



Avaliação para Analista de Teste – FPF Tech

Nome:

Marque as questões de múltipla escolha na tabela abaixo, as demais questões de implementação podem ser incluídas no final da prova com os seus devidos links. Boa Sorte!

1	2	3	4	5	6	7	8

1. No processo de teste de software, o beta teste é:

- a) realizado por uma equipe de teste independente.
- b) realizado pelos clientes no local de trabalho do desenvolvedor de software.
- c) útil para testar software sob medida.
- d) executado o mais cedo possível no ciclo de vida.
- e) realizado por clientes em seu próprio local de trabalho.**

2. Por motivos que ainda não são completamente compreendidos por autores e estudiosos de Ciências da Computação, um grande número de erros em software tende a ocorrer nas fronteiras do domínio da entrada de dados. Engenheiros desenvolveram uma técnica de controle de qualidade que leva à seleção de casos de teste que exercitem esses valores. Assinalar a alternativa que tem como objetivo realizar este tipo de teste:

- a) Teste de comparação
- b) Análise de valor limite.**
- c) Teste de emparelhamento.
- d) Teste de matriz ortogonal.
- e) Teste de tabela de decisão.

3. A automatização de software é um processo, em qual, de alto custo financeiro e que envolve várias etapas de teste. Alguns dos testes de software, aplicados nesse processo, são os testes de caixa branca e os testes de caixa preta. Assinale a alternativa correta sobre o teste de caixa preta.

- a) É um teste de software, que visa aprimorar a interface homem – máquina.
- b) É um teste de interação com o banco de dados.**

- c) É um teste de verificação, segundo o Designer Pattern.
- d) Modelo de teste, adotado pelos modelos orientados a objeto.
- e) Teste que verifica as saídas de dados, usando diversos tipos de entradas de dados.

4. Os testes de software são executados, usando os procedimentos e documentos de script de teste. Para que a fase de execução de teste, seja realizada com sucesso deve(m) ser executado(s):

- a) os casos de uso
- b) os diagramas de atividade
- c) os testes de Turing
- d) o teste de COMA.
- e) os casos de teste

5. Assinale a alternativa que não corresponde a um dos testes de software, realizados em softwares comerciais.

- a) Teste de Sistema
- b) Teste de validação.
- c) Teste de integração
- d) Teste de volatilidade de requisitos
- e) Teste de unidade.

6. A execução de um sistema com o objetivo de encontrar falhas sob condições que demandam recursos em quantidade, frequência ou volume anormais é definida como:

- a) payload.
- b) teste de desempenho.
- c) teste de estresse.
- d) latência da falha.
- e) workload.

7. Como é chamado o tipo de teste que não utiliza o aprendizado empírico?

- a) Teste Ad-hoc.

- b) Teste de Exploratório.
- c) Teste de Estresse.
- d) Teste de Usabilidade.
- e) Teste de Segurança.

8. Em determinado formulário de um sistema WEB, é apresentado um campo de entrada que deve aceitar números inteiros de 10 a 500. De acordo com a análise de valor limite, que valores devem ser testados?

- a) 0, 455, 1000
- b) ?1, 20, 490, 1010
- c) ?10, 0, 455
- d) 9, 10, 500, 501**
- e) ?32768, ?1024, 18, 455, 492, 1024, 32768

9. Crie os cenários de teste e implemente a automação de testes funcionais utilizando o Selenium WebDriver e algum framework de teste unitário em qualquer linguagem de programação para a aplicação triângulo disponível em <http://www.vanilton.net/triangulo/>. Os testes devem cobrir o requisito:

Dados os três lados de um triângulo, o programa informará se o triângulo é equilátero, isósceles ou escaleno. Lembre-se que os lados só formam um triângulo se o comprimento de um lado for sempre menor do que a soma dos outros dois (o programa deve exibir uma mensagem de erro caso essa propriedade não seja satisfeita)

Por fim **compartilhe o link do código fonte no github ou gitlab** e documente o que é preciso e como realizar a sua execução.

10. Utilize uma ferramenta para teste em APIs Rest, sugerido JMeter ou Postman. Abaixo os tópicos a serem seguidos:

- Subir o serviço em Django Rest Framework para teste (containers Docker)
 - Faça o clone do projeto git: <https://github.com/Vanilton18/loja-api.git>
 - **Siga os passos do README.md** a partir da seção “Executar container API”
- Comportamentos a serem testados abaixo, lembre-se que todos devem possuir ao menos uma verificação
 - Criar um produto
 - <http://localhost:8000/produtos/>

Metod: POST

Content-Type: application/json

Body da Request:

```
{  
  "nome": "Mouse",  
  "preco": "3.00",  
  "quantidade": 2,  
  "categoria": 1  
}
```

Expected Status Code: 201

Para referência quanto os códigos de categorias.

```
CATEGORIAS = (  
  (1, 'Acessórios de Informática'),  
  (2, 'Memória RAM'),  
  (3, 'Disco Rígido/SSD'),  
  (4, 'Placa de Vídeo'),  
  (5, 'Gabinete'),  
  (6, 'Placa Mãe'),  
  (7, 'Fonte'),  
  (8, 'Processadores'),  
  (9, 'Eletrodomésticos'),  
  (10, 'Cama e Mesa'),  
)
```

- Filtrar um produto
 - http://localhost:8000/produtos/ID_PRODUTO/?format=json
 - Deve ser verificado o nome, preço, quantidade e categoria do produto

Por fim **compartilhe o link do código fonte no github ou gitlab** e documente o que é preciso e como realizar a sua execução.

11. No que tange seu conhecimento em Linux crie um script .sh que faça os seguintes passos quando executado:

- a. Crie uma pasta com seu nome
- b. Dentro da pasta com seu nome crie uma pasta com o nome “resultado”
- c. Baixe o arquivo hospedado em <https://vanilton.net/v1/download/zip.zip>
- d. Descompacte-o na raiz da pasta com seu nome

- e. Mova o arquivo descompactado para a pasta “resultado”
- f. Remova o arquivo baixado

Por fim **compartilhe o link para download** do script criado