

Universidade do Estado do Amazonas  
Escola Superior de Tecnologia

**Trabalho Prático 1:**  
**Roteiros de Casos de Teste**

Manaus, AM  
15 de abril de 2022

**Yasmin Maria Muniz de Oliveira**

**Trabalho Prático 1:  
Roteiros de Casos de Teste**

Trabalho prático apresentado à  
disciplina Tópicos Especiais em  
Engenharia de Software como  
requisito para obtenção de nota na  
Avaliação Parcial 1.

Manaus, AM  
15 de abril de 2022

## 1. Considerações iniciais

### 1.1. Método avaliado

O método escolhido para ser testado neste trabalho faz parte de um programa em Python 3 cujo objetivo é verificar se um e-mail é válido (ham) ou inválido (spam), usando expressões regulares. A função desse método é analisar a mensagem presente no corpo do e-mail. Para isso, recebe como entrada uma string contendo o nome do arquivo .txt em que a mensagem está e retorna “ham” caso a mensagem seja genuína ou “spam” caso contrário.

Uma mensagem é considerada “spam” se atender a algum ou alguns dos seguintes critérios:

- Possuir tags específicas HTML (head, body, img, alt, href) quer sejam de abertura ou fechamento, que podem dar indícios de vínculo a conteúdo externo malicioso;
- Possuir palavras suspeitas específicas (milionário, empréstimo, loteria, banco, herança, seguidor, desconto), sejam maiúsculas ou minúsculas, com os devidos acentos ou não;
- Possuir uma quantidade de consoantes que equivale a mais do que o dobro da quantidade de vogais;
- Possuir mais de 15 sinais de pontuação (ponto final, vírgula, ponto-e-vírgula, interrogação, exclamação) no corpo inteiro da mensagem;
- Ser vazia (não possuir nenhum caractere).

O método em questão se chama *mensagem* e está contido no script `msg.py`, dentro do diretório `/src`.

### 1.2. Frameworks de teste

O framework utilizado para efetuar os testes sobre o método apresentado foi o *pytest*, junto com o framework de testes unitários nativo da linguagem Python, o *unittest*. Também foi utilizado o plugin *pytest-cov* para analisar a cobertura dos testes. Então, o comando aplicado foi `python3 -m pytest -rp --cov-report term-missing --cov=src/ test-{roteiro}.py`.

## 2. Roteiro de teste 1: técnica funcional

### 2.1. Particionamento de equivalência

Considerando os critérios para que uma mensagem seja considerada como “spam”, as mensagens podem pertencer a seis conjuntos, cinco deles inválidos e um válido. Usando apenas um caso de teste para cada conjunto, foram obtidos os seguintes casos de teste:

- CT01 (ct01.txt, spam), sendo que o arquivo em questão contém uma mensagem vazia.
- CT02 (ct02.txt, spam), sendo que o arquivo em questão contém o texto “Um gatinho para alegrar seu dia :) ”.
- CT03 (ct03.txt, spam), sendo que o arquivo em questão contém o texto “Quer se tornar o mais novo milionário do Brasil? Então, corre agora para o nosso site!”.
- CT04 (ct04.txt, spam), sendo que o arquivo em questão contém o texto “,,;,,,;?..?!..?!..!.;,;,;,,”.
- CT05 (ct05.txt, spam), sendo que o arquivo em questão contém o texto “sua chave XPOLfvbMsDtwZxcQWLpamzZxncYGsWHplmBvDrTyZn”.
- CT06 (ct06.txt, ham), sendo que o arquivo em questão contém o texto “Oi, tudo bem? Seu problema foi resolvido?”.

