

UNIVERSIDADE FEDERAL DE SERGIPE

VINÍCIUS DIAS VALENÇA

Teste DE SOFTWARE

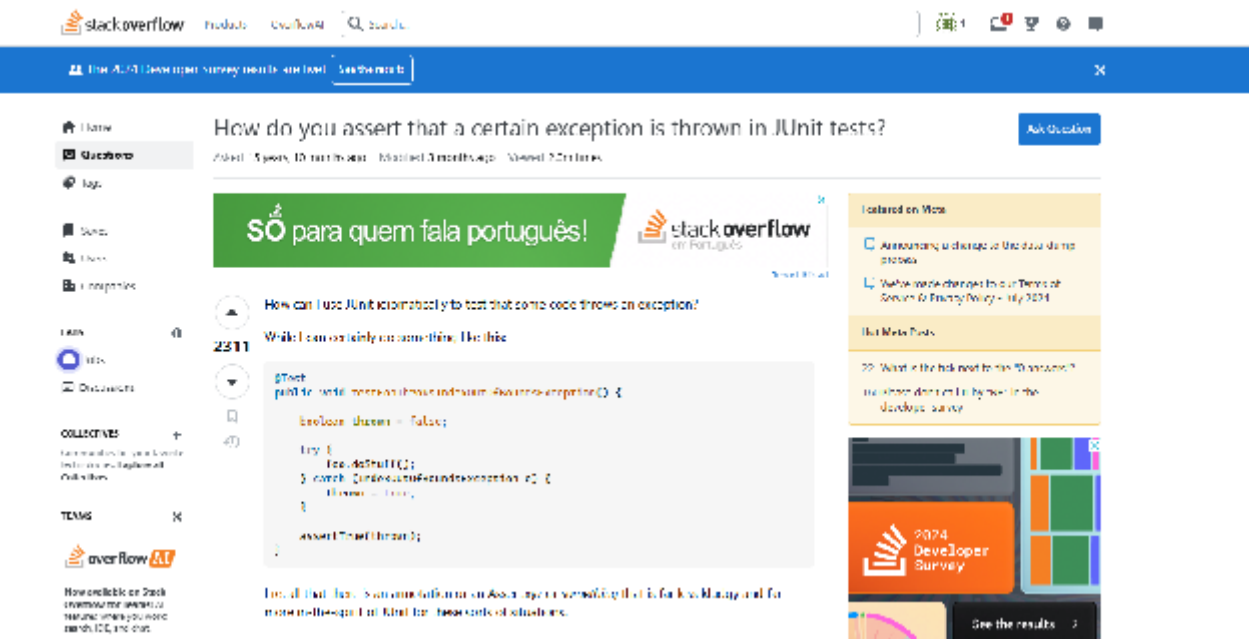
ATIVIDADE 1

Problema do Stack Overflow

1. Motivação

How do you assert that a certain exception is thrown in JUnit tests?

<https://stackoverflow.com/questions/156503/how-do-you-assert-that-a-certain-exception-is-thrown-in-junit-tests>



```
@Test
public void testFooThrowsIndexOutOfBoundsException() {

    boolean thrown = false;

    try {
        foo.doStuff();
    } catch (IndexOutOfBoundsException e) {
        thrown = true;
    }

    assertTrue(thrown);
}
```

Ao escrever testes unitários, um cenário comum é garantir que uma exceção específica seja lançada quando certas condições são atendidas. Por exemplo, se um método deve lançar uma exceção quando recebe parâmetros inválidos, queremos testar e garantir esse comportamento.

Problemas com a Abordagem Inicial

1. A necessidade de um bloco **try-catch** e a variável **thrown** adicionam código desnecessário, tornando o teste mais verboso e difícil de ler.
2. O teste depende da variável **thrown** ser corretamente definida no bloco **catch**. Qualquer alteração acidental pode fazer o teste falhar silenciosamente.
3. Se o teste falhar, a mensagem de erro padrão pode não ser muito clara sobre o que deu errado.

