

Pytest fixtures e classes de testes

No vídeo, vimos um pouco sobre as fixtures do `pytest`. Existem outras fixtures que podemos utilizar além daquelas. Neste post <https://blog.alura.com.br/montando-cenarios-de-testes-com-o-pytest> (<https://blog.alura.com.br/montando-cenarios-de-testes-com-o-pytest>) são mostrados algumas outras fixtures.

É interessante notar que utilizamos funções para escrever os testes. Porém, se quiséssemos, poderíamos utilizar classes de testes também. A implementação do mesmo módulo de testes como uma classe, ficaria parecida com essa:

```
from src.leilao.dominio import Usuario, Leilao

import pytest

from src.leilao.excecoes import LanceInvalido

class TestUsuario:

    @pytest.fixture
    def vini(self):
        return Usuario('Vini', 100.0)

    @pytest.fixture
    def leilao(self):
        return Leilao('Celular')

    def test_deve_subtrair_valor_da_carteira_do_usuario_quando_este_propor_um_lance(self, vini, leilao):
        vini.propoe_lance(leilao, 50.0)

        assert vini.carteira == 50.0

    def test_deve_permitir_propor_lance_quando_o_valor_eh_menor_que_o_valor_da_carteira(self, vini, leilao):
        vini.propoe_lance(leilao, 1.0)

        assert vini.carteira == 99.0

    def test_deve_permitir_propor_lance_quando_o_valor_eh_igual_ao_valor_da_carteira(self, vini, leilao):
        vini.propoe_lance(leilao, 100.0)

        assert vini.carteira == 0.0

    def test_nao_deve_permitir_propor_lance_com_valor_maior_que_o_da_carteira(self, vini, leilao):
        with pytest.raises(LanceInvalido):

            vini.propoe_lance(leilao, 200.0)
```

