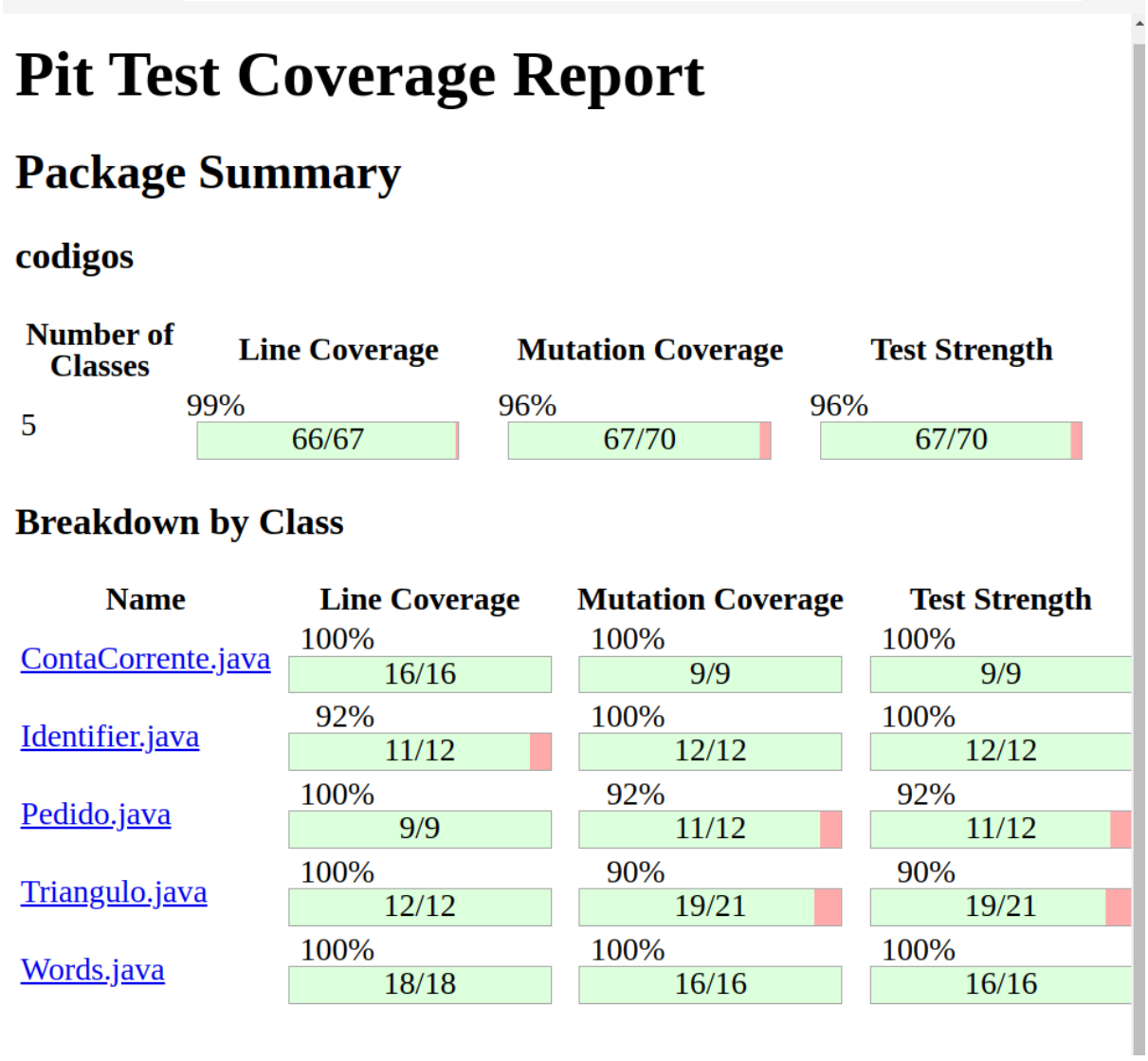
ENG II - Testes Mutantes

Resultados obtidos nos testes mutantes com os testes criados na atividade anterior



Pit Test Coverage Report

Package Summary

codigos

Number of Classes	Line Coverage	Mutation Coverage	Test Strength
5	99% 66/67	96% 67/70	96% 67/70

Breakdown by Class

Name	Line Coverage	Mutation Coverage	Test Strength
ContaCorrente.java	100% 16/16	100% 9/9	100% 9/9
Identifier.java	92% 11/12	100% 12/12	100% 12/12
Pedido.java	100% 9/9	92% 11/12	92% 11/12
Triangulo.java	100% 12/12	90% 19/21	90% 19/21
Words.java	100% 18/18	100% 16/16	100% 16/16

Testes criados para cobrir partes de código não verificadas pelos testes mutantes

Identifer.java

```
@Test
   public void identvalidar(){
       Identifier identifier = new Identifier();
       Assert.assertNotNull( "A instância de Identifier não
deveria ser nula.", identifier);
   }
```

O erro estava somente na linha de “criação da classe”, para verificar isso era necessário criar uma instância e verificar se é Nula.

Pedido.java

```
@Test
   public void testTaxa5PeloValorCompraMaiorIgual200(){
       // valorCompra >= 200, < 400, cliente nao "ouro", "prata",
nao primeiraCompra
       float taxa = pedido.calculaTaxaDesconto(false,"",200);
       assertEquals(5,taxa,0.001);
   }
```

Antes não havia um teste para verificar na fronteira da compra com valor = 200. Foi adicionado um teste para verificar essa condição.

Triangulo.java

```
@Test
   public void testNaoFormaIgual() throws LadoInvalidoException {
       String resultado = Triangulo.classificaTriangulo(3, 2, 1);
       assertEquals("NAO FORMA TRIANGULO", resultado);
   }

   @Test
   public void testNaoFormaIgual2() throws LadoInvalidoException
{
       String resultado = Triangulo.classificaTriangulo(2, 3, 1);
```

```
        assertEquals("NAO FORMA TRIANGULO", resultado);
    }
@Test
    public void testNegativoZero() throws LadoInvalidoException{
        thrown.expect(LadoInvalidoException.class);
        thrown.expectMessage("lado invalido");
        Triangulo.classificaTriangulo(-1, -1, 0);
    }
```

O problema se deu na verificação do tamanho dos lados do triângulo e se é possível formá-lo, foi adicionado casos de fronteira (quando a soma dos dois for igual ao 3 lado) e também caso negativo.

Resultado final após alterações

Pit Test Coverage Report

Package Summary

codigos

Number of Classes	Line Coverage	Mutation Coverage	Test Strength
5	100% 67/67	100% 70/70	100% 70/70

Breakdown by Class

Name	Line Coverage	Mutation Coverage	Test Strength
ContaCorrente.java	100% 16/16	100% 9/9	100% 9/9
Identifier.java	100% 12/12	100% 12/12	100% 12/12
Pedido.java	100% 9/9	100% 12/12	100% 12/12
Triangulo.java	100% 12/12	100% 21/21	100% 21/21
Words.java	100% 18/18	100% 16/16	100% 16/16