# **Documentação do Projeto: Diário oficial**

## **Objetivo**

O objetivo deste projeto é automatizar a visualização do Diário Oficial para ser colocado no sharepoint das bibliotecas.

## **Processos da Automação**

* **Pega texto do Diário Oficial:**
  + Acessa as mensagens do e-mail pelo através do protocolo imap(Protocolo de acesso de mensagens via internet – Permite com que possamos acessar mensagens de e-mail).
  + Filtramos as mensagens enviadas somente pelo e-mail do Diário Oficial e no dia atual.
  + Identifica e armazena o texto do e-mail selecionado.
  + Decodifica os textos e envia como resposta da requisição.
  + Caso não tenha recebido nenhum e-mail relacionado, ou tenha ocorrido algum erro no processo, será retornado uma página html sobre o não recebimento do e-mail ou sobre o erro, respectivamente.
* **Código hospedado na web:**
  + O código está hospedado na web através da plataforma “Vercel”, ele realiza o processo e mostra a resposta quando o site é acessado ou requisitado.
  + Nele é apresentado o conteúdo html do e-mail ou das mensagens e problemas que possam ter ocorrido.
  + O site pode ser acessado através do seguinte link: https://pegadiarioofcsmc.vercel.app/
* **Dados no SharePoint:**
  + Como os dados são apresentados ao acessarmos a página, no SharePoint de cada biblioteca está presente um iframe que referência o site e mostra suas informações atualizadas na própria página.

## **Arquivos do Projeto**

* **Arquivo app.py:** Este é o arquivo que cria o sistema de endereços na url. Através da biblioteca Flask, ele roda o código do processo e renderiza o HTML na página quando a url é acessada (Através da requisição “GET”).
* **Arquivo pegar\_dados.py:** Arquivo que contém o código onde todo o processo é realizado, desde a definição do e-mail até o retorno do conteúdo HTML.
* **Pasta templates:** A pasta contém arquivos com conteúdo HTML e CSS para os outros tipos de respostas que podem ocorrer no site.
* **Arquivo requirements.txt:** Arquivo que contém bibliotecas usadas e suas versões, necessárias para que o Vercel consiga entender e executar o código.
* **Arquivo vercel.json:** Arquivo de configuração para o Vercel.

**Requisitos do Sistema**

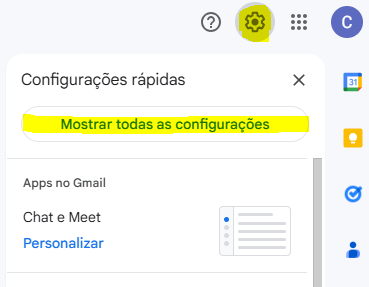
* Python 3.8.10
* Pacotes importados: Flask, imap\_tools, datetime, base64.

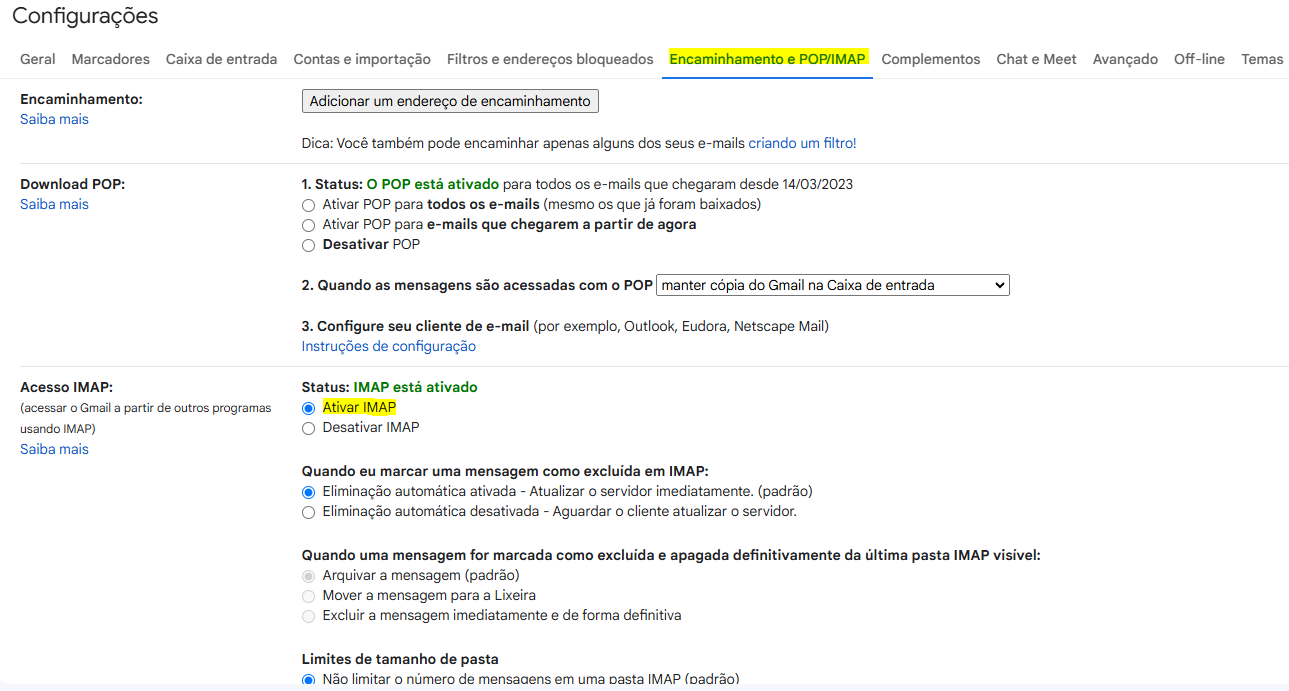
## **Explicação de Funções**

* **atualizar\_dados():**
  + É definido o e-mail e a senha de aplicativo de onde serão pegas as mensagens de e-mail.
  + Filtra e pega os e-mails com a data atual e enviados pelo Diário Oficial.
  + Caso um e-mail seja encontrado, ele pega o conteúdo, trata para encaixar na formatação, e retorna o conteúdo HTML.
  + Caso não tenha encontrado ou tenha ocorrido algum erro, ele retorna o conteúdo HTML correspondente da pasta templates.
* **atualizar\_dados\_pagina():**
  + É a função que é executada quando é feito um acesso na página.
  + É definido o e-mail do Diário Oficial, para fazer a filtragem de e-mails.
  + Conecta no servidor imap para pegar as informações dos e-mails.
  + Pega o HTML retornado pela função “atualizar\_dados()” e renderiza o conteúdo na página.

## **Uso do Projeto**

* Para que o código consiga pegar as mensagens de um e-mail, é necessário ativar a conexão imap na conta.





* Agora é necessário criar uma senha de aplicativo para ser usada pela aplicação, para isso, é necessário acessar as configurações de segurança da conta e ativar a verificação em duas etapas, após isso é possível criar uma senha de aplicativo, para criar, basta ir na barra de pesquisa das configurações e pesquisar “Senhas de app”, após a criação, vc ira colocar seu email e senha de app dentro do código “pegar\_dados.py” nos campos “email” e “senha”, respectivamente (Colocar entre aspas simples ou duplas).
* O código está rodando em um servidor do vercel na web, isso significa que a plataforma vercel lê o código através do gihub hub em tempo real, então caso queira que alguma alteração feita no código seja atribuída ao site, é necessário passar os códigos atualizados para o github, na branch “main”, e realizar o commit.