EFICIENCIA DE TESTES:

EVOSUITE VS MODELOS I.A.

A91672 LUIS FERREIRA
A93258 BERNARDO LIMA
A94523 ALEXANDRA SANTOS

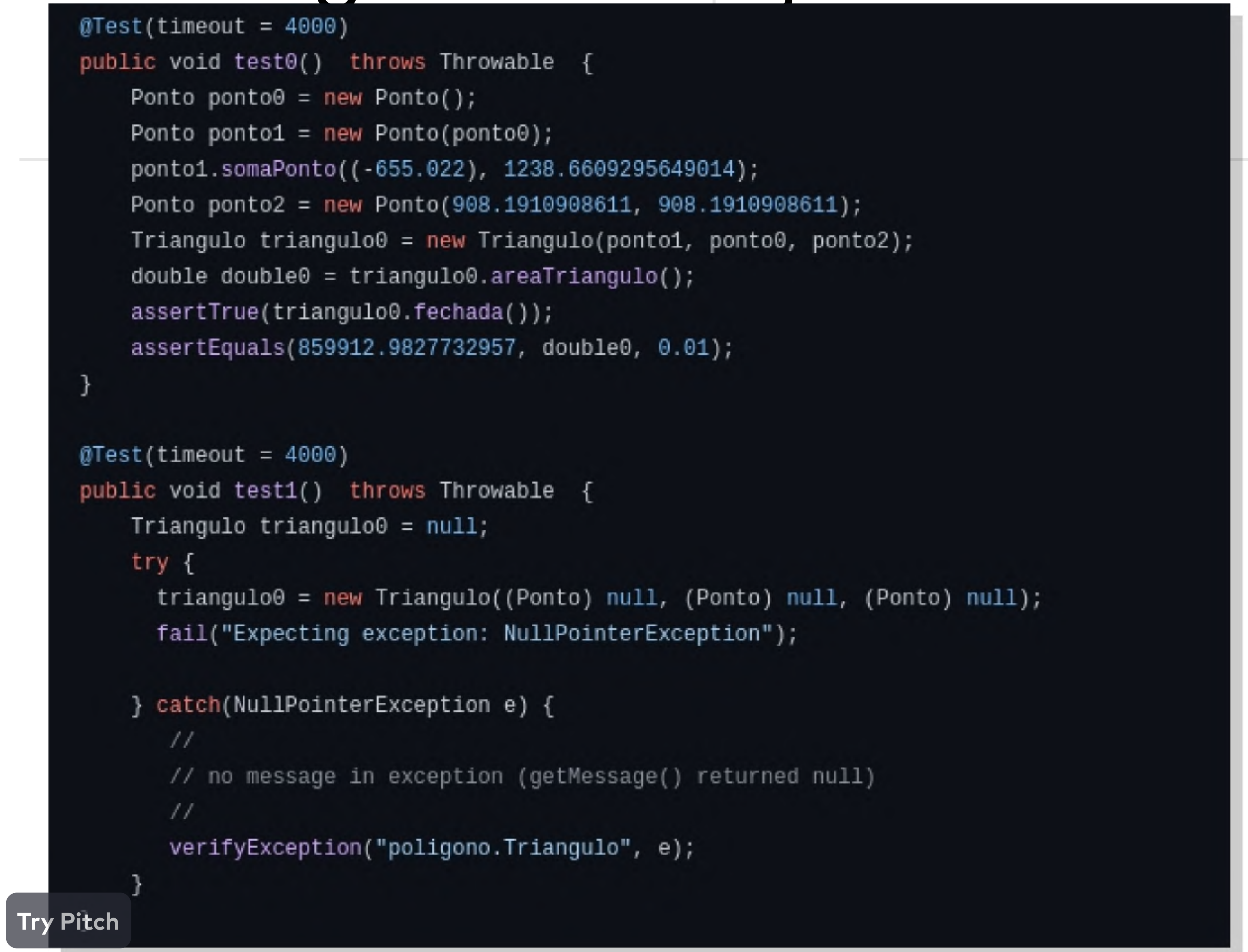
INTRODUCAO

Os testes de software desempenham um papel crucial no desenvolvimento, assegurando qualidade e segurança, auxiliando a identificação de erros à redução de custos e tempo de correção.

Para esse efeito, usamos JaCoCo e Pitest para avaliar a qualidade de testes gerados automaticamente pelo Maven Evosuite e pela *IA*, nomeadamente o *GPT* e o *Gemini*.

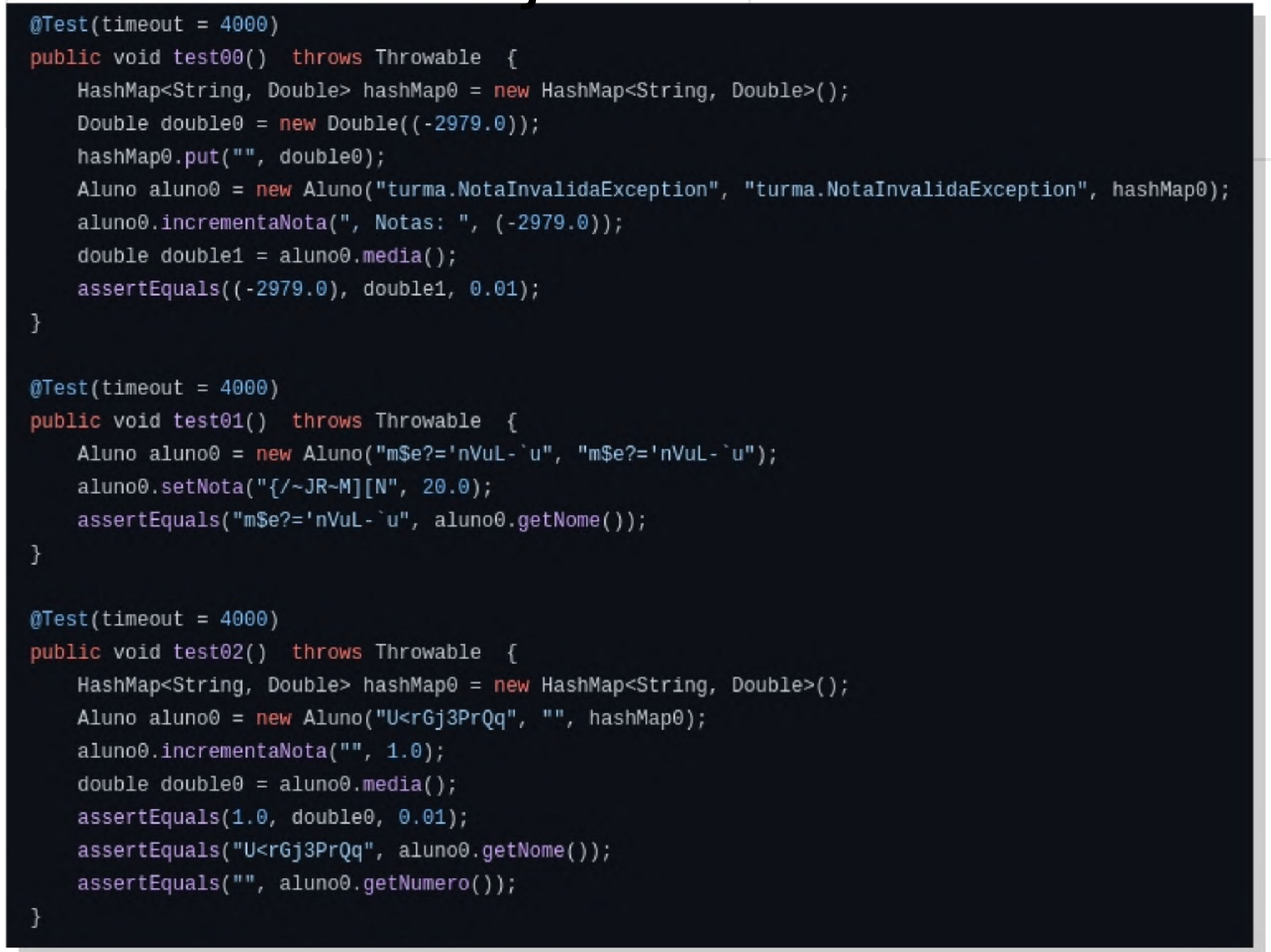
EXEMPLOS DOS TESTES GERADOS: EVOSUITE

Projeto: Poligono Triangulo_ESTest.java

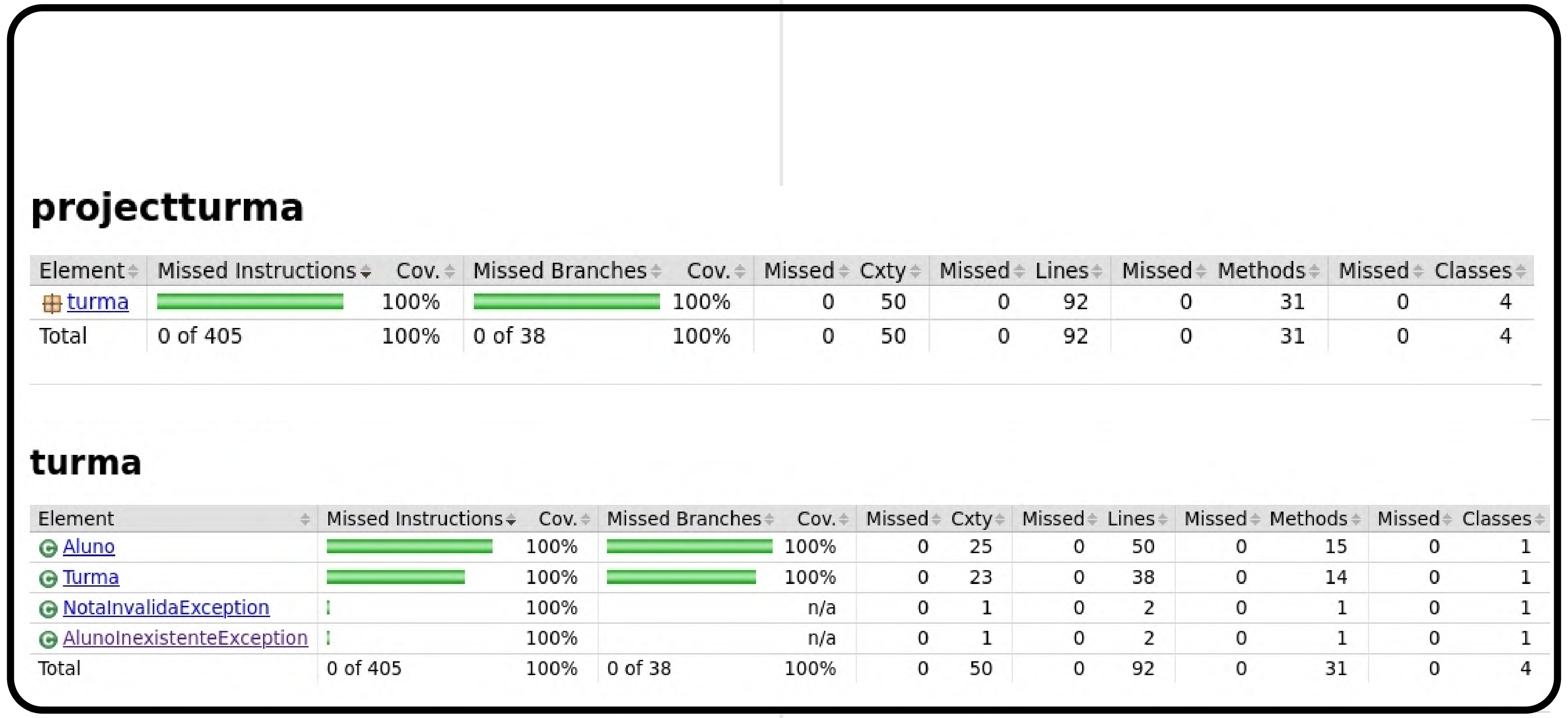


```java
@Test(timeout = 4000)
public void test0()  throws Throwable  {
    Ponto ponto0 = new Ponto();
    Ponto ponto1 = new Ponto(ponto0);
    ponto1.somaPonto((-655.022), 1238.6609295649014);
    Ponto ponto2 = new Ponto(908.1910908611, 908.1910908611);
    Triangulo triangulo0 = new Triangulo(ponto1, ponto0, ponto2);
    double double0 = triangulo0.areaTriangulo();
    assertTrue(triangulo0.fechada());
    assertEquals(859912.9827732957, double0, 0.01);
}

@Test(timeout = 4000)
public void test1()  throws Throwable  {
    Triangulo triangulo0 = null;
    try {
      triangulo0 = new Triangulo((Ponto) null, (Ponto) null, (Ponto) null);
      fail("Expecting exception: NullPointerException");

    } catch(NullPointerException e) {
       //
       // no message in exception (getMessage() returned null)
       //
       verifyException("poligono.Triangulo", e);
    }
```

Projeto: Turma Aluno_ESTest.java



```java
@Test(timeout = 4000)
public void test00()  throws Throwable  {
    HashMap<String, Double> hashMap0 = new HashMap<String, Double>();
    Double double0 = new Double((-2979.0));
    hashMap0.put("", double0);
    Aluno aluno0 = new Aluno("turma.NotaInvalidaException", "turma.NotaInvalidaException", hashMap0);
    aluno0.incrementaNota(", Notas: ", (-2979.0));
    double double1 = aluno0.media();
    assertEquals((-2979.0), double1, 0.01);
}

@Test(timeout = 4000)
public void test01()  throws Throwable  {
    Aluno aluno0 = new Aluno("m$e?='nVuL-`u", "m$e?='nVuL-`u");
    aluno0.setNota("{/~JR~M][N", 20.0);
    assertEquals("m$e?='nVuL-`u", aluno0.getNome());
}

@Test(timeout = 4000)
public void test02()  throws Throwable  {
    HashMap<String, Double> hashMap0 = new HashMap<String, Double>();
    Aluno aluno0 = new Aluno("U<rGj3PrQq", "", hashMap0);
    aluno0.incrementaNota("", 1.0);
    double double0 = aluno0.media();
    assertEquals(1.0, double0, 0.01);
    assertEquals("U<rGj3PrQq", aluno0.getNome());
    assertEquals("", aluno0.getNumero());
}
```