2. Reproduzindo o problema

```
4
5
6 package com.example;
7 import java.util.List;
8
9 public class ClassToTest {
10     public void checkNames(List<String> names) {
11         for (String name : names) {
12             if (name.length() <= 2) {
13                 throw new IllegalArgumentException("O nome precisa ter pelo menos 3 letras: " + name);
14             }
15         }
16     }
17 }
```

```
1 package com.example;
2 import org.junit.jupiter.api.Test;
3
4 import java.util.Arrays;
5 import java.util.List;
6
7 import static org.junit.jupiter.api.Assertions.assertEquals;
8 import static org.junit.jupiter.api.Assertions.assertThrows;
9
10 public class TestClass {
11
12     @Test
13     public void testCheckNamesThrowsExceptionForShortNames() {
14         ClassToTest foo = new ClassToTest();
15         List<String> names = Arrays.asList("Joe", "Al", "Sam");
16
17         IllegalArgumentException exception = assertThrows(IllegalArgumentException.class, () -> foo.checkNames(names));
18         assertEquals("O nome precisa ter pelo menos 3 letras: Al", exception.getMessage());
19     }
20
21     @Test
22     public void testCheckNamesDoesNotThrowExceptionForValidNames() {
23         ClassToTest foo = new ClassToTest();
24         List<String> names = Arrays.asList("Joe", "Sam", "Ann");
25
26         foo.checkNames(names);
27     }
28 }
```

testCheckNamesThrowsExceptionForShortNames:

Cria uma instância de **ClassToTest** e uma lista de nomes onde um dos nomes é muito curto.

Usa **assertThrows** para verificar que **IllegalArgumentException** é lançada.

Verifica se a mensagem da exceção corresponde à esperada.

testCheckNamesDoesNotThrowExceptionForValidNames:

Cria uma instância de **ClassToTest** e uma lista de nomes válidos.

Chama **checkNames** e verifica implicitamente que nenhuma exceção é lançada.

```

1 <project xmlns="http://maven.apache.org/POM/4.0.0"
2       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3       xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
4   <modelVersion>4.0.0</modelVersion>
5
6   <groupId>com.example</groupId>
7   <artifactId>atividade1-project</artifactId>
8   <version>1.0-SNAPSHOT</version>
9
10  <dependencies>
11    <dependency>+
12      <groupId>org.junit.jupiter</groupId>
13      <artifactId>junit-jupiter-api</artifactId>
14      <version>5.8.2</version>
15      <scope>test</scope>
16    </dependency>
17    <dependency>
18      <groupId>org.junit.jupiter</groupId>
19      <artifactId>junit-jupiter-engine</artifactId>
20      <version>5.8.2</version>
21      <scope>test</scope>
22    </dependency>
23  </dependencies>
24
25  <build>
26    <plugins>
27      <plugin>
28        <groupId>org.apache.maven.plugins</groupId>
29        <artifactId>maven-surefire-plugin</artifactId>
30        <version>3.0.0-M5</version>
31      </plugin>
32    </plugins>
33  </build>
34 </project>

```

3. Outras soluções:



in junit, there are four ways to test exception.

248

junit5.x



- for junit5.x, you can use `assertThrows` as following

```

@Test
public void testFooThrowsIndexOutOfBoundsException() {
    Throwable exception = assertThrows(IndexOutOfBoundsException.class, () -> foo);
    assertEquals("expected messages", exception.getMessage());
}

```

Utilizar uma exceção específica, como `IllegalArgumentException`, comunica claramente a intenção do código e o tipo de erro que está sendo tratado. Isso torna o código mais legível e fácil de entender para outros desenvolvedores que possam trabalhar nele no futuro.

`Throwable` é a superclasse de todas as exceções e erros em Java. Utilizá-la para capturar exceções pode levar a capturar exceções não intencionais, como `NullPointerException`, `IOException`, ou mesmo `OutOfMemoryError`, que são casos que normalmente não deveriam ser tratados da mesma forma.

