06

Do as I did: Test-Driven Development

It's time for you to implement what you saw in class!

In this class we learned what ***Test-Driven Development*** **is** , we saw what each stage of the TDD cycle is (**Test Construction** , **Code Implementation** and **Refactoring**) and we helped Dominique create a new feature from scratch in her project through the use of this methodology.

So, have you gotten your hands dirty yet?

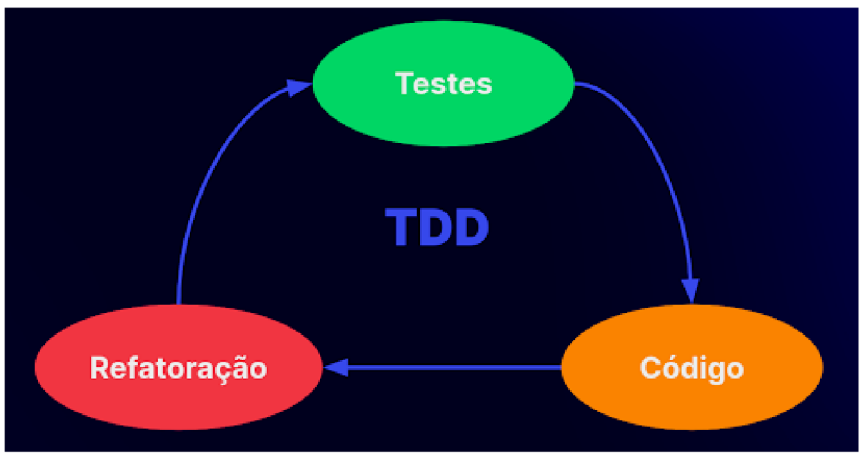
It's time for you to develop using the TDD method!

If you have any questions, check the progress of your project by clicking on **Instructor's Opinion** .

Instructor's opinion

Check the progress of your project according to the instructions:

1) First, we learn a little about the TDD cycle, which is: Test Construction, Code Implementation and Refactoring;



2) Dominique's boss contacted her and asked her to develop a new feature for the class `Funcionario` . It is necessary to implement a method that identifies who the company's directors are by their last name. In addition, she must also identify who receives a salary equal to or greater than R$100,000.00 and make a deduction equivalent to 10% of this amount;

3) We start with the first step of the TDD cycle, **building tests** . We create a test in the file `test_bytebank.py` to fulfill these business rules;

```
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', entrac
        funconario_teste.decrescimo_salario() # when
        resultado = funconario_teste.salario

        assert resultado == esperado  # then
```

COPY CODE


```
def decrescimo_salario(self):
        sobrenomes = ['Bragança', 'Windsor', 'Bourbon', 'Yamato', 'Al Sau
        if self._salario >= 100000 and (self.sobrenome() in sobrenomes):
            decrescimo = self._salario * 0.1
            self._salario = self._salario - decrescimo
```

COPY CODE

5) We noticed that, although the test passes and the method works as it should, `decrescimo_salario()` it does not present the best coding practices in Python;

The method is performing two functions: identifying whether the employee is a partner, that is, if he has a surname among the list of partner surnames (Bragança, Windsor, Bourbon, Yamato, Al Saud, Khan, Tudor, Ptolemy) and has a salary equal to or above R$100,000.00; and making a reduction in the salary equivalent to 10% of the employee's salary.

6) We begin the third stage of the TDD cycle, **refactoring** . We create a new method in the file `bytebank.py` called `_eh_socio()` that checks if the employee is, in fact, a partner;

```
def _eh_socio(self):
        sobrenomes = ['Bragança', 'Windsor', 'Bourbon', 'Yamato', 'Al Sau
        return (self._salario >= 100000) and (self.sobrenome() in sobrenc
```

COPY CODE

7) Next, we modify the method `decrescimo_salario()` to use the method `_eh_socio()` ;

```
def decrescimo_salario(self):
        if self._eh_socio():
            decrescimo = self._salario * 0.1
            self._salario = self._salario - decrescimo
```

COPY CODE