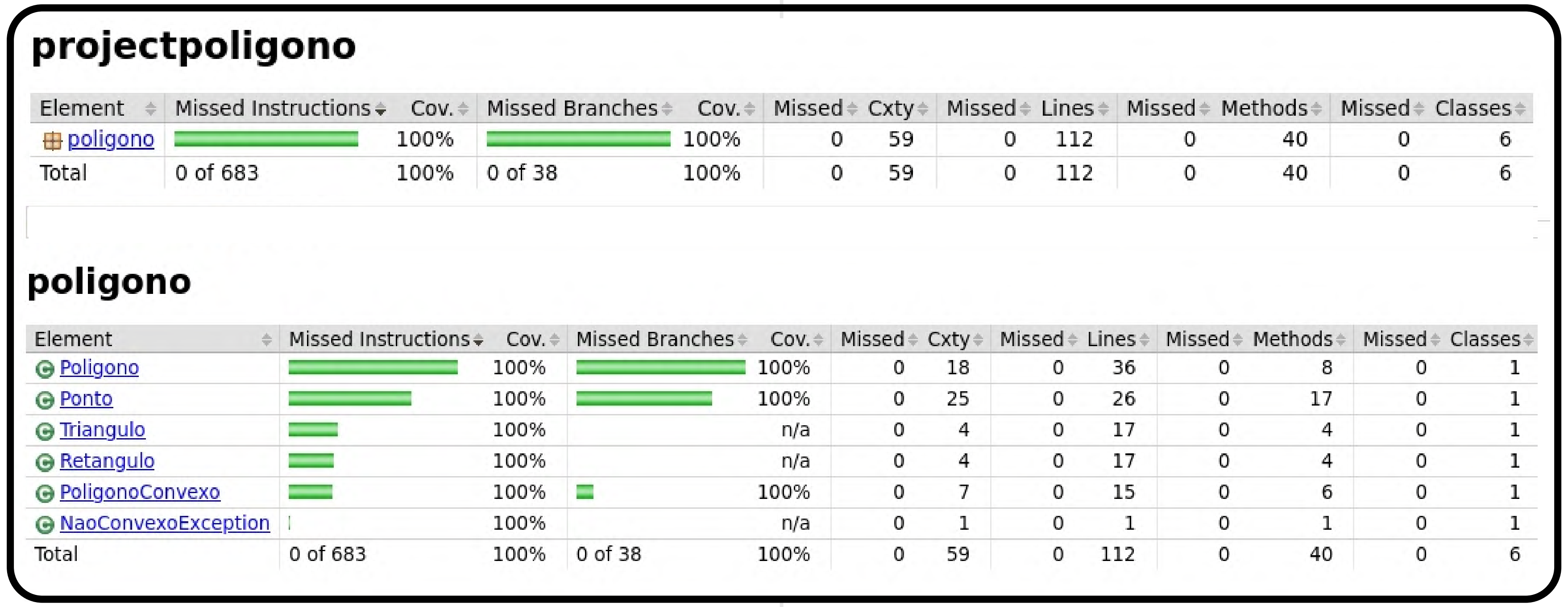


projectturma

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
turma	🟩	100%	🟩	100%	0	50	0	92	0	31	0	4
Total	0 of 405	100%	0 of 38	100%	0	50	0	92	0	31	0	4

turma

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
Aluno	🟩	100%	🟩	100%	0	25	0	50	0	15	0	1
Turma	🟩	100%	🟩	100%	0	23	0	38	0	14	0	1
NotaInvalidaException	🟩	100%		n/a	0	1	0	2	0	1	0	1
AlunoInexistenteException	🟩	100%		n/a	0	1	0	2	0	1	0	1
Total	0 of 405	100%	0 of 38	100%	0	50	0	92	0	31	0	4



projectpoligono

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
poligono	🟩	100%	🟩	100%	0	59	0	112	0	40	0	6
Total	0 of 683	100%	0 of 38	100%	0	59	0	112	0	40	0	6

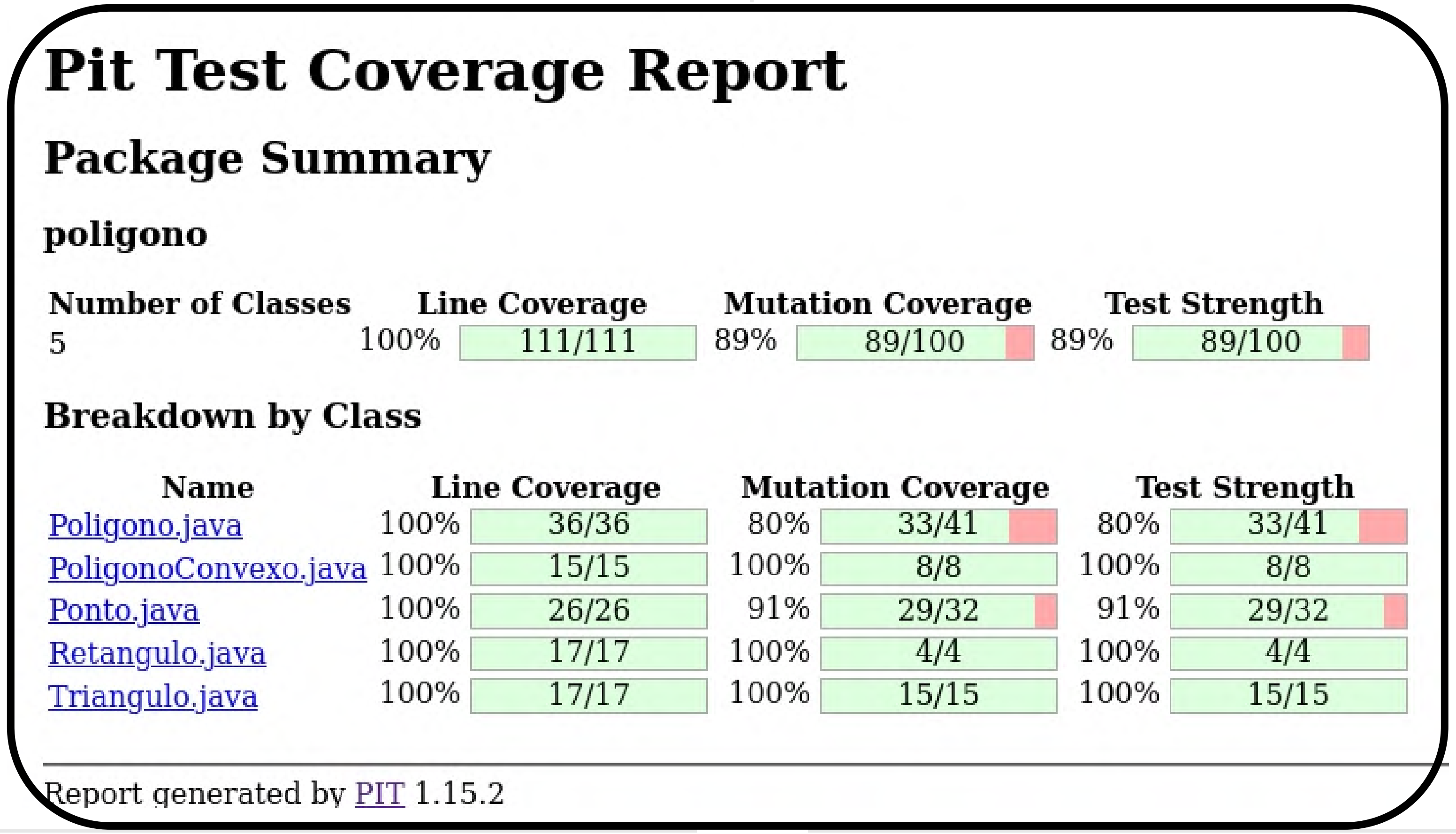
poligono

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
Poligono	🟩	100%	🟩	100%	0	18	0	36	0	8	0	1
Ponto	🟩	100%	🟩	100%	0	25	0	26	0	17	0	1
Triangulo	🟩	100%		n/a	0	4	0	17	0	4	0	1
Retangulo	🟩	100%		n/a	0	4	0	17	0	4	0	1
PoligonoConvexo	🟩	100%	🟩	100%	0	7	0	15	0	6	0	1
NaoConvexoException	🟩	100%		n/a	0	1	0	1	0	1	0	1
Total	0 of 683	100%	0 of 38	100%	0	59	0	112	0	40	0	6

JACOCO: EVOSUITE

Testes quantitativos

- 100% complexidades ciclomáticas (cxty) e cobertura linhas, *branches* para ambas as classes.



Pit Test Coverage Report

Package Summary

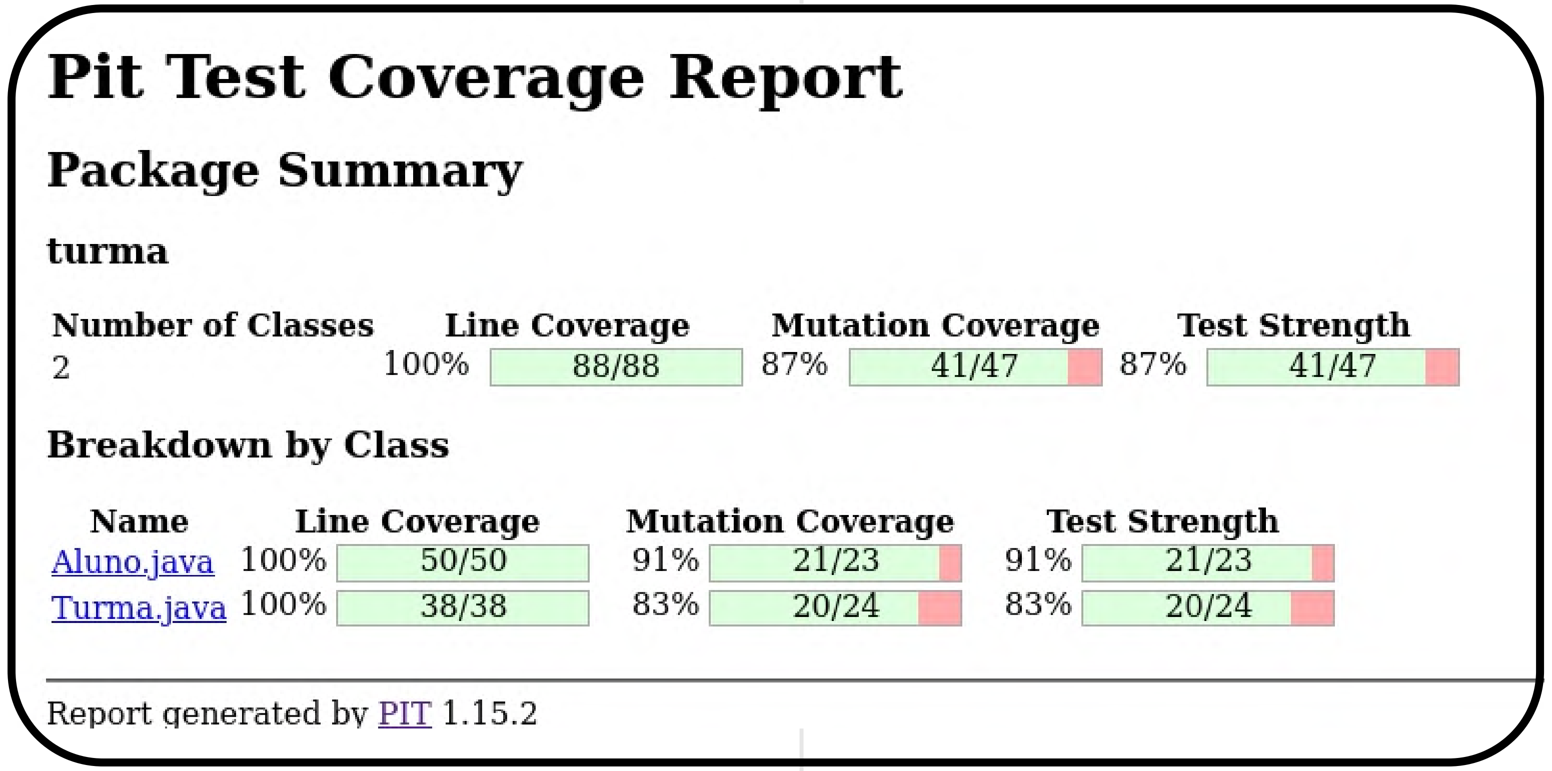
poligono

Number of Classes	Line Coverage	Mutation Coverage	Test Strength
5	100% 111/111	89% 89/100	89% 89/100

Breakdown by Class

Name	Line Coverage	Mutation Coverage	Test Strength
Poligono.java	100% 36/36	80% 33/41	80% 33/41
PoligonoConvexo.java	100% 15/15	100% 8/8	100% 8/8
Ponto.java	100% 26/26	91% 29/32	91% 29/32
Retangulo.java	100% 17/17	100% 4/4	100% 4/4
Triangulo.java	100% 17/17	100% 15/15	100% 15/15

Report generated by PIT 1.15.2



Pit Test Coverage Report

Package Summary

turma

Number of Classes	Line Coverage	Mutation Coverage	Test Strength
2	100% 88/88	87% 41/47	87% 41/47

Breakdown by Class

Name	Line Coverage	Mutation Coverage	Test Strength
Aluno.java	100% 50/50	91% 21/23	91% 21/23
Turma.java	100% 38/38	83% 20/24	83% 20/24

Report generated by PIT 1.15.2

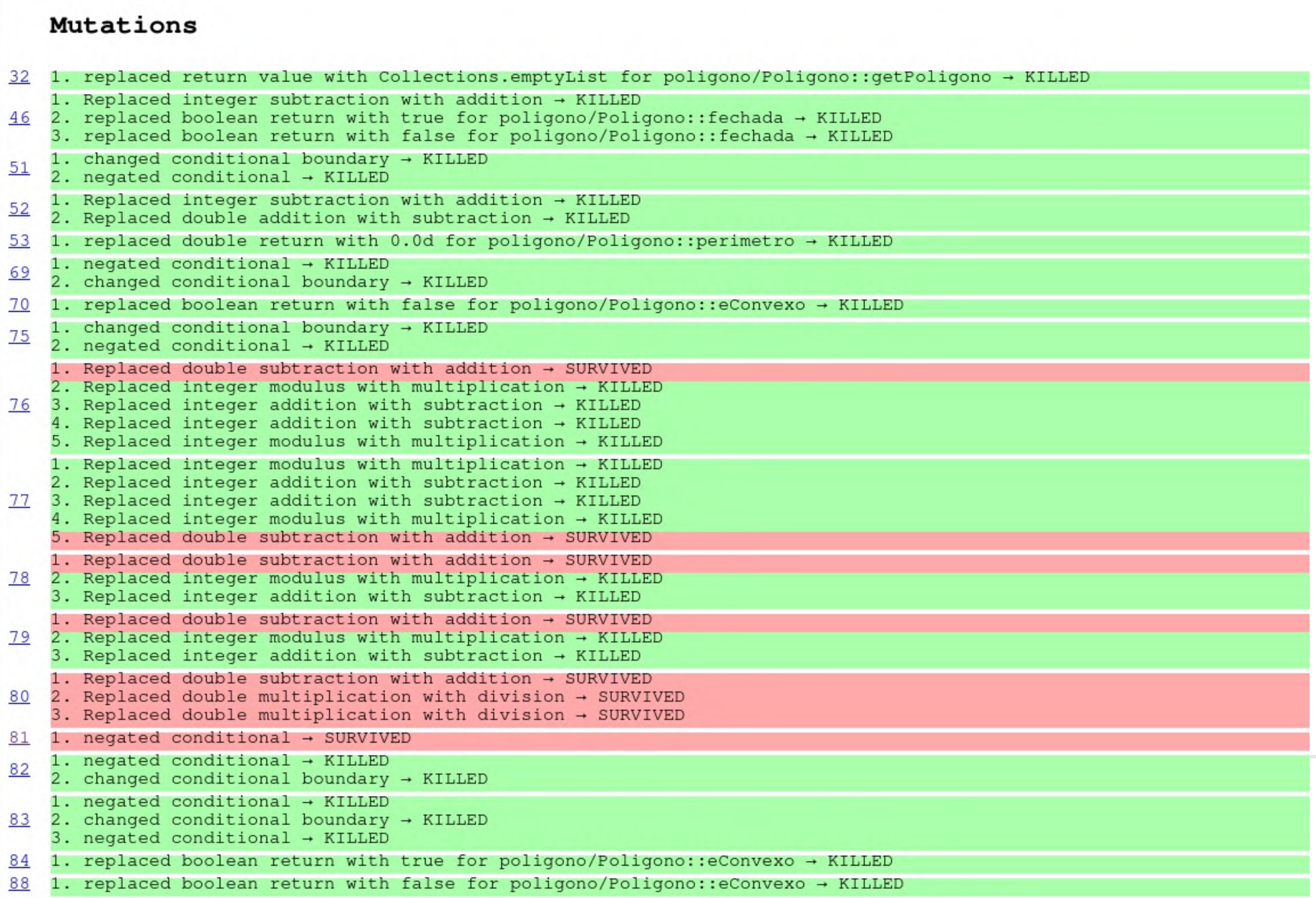
PITEST: EVOSUITE

Testes qualitativos

- Melhorias necessárias na deteção e na morte das modificações do código, como é evidente nos resultados de cobertura de mutações.