ATIVIDADE 2 - Aplicação de Testes de Mutação em Python

Etapa 1: Acessando o Repositório

- Acesse o repositório do exemplo disponível no GitHub. (<https://github.com/alura-cursos/2622-python-tdd>)
- **Clone o repositório:**
 - <https://github.com/alura-cursos/2622-python-tdd.git>

Etapa 2: Preparando o Ambiente de Desenvolvimento

- Descompacte o conteúdo do repositório em uma pasta local.
- Abra o editor de código de sua preferência. No vídeo, é utilizado o **Visual Studio Code** (VS Code).
 - **Nota:** Para utilizar o VS Code com Python, algumas extensões específicas devem estar instaladas.

Etapa 3: Configuração do Ambiente Virtual

- Instale o **Python 3.7** (ou superior) e o **virtualenv**.
- Crie um ambiente virtual para isolar as bibliotecas necessárias.
 - Comando para criar o ambiente virtual:

```
python3 -m venv venv
```

- Ative o ambiente virtual:

```
source venv/bin/activate # Linux/Mac  
venv\Scripts\activate # Windows
```

Etapa 4: Instalando as Dependências

- Instale as bibliotecas necessárias listadas no arquivo **requirements.txt**:

```
pip install -r requirements.txt
```

- As principais bibliotecas são:
 - **pytest**: Framework para testes.
 - **pytest-cov**: Plugin do pytest para medir a cobertura de código.
 - **mutmut**: Ferramenta para aplicar teste de mutação.

Etapa 5: Executando Testes e Medindo Cobertura

- Execute os casos de teste utilizando o pytest:

```
pytest -vv [caminho_para_o_arquivo]
```

- Verifique a cobertura de código:

```
pytest -vv [caminho_para_o_arquivo] --cov=cal
```

- Gere um relatório HTML de cobertura:

```
pytest -vv [caminho_para_o_arquivo] --cov=cal --cov-report=html
```

Etapa 6: Classe Funcionário

O código define uma classe **Funcionario** que representa um funcionário com alguns atributos e métodos para manipular e acessar suas informações.

Atributos

- **nome**: Nome do funcionário.
- **data_nascimento**: Data de nascimento do funcionário no formato 'DD/MM/AAAA'.
- **salario**: Salário do funcionário.

Métodos

- **idade**: Calcula a idade do funcionário com base no ano de nascimento.
- **sobrenome**: Retorna o sobrenome do funcionário, assumindo que é a última palavra no nome completo.
- **_eh_socio**: Método privado que verifica se o funcionário é considerado um "sócio" com base no salário e sobrenome.
- **decrescimo_salario**: Aplica um desconto de 10% no salário se o funcionário for um "sócio".
- **calcular_bonus**: Calcula um bônus de 10% sobre o salário. Se o valor do bônus for maior que 1000, lança uma exceção.

Etapa 7: Resumo do Código de Testes

O código fornece uma série de testes unitários para a classe **Funcionario**, utilizando o framework de testes **pytest**. Os testes verificam o comportamento dos métodos da classe **Funcionario** em diferentes cenários.

1. Teste de Idade

- **Descrição**: Verifica se o método **idade** retorna o valor correto para uma data de nascimento específica.
- **Entrada**: '13/03/2000'
- **Esperado**: 24 (nota: o valor original no código é 22, indicando um erro)
- **Verificação**: Compara o resultado da idade com o valor esperado.

2. Teste de Sobrenome

- **Descrição**: Verifica se o método **sobrenome** retorna o sobrenome correto a partir de um nome completo.
- **Entrada**: ' Lucas Carvalho '

- **Esperado:** 'Carvalho'
- **Verificação:** Compara o sobrenome obtido com o valor esperado.