### 2.2. Análise do valor limite

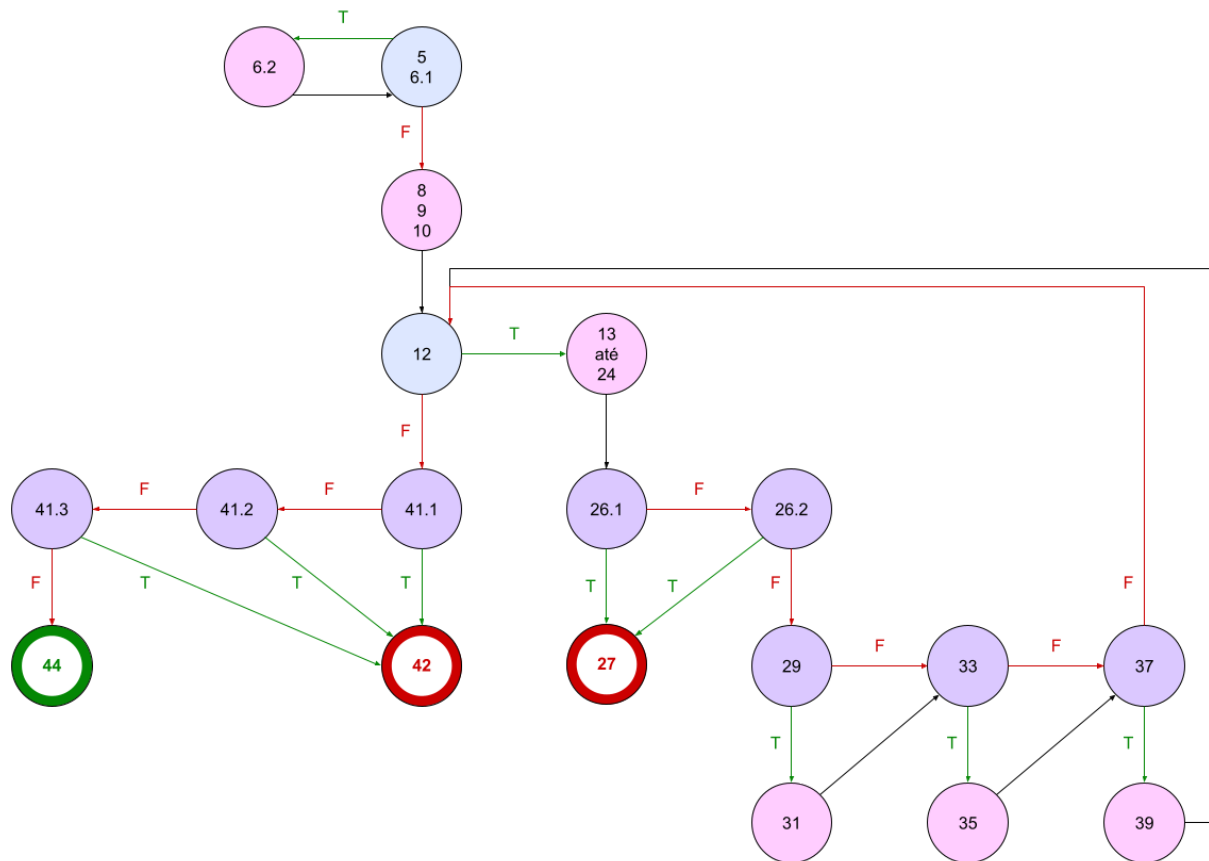
Analisando os valores críticos que cercam as classes consideradas inválidas, houve o acréscimo de seis casos de teste:

- CT01 (ct01.txt, spam), sendo que o arquivo em questão contém uma mensagem vazia.
- CT02 (ct02.txt, spam), sendo que o arquivo em questão contém o texto “Um gatinho para alegrar seu dia :) ”.

- CT03 (ct03.txt, spam), sendo que o arquivo em questão contém o texto “Quer se tornar o mais novo milionário do Brasil? Então, corre agora para o nosso site!”.
- CT04 (ct04.txt, spam), sendo que o arquivo em questão contém o texto “,,;,,,;;??.?!.?!.?!.?!.!.;;,,,;;,”.
- CT05 (ct05.txt, spam), sendo que o arquivo em questão contém o texto “sua chave XPOLfvbMsDtwZxcQWLpamzZxncYGsWHplmBvDrTyZn”.
- CT06 (ct06.txt, ham), sendo que o arquivo em questão contém o texto “Oi, tudo bem? Seu problema foi resolvido?”.
- CT07 (ct07.txt, ham), sendo que o arquivo em questão contém o texto “,.;,.;,.;,.;,” com 14 sinais de pontuação.
- CT08 (ct08.txt, ham), sendo que o arquivo em questão contém o texto “,.;,.;,.;,.;,.;,” com 15 sinais de pontuação.
- CT09 (ct09.txt, spam), sendo que o arquivo em questão contém o texto “,.;,.;,.;,.;,.;,” com 16 sinais de pontuação.
- CT10 (ct10.txt, spam), sendo que o arquivo em questão contém o texto “5bcdfg 2ae” com uma consoante a mais do que o dobro de vogais.
- CT11 (ct11.txt, ham), sendo que o arquivo em questão contém o texto “4bcdf 2ae” com a quantidade de consoantes igual ao dobro de vogais.
- CT12 (ct12.txt, ham), sendo que o arquivo em questão contém o texto “3bcd 2ae” com uma consoante a menos do que o dobro de vogais.

### 3. Roteiro de teste 2: técnica estrutural

#### 3.1. Grafo do fluxo de controle do método



Os nós do grafo estão numerados de acordo com as linhas do script em `msg.py`. Nós azuis indicam laços de repetição, nós rosas indicam atribuições contínuas de valores e nós roxos indicam condicionais. Quanto às saídas do método, os nós vermelhos designam a saída “spam” e o nó verde a saída “ham”.

