

Avaliação de conhecimento em Desenvolvimento de Software

segunda-feira, 5 de julho de 2021

17:54

Requisitos

Ferramentas, frameworks, serviços e pontos importantes que devem ser utilizados e considerados na implementação do projeto:

- Linguagem a ser utilizada: Python
- Data de Entrega: 15/07/2021
- Utilizar micro serviços
- Implementar no mínimo um dos pontos do projeto
- Documentar todo o projeto de forma sintética, porém adentrando cada ponto de forma geral.
- Após a conclusão do projeto entregá-lo através de uma PR.

Escopo do Projeto

Criação de Função para Geração de Senha:

Para geração de senha deve-se seguir a política de complexidade para padronização seguiu-se o mesmo escopo utilizado para a Microsoft em que consiste que a senha deve seguir os seguintes atributos:

- Colocar a quantidade de caracteres da senha como 9
- Conter Letras Maiúsculas de Idiomas Europeus (A a Z)
- Conter Letras Minúsculas de Idiomas Europeus (A a z)
- Caracteres não alfanuméricos
- Qualquer caractere Unicode
- O nome do usuário ou e-mail não deve fazer parte do conjunto da senha.

Modelagem do Projeto:

Neste projeto foi visado a utilização de dois frameworks para o desenvolvimento web bem conhecidos como o Django e o flask, como o objetivo do projeto é simples e não requer inicialmente uma maior abstração de outros fatores externos como a utilização de banco de dados, formulários etc., foi definido que o mini framework flask e o ideal já que este apresenta simplicidade, rapidez no desenvolvimento e velocidade para criação de micro serviços.

Foi separado por implementações seguindo da seguinte forma:

- Index - que este é responsável pela chamada inicial da tela e pela rota inicial, sendo que nesta tela encontra-se os dados para o usuário definir como os parâmetros de acesso a url e a geração da própria senha randômica
- Password - função responsável pela verificação dos request no momento de chamada da url criada com a senha
- Error - Redirecionamento para request de rotas inexistentes exibindo uma mensagem ao usuário.

Após a definição das rotas e suas respectivas funções, o próximo passo era definir como controlar os requests para cada chamada que iria ser feita através da url criada

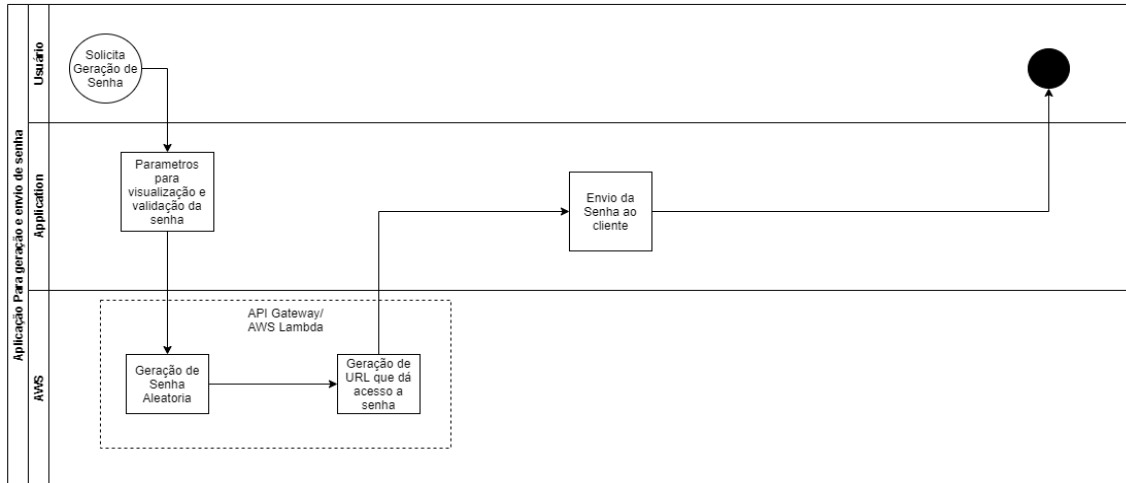
contendo a senha, então através de pesquisas foi procurada uma maneira de realizar este processo, dentro do próprio flask é disponibilizado (<https://www.section.io/engineering-education/implementing-rate-limiting-in-flask/>) em através do flask limiter definir pelos parâmetros do usuário a quantidade de requisições, porem devido à demora a encontrar essa funcionalidade não foi implementado ao projeto final, porém fica como uma implementação futura.

Escopo para implementação de uso serverless

Para a estrutura serverless foi pensando na AWS já que dentre as prestadoras destes serviços a AWS possui um custo-benefício por uso de recursos menor, além e claro da familiaridade com a ferramenta.

dentre as ferramentas disponíveis o uso da API Gateway é o mais apropriado já que esta ferramenta além de disponibilizar a criação de estruturas backend para monitoração e controle de requests para o uso da geração de senhas, além e claro que ele e integrado para uso do lambda da aws.

Porem dentre as pesquisas para implementação serverless foram encontradas além da implementação através da interface da AWS, a possibilidade de fazer a integração através de plugins lambda disponibilizados por npm, de modo a trazer a conexão e implementações do serverless incorporados ao seu backend, inicialmente seguindo o exemplo do (<https://www.serverless.com/blog/serverless-python-packaging>), foi testado o projeto inicial, porem devido a algum problema relacionado as credenciais da AWS não foi possível dar continuidade a implementação seguindo com a primeira proposta de criar a conexão e testes na própria interface do serviço.



Design do Projeto:

Para layout do projeto foi definido o uso do bootstrap já que é um framework de implementação simples

Implementações Futuras:

- Implementação de Micro serviços
- Implementação do Serverless através do Lambda da AWS
- Revisão de Pontos de Segurança
- Implementação de Quantidade de Request para acesso a uma url.