3. Teste de Desconto no Salário

- **Descrição:** Verifica se o método `decrescimo_salario` aplica corretamente um desconto no salário se o funcionário for um "sócio".
- **Entrada:** Salário de 100000 e nome 'Paulo Bragança'
- **Esperado:** 90000
- **Verificação:** Compara o salário após o desconto com o valor esperado.

4. Teste de Cálculo de Bônus

- **Descrição:** Verifica se o método `calcular_bonus` calcula corretamente o bônus para um salário dentro do limite.
- **Entrada:** Salário de 1000
- **Esperado:** 100
- **Verificação:** Compara o bônus calculado com o valor esperado.

5. Teste de Exceção no Cálculo de Bônus

- **Descrição:** Verifica se o método `calcular_bonus` lança uma exceção quando o salário é muito alto.
- **Entrada:** Salário de 1000000000
- **Esperado:** Lançar uma exceção
- **Verificação:** Certifica-se de que uma exceção é lançada.

Etapa 8: Executando o Teste de Mutação

- Utilize o **Mutmut** para aplicar o teste de mutação e verificar a robustez dos casos de teste:

```
mutmut run --paths-to-mutate [caminho_para_o_arquivo]
```

- O Mutmut criará várias versões mutantes do código original e executará os testes contra essas versões.

Etapa 9: Analisando os Resultados do Teste de Mutação

- Ao final da execução, o Mutmut fornecerá um resumo dos mutantes que foram "mortos" e aqueles que sobreviveram.
- Utilize comandos do Mutmut para obter mais detalhes sobre os mutantes que sobreviveram:

```
mutmut results
```

- **Dica:** Utilize para gerar um arquivo html contendo todos os mutantes sobreviventes. Dessa forma, é possível analisá-los de forma mais detalhada.

```
mutmut html
```

Resultados

- Foram adicionados alguns testes no conjunto de testes já existentes, aumentando a detecção dos mutantes de 22 para 34.
- **Conclusão**
 - Com a utilização do mutmut repetidamente é possível reduzir drasticamente a quantidade de vulnerabilidade nos testes envolvidos no problema em questão.

- Mutações com o código original

codigo/bytebank.py

Killed 22 out of 42 mutants

Survived

Survived mutation testing. These mutants show holes in your test suite.

Mutant 5

```
--- codigo/bytebank.py
+++ codigo/bytebank.py
@@ -10,7 +10,6 @@
     def nome(self):
         return self._nome

-    @property
     def salario(self):
         return self._salario
```

Mutant 16

```
--- codigo/bytebank.py
+++ codigo/bytebank.py
@@ -23,7 +23,7 @@
     def sobrenome(self):
         nome_completo = self.nome.strip()
         nome_quebrado = nome_completo.split(' ')
-        return nome_quebrado[-1]
+        return nome_quebrado[+1]
```

def eh_socio(self):

Mutant 18

```
--- codigo/bytebank.py
+++ codigo/bytebank.py
@@ -27,7 +27,7 @@

     def eh_socio(self):
         sobrenomes = ['Braganca', 'Windsor', 'Bourbon', 'Yamato', 'Al Saud', 'Khan', 'Tudor', 'Ptolomeu']
+        sobrenomes = ['XXBragancaXX', 'Windsor', 'Bourbon', 'Yamato', 'Al Saud', 'Khan', 'Tudor', 'Ptolomeu']
         return (self._salario >= 100000) and (self.sobrenome() in sobrenomes)
```

Mutant 19

```
--- codigo/bytebank.py
+++ codigo/bytebank.py
@@ -27,7 +27,7 @@

     def eh_socio(self):
         sobrenomes = ['Braganca', 'Windsor', 'Bourbon', 'Yamato', 'Al Saud', 'Khan', 'Tudor', 'Ptolomeu']
+        sobrenomes = ['Braganca', 'XXWindsorXX', 'Bourbon', 'Yamato', 'Al Saud', 'Khan', 'Tudor', 'Ptolomeu']
         return (self._salario >= 100000) and (self.sobrenome() in sobrenomes)
```