#### 3.2. Níveis de cobertura

##### 3.2.1. Nível 1, 2 e 3: cobertura dos nós, de decisões e de condições

Para cobrir todos os nós do grafo, para que todas as decisões assumissem os valores TRUE e FALSE e para que todas as condições fossem alcançadas, foram utilizados exatamente os mesmos casos de teste:

- ### 3.2.2. Nível 4: cobertura de decisões/condições

- CT01 (ct01.txt, spam), sendo que o arquivo em questão contém uma mensagem vazia.
- CT02 (ct02.txt, ham), sendo que o arquivo em questão contém o texto “3212-8000”.
- CT03 (ct03.txt, spam), sendo que o arquivo em questão contém o texto “Loteriaaaa! ”.
- CT04 (ct04.txt, spam), sendo que o arquivo em questão contém o texto “ahsd????????????????????”.
- CT05 (ct05.txt, spam), sendo que o arquivo em questão contém o texto “Um gatinho para alegrar seu dia :) ”.
- CT06 (ct06.txt, spam), sendo que o arquivo em questão contém o texto “Quer se tornar o mais novo milionário do Brasil? Então, corre agora para o nosso site!”.
- CT07 (ct07.txt, spam), sendo que o arquivo em questão contém o texto “,.i,.i,.i,.i,.i”.
- CT08 (ct08.txt, spam), sendo que o arquivo em questão contém o texto “sua chave XPOLfvbMsDtwZxcQWLpamzZxncYGsWHplmBvDrTyZn”.

### 3.2.3. Nível 5: cobertura de condições múltiplas

Para cobrir todas as condições múltiplas, levando em consideração o funcionamento de compiladores, puderam ser removidos dois casos de teste do nível anterior (CT03 e CT04), já que a última condição deles não seria executada. Restaram os seguintes casos:

- CT01 (ct01.txt, spam), sendo que o arquivo em questão contém uma mensagem vazia.
- CT02 (ct02.txt, ham), sendo que o arquivo em questão contém o texto “3212-8000”.

CT03 (ct03.txt, spam), sendo que o arquivo em questão contém o texto “Um gatinho para alegrar seu dia :) ”.

- CT04 (ct04.txt, spam), sendo que o arquivo em questão contém o texto “Quer se tornar o mais novo milionário do Brasil? Então, corre agora para o nosso site!”.
- CT05 (ct05.txt, spam), sendo que o arquivo em questão contém o texto “,.i,.i,.i,.i,.i,.i,.i,.i,.i,.i”.
- CT06 (ct06.txt, spam), sendo que o arquivo em questão contém o texto “sua chave XPOLfvbMsDtwZxcQWLpamzZxncYGsWHplmBvDrTyZn”.

### 3.2.4. Nível 6: cobertura de loops

Um dos casos de teste do nível anterior já considerava a não execução de ambos os loops. Foram acrescentados testes para verificar a execução dos loops 1 vez, 2 vezes e n vezes:

- CT01 (ct01.txt, spam), sendo que o arquivo em questão contém uma mensagem vazia.
- CT02 (ct02.txt, ham), sendo que o arquivo em questão contém o texto “3212-8000”.
- CT03 (ct03.txt, spam), sendo que o arquivo em questão contém o texto “Um gatinho para alegrar seu dia :) ”.



- CT04 (ct04.txt, spam), sendo que o arquivo em questão contém o texto “Quer se tornar o mais novo milionário do Brasil? Então, corre agora para o nosso site!”.
- CT05 (ct05.txt, spam), sendo que o arquivo em questão contém o texto “,.i,.i,.i,.i,.i,.i,.i,.i,”.
- CT06 (ct06.txt, spam), sendo que o arquivo em questão contém o texto “sua chave XPOLfvbMsDtwZxcQWLpamzZxncYGsWHplmBvDrTyZn”.
- CT07 (ct07.txt, ham), sendo que o arquivo em questão contém o texto “Oi? Recebeu o arquivo?”, que fará os loops serem executados apenas uma vez.
- CT08 (ct08.txt, ham), sendo que o arquivo em questão contém o texto “Oi, tudo bem? Seu problema foi resolvido?”, que fará os loops serem executados duas vezes.
- CT09 (ct09.txt, ham), sendo que o arquivo em questão contém o texto “Monitores, bom dia! preciso dos casos de teste hoje a tarde. Atenciosamente, Bianca”, que fará os loops serem executados cinco vezes.

### 3.2.5. Nível 7: cobertura de todos os caminhos

Como o número de caminhos de execução seria muito grande para que eles fossem cobertos, os casos de teste finais obtidos foram:

- CT01 (ct01.txt, spam), sendo que o arquivo em questão contém uma mensagem vazia.
- CT02 (ct02.txt, ham), sendo que o arquivo em questão contém o texto “3212-8000”, que fará os loops serem executados apenas uma vez.
- CT03 (ct03.txt, spam), sendo que o arquivo em questão contém o texto “Um gatinho para alegrar seu dia :) ”.
- CT04 (ct04.txt, spam), sendo que o arquivo em questão contém o texto “Quer se tornar o mais novo milionário do Brasil? Então, corre agora para o nosso site!”.

- CT05 (ct05.txt, spam), sendo que o arquivo em questão contém o texto “`,.i,.i,.i,.i,.i,`”, que fará os loops serem executados apenas uma vez.
- CT06 (ct06.txt, spam), sendo que o arquivo em questão contém o texto “`sua chave XPOLfvbMsDtwZxcQWLpamzZxncYGsWHplmBvDrTyZn`”, que fará os loops serem executados duas vezes.
- CT07 (ct07.txt, ham), sendo que o arquivo em questão contém o texto “`Monitores, bom dia! preciso dos casos de teste hoje a tarde. Atenciosamente, Bianca`”, que fará os loops serem executados cinco vezes.