**Descrição do projeto**

Parabéns! Você concluiu a seção sobre ferramentas de desenvolvimento de software. Agora é hora de aplicar o conhecimento e as habilidades que você adquiriu em um projeto: vamos criar e implementar um dashboard de aplicativo web em um serviço de nuvem.

Quando você terminar o projeto, lembre-se de enviar seu trabalho ao revisor para avaliação. Você vai receber feedback em até 48 horas. Use esse feedback para fazer alterações e envie a nova versão de volta ao revisor.

Você talvez receba mais feedback referente à nova versão. Isso é completamente normal. Não é incomum passar por vários ciclos de feedback e revisão.

Seu projeto será considerado concluído assim que o revisor o aprovar.

**Descrição do projeto**

O foco desse projeto é que você tenha mais prática com tarefas comuns de engenharia de software. Isso vai aumentar e complementar suas habilidades de dados e vai fazer com que você seja uma opção mais atrativa para possíveis empregadores.

As tarefas são: criar e gerenciar ambientes virtuais Python, desenvolver um aplicativo web e implantá-lo em um serviço de nuvem que o tornará acessível ao público.

Neste projeto, forneceremos um conjunto de dados de anúncios de vendas de carros. No entanto, neste projeto, o foco não será no conjunto de dados nem na análise, portanto, você está livre a escolher o conjunto de dados que desejar.

# Instruções para concluir o projeto

### Etapa 1. Configurar tudo

1. Crie uma conta em [github.com](http://github.com/). Se você já tem uma conta ou criou uma no [capítulo sobre o Git e GitHub](https://tripleten.com/trainer/data-scientist/lesson/4eaef0c8-0849-4290-9f70-7024aad628b4/), pode pular essa etapa.
2. Crie um novo repositório git com um arquivo README.md e um arquivo .gitignore (escolha um modelo Python). Se você precisar de uma recapitulação, consulte [esta lição](https://tripleten.com/trainer/data-scientist/lesson/9b3a2508-a317-4393-af9c-1ce27b53d250/).
3. Crie uma conta em [render.com](http://render.com/), se você ainda não fez isso. Falamos sobre o Render e aplicativos web [nesta lição](https://tripleten.com/trainer/data-scientist/lesson/58a5617c-a749-443a-b7ff-958515c1f049/). Quando estiver criando uma conta no Render, selecione a opção "GitHub" e siga os passos de inscrição. Isso é exatamente o que você precisa para vincular sua conta no Render à sua conta no GitHub.
4. Neste projeto, você vai realizar uma análise exploratória de dados. Para fazer isso, você precisa ter os pacotes pandas e plotly-express instalados. Já falamos sobre plotly [nesta lição](https://tripleten.com/trainer/data-scientist/lesson/2dde32d5-6918-43d7-9cc9-901d78c66695/). plotly-express é projetado com padrões razoáveis e opções configuráveis para tipos comuns de gráficos, o que o torna um ótimo ponto de entrada para iniciantes, em comparação com o próprio plotly. Se você ainda não o conhece bem, não se preocupe — vamos guiar você por todo o processo! Além disso, você vai precisar do pacote streamlit para desenvolver um aplicativo web. Crie um novo ambiente virtual e escolha um nome significativo que seja relacionado ao conjunto de dados com o qual você vai trabalhar. Por exemplo, você poderia chamá-lo de vehicles\_env. Certifique-se de que você instalou pelo menos os seguintes pacotes no ambiente: pandas, streamlit e plotly-express.

Se você precisar de uma recapitulação sobre como criar um ambiente virtual e instalar pacotes, consulte [a lição sobre este tópico](https://tripleten.com/trainer/data-scientist/lesson/2dde32d5-6918-43d7-9cc9-901d78c66695/).