Mutant 20

```
--- codigo/bytebank.py
+++ codigo/bytebank.py
@@ -27,7 +27,7 @@
```

Mutant 27

```
--- codigo/bytebank.py
+++ codigo/bytebank.py
@@ -28,7 +28,7 @@

     def eh_socio(self):
         sobrenomes = ['Braganca', 'Windsor', 'Bourbon', 'Yamato', 'Al Saud', 'Khan', 'Tudor', 'Ptolomeu']
-        return (self._salario >= 100000) and (self.sobrenome() in sobrenomes)
+        return (self._salario >= 100000) and (self.sobrenome() in sobrenomes)
```

def decrescimo_salario(self):

Mutant 28

```
--- codigo/bytebank.py
+++ codigo/bytebank.py
@@ -28,7 +28,7 @@

     def eh_socio(self):
         sobrenomes = ['Braganca', 'Windsor', 'Bourbon', 'Yamato', 'Al Saud', 'Khan', 'Tudor', 'Ptolomeu']
-        return (self._salario >= 100000) and (self.sobrenome() in sobrenomes)
+        return (self._salario >= 100001) and (self.sobrenome() in sobrenomes)
```

def decrescimo_salario(self):

Mutant 31

```
--- codigo/bytebank.py
+++ codigo/bytebank.py
@@ -33,7 +33,7 @@

     def decrescimo_salario(self):
         if self._eh_socio():
-            decrescimo = self._salario * 0.1
+            decrescimo = self._salario / 0.1
         self._salario = self._salario - decrescimo
```

Mutant 32

```
--- codigo/bytebank.py
+++ codigo/bytebank.py
@@ -33,7 +33,7 @@

     def decrescimo_salario(self):
         if self._eh_socio():
-            decrescimo = self._salario * 0.1
+            decrescimo = self._salario * 1.1
         self._salario = self._salario - decrescimo
```

Mutant 33

```
--- codigo/bytebank.py
+++ codigo/bytebank.py
@@ -33,7 +33,7 @@

     def decrescimo_salario(self):
         if self._eh_socio():
-            decrescimo = self._salario * 0.1
```

```
def eh_socio(self):
    sobrenomes = ['Braganca', 'Windsor', 'Bourbon', 'Yamato', 'Al Saud', 'Khan', 'Tudor', 'Ptolomeu']
+    sobrenomes = ['Braganca', 'Windsor', 'XXBourbonXX', 'Yamato', 'Al Saud', 'Khan', 'Tudor', 'Ptolomeu']
    return (self._salario >= 100000) and (self.sobrenome() in sobrenomes)
```

Mutant 21

```
--- codigo/bytebank.py
+++ codigo/bytebank.py
@@ -27,7 +27,7 @@
```

```
def eh_socio(self):
    sobrenomes = ['Braganca', 'Windsor', 'Bourbon', 'Yamato', 'Al Saud', 'Khan', 'Tudor', 'Ptolomeu']
+    sobrenomes = ['Braganca', 'Windsor', 'Bourbon', 'XXYamatoXX', 'Al Saud', 'Khan', 'Tudor', 'Ptolomeu']
    return (self._salario >= 100000) and (self.sobrenome() in sobrenomes)
```

Mutant 22

```
--- codigo/bytebank.py
+++ codigo/bytebank.py
@@ -27,7 +27,7 @@
```

```
def eh_socio(self):
    sobrenomes = ['Braganca', 'Windsor', 'Bourbon', 'Yamato', 'Al Saud', 'Khan', 'Tudor', 'Ptolomeu']
+    sobrenomes = ['Braganca', 'Windsor', 'Bourbon', 'Yamato', 'XXAl SaudXX', 'Khan', 'Tudor', 'Ptolomeu']
    return (self._salario >= 100000) and (self.sobrenome() in sobrenomes)
```

Mutant 23

```
--- codigo/bytebank.py
+++ codigo/bytebank.py
@@ -27,7 +27,7 @@
```

```
def eh_socio(self):
    sobrenomes = ['Braganca', 'Windsor', 'Bourbon', 'Yamato', 'Al Saud', 'Khan', 'Tudor', 'Ptolomeu']
+    sobrenomes = ['Braganca', 'Windsor', 'Bourbon', 'Yamato', 'Al Saud', 'XXXhankX', 'Tudor', 'Ptolomeu']
    return (self._salario >= 100000) and (self.sobrenome() in sobrenomes)
```

Mutant 24

```
--- codigo/bytebank.py
+++ codigo/bytebank.py
@@ -27,7 +27,7 @@
```

```
def eh_socio(self):
    sobrenomes = ['Braganca', 'Windsor', 'Bourbon', 'Yamato', 'Al Saud', 'Khan', 'Tudor', 'Ptolomeu']
+    sobrenomes = ['Braganca', 'Windsor', 'Bourbon', 'Yamato', 'Al Saud', 'Khan', 'XXTudorXX', 'Ptolomeu']
    return (self._salario >= 100000) and (self.sobrenome() in sobrenomes)
```

Mutant 25

```
--- codigo/bytebank.py
+++ codigo/bytebank.py
@@ -27,7 +27,7 @@
```

```
+        decrescimo = None
+        self._salario = self._salario - decrescimo
```

Mutant 34

```
--- codigo/bytebank.py
+++ codigo/bytebank.py
@@ -34,7 +34,7 @@
     def decrescimo_salario(self):
         if self._eh_socio():
-            decrescimo = self._salario * 0.1
+            self._salario = self._salario - decrescimo
+            self._salario = self._salario + decrescimo
```

def calcular_bonus(self):

Mutant 35

```
--- codigo/bytebank.py
+++ codigo/bytebank.py
@@ -34,7 +34,7 @@
     def decrescimo_salario(self):
         if self._eh_socio():
-            decrescimo = self._salario * 0.1
+            self._salario = self._salario - decrescimo
+            self._salario = None
```

def calcular_bonus(self):

Mutant 39

```
--- codigo/bytebank.py
+++ codigo/bytebank.py
@@ -39,7 +39,7 @@

     def calcular_bonus(self):
         valor = self._salario * 0.1
-        if valor > 1000:
+        if valor >= 1000:
             raise Exception('O salário é muito alto para receber um bônus')
         return valor
```

Mutant 40

```
--- codigo/bytebank.py
+++ codigo/bytebank.py
@@ -39,7 +39,7 @@

     def calcular_bonus(self):
         valor = self._salario * 0.1
-        if valor > 1000:
+        if valor >= 1001:
             raise Exception('O salário é muito alto para receber um bônus')
         return valor
```

Mutant 41

```
--- codigo/bytebank.py
+++ codigo/bytebank.py
@@ -40,7 +40,7 @@

     def calcular_bonus(self):
         valor = self._salario * 0.1
-        if valor > 1000:
+        raise Exception('O salário é muito alto para receber um bônus')
+        raise Exception('XOD salário é muito alto para receber um bônusXX')
         return valor
```


- Mutações com testes adicionados

codigo/bytebank.py

Killed 34 out of 42 mutants

Survived

Survived mutation testing. These mutants show holes in your test suite.

