

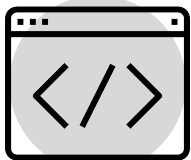
Profissão Cientista de Dados



BOAS PRÁTICAS



Streamlit III



- **Entenda o Deploy**
- **Configure o Personal Access Token (PAT) no GitHub**
- **Adicione arquivos no repositório**
- **Configure sua conta no Heroku**
- **Crie e configure o Procfile**
- **Crie e configure o Requirements.txt**
- **Realize o Deploy via Linha de Comando**



Entenda o Deploy

- Utilize ferramentas adequadas para o deploy: O processo de disponibilizar o que foi desenvolvido na máquina do desenvolvedor para o usuário é chamado de deploy. Existem várias maneiras de disponibilizar o resultado de uma análise, como tabelas do Excel, tabelas em bancos de dados, gráficos em PNG, ferramentas de BI, APIs e Streamlit. Escolha a ferramenta que melhor atenda às necessidades do projeto e do usuário final.
- Prototipação e desenvolvimento devem ser feitos na máquina do desenvolvedor: A arquitetura desejada é que a prototipação e o desenvolvimento do arquivo .py continuem sendo feitos na máquina do desenvolvedor. O armazenamento do arquivo .py será feito no GitHub.



Entenda o Deploy

- Utilize servidores para execução da aplicação: A inicialização do Streamlit e a execução da aplicação devem ser feitas em um servidor, como o Heroku. Isso permite que o usuário final acesse a aplicação através de seu próprio navegador, sem precisar ter Python instalado em sua máquina ou saber como escrever Python.



Configure o Personal Access Token (PAT) no GitHub

- Sempre crie um novo repositório no GitHub para iniciar o desenvolvimento de uma nova aplicação. Isso ajuda a manter o código organizado e facilita o rastreamento de alterações.
- Clone o repositório para o seu computador local. Isso permite que você trabalhe no código mesmo quando estiver offline.
- Mova todos os arquivos necessários para a aplicação para dentro do repositório. Isso garante que todos os arquivos estejam no mesmo local e possam ser facilmente acessados.



Adicione Arquivos ao Repositório

- Não inclua todos os pacotes listados pelo comando 'pip freeze' no seu arquivo .txt. Selecione apenas os pacotes que são relevantes para o seu projeto.
- Se um pacote foi atualizado durante o desenvolvimento, certifique-se de incluí-lo no arquivo .txt.



Configure sua Conta no Heroku

- Sempre crie um token para subir arquivos para o Heroku. Isso aumenta a segurança do seu projeto.
- Use o token criado para enviar todos os arquivos do repositório local para o repositório remoto. Isso garante que todos os arquivos necessários para o projeto estejam disponíveis no Heroku.
- Ao criar um token no GitHub, dê uma descrição rápida para ele e nomeie-o de acordo com o projeto. Isso facilita a organização e o gerenciamento de vários tokens.
- Depois de criar o token, copie-o e guarde-o em um local seguro. Lembre-se de que essa é a única vez que ele será visível, então é importante não perdê-lo.



Crie e Configure o Profile

- Após adicionar os arquivos, faça um 'commit' e um 'push' para enviar os arquivos para o repositório remoto. Atualize o repositório para confirmar que todos os arquivos foram enviados com sucesso.
- Se houver algum problema com o comando 'git', tente abrir uma caixa de diálogo e usar o 'token' como senha.



Crie e Configure o Requirements .txt

- Conecte seu repositório ao Render para facilitar o acesso à aplicação. Isso pode ser feito criando uma conta no Render e seguindo o processo de conexão.
- Esteja preparado para resolver problemas que possam surgir durante a criação da aplicação, como a não localização de versões específicas de bibliotecas. Isso pode ser feito editando o arquivo 'requirements.txt' e substituindo as versões das bibliotecas.



Crie e Configure o Requirements .txt

- O objetivo final deve ser criar uma aplicação na máquina local, fazer upload para o Render, fazer as alterações necessárias e, em seguida, disponibilizar um link para que qualquer pessoa possa utilizar a aplicação. Esteja preparado para fazer alterações no repositório e observar a reconstrução da aplicação em tempo real.



Realize o Deploy via Linha de Comando

- Conecte seu repositório do GitHub ao Heroku. Dessa forma, sempre que um novo commit for feito no GitHub, o Heroku automaticamente construirá uma nova versão da aplicação. Isso garante que qualquer alteração feita no código seja automaticamente refletida na versão da aplicação disponível para os usuários.
- Sempre verifique como sua aplicação aparece no navegador após o deploy. Isso garante que a aplicação está funcionando corretamente e que os usuários estão vendo a versão mais recente.



Bons estudos!

