

**Engenharia de Software - Trabalho Prático (30% nota)**

**2º ano – Tecnologias de Informação**

**Relatório**

Gonçalo Farias -109180

Índice

**Introdução1**

Objetivo do Teste

**Estratégias de Teste2**

Testes Unitários

Estratégias Abordadas

**Justificação de Casos de Teste3**

**Conclusão4**

1. **Introdução**

O objetivo deste relatório é descrever as estratégias de teste adotadas e justificar os casos de teste implementados para o método ‘run’ da classe ‘PimPamPum’. O método ‘run’ é responsável por receber um número inteiro como parâmetro e retornar uma string de acordo com as seguintes regras: substituir números divisíveis por 3 pela palavra “Pim”, números divisíveis por 5 pela palavra 2Pam” e números divisíveis por 7 pela palavra “Pum”. Além disso, os dígitos 3, 5 e 7 também são substituídos pelos mesmos valores correspondentes.

* **Objetivo do Teste**

O objetivo principal dos testes feitos por mim é assegurar que a implementação do método “run” na classe “PimPamPum” esteja correta e atenda sempre aos requisitos que foram estabelecidos no âmbito do projeto.

1. **Estratégias de Teste**

* **Testes Unitários**

A estratégia de utilizar os testes unitários, teve como base o facto de pretender garantir que cada unidade do código fosse testada de forma isolada e individual, facilitando a identificação de falhas em certas partes do código.

* **Estratégias Abordadas**

**1.** Testar se o método retorna a palavra “Pim” para números divisíveis por 3;

**2.** Testar se o método retorna a palavra “Pam” para números divisíveis por 5;

**3.** Testar se o método retorna a palavra “Pum” para números divisíveis por 7;

**4.** Testar se o método retorna corretamente as palavras “Pim” e “Pam” para números que contenham os dígitos 3 e 5, respetivamente;

**5.** Testar se o método retorna corretamente as palavras “Pim”, “Pam” e “Pum” para números que contenham os dígitos 3, 5 e 7, respetivamente;

**6.** Testar se o método retorna o próprio número quando ele não é divisível por nenhum dos valores (3, 5 ou 7) e não contém os dígitos 3, 5 ou 7.

1. **Justificação de Casos de Teste**

Foram desenvolvidos casos de teste específicos para garantir uma cobertura abrangente, incluindo:

1. **testRun\_DivisivelPor3:** Este teste tem como objetivo verificar se o método retorna corretamente a palavra “Pim” para números divisíveis por 3.
2. **testRun\_DivisivelPor5:** O objetivo deste teste é verificar se o método retorna corretamente a palavra “Pam” para números divisíveis por 5.
3. **testRun\_DivisivelPor7:** O objetivo deste teste é verificar se o método retorna corretamente a palavra “Pum” para números divisíveis por 7.
4. **testRun\_Conter3:** O objetivo deste teste é verificar se o método retorna corretamente a palavra “Pim” para números que contêm o dígito 3.
5. **testRun\_Conter5:** O objetivo deste teste é verificar se o método retorna corretamente a palavra “Pam” para números que contêm o dígito 5.
6. **testRun\_Conter7:** O objetivo deste teste é verificar se o método retorna corretamente a palavra “Pum” para números que contêm o dígito 7.
7. **testRun\_DivisivelPor3Conter3:** O objetivo deste teste é verificar se o método retorna corretamente a palavra "Pim" para números que são divisíveis por 3 e também contêm o dígito 3.
8. **testRun\_DivisivelPor3Por5Conter5:** O objetivo deste teste é verificar se o método retorna corretamente as palavras “Pim” e “Pam” para números que são divisíveis por 3 e por 5, e também contêm o dígito 5.
9. **testRun\_DivisivelPor3Por7:** O objetivo deste teste é verificar se o método retorna corretamente as palavras “Pim” e “Pum” para números que são divisíveis por 3 e por 7.
10. **testRun\_Conter3Conter5:** O objetivo deste teste é verificar se o método retorna corretamente as palavras “Pam” e "Pim" para números que contêm os dígitos 3 e 5, respetivamente.
11. **testRun\_NaoDivisivelPor3Por5:** O objetivo deste teste é verificar que o método não retorna “Pim” nem “Pam” para números não divisíveis por 3 e 5.
12. **testRun\_DivisivelPor3Conter3Conter5Conter7:** O objetivo deste teste é verificar se o método retorna corretamente as palavras “Pim”, “Pum”, “Pam” e “Pim” para números que são divisíveis por 3, e também contêm os dígitos 3, 5 e 7.
13. **testRun\_NaoDivisivel:** O objetivo deste teste é verificar se o método retorna o próprio número quando ele não é divisível por nenhum dos valores (3, 5 ou 7) e não contém os dígitos 3, 5 ou 7. Isso é importante para garantir que o método esteja tratando corretamente casos em que nenhuma das regras se aplica.
14. **testRun\_NaoConterDigitosIguais:** O objetivo deste teste é verificar se o método retorna o próprio número quando ele não contém nenhum dos dígitos 3, 5 ou 7. Isso é importante para garantir que o método esteja tratando corretamente casos em que não há necessidade de substituição por palavras.
15. **testRun\_NumeroNegativo:** O objetivo deste teste é verificar se o método lança uma exceção ‘IllegalArgumentException’ quando o número passado como parâmetro é negativo. Isso é importante para garantir que o método esteja tratando adequadamente casos inválidos.
16. **testRun\_NumeroZero:** O objetivo deste teste é verificar se o método retorna corretamente as palavras “Pim”, “Pam” e “Pum” para o número zero. Isso é importante para garantir que o método esteja tratando adequadamente casos especiais, como zero.
17. **Conclusão**

As estratégias de teste adotadas e justificativas dos casos de teste implementados visam garantir que o método ‘run’ da classe ‘PimPamPum’ esteja funcionando corretamente de acordo com as especificações do problema. Os casos de teste foram selecionados com base nas diferentes regras definidas no enunciado do problema e em possíveis cenários alternativos. Ao realizar os testes, foi possível garantir que todas as regras estão sendo aplicadas corretamente e que o método trata adequadamente possíveis casos excecionais.