Mutant 5

```
--- codigo/bytebank.py
+++ codigo/bytebank.py
@@ -18,7 +18,6 @@
     def nome(self):
         return self._nome

- @property
- def salario(self):
-     return self._salario
```

Mutant 19

```
--- codigo/bytebank.py
+++ codigo/bytebank.py
@@ -27,7 +27,7 @@

     def _eh_socio(self):
-         sobrenomes = ['Bragança', 'Windsor', 'Bourbon', 'Yamato', 'Al Saud', 'Khan', 'Tudor', 'Ptolomeu']
+         sobrenomes = ['Bragança', 'XXWindsorXX', 'Bourbon', 'Yamato', 'Al Saud', 'Khan', 'Tudor', 'Ptolomeu']
         return (self._salario >= 100000) and (self.sobrenome() in sobrenomes)
```

Mutant 20

```
--- codigo/bytebank.py
+++ codigo/bytebank.py
@@ -27,7 +27,7 @@

     def _eh_socio(self):
-         sobrenomes = ['Bragança', 'Windsor', 'Bourbon', 'Yamato', 'Al Saud', 'Khan', 'Tudor', 'Ptolomeu']
+         sobrenomes = ['Bragança', 'Windsor', 'XXBourbonXX', 'Yamato', 'Al Saud', 'Khan', 'Tudor', 'Ptolomeu']
         return (self._salario >= 100000) and (self.sobrenome() in sobrenomes)
```

Mutant 21

```
--- codigo/bytebank.py
+++ codigo/bytebank.py
@@ -27,7 +27,7 @@

     def _eh_socio(self):
-         sobrenomes = ['Bragança', 'Windsor', 'Bourbon', 'Yamato', 'Al Saud', 'Khan', 'Tudor', 'Ptolomeu']
+         sobrenomes = ['Bragança', 'Windsor', 'Bourbon', 'XXYamatoXX', 'Al Saud', 'Khan', 'Tudor', 'Ptolomeu']
         return (self._salario >= 100000) and (self.sobrenome() in sobrenomes)
```

Mutant 22

```
--- codigo/bytebank.py
+++ codigo/bytebank.py
@@ -27,7 +27,7 @@
```

```
def _eh_socio(self):
-     sobrenomes = ['Bragança', 'Windsor', 'Bourbon', 'Yamato', 'Al Saud', 'Khan', 'Tudor', 'Ptolomeu']
+     sobrenomes = ['Bragança', 'Windsor', 'Bourbon', 'Yamato', 'XXAl SaudXX', 'Khan', 'Tudor', 'Ptolomeu']
     return (self._salario >= 100000) and (self.sobrenome() in sobrenomes)
```

Mutant 23

```
--- codigo/bytebank.py
+++ codigo/bytebank.py
@@ -27,7 +27,7 @@

     def _eh_socio(self):
-         sobrenomes = ['Bragança', 'Windsor', 'Bourbon', 'Yamato', 'Al Saud', 'Khan', 'Tudor', 'Ptolomeu']
+         sobrenomes = ['Bragança', 'Windsor', 'Bourbon', 'Yamato', 'Al Saud', 'XXXhanXX', 'Tudor', 'Ptolomeu']
         return (self._salario >= 100000) and (self.sobrenome() in sobrenomes)
```

Mutant 24

```
--- codigo/bytebank.py
+++ codigo/bytebank.py
@@ -27,7 +27,7 @@

     def _eh_socio(self):
-         sobrenomes = ['Bragança', 'Windsor', 'Bourbon', 'Yamato', 'Al Saud', 'Khan', 'Tudor', 'Ptolomeu']
+         sobrenomes = ['Bragança', 'Windsor', 'Bourbon', 'Yamato', 'Al Saud', 'Khan', 'XXTudorXX', 'Ptolomeu']
         return (self._salario >= 100000) and (self.sobrenome() in sobrenomes)
```

Mutant 25

```
--- codigo/bytebank.py
+++ codigo/bytebank.py
@@ -27,7 +27,7 @@

     def _eh_socio(self):
-         sobrenomes = ['Bragança', 'Windsor', 'Bourbon', 'Yamato', 'Al Saud', 'Khan', 'Tudor', 'Ptolomeu']
+         sobrenomes = ['Bragança', 'Windsor', 'Bourbon', 'Yamato', 'Al Saud', 'Khan', 'Tudor', 'XXPtolomeuXX']
         return (self._salario >= 100000) and (self.sobrenome() in sobrenomes)
```