

Documentação Técnica

Criação da aplicação

Para a criação da *api* foi escolhido o framework **Flask**, pois é o framework *python* que possui familiaridade.

A aplicação foi construída utilizando o padrão *application factory*.

Para os tests foi utilizada o módulo **unittests**

A api foi disponibilizada no *azure* e no *heroku*.

Também é possível utilizar adicionar em um container conforme arquivo *readme.md*.

A especificação da *api* pode ser encontrada no *endpoint* **/docs** para isso foi utilizado **Swagger**.

Resolução do problema

A principal parte da resolução do problema se encontra no módulo *words* e segue o fluxo de execução:

1. O texto é limpo removendo-se os caracteres especiais e transformado em minúsculo para evitar duplicidade
2. São removidos as *stopwords* (brasileiras) para cada texto é transformado em uma lista de palavras (*list*), cada item que for uma *stopword* é removido o texto volta a ser uma *string*
3. Para geração das listas de palavras também são utilizadas listas de strings
4. Após a geração das listas de palavras, também é gerado as listas de ocorrência também utilizando *list*
5. Cada item gerado é adicionado a um dicionário que pode ser filtrado na chamada da API

Foram utilizados mais objetos do tipo *list*, porque apesar dos *sets* já resolverem o problema de duplicidade, alteravam a ordem das palavras.

A *api* recebe em seu *endpoint* via *post* um *json* contendo os textos e um filtro de resultado (opcional) e retorna um *json* com os resultados (que podem ser removidos pelo filtro):

- textos
 - Os textos enviados já limpos e com *stopwords* removidas
- listaDePalavras
 - A lista de palavras gerada para todos os textos
- listaDeDuasPalavras
 - A lista de duas palavras gerada para todos os textos
- vetorDePalavras

- Os vetores de ocorrência de palavras para cada texto
- vetorDeDuasPalavras
 - Os vetores de ocorrência de palavras para cada texto

Para salvar o histórico foi utilizado **mongodb** como serviço da **mlab**, como estava na camada gratuita foi criado um índice para remover o documento da coleção após 1 dia.

Tecnologias Utilizadas

Nome	Descrição
Visual Studio Code	Editor de Texto utilizado para criação dos scripts de workflow
Py Charm Professional	IDE utilizada para desenvolvimento
Docker for Windows	Utilizado para testar um container contendo a aplicação
WSL 2	Utilizado GIT e ssh + gpg para assinatura dos commits
Python	Python para windows versão 3.8
GitHub	Utilizado para disponibilizar código e executar workflows de teste e deploy
Codecov	Utilizado para subir a cobertura de testes do fonte
Azure	Utilizado para hospedar a aplicação [camada gratuita]
Heroku	Utilizado para hospedar a aplicação [camada gratuita]
Mlab	Mongo db como serviço (camada gratuita)

Agradecimentos

Agradeço a empresa pela disponibilização do teste e teste e pela oportunidade. A construção da aplicação me ajudou a testar novas tecnologias e adquirir novos conhecimentos.

Todo feedback será bem-vindo. Sempre podemos melhorar 🙏🐸

Obrigado!