

Implemente um Test Data Builder para a classe Leilao, com as mesmas funcionalidades apresentadas na explicação desta seção.

Modifique sua bateria de testes para fazer uso deste Builder.

Opinião do instrutor

O Builder ficou parecido com isso:

```
public class CriadorDeLeilao {

    private Leilao leilao;

    public CriadorDeLeilao() { }

    public CriadorDeLeilao para(String descricao) {
        this.leilao = new Leilao(descricao);
        return this;
    }

    public CriadorDeLeilao lance(Usuario usuario, double valor) {
        leilao.propoe(new Lance(usuario, valor));
        return this;
    }

    public Leilao constroi() {
        return leilao;
    }
}
```

E nossas classes de teste:

```
public class LeilaoTest {

    @Test
    public void deveReceberUmLance() {
        Leilao leilao = new CriadorDeLeilao().para("Macbook Pro 15").constroi();
        assertEquals(0, leilao.getLances().size());

        leilao.propoe(new Lance(new Usuario("Steve Jobs"), 2000));

        assertEquals(1, leilao.getLances().size());
        assertEquals(2000.0, leilao.getLances().get(0).getValor(), 0.00001);
    }

    @Test
    public void deveReceberVariosLances() {
        Leilao leilao = new CriadorDeLeilao()
            .para("Macbook Pro 15")
            .lance(new Usuario("Steve Jobs"), 2000)
            .lance(new Usuario("Steve Wozniak"), 3000)
            .constroi();

        assertEquals(2, leilao.getLances().size());
        assertEquals(2000.0, leilao.getLances().get(0).getValor(), 0.00001);
        assertEquals(3000.0, leilao.getLances().get(1).getValor(), 0.00001);
    }

    @Test
    public void naoDeveAceitarDoisLancesSeguidosDoMesmoUsuario() {
        Usuario steveJobs = new Usuario("Steve Jobs");
        Leilao leilao = new CriadorDeLeilao()
            .para("Macbook Pro 15")
            .lance(steveJobs, 2000.0)
            .lance(steveJobs, 3000.0)
            .constroi();

        assertEquals(1, leilao.getLances().size());
        assertEquals(2000.0, leilao.getLances().get(0).getValor(), 0.00001);
    }

    @Test
    public void naoDeveAceitarMaisDoQue5LancesDeUmMesmoUsuario() {
        Usuario steveJobs = new Usuario("Steve Jobs");
        Usuario billGates = new Usuario("Bill Gates");

        Leilao leilao = new CriadorDeLeilao().para("Macbook Pro 15")
            .lance(steveJobs, 2000)
            .lance(billGates, 3000)
            .lance(steveJobs, 4000)
            .lance(billGates, 5000)
            .lance(steveJobs, 6000)
            .lance(billGates, 7000)
            .lance(steveJobs, 8000)
            .lance(billGates, 9000)
            .lance(steveJobs, 10000)
            .lance(billGates, 11000)
            .lance(steveJobs, 12000)
            .constroi();

        assertEquals(10, leilao.getLances().size());
        int ultimo = leilao.getLances().size()-1;
        assertEquals(11000.0, leilao.getLances().get(ultimo).getValor(), 0.00001);
    }
}

public class AvaliadorTest {

    private Avaliador leiloeiro;
    private Usuario maria;
    private Usuario jose;
    private Usuario joao;

    @Before
    public void setUp() {
        this.leiloeiro = new Avaliador();
        this.joao = new Usuario("João");
        this.jose = new Usuario("José");
        this.maria = new Usuario("Maria");
    }

    @Test
    public void deveEntenderLancesEmOrdemCrescente() {

        Leilao leilao = new CriadorDeLeilao()
            .para("Playstation 3 Novo")
            .lance(joao, 250)
            .lance(jose, 300)
            .lance(maria, 400)
            .constroi();

        leiloeiro.avalialance(leilao);

        assertEquals(400.0, leiloeiro.getMaiorLance(), 0.00001);
        assertEquals(250.0, leiloeiro.getMenorLance(), 0.00001);
    }

    @Test
    public void deveEntenderLeilaoComApenasUmLance() {
        Leilao leilao = new CriadorDeLeilao()
            .para("Playstation 3 Novo")
            .lance(joao, 1000)
            .constroi();

        leiloeiro.avalialance(leilao);

        assertEquals(1000.0, leiloeiro.getMaiorLance(), 0.00001);
        assertEquals(1000.0, leiloeiro.getMenorLance(), 0.00001);
    }

    @Test
    public void deveEncontrarOsTresMajoresLances() {
        Leilao leilao = new CriadorDeLeilao()
            .para("Playstation 3 Novo")
            .lance(joao, 100)
            .lance(maria, 200)
            .lance(joao, 300)
            .lance(maria, 400)
            .constroi();

        leiloeiro.avalialance(leilao);

        List<Lance> maiores = leiloeiro.getTresMajores();
        assertEquals(3, maiores.size());
        assertEquals(400.0, maiores.get(0).getValor(), 0.00001);
        assertEquals(300.0, maiores.get(1).getValor(), 0.00001);
        assertEquals(200.0, maiores.get(2).getValor(), 0.00001);
    }
}
```