRUTOR

**Opinião do instrutor**

Confira o andamento do seu projeto de acordo com o que foi instruído:

1) Primeiramente, aprendemos um pouco sobre o ciclo do TDD, sendo ele: Construção de Testes, Implementação de Código e Refatoração;



2) O chefe da Dominique entrou em contato com ela e pediu que ela desenvolvesse uma nova funcionalidade para a classe Funcionario. É necessário que seja implementado um método que identifique quem são os diretores da empresa pelo sobrenome. Além disso, ela também deve identificar quem recebe salário igual ou superior a R$100000,00 e fazerum decréscimo equivalente a 10% deste valor;

3) Começamos com a primeira etapa do ciclo do TDD, a **construção de testes**. Criamos um teste no arquivo test\_bytebank.py para cumprir essas regras de negócio;

def test\_quando\_decrescimo\_salario\_recebe\_100000\_deve\_retornar\_90000(self):

entrada\_salario = 100000 #given

entrada\_nome = 'Paulo Bragança'

esperado = 90000

funconario\_teste = Funcionario(entrada\_nome, '11/11/2000', entrada\_salario)

funconario\_teste.decrescimo\_salario() # when

resultado = funconario\_teste.salario

assert resultado == esperado # then

4) Em seguida, vamos para a etapa de **implementação de código** criar o método decrescimo\_salario() no arquivo bytebank.py;

def decrescimo\_salario(self):

sobrenomes = ['Bragança', 'Windsor', 'Bourbon', 'Yamato', 'Al Saud', 'Khan', 'Tudor', 'Ptolomeu']

if self.\_salario >= 100000 and (self.sobrenome() in sobrenomes):

decrescimo = self.\_salario \* 0.1

self.\_salario = self.\_salario - decrescimo

5) Notamos que, apesar do teste passar e o método funcionar como deveria, decrescimo\_salario() não apresenta as melhores práticas de código em Python;

O método está desempenhando duas funções: identificar se o funcionário é um sócio, ou seja, se ele possui o sobrenome entre entre a lista de sobrenomes de sócios (Bragança, Windsor, Bourbon, Yamato, Al Saud, Khan, Tudor, Ptolomeu) e possui o seu salário igual ou acima de R$100000,00; e fazer um decréscimo no salário equivalente a 10% do salário do funcionário.

6) Começamos a terceira etapa do ciclo do TDD, a **refatoração**. Criamos um novo método no arquivo bytebank.py chamado \_eh\_socio() que verifica se o funcionário é, de fato, um sócio;

def \_eh\_socio(self):

sobrenomes = ['Bragança', 'Windsor', 'Bourbon', 'Yamato', 'Al Saud', 'Khan', 'Tudor', 'Ptolomeu']

return (self.\_salario >= 100000) and (self.sobrenome() in sobrenomes)

7) Em seguida, modificamos o método decrescimo\_salario() para utilizar o método \_eh\_socio();

def decrescimo\_salario(self):

if self.\_eh\_socio():

decrescimo = self.\_salario \* 0.1

self.\_salario = self.\_salario - decrescimo