1. [Instale o VS Code](https://code.visualstudio.com/download), se você ainda não fez isso. Clone seu repositório do projeto do GitHub e abra-o como um projeto no VS Code. Esse será o diretório do seu projeto. Defina o interpretador Python para aquele usado pelo ambiente virtual que você criou anteriormente.
2. Para fins de simplicidade, em vez de salvar seu ambiente no arquivo requirements.txt no diretório do projeto, crie manualmente o arquivo requirement.txt e adicione três bibliotecas ali sem especificar as versões delas:
3. pandas
4. plotly\_express
5. streamlit

### Etapa 2. Baixar **o arquivo de dados**

1. [Baixe o conjunto de dados de anúncios de carros](https://practicum-content.s3.us-west-1.amazonaws.com/new-markets/Data_NEW_4_sprint/vehicles.csv) (vehicles\_us.csv) ou encontre seu próprio conjunto de dados em formato CSV.
2. Coloque o conjunto de dados no diretório do projeto.

### Etapa 3. Análise exploratória de dados

1. Crie um diretório chamado notebooks no diretório do seu projeto.
2. Crie um caderno Jupyter chamado EDA.ipynb no VS Code e salve-o no diretório notebooks do projeto. Lembre-se de que .ipynb é uma extensão de arquivo usada para cadernos Jupyter.
3. Abra o caderno Jupyter EDA.ipynb e experimente com plotly-express para criar visualizações para uma análise exploratória básica do conjunto de dados no caderno. Aqui estão alguns exemplos de:

### Etapa 4. Desenvolver o dashboard do aplicativo web

1. Crie um arquivo app.py no diretório raiz do projeto. Para criar um arquivo .py, clique em "New File" (Novo arquivo) no VS Code e armazene-o no diretório do projeto com o nome desejado e a extensão .py.
2. Importe streamlit como st, pandas e plotly\_express no início do arquivo.
3. Leia o arquivo CSV do conjunto de dados em um DataFrame. O código será o mesmo que você tinha no Jupyter Notebook ao explorar o conjunto de dados.
4. Agora, vamos criar o conteúdo do aplicativo baseado em Streamlit. Aqui está o que queremos que você inclua nele:
   * Pelo menos um cabeçalho (você pode usar [st.header()](https://docs.streamlit.io/library/api-reference/text/st.header) *(os materiais estão em inglês)* para fazer isso. Na [lição sobre aplicativos web](https://tripleten.com/trainer/data-scientist/lesson/58a5617c-a749-443a-b7ff-958515c1f049/), mostramos como criar um cabeçalho).
   * Crie um botão que, ao ser clicado, vai criar um histograma plotly-express. Para fazer isso, considere usar as funções [st.write()](https://docs.streamlit.io/library/api-reference/write-magic/st.write) e [st.plotly\_chart()](https://docs.streamlit.io/library/api-reference/charts/st.plotly_chart) *(os materiais estão em inglês)*.

Aqui está um exemplo de como você pode fazer isso:

import pandas as pd

import plotly.express as px

import streamlit as st

car\_data = pd.read\_csv('vehicles\_us.csv') *# lendo os dados*

hist\_button = st.button('Criar histograma') *# criar um botão*

if hist\_button: *# se o botão for clicado*

*# escrever uma mensagem*

st.write('Criando um histograma para o conjunto de dados de anúncios de vendas de carros')

*# criar um histograma*

fig = px.histogram(car\_data, x="odometer")

*# exibir um gráfico Plotly interativo*

st.plotly\_chart(fig, use\_container\_width=True)

* + Adicione outro botão que, ao ser clicado, cria um gráfico de dispersão plotly-express. Сonsidere usar as funções [st.write()](https://docs.streamlit.io/library/api-reference/write-magic/st.write) e [st.plotly\_chart()](https://docs.streamlit.io/library/api-reference/charts/st.plotly_chart) *(os materiais estão em inglês)*.

Esta etapa é opcional, mas se você quiser um desafio extra, tente substituir botões por caixas de seleção, disponíveis em streamlit através de st.checkbox(). Você pode solicitar que o usuário selecione a caixa de seleção que corresponda a um histograma ou um gráfico de dispersão, e então um gráfico será criado dependendo da caixa selecionada. Aqui está um exemplo simples de como caixas de seleção funcionam em streamlit:

import streamