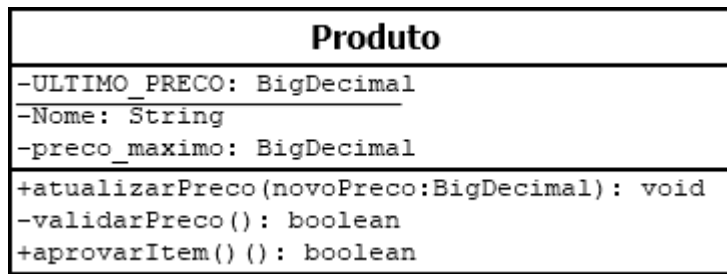
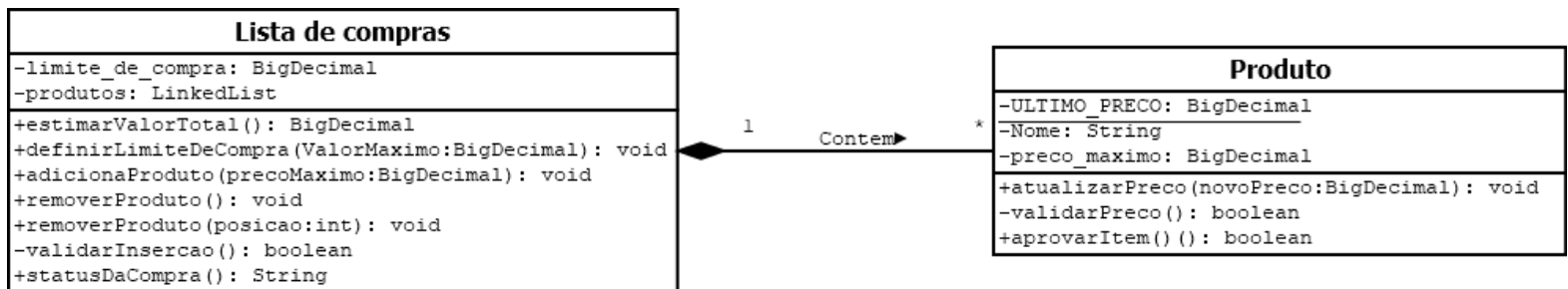


1) UML Produto



2) UML Lista de Compra -> Produto



3) Método adicionaProduto

```
public void adicionaProduto(String nome, BigDecimal precoMaximo){
    if(validaInsercao()){
        Produto tmp = new Produto(nome, precoMaximo);
        if(tmp.aprovarItem()){
            itens.add(new Produto(nome, precoMaximo));
        }
    }
}
```

4) Teste da classe Produto na proxima pagina.

```
public class ProdutoTest {
    Produto meuProdutoTeste;

    @BeforeEach
    public void setup() {
        meuProdutoTeste = new Produto();
    }

    @Test
    public void deveIniciarComoDesconhecido() {
        assertEquals("Desconhecido", meuProdutoTeste.getNome());
    }

    @Test
    public void deveIniciarComValorZero(){
        BigDecimal valorEsperado = new BigDecimal(0.0);
        BigDecimal valorRecebido = meuProdutoTeste.getPreco();
        assertEquals(valorEsperado, valorRecebido);
    }

    @Test
    public void deveIniciarComValorDeterminado(){
        Produto produtoTeste = new Produto("Teste", new BigDecimal(5.5));
        BigDecimal valorEsperado = new BigDecimal(5.5);
        BigDecimal valorRecebido = produtoTeste.getPreco();
        assertEquals(valorEsperado, valorRecebido);
    }

    @Test
    public void deveIniciarComNomeDeterminado(){
        Produto produtoTeste = new Produto("Teste", new BigDecimal(0.0));
        assertEquals("Teste", produtoTeste.getNome());
    }

    @Test
    public void deveAtualizarPrecoMaximo(){
        meuProdutoTeste.atualizaPrecoMaximo(new BigDecimal(1.99));
        BigDecimal valorEsperado = new BigDecimal(1.99);
        BigDecimal valorRecebido = meuProdutoTeste.getPreco();
        assertEquals(valorEsperado, valorRecebido);
    }

    @Test
    public void deveManterNegarProdutoCasoMaiorQueMaximo(){
        /*O produto caro serve para subir o valor do ultimo preco válido do produto */
        Produto caro = new Produto("Caro", new BigDecimal(7.99));
        Produto barato = new Produto("Barato", new BigDecimal(5.00));
        /*Espera falso pois o valor maximo que queremos pagar é 5.00, porem está em 7.99 */
        assertFalse(barato.aprovarItem());
    }
}
```