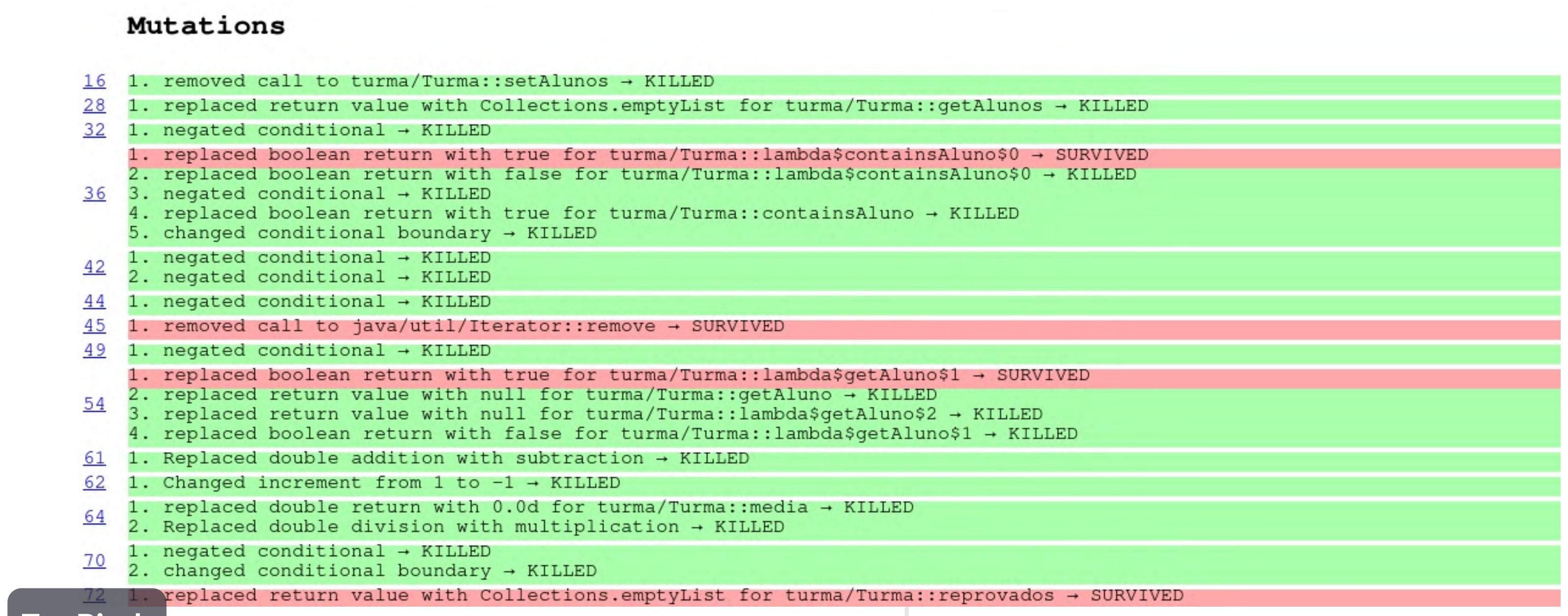
ANALISE DETALHADA PITEST: EVOSUITE

Poligono.java



```
Mutations

32  1. replaced return value with Collections.emptyList for poligono/Poligono::getPoligono → KILLED
    1. Replaced integer subtraction with addition → KILLED
46  2. replaced boolean return with true for poligono/Poligono::fechada → KILLED
    3. replaced boolean return with false for poligono/Poligono::fechada → KILLED
51  1. changed conditional boundary → KILLED
    2. negated conditional → KILLED
52  1. Replaced integer subtraction with addition → KILLED
    2. Replaced double addition with subtraction → KILLED
53  1. replaced double return with 0.0d for poligono/Poligono::perimetro → KILLED
69  1. negated conditional → KILLED
    2. changed conditional boundary → KILLED
70  1. replaced boolean return with false for poligono/Poligono::eConvexo → KILLED
75  1. changed conditional boundary → KILLED
    2. negated conditional → KILLED
    1. Replaced double subtraction with addition → SURVIVED
    2. Replaced integer modulus with multiplication → KILLED
76  3. Replaced integer addition with subtraction → KILLED
    4. Replaced integer addition with subtraction → KILLED
    5. Replaced integer modulus with multiplication → KILLED
    1. Replaced integer modulus with multiplication → KILLED
    2. Replaced integer addition with subtraction → KILLED
77  3. Replaced integer addition with subtraction → KILLED
    4. Replaced integer modulus with multiplication → KILLED
    5. Replaced double subtraction with addition → SURVIVED
    1. Replaced double subtraction with addition → SURVIVED
78  2. Replaced integer modulus with multiplication → KILLED
    3. Replaced integer addition with subtraction → KILLED
    1. Replaced double subtraction with addition → SURVIVED
79  2. Replaced integer modulus with multiplication → KILLED
    3. Replaced integer addition with subtraction → KILLED
    1. Replaced double subtraction with addition → SURVIVED
80  2. Replaced double multiplication with division → SURVIVED
    3. Replaced double multiplication with division → SURVIVED
81  1. negated conditional → SURVIVED
82  1. negated conditional → KILLED
    2. changed conditional boundary → KILLED
    1. negated conditional → KILLED
83  2. changed conditional boundary → KILLED
    3. negated conditional → KILLED
84  1. replaced boolean return with true for poligono/Poligono::eConvexo → KILLED
88  1. replaced boolean return with false for poligono/Poligono::eConvexo → KILLED
```

Turma.java



```
Mutations

16  1. removed call to turma/Turma::setAlunos → KILLED
28  1. replaced return value with Collections.emptyList for turma/Turma::getAlunos → KILLED
32  1. negated conditional → KILLED
    1. replaced boolean return with true for turma/Turma::lambda$containsAluno$0 → SURVIVED
    2. replaced boolean return with false for turma/Turma::lambda$containsAluno$0 → KILLED
36  3. negated conditional → KILLED
    4. replaced boolean return with true for turma/Turma::containsAluno → KILLED
    5. changed conditional boundary → KILLED
42  1. negated conditional → KILLED
    2. negated conditional → KILLED
44  1. negated conditional → KILLED
45  1. removed call to java/util/Iterator::remove → SURVIVED
49  1. negated conditional → KILLED
    1. replaced boolean return with true for turma/Turma::lambda$getAluno$1 → SURVIVED
54  2. replaced return value with null for turma/Turma::getAluno → KILLED
    3. replaced return value with null for turma/Turma::lambda$getAluno$2 → KILLED
    4. replaced boolean return with false for turma/Turma::lambda$getAluno$1 → KILLED
61  1. Replaced double addition with subtraction → KILLED
62  1. Changed increment from 1 to -1 → KILLED
64  1. replaced double return with 0.0d for turma/Turma::media → KILLED
    2. Replaced double division with multiplication → KILLED
70  1. negated conditional → KILLED
    2. changed conditional boundary → KILLED
72  1. replaced return value with Collections.emptyList for turma/Turma::reprovados → SURVIVED
```

- Destacam-se substituições de operadores matemáticos em operações com números em vírgula flutuante.

- Falhas frequentes em testes sobre estruturas de dados.

CONCLUSÕES: EVOSUITE

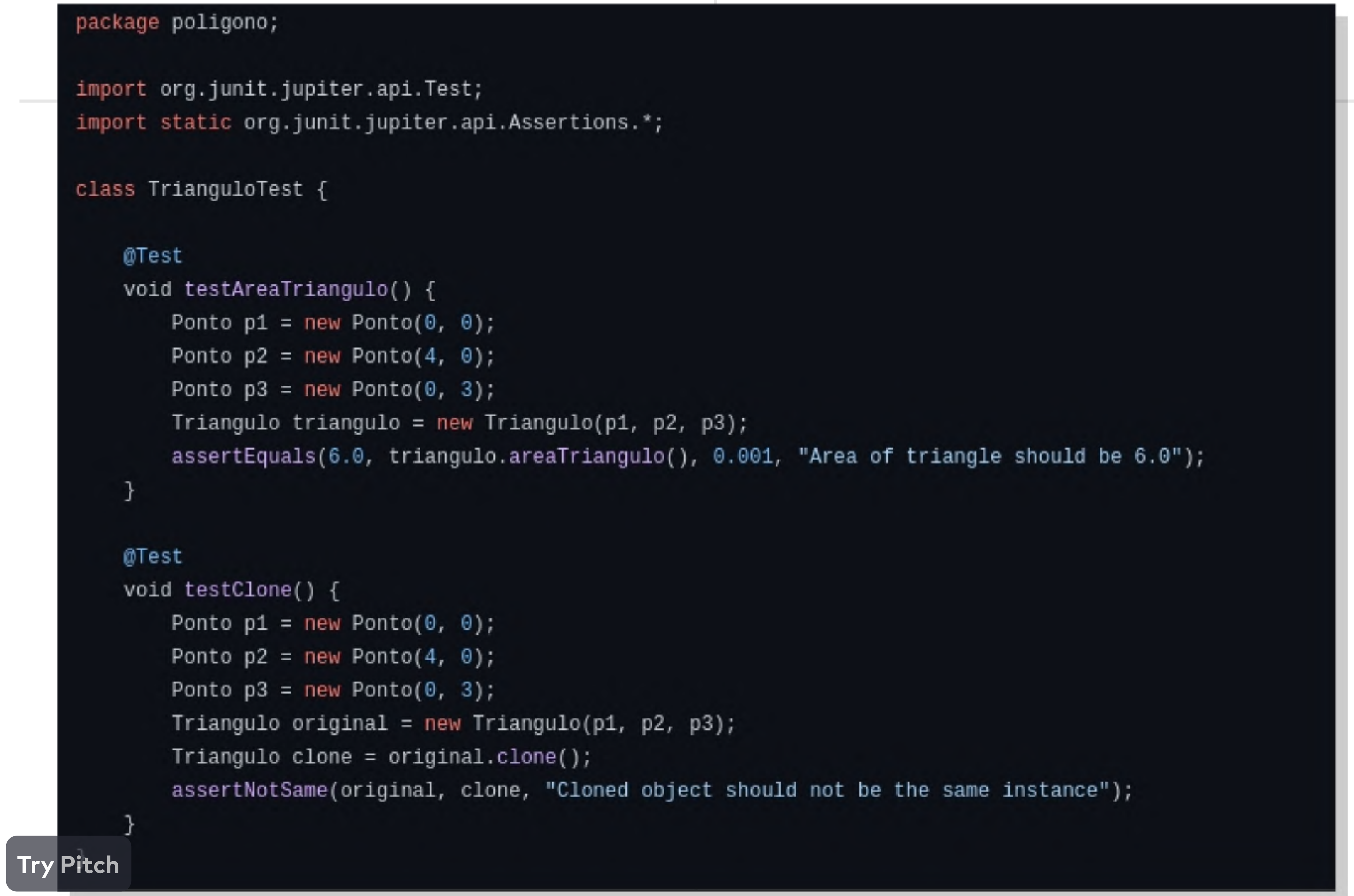
- Configuração desafiante do JaCoCo e EvoSuite.

- Apresenta capacidade de gerar testes facilmente.

- A análise pelo JaCoCo e Pitest revela espaço para melhorias na cobertura de testes.

