A API a ser testada é de falsas publicações, em que os usuários podem gerir publicações.

Neste documento, deixo em detalhes os procedimentos que serão seguidos para testá-la de modo a garantir a segurança da mesma, para os utilizadores, seguindo os passos de avaliação já conhecidos (deixados no documento do desafio) bem como algumas recomendações minhas para melhoria.

1. **Definição da estratégia de teste a API**

**Teste de caixa preta de forma automatizada** - uma vez que não tenho acesso ao código, este teste vai ser de alto nível (apenas da análise de respostas a requisições) para garantir que em função de entradas que serão feitas por utilizadores, as saídas são as desejáveis.

**Ambiente de testes**

Um container Docker para rodar a API e expô-la de modo a possuir uma comunicação com a mesma.

**Objetivos do teste**

Garantir que existe uma comunicação a todos endpoints da API

Garantir que um usuário autenticado consegue criar post e estes ficam associados a ele mesmo.

Garantir que todo usuário autenticado ou não consegue listar post de todos.

Garantir que a atualização de post só é feita pelo dono do post autenticado e ninguém mais.

Garantir que a eliminação de post só é feita pelo dono do post autenticado e ninguém mais.

Garantir a integridade dos dados.

Garantir que as passwords são salvas de forma encriptada.

**Tipo de teste**

Vou optar pelos seguintes testes

| **Tipo** | **Descrição** |
| --- | --- |
| Funcional | Garantir que haja uma cobertura a nível de CRUD, bem como a validação dos inputs a cada um dos endpoints |
| Teste negativo | Garantir que não posso inserir dados nulos, requisições sem token |
| Segurança | Acesso baseado em roles (cada um responsável por fazer modificação no seu próprio post) e Injeção em SQL |
| Tratamento de erros | Garantir que a API responda com códigos HTTP perceptíveis, incluindo mensagens |

**Observação pessoal**: não irei fazer teste de performance porque esta API está no meu computador e não remotamente, sendo assim vai ser rápido, mas se estivesse remotamente faria.

**Nível de teste**

Apenas a nível de api de forma automatizada, com Postman

**Dados**

Os dados de teste usados, serão os que irei criar via api

**Ferramentas**

**Robot Framework** - para automatização da realização de testes aos endpoints da aplicação.

**Técnicas de desenho de dados**

***Análise De Valores Limites (AVL)***

Sendo que a API precisa de login, vou verificar se a API aceita as passwords que obrigatoriamente levam pelo menos tem 6 caracteres para melhor segurança, é um valor padronizado e muito aceitável. O valor máximo é muito variável.

| **AVL** | **Descrição** |
| --- | --- |
| AVL\_001 | Password com 5 caracteres |
| AVL\_002 | Password com mais de 5 caracteres |

***Partição de equivalência (PE)***

Vou manipular o email durante a criação de dados

| **PE** | **Descrição** |
| --- | --- |
| PE\_001 | email sem @ |
| PE\_002 | email sem domínio |
| PE\_003 | email vazio |
| PE\_004 | email com caracteres especiais |

**Bug**

Um user consegue apagar post do outro

É arriscado ter uma api que cria users com password de 1 caracter