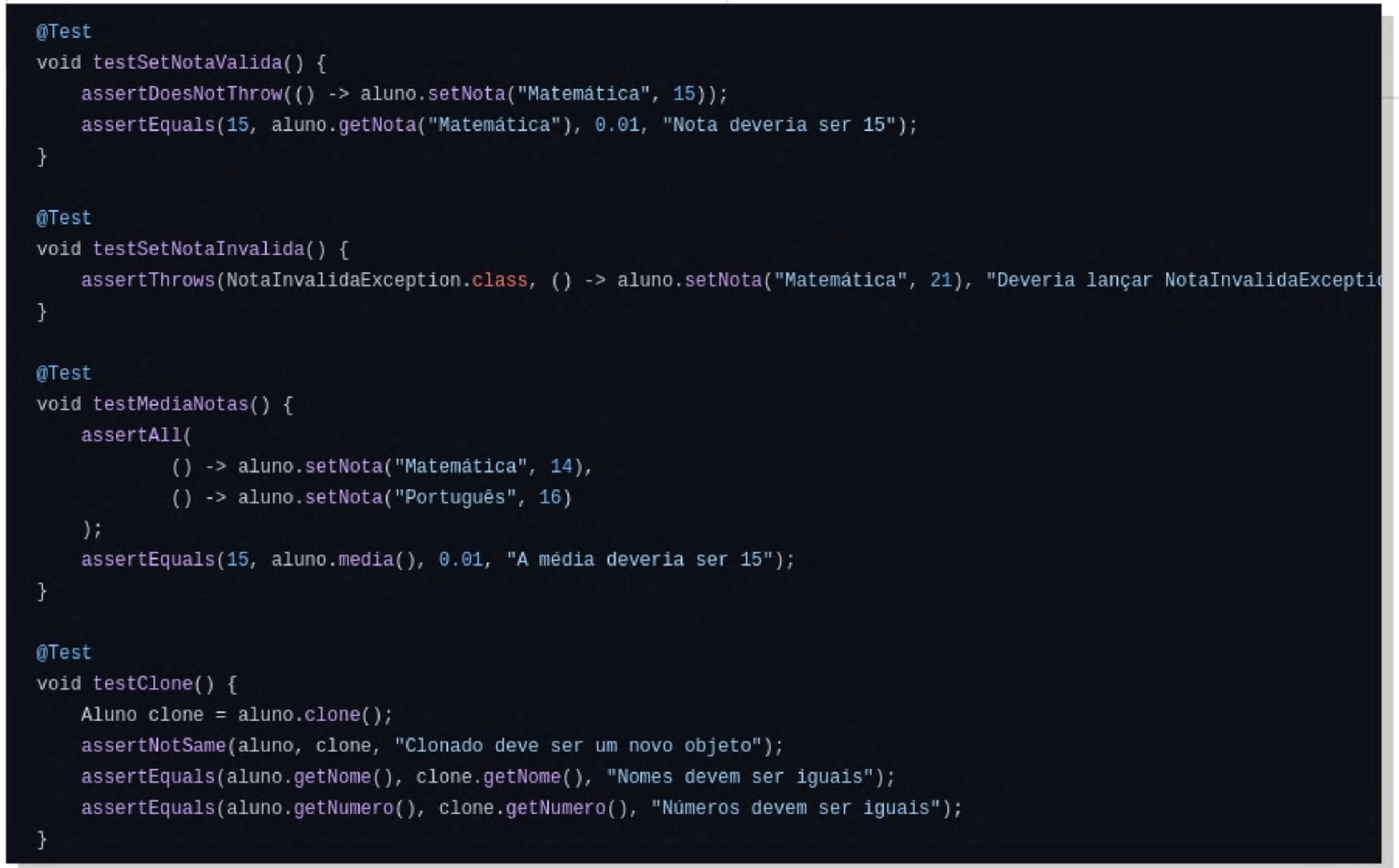
EXEMPLOS DOS TESTES GERADOS: CHATGPT

Projeto: Poligno
TrianguloTest.java



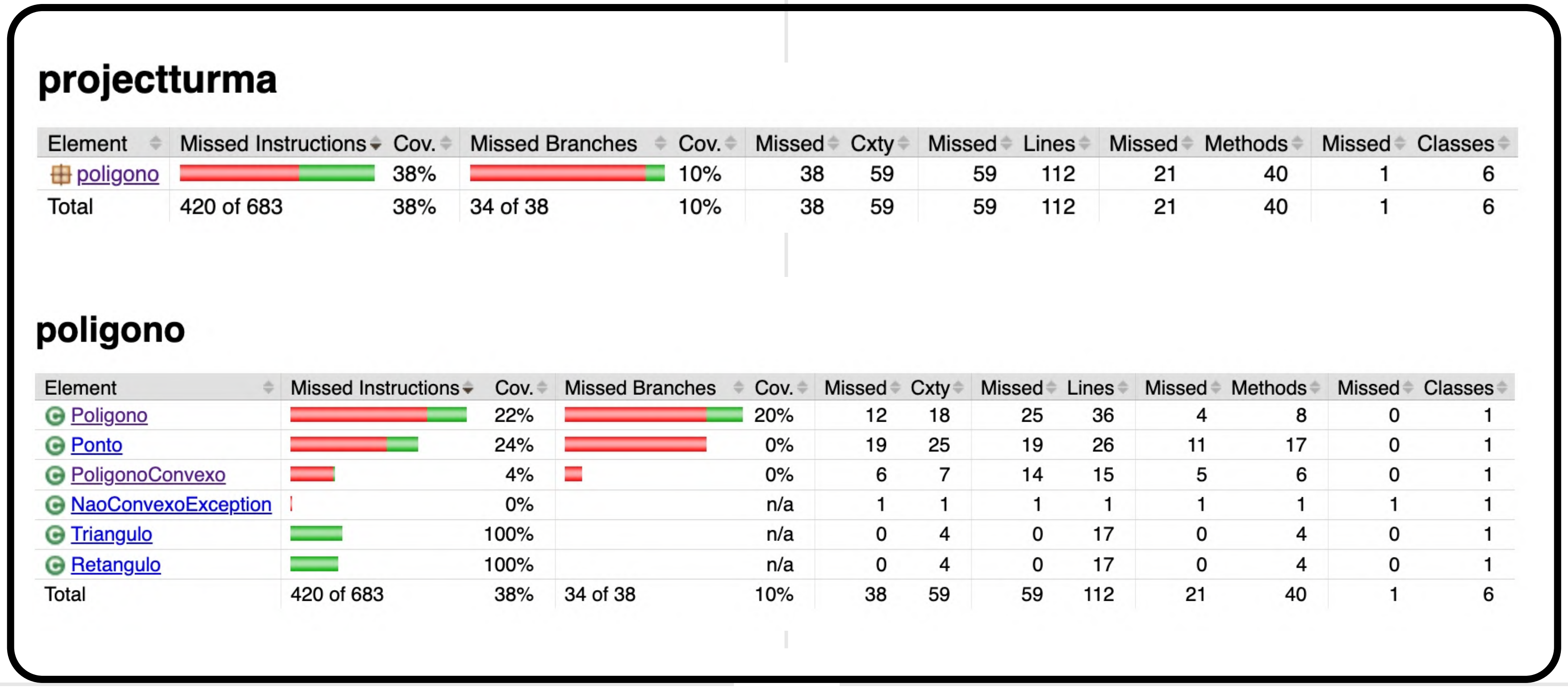
```java
package poligono;

import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

class TrianguloTest {

    @Test
    void testAreaTriangulo() {
        Ponto p1 = new Ponto(0, 0);
        Ponto p2 = new Ponto(4, 0);
        Ponto p3 = new Ponto(0, 3);
        Triangulo triangulo = new Triangulo(p1, p2, p3);
        assertEquals(6.0, triangulo.areaTriangulo(), 0.001, "Area of triangle should be 6.0");
    }

    @Test
    void testClone() {
        Ponto p1 = new Ponto(0, 0);
        Ponto p2 = new Ponto(4, 0);
        Ponto p3 = new Ponto(0, 3);
        Triangulo original = new Triangulo(p1, p2, p3);
        Triangulo clone = original.clone();
        assertNotSame(original, clone, "Cloned object should not be the same instance");
    }
```

Projeto: Turma
AlunoTest.java



```java
@Test
void testSetNotaValida() {
    assertDoesNotThrow(() -> aluno.setNota("Matemática", 15));
    assertEquals(15, aluno.getNota("Matemática"), 0.01, "Nota deveria ser 15");
}

@Test
void testSetNotaInvalida() {
    assertThrows(NotaInvalidaException.class, () -> aluno.setNota("Matemática", 21), "Deveria lançar NotaInvalidaExceptio
}

@Test
void testMediaNotas() {
    assertAll(
            () -> aluno.setNota("Matemática", 14),
            () -> aluno.setNota("Português", 16)
    );
    assertEquals(15, aluno.media(), 0.01, "A média deveria ser 15");
}

@Test
void testClone() {
    Aluno clone = aluno.clone();
    assertNotSame(aluno, clone, "Clonado deve ser um novo objeto");
    assertEquals(aluno.getNome(), clone.getNome(), "Nomes devem ser iguais");
    assertEquals(aluno.getNumero(), clone.getNumero(), "Números devem ser iguais");
}
```

JACOCO: CHATGPT

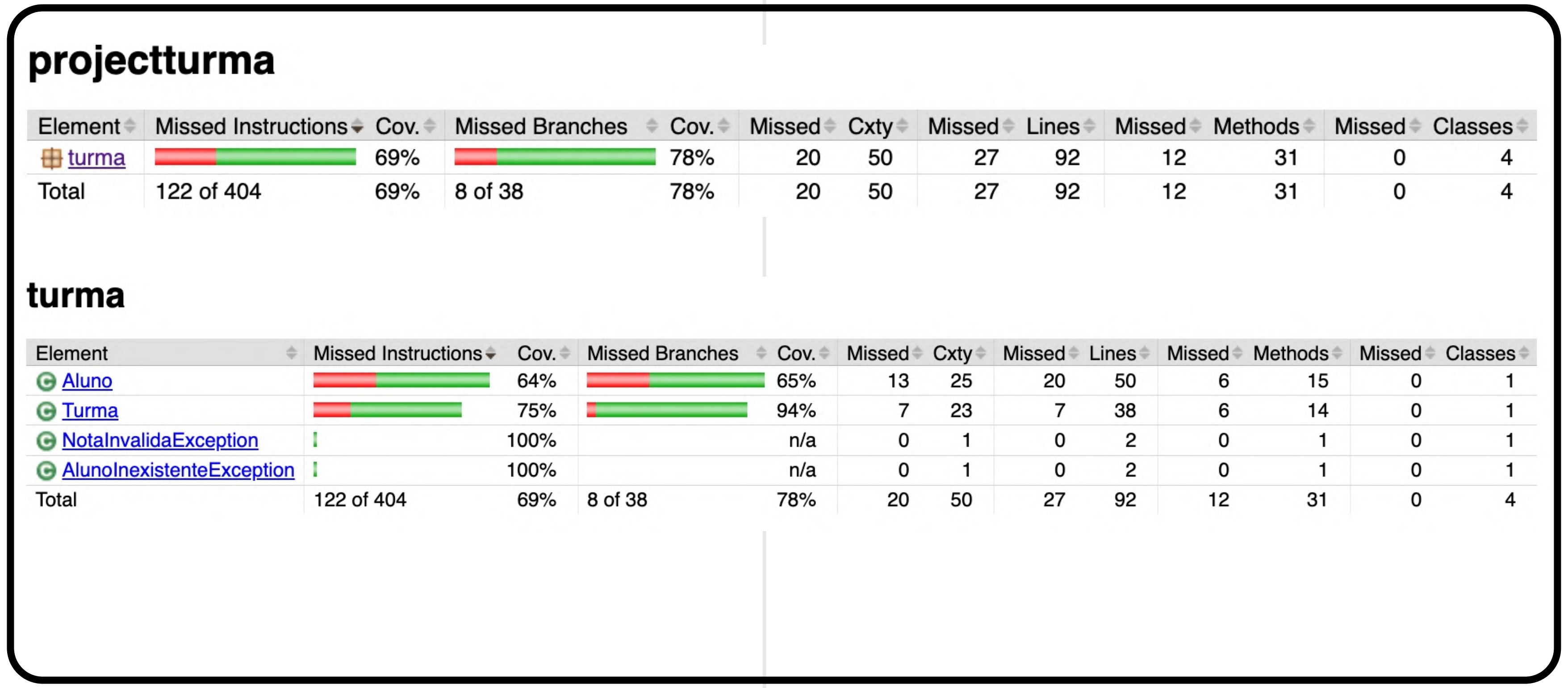


projectturma

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
poligono		38%		10%	38	59	59	112	21	40	1	6
Total	420 of 683	38%	34 of 38	10%	38	59	59	112	21	40	1	6

poligono

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
Poligono		22%		20%	12	18	25	36	4	8	0	1
Ponto		24%		0%	19	25	19	26	11	17	0	1
PoligonoConvexo		4%		0%	6	7	14	15	5	6	0	1
NaoConvexoException		0%		n/a	1	1	1	1	1	1	1	1
Triangulo		100%		n/a	0	4	0	17	0	4	0	1
Retangulo		100%		n/a	0	4	0	17	0	4	0	1
Total	420 of 683	38%	34 of 38	10%	38	59	59	112	21	40	1	6



projectturma

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
turma		69%		78%	20	50	27	92	12	31	0	4
Total	122 of 404	69%	8 of 38	78%	20	50	27	92	12	31	0	4

turma

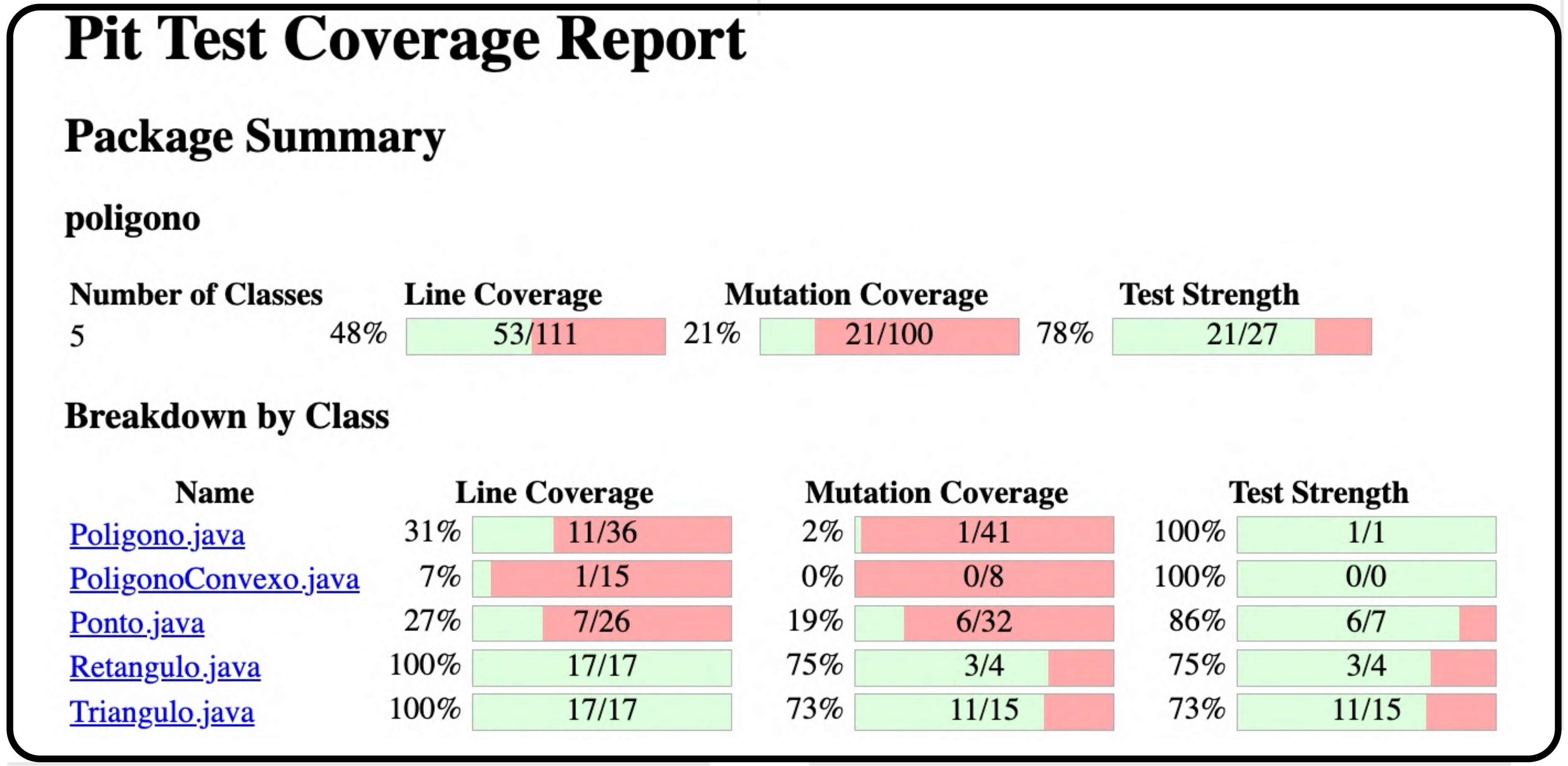
Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
Aluno		64%		65%	13	25	20	50	6	15	0	1
Turma		75%		94%	7	23	7	38	6	14	0	1
NotaInvalidaException		100%		n/a	0	1	0	2	0	1	0	1
AlunoInexistenteException		100%		n/a	0	1	0	2	0	1	0	1
Total	122 of 404	69%	8 of 38	78%	20	50	27	92	12	31	0	4

Testes quantitativos

- Apenas 38% das instruções executadas pelos testes.
- Apenas 10% de *branches* executados.
- Projeto *Turma* com cobertura satisfatória de instruções(69%) e cobertura de branches(78%).

PITEST: CHATGPT

Testes qualitativos



Pit Test Coverage Report

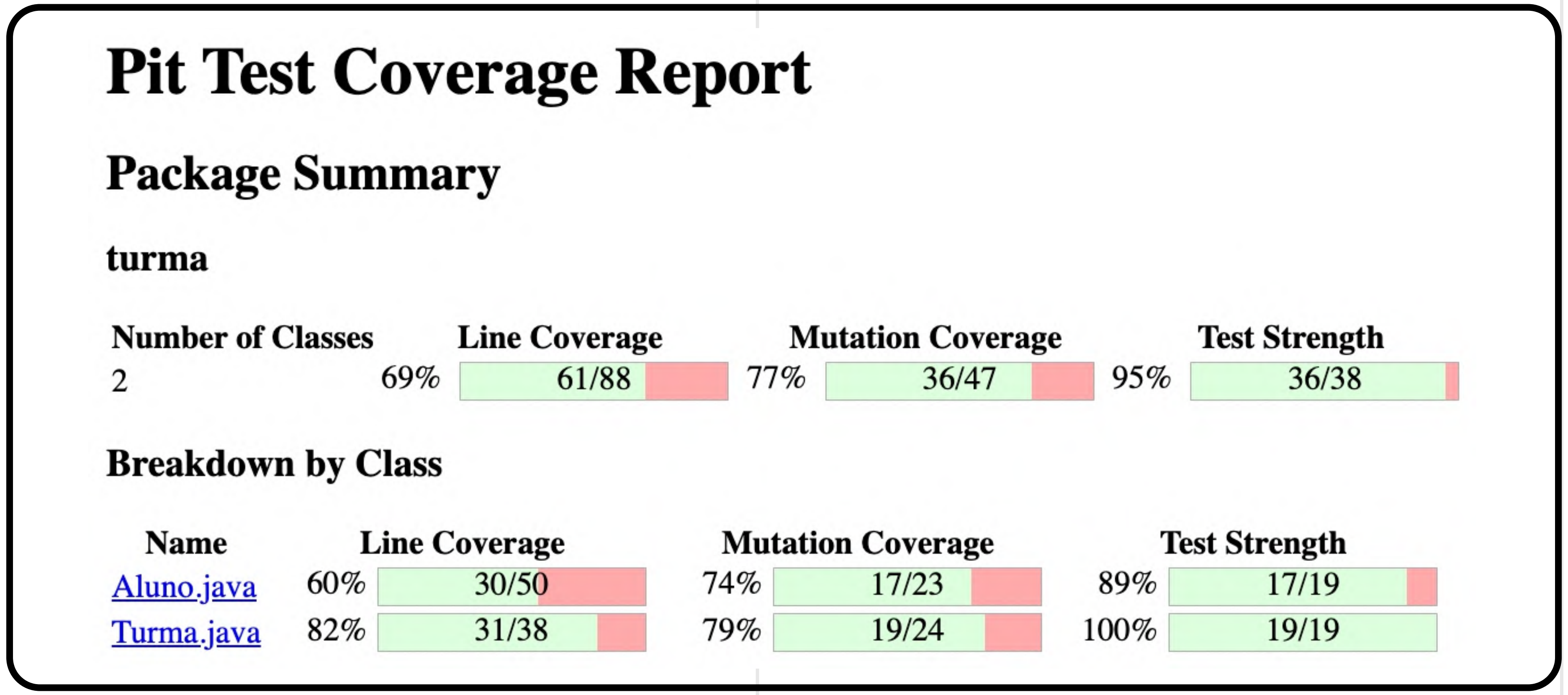
Package Summary

poligono

Number of Classes	Line Coverage		Mutation Coverage		Test Strength	
5	48%	53/111	21%	21/100	78%	21/27

Breakdown by Class

Name	Line Coverage		Mutation Coverage		Test Strength	
Poligono.java	31%	11/36	2%	1/41	100%	1/1
PoligonoConvexo.java	7%	1/15	0%	0/8	100%	0/0
Ponto.java	27%	7/26	19%	6/32	86%	6/7
Retangulo.java	100%	17/17	75%	3/4	75%	3/4
Triangulo.java	100%	17/17	73%	11/15	73%	11/15



Pit Test Coverage Report

Package Summary

turma

Number of Classes	Line Coverage		Mutation Coverage		Test Strength	
2	69%	61/88	77%	36/47	95%	36/38

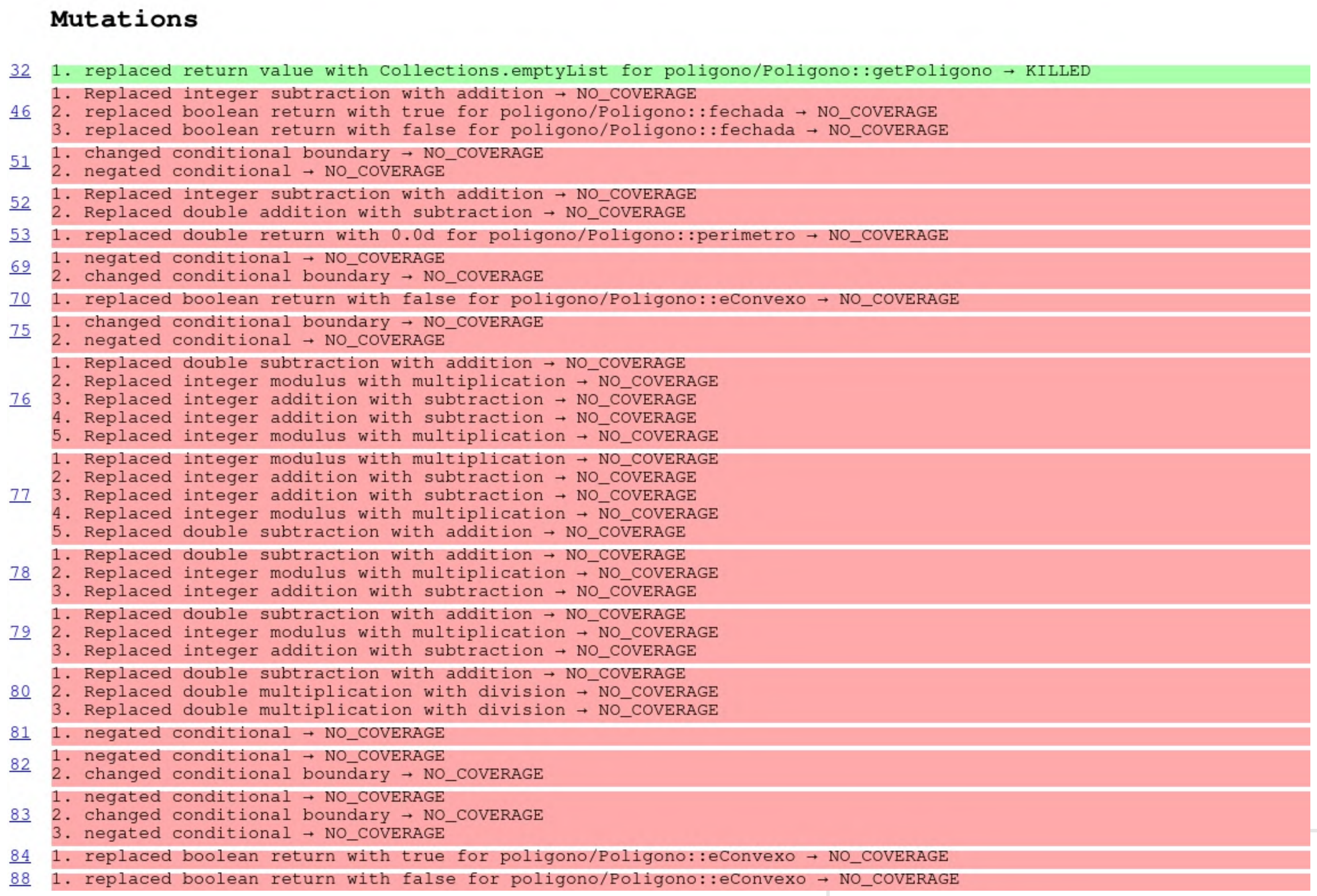
Breakdown by Class

Name	Line Coverage		Mutation Coverage		Test Strength	
Aluno.java	60%	30/50	74%	17/23	89%	17/19
Turma.java	82%	31/38	79%	19/24	100%	19/19

Em comparação às outras ferramentas:

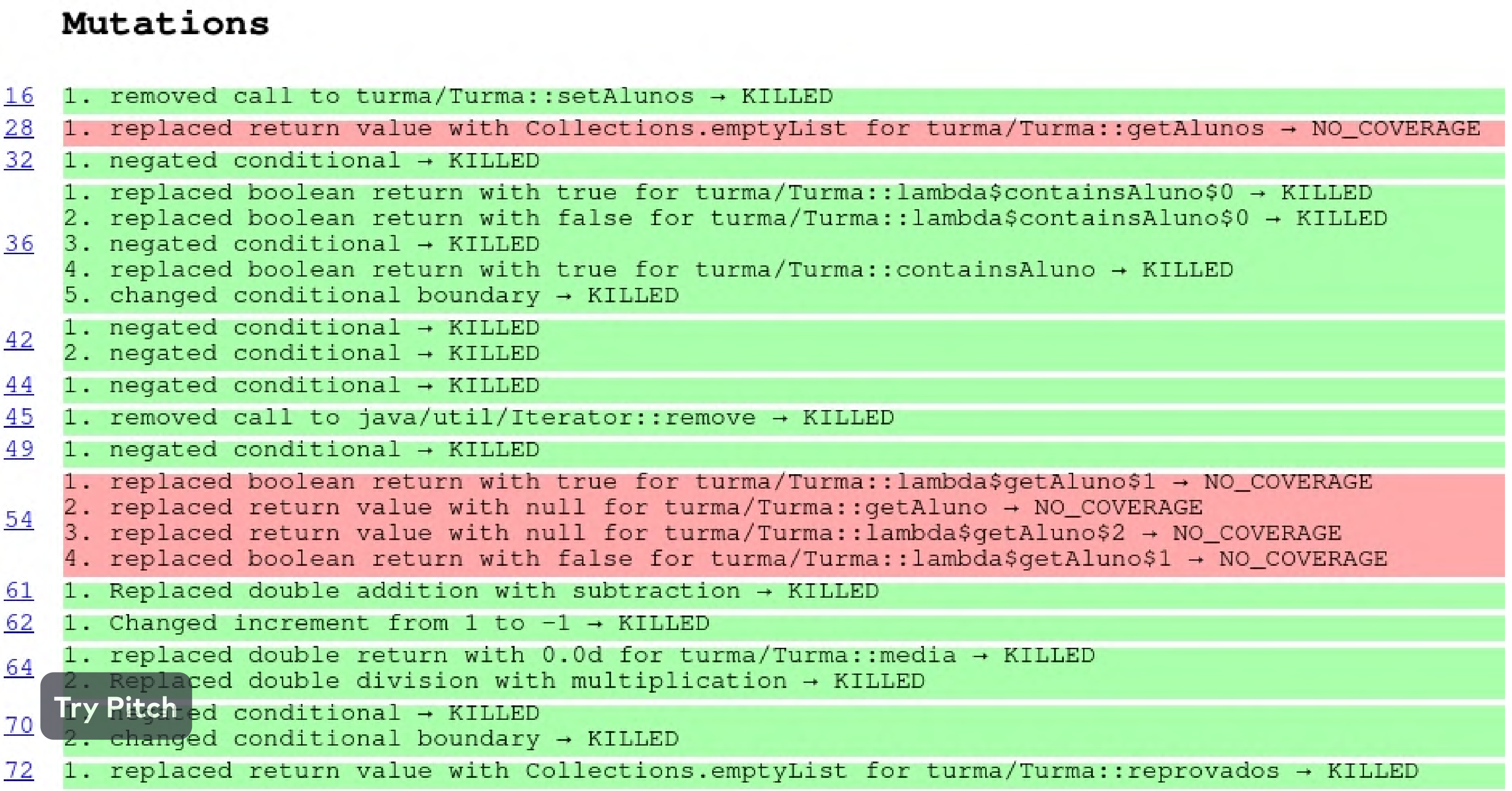
- Cobertura de mutação mais baixa.
- Testes menos eficazes.

Poligono.java



```
    Mutations

32  1. replaced return value with Collections.emptyList for poligono/Poligono::getPoligono → KILLED
    1. Replaced integer subtraction with addition → NO_COVERAGE
46  2. replaced boolean return with true for poligono/Poligono::fechada → NO_COVERAGE
    3. replaced boolean return with false for poligono/Poligono::fechada → NO_COVERAGE
51  1. changed conditional boundary → NO_COVERAGE
    2. negated conditional → NO_COVERAGE
52  1. Replaced integer subtraction with addition → NO_COVERAGE
    2. Replaced double addition with subtraction → NO_COVERAGE
53  1. replaced double return with 0.0d for poligono/Poligono::perimetro → NO_COVERAGE
69  1. negated conditional → NO_COVERAGE
    2. changed conditional boundary → NO_COVERAGE
70  1. replaced boolean return with false for poligono/Poligono::eConvexo → NO_COVERAGE
75  1. changed conditional boundary → NO_COVERAGE
    2. negated conditional → NO_COVERAGE
    1. Replaced double subtraction with addition → NO_COVERAGE
    2. Replaced integer modulus with multiplication → NO_COVERAGE
76  3. Replaced integer addition with subtraction → NO_COVERAGE
    4. Replaced integer addition with subtraction → NO_COVERAGE
    5. Replaced integer modulus with multiplication → NO_COVERAGE
    1. Replaced integer modulus with multiplication → NO_COVERAGE
    2. Replaced integer addition with subtraction → NO_COVERAGE
77  3. Replaced integer addition with subtraction → NO_COVERAGE
    4. Replaced integer modulus with multiplication → NO_COVERAGE
    5. Replaced double subtraction with addition → NO_COVERAGE
    1. Replaced double subtraction with addition → NO_COVERAGE
78  2. Replaced integer modulus with multiplication → NO_COVERAGE
    3. Replaced integer addition with subtraction → NO_COVERAGE
    1. Replaced double subtraction with addition → NO_COVERAGE
79  2. Replaced integer modulus with multiplication → NO_COVERAGE
    3. Replaced integer addition with subtraction → NO_COVERAGE
    1. Replaced double subtraction with addition → NO_COVERAGE
80  2. Replaced double multiplication with division → NO_COVERAGE
    3. Replaced double multiplication with division → NO_COVERAGE
81  1. negated conditional → NO_COVERAGE
82  1. negated conditional → NO_COVERAGE
    2. changed conditional boundary → NO_COVERAGE
    1. negated conditional → NO_COVERAGE
83  2. changed conditional boundary → NO_COVERAGE
    3. negated conditional → NO_COVERAGE
84  1. replaced boolean return with true for poligono/Poligono::eConvexo → NO_COVERAGE
88  1. replaced boolean return with false for poligono/Poligono::eConvexo → NO_COVERAGE
```

Turma.java



```
    Mutations

16  1. removed call to turma/Turma::setAlunos → KILLED
28  1. replaced return value with Collections.emptyList for turma/Turma::getAlunos → NO_COVERAGE
32  1. negated conditional → KILLED
    1. replaced boolean return with true for turma/Turma::lambda$containsAluno$0 → KILLED
    2. replaced boolean return with false for turma/Turma::lambda$containsAluno$0 → KILLED
36  3. negated conditional → KILLED
    4. replaced boolean return with true for turma/Turma::containsAluno → KILLED
    5. changed conditional boundary → KILLED
42  1. negated conditional → KILLED
    2. negated conditional → KILLED
44  1. negated conditional → KILLED
45  1. removed call to java/util/Iterator::remove → KILLED
49  1. negated conditional → KILLED
    1. replaced boolean return with true for turma/Turma::lambda$getAluno$1 → NO_COVERAGE
54  2. replaced return value with null for turma/Turma::getAluno → NO_COVERAGE
    3. replaced return value with null for turma/Turma::lambda$getAluno$2 → NO_COVERAGE
    4. replaced boolean return with false for turma/Turma::lambda$getAluno$1 → NO_COVERAGE
61  1. Replaced double addition with subtraction → KILLED
62  1. Changed increment from 1 to -1 → KILLED
64  1. replaced double return with 0.0d for turma/Turma::media → KILLED
    2. Replaced double division with multiplication → KILLED
70  1. negated conditional → KILLED
    2. changed conditional boundary → KILLED
72  1. replaced return value with Collections.emptyList for turma/Turma::reprovados → KILLED
```

ANALISE DETALHADA PITEST: CHATGPT

Poligono

- Fraca qualidade e cobertura de testes
- Verificou-se uma frequente substituição de operadores matemáticos e booleanos

Turma

- Menor dificuldade criação testes, com melhor qualidade

DIFICULDADES DE TESTES: INTELIGENCIA ARTIFICIAL

- Testes com frequentes erros de escrita.

- Destaque na importância da validação manual e compreensão do contexto global do projeto.

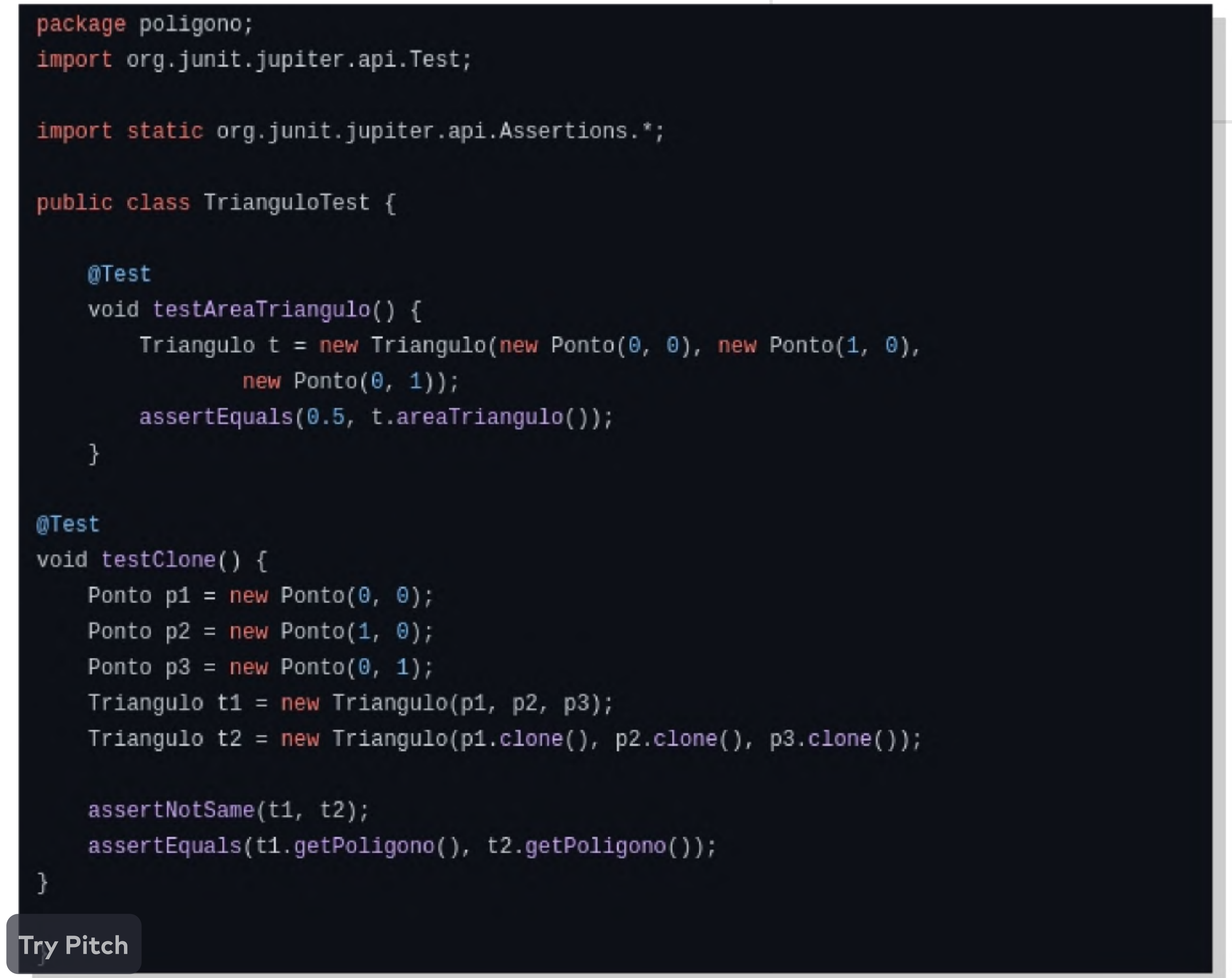
CONCLUSOES: CHAT GPT

- Melhor implementação de código em *IA.*
- Testes gerados piores comparativamente ao *EvoSuite*.
- Necessária muita "manutenção" pelo utilizador.
- Análise limitada pela versão gratuita

EXEMPLOS DOS TESTES GERADOS:

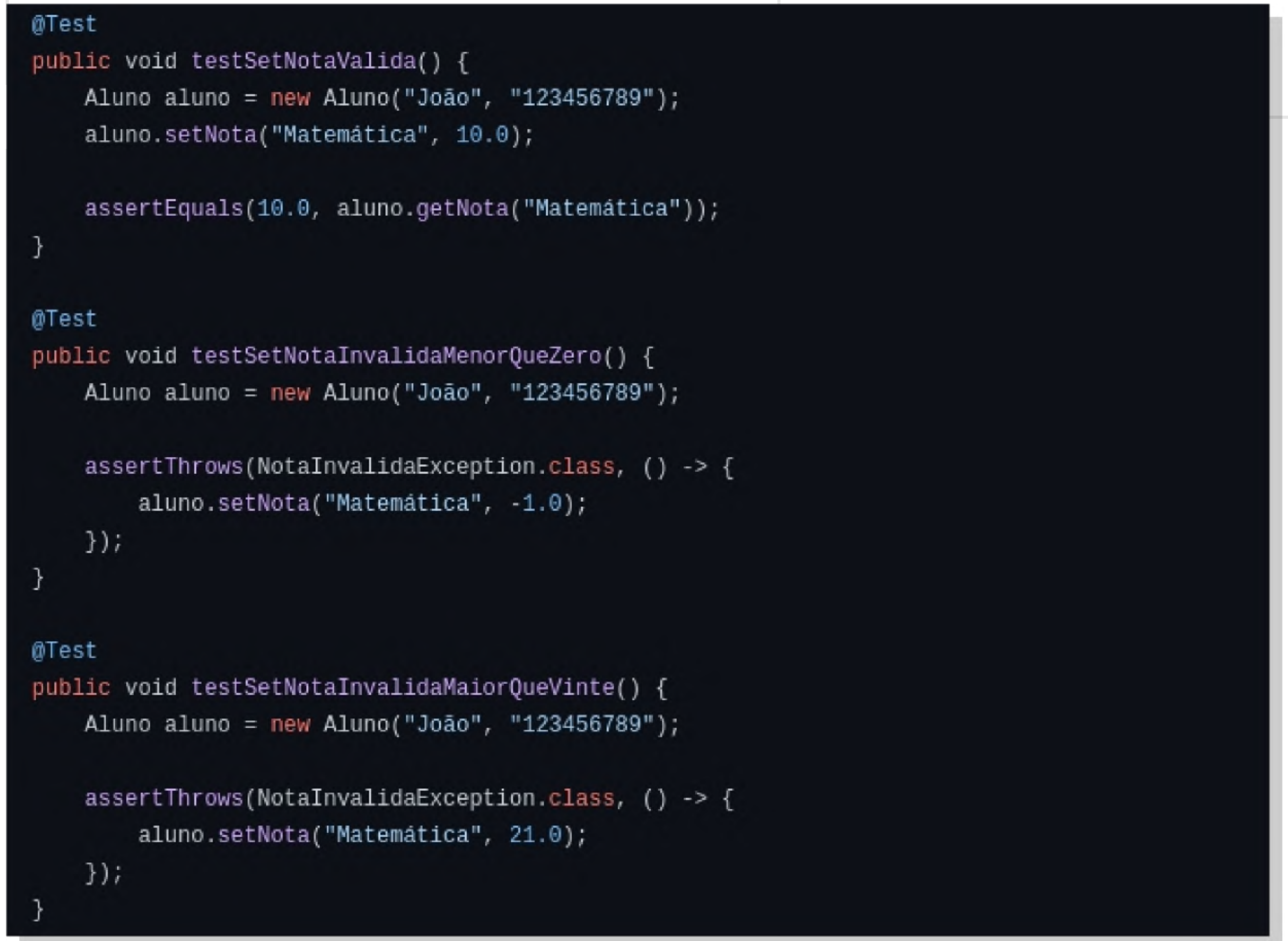
GEMINI Testes dados

Projeto: Poligno
TrianguloTest.java



```java
package poligono;
import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.*;

public class TrianguloTest {

    @Test
    void testAreaTriangulo() {
        Triangulo t = new Triangulo(new Ponto(0, 0), new Ponto(1, 0),
                new Ponto(0, 1));
        assertEquals(0.5, t.areaTriangulo());
    }

@Test
void testClone() {
    Ponto p1 = new Ponto(0, 0);
    Ponto p2 = new Ponto(1, 0);
    Ponto p3 = new Ponto(0, 1);
    Triangulo t1 = new Triangulo(p1, p2, p3);
    Triangulo t2 = new Triangulo(p1.clone(), p2.clone(), p3.clone());

    assertNotSame(t1, t2);
    assertEquals(t1.getPoligono(), t2.getPoligono());
}
```

Projeto: Turma
AlunoTest.java

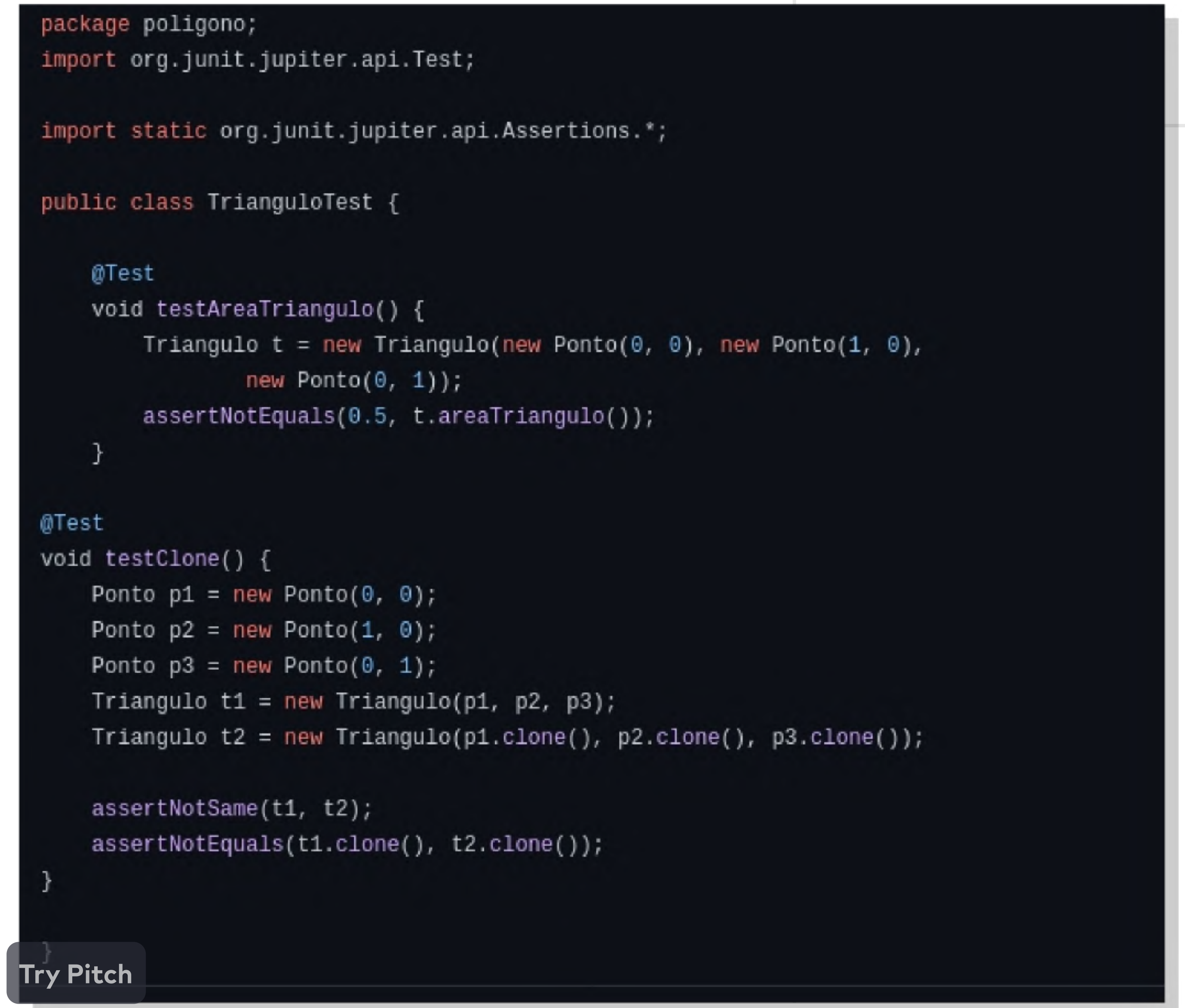


```java
@Test
public void testSetNotaValida() {
    Aluno aluno = new Aluno("João", "123456789");
    aluno.setNota("Matemática", 10.0);

    assertEquals(10.0, aluno.getNota("Matemática"));
}

@Test
public void testSetNotaInvalidaMenorQueZero() {
    Aluno aluno = new Aluno("João", "123456789");

    assertThrows(NotaInvalidaException.class, () -> {
        aluno.setNota("Matemática", -1.0);
    });
}

@Test
public void testSetNotaInvalidaMaiorQueVinte() {
    Aluno aluno = new Aluno("João", "123456789");

    assertThrows(NotaInvalidaException.class, () -> {
        aluno.setNota("Matemática", 21.0);
    });
}
```

EXEMPLOS DOS TESTES GERADOS:

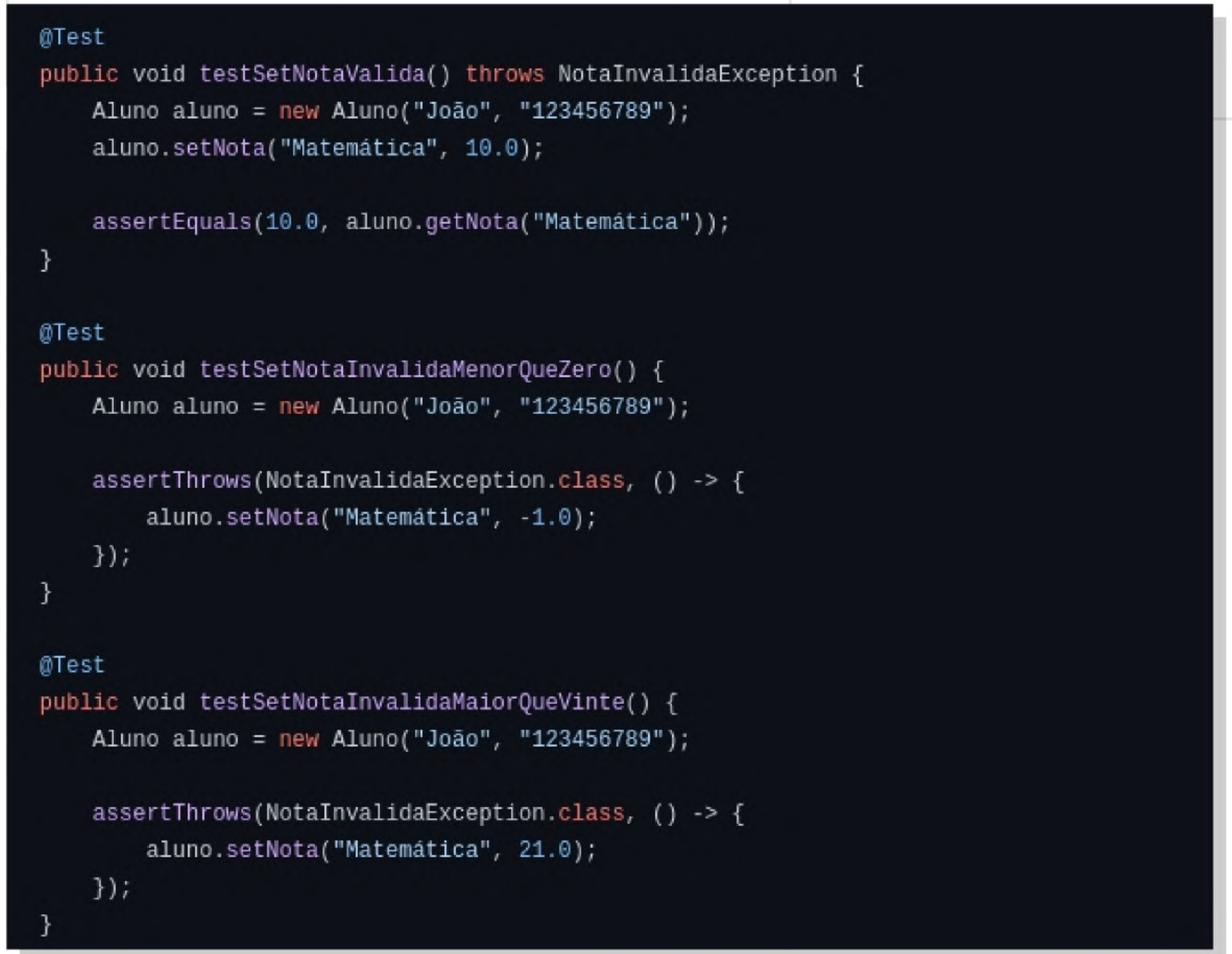
GEMINI Testes corrigidos

Projeto: Poligno
TrianguloTest.java



```
package poligono;
import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.*;

public class TrianguloTest {

    @Test
    void testAreaTriangulo() {
        Triangulo t = new Triangulo(new Ponto(0, 0), new Ponto(1, 0),
                new Ponto(0, 1));
        assertNotEquals(0.5, t.areaTriangulo());
    }

@Test
void testClone() {
    Ponto p1 = new Ponto(0, 0);
    Ponto p2 = new Ponto(1, 0);
    Ponto p3 = new Ponto(0, 1);
    Triangulo t1 = new Triangulo(p1, p2, p3);
    Triangulo t2 = new Triangulo(p1.clone(), p2.clone(), p3.clone());

    assertNotSame(t1, t2);
    assertNotEquals(t1.clone(), t2.clone());
}

}
```

Projeto: Turma
AlunoTest.java



```
@Test
public void testSetNotaValida() throws NotaInvalidaException {
    Aluno aluno = new Aluno("João", "123456789");
    aluno.setNota("Matemática", 10.0);

    assertEquals(10.0, aluno.getNota("Matemática"));
}

@Test
public void testSetNotaInvalidaMenorQueZero() {
    Aluno aluno = new Aluno("João", "123456789");

    assertThrows(NotaInvalidaException.class, () -> {
        aluno.setNota("Matemática", -1.0);
    });
}

@Test
public void testSetNotaInvalidaMaiorQueVinte() {
    Aluno aluno = new Aluno("João", "123456789");

    assertThrows(NotaInvalidaException.class, () -> {
        aluno.setNota("Matemática", 21.0);
    });
}
```

CORRECOES DOS TESTES GERADOS: GEMINI

- Falhas frequentes em elementos de *POO* e testes unitários

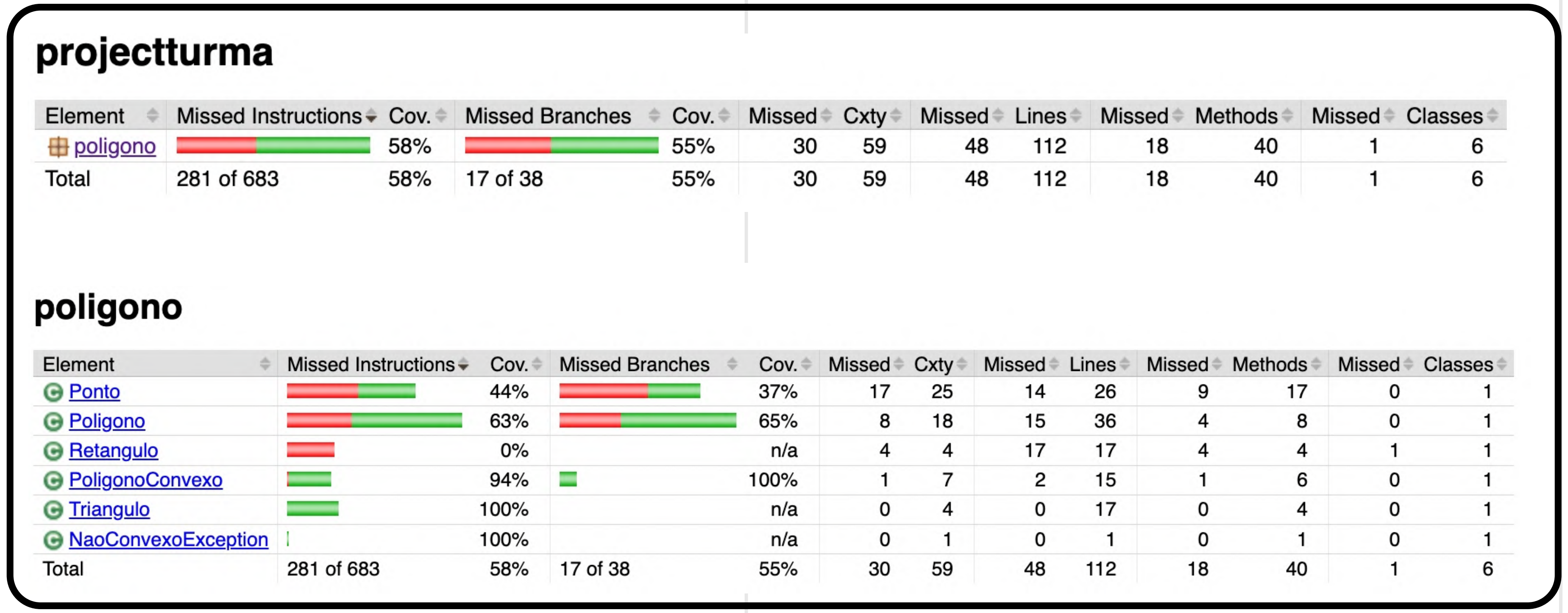
Projeto Turma

- Falha na utilização de exceções do projeto.
- Utilização de métodos inválidos.
- Falha nos *imports* de bibliotecas.

Projeto Poligono

- Tentativa direta de testagem em classe abstrata.
- Tentativa de utilização de métodos sem objetos necessários.
- Incapacidade de gerar testes em métodos com unidades *float*, levando a erros de arredondamento.

JACOCO: GEMINI

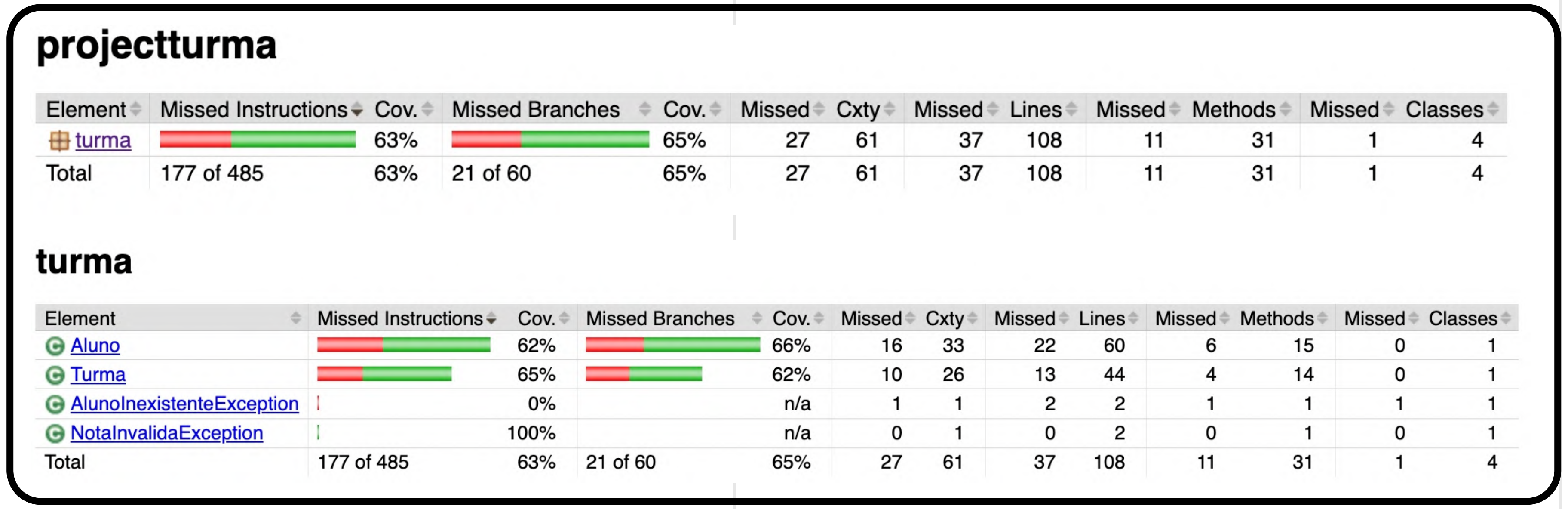


projectturma

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
poligono	🟥🟩	58%	🟥🟩	55%	30	59	48	112	18	40	1	6
Total	281 of 683	58%	17 of 38	55%	30	59	48	112	18	40	1	6

poligono

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
Ponto	🟥🟩	44%	🟥🟩	37%	17	25	14	26	9	17	0	1
Poligono	🟥🟩	63%	🟥🟩	65%	8	18	15	36	4	8	0	1
Retangulo	🟥	0%		n/a	4	4	17	17	4	4	1	1
PoligonoConvexo	🟩	94%	🟩	100%	1	7	2	15	1	6	0	1
Triangulo	🟩	100%		n/a	0	4	0	17	0	4	0	1
NaoConvexoException	🟩	100%		n/a	0	1	0	1	0	1	0	1
Total	281 of 683	58%	17 of 38	55%	30	59	48	112	18	40	1	6



projectturma

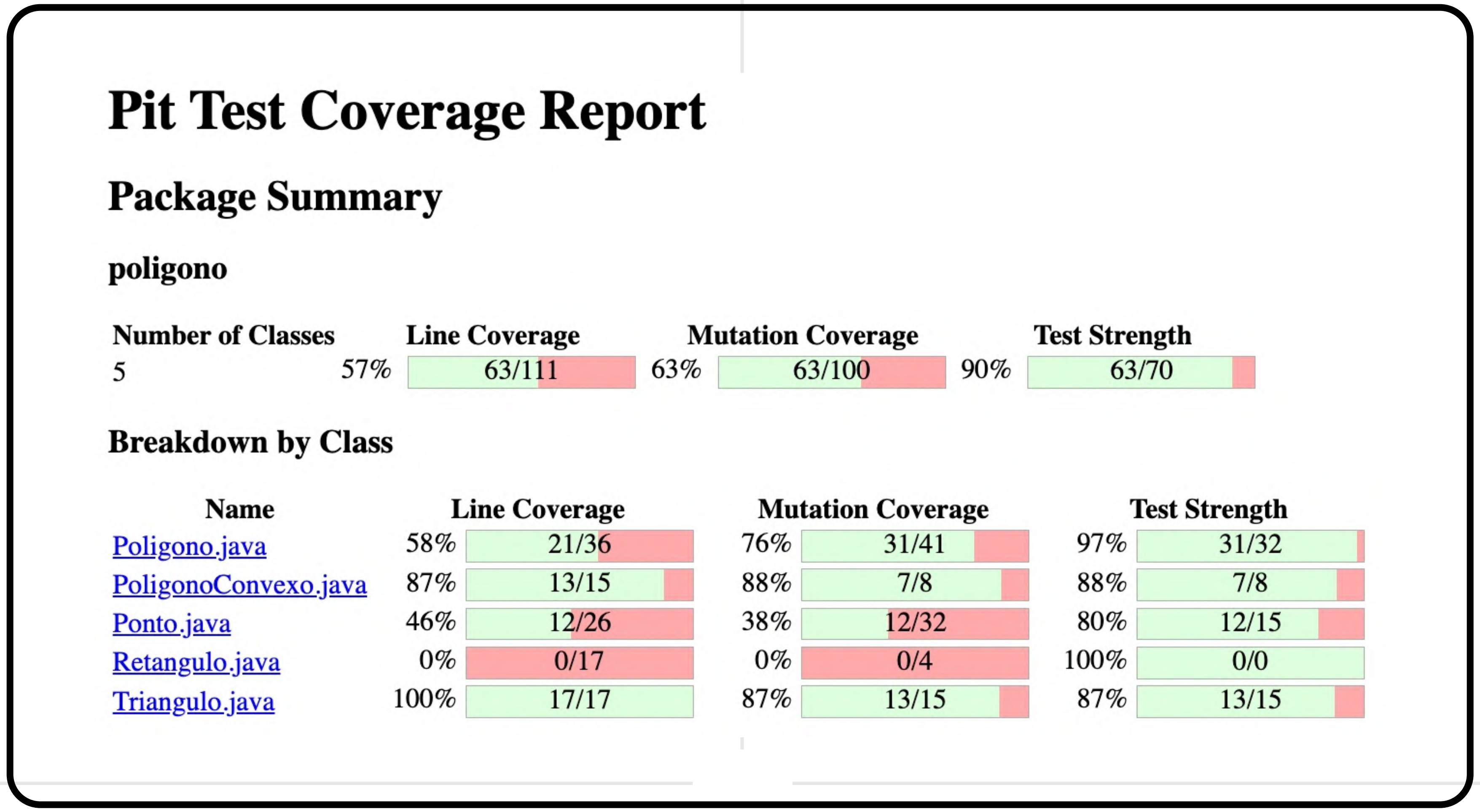
Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
turma	🟥🟩	63%	🟥🟩	65%	27	61	37	108	11	31	1	4
Total	177 of 485	63%	21 of 60	65%	27	61	37	108	11	31	1	4

turma

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
Aluno	🟥🟩	62%	🟥🟩	66%	16	33	22	60	6	15	0	1
Turma	🟥🟩	65%	🟥🟩	62%	10	26	13	44	4	14	0	1
AlunoInexistenteException	🟥	0%		n/a	1	1	2	2	1	1	1	1
NotaInvalidaException	🟩	100%		n/a	0	1	0	2	0	1	0	1
Total	177 of 485	63%	21 of 60	65%	27	61	37	108	11	31	1	4

Testes quantitativos

- Os testes das classes "Poligno" e "Turma" revelam padrões diferentes de cobertura de instruções e branches ausentes.
- A classe "Turma" demonstra uma consistência maior em termos de cobertura de testes.
- Por outro lado, a classe "Poligno" exibe uma variação mais significativa na cobertura de testes.



Pit Test Coverage Report

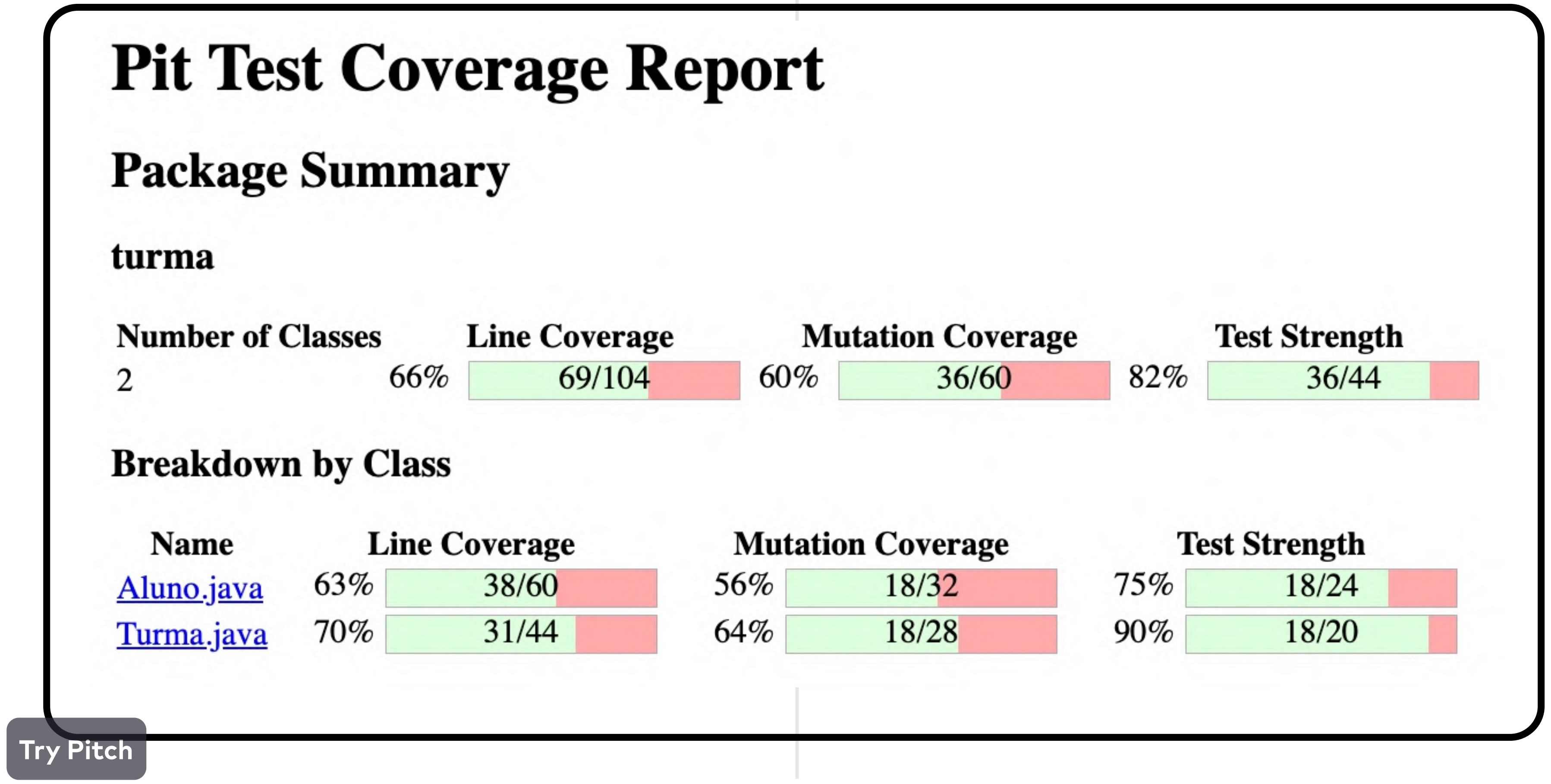
Package Summary

poligono

Number of Classes	Line Coverage	Mutation Coverage	Test Strength
5	57% 63/111	63% 63/100	90% 63/70

Breakdown by Class

Name	Line Coverage	Mutation Coverage	Test Strength
Poligono.java	58% 21/36	76% 31/41	97% 31/32
PoligonoConvexo.java	87% 13/15	88% 7/8	88% 7/8
Ponto.java	46% 12/26	38% 12/32	80% 12/15
Retangulo.java	0% 0/17	0% 0/4	100% 0/0
Triangulo.java	100% 17/17	87% 13/15	87% 13/15



Pit Test Coverage Report

Package Summary

turma

Number of Classes	Line Coverage	Mutation Coverage	Test Strength
2	66% 69/104	60% 36/60	82% 36/44

Breakdown by Class

Name	Line Coverage	Mutation Coverage	Test Strength
Aluno.java	63% 38/60	56% 18/32	75% 18/24
Turma.java	70% 31/44	64% 18/28	90% 18/20

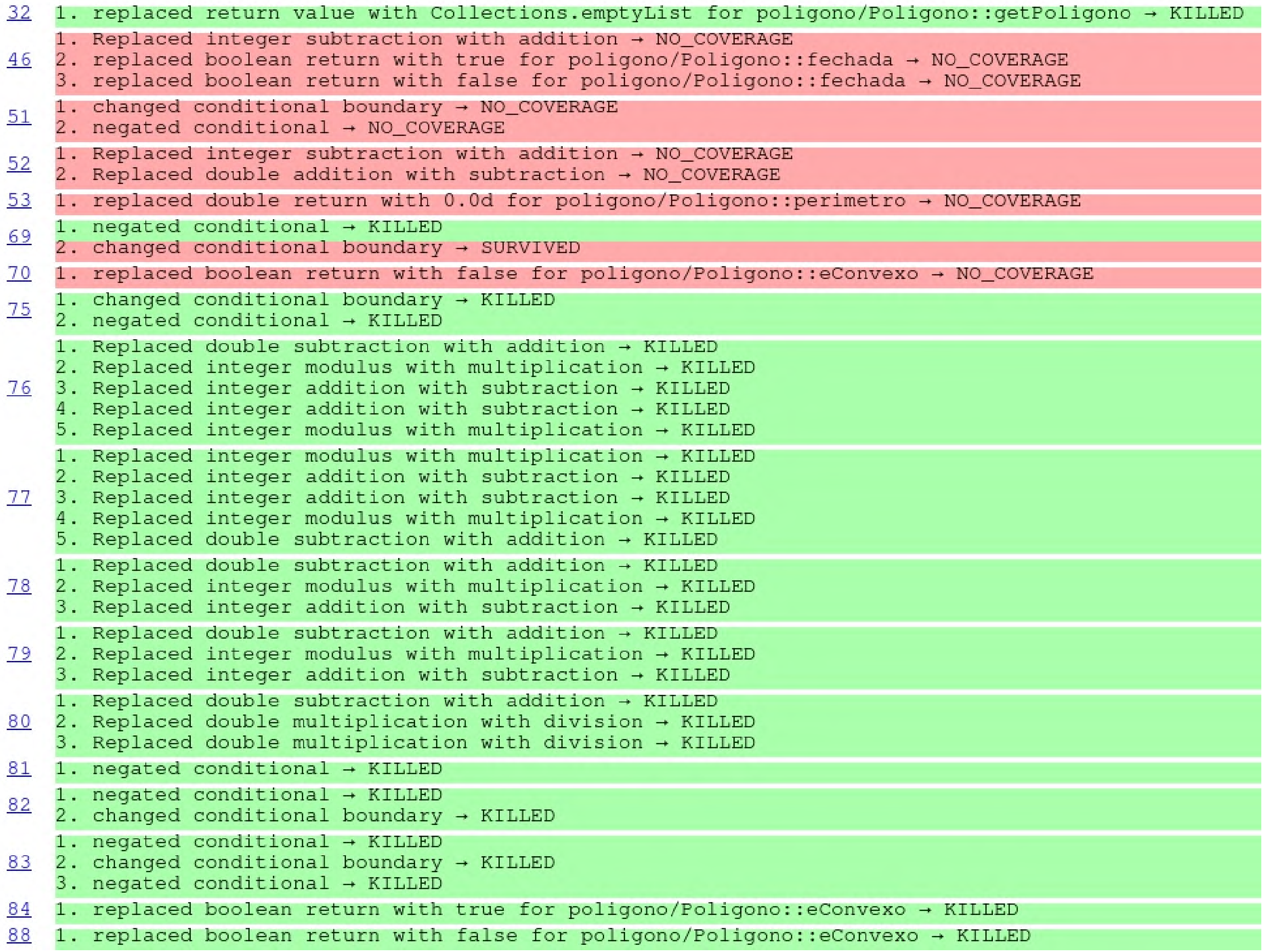
PITEST: GEMINI

Testes qualitativos

- Cobertura de testes razoável depois de correções.
- Cobertura de mutações mediocre, influenciando resultados em ambos os projetos.

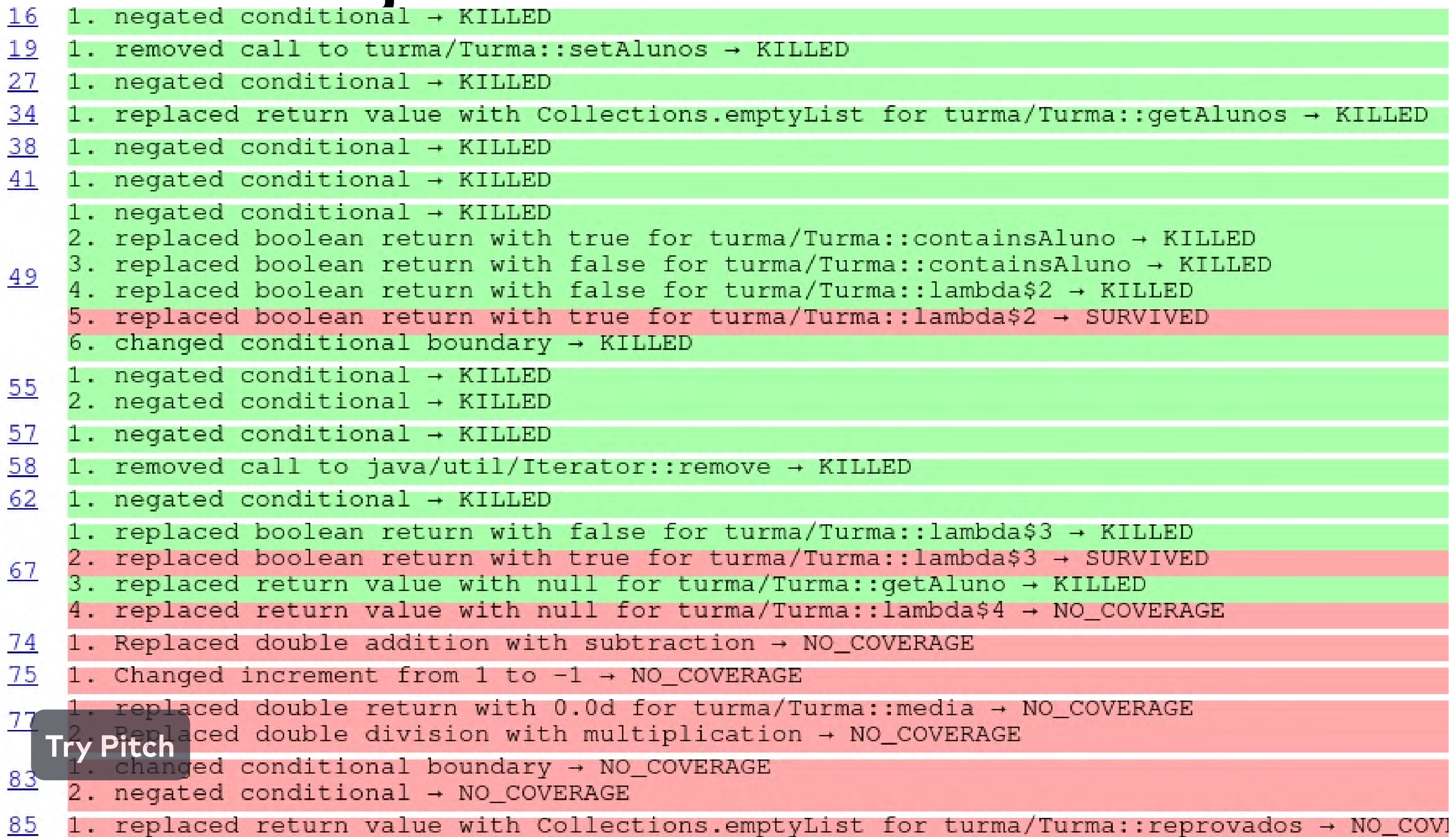
ANALISE DETALHADA PITEST: GEMINI

Poligono.java



```
32  1. replaced return value with Collections.emptyList for poligono/Poligono::getPoligono → KILLED
    1. Replaced integer subtraction with addition → NO_COVERAGE
46  2. replaced boolean return with true for poligono/Poligono::fechada → NO_COVERAGE
    3. replaced boolean return with false for poligono/Poligono::fechada → NO_COVERAGE
51  1. changed conditional boundary → NO_COVERAGE
    2. negated conditional → NO_COVERAGE
52  1. Replaced integer subtraction with addition → NO_COVERAGE
    2. Replaced double addition with subtraction → NO_COVERAGE
53  1. replaced double return with 0.0d for poligono/Poligono::perimetro → NO_COVERAGE
69  1. negated conditional → KILLED
    2. changed conditional boundary → SURVIVED
70  1. replaced boolean return with false for poligono/Poligono::eConvexo → NO_COVERAGE
75  1. changed conditional boundary → KILLED
    2. negated conditional → KILLED
    1. Replaced double subtraction with addition → KILLED
    2. Replaced integer modulus with multiplication → KILLED
76  3. Replaced integer addition with subtraction → KILLED
    4. Replaced integer addition with subtraction → KILLED
    5. Replaced integer modulus with multiplication → KILLED
    1. Replaced integer modulus with multiplication → KILLED
    2. Replaced integer addition with subtraction → KILLED
77  3. Replaced integer addition with subtraction → KILLED
    4. Replaced integer modulus with multiplication → KILLED
    5. Replaced double subtraction with addition → KILLED
    1. Replaced double subtraction with addition → KILLED
78  2. Replaced integer modulus with multiplication → KILLED
    3. Replaced integer addition with subtraction → KILLED
    1. Replaced double subtraction with addition → KILLED
79  2. Replaced integer modulus with multiplication → KILLED
    3. Replaced integer addition with subtraction → KILLED
    1. Replaced double subtraction with addition → KILLED
80  2. Replaced double multiplication with division → KILLED
    3. Replaced double multiplication with division → KILLED
81  1. negated conditional → KILLED
82  1. negated conditional → KILLED
    2. changed conditional boundary → KILLED
    1. negated conditional → KILLED
83  2. changed conditional boundary → KILLED
    3. negated conditional → KILLED
84  1. replaced boolean return with true for poligono/Poligono::eConvexo → KILLED
88  1. replaced boolean return with false for poligono/Poligono::eConvexo → KILLED
```

Turma.java



```
16  1. negated conditional → KILLED
19  1. removed call to turma/Turma::setAlunos → KILLED
27  1. negated conditional → KILLED
34  1. replaced return value with Collections.emptyList for turma/Turma::getAlunos → KILLED
38  1. negated conditional → KILLED
41  1. negated conditional → KILLED
    1. negated conditional → KILLED
    2. replaced boolean return with true for turma/Turma::containsAluno → KILLED
    3. replaced boolean return with false for turma/Turma::containsAluno → KILLED
49  4. replaced boolean return with false for turma/Turma::lambda$2 → KILLED
    5. replaced boolean return with true for turma/Turma::lambda$2 → SURVIVED
    6. changed conditional boundary → KILLED
55  1. negated conditional → KILLED
    2. negated conditional → KILLED
57  1. negated conditional → KILLED
58  1. removed call to java/util/Iterator::remove → KILLED
62  1. negated conditional → KILLED
    1. replaced boolean return with false for turma/Turma::lambda$3 → KILLED
67  2. replaced boolean return with true for turma/Turma::lambda$3 → SURVIVED
    3. replaced return value with null for turma/Turma::getAluno → KILLED
    4. replaced return value with null for turma/Turma::lambda$4 → NO_COVERAGE
74  1. Replaced double addition with subtraction → NO_COVERAGE
75  1. Changed increment from 1 to -1 → NO_COVERAGE
77  1. replaced double return with 0.0d for turma/Turma::media → NO_COVERAGE
    2. Replaced double division with multiplication → NO_COVERAGE
83  1. changed conditional boundary → NO_COVERAGE
    2. negated conditional → NO_COVERAGE
85  1. replaced return value with Collections.emptyList for turma/Turma::reprovados → NO_COVI
```

- Notável falha na prevenção de alterações diretas em *returns* de booleanos e *doubles*.
- Falhas na prevenção de mutações à alteração da lógica de certos métodos, construtores.

CONCLUSOES: GEMINI

- Falta de conhecimentos na área de programação orientada a objetos;
- Falta de conhecimento na ferramenta de testes unitários *Junit*, destacando a não utilização de @beforeeach;
- Recusou diversas vezes a geração de testes comparada ao *GPT*;
- Testes revelaram-se ligeiramente superiores ao *GPT* em termos de cobertura e qualidade;
- Através da utilização desta ferramenta também foi de realçar o nível precário que a I.A. ainda se encontra;

RAZAO DA SUPERIORIDADE DO EVOSUITE

- Cobertura de testes superior, devido à especialização da ferramenta.
- Comparativamente, o *GPT*, demonstrou uma razoável capacidade de criação de testes mas pior cobertura.
- O Gemini, criou testes com mais falhas no seu código, mas uma cobertura razoável.

CONCLUSAO

Ao concentrar-se na produção de testes unitários em JUnit, foram notadas diferenças significativas entre as diversas ferramentas disponíveis.

O EvoSuite destacou-se quantitativa e qualitativamente.

O GPT mostrou superioridade sobre o Gemini, mas explorar diferentes configurações do EvoSuite e versões do GPT/Gemini poderia ter enriquecido a análise.

ChatGPT
Gemini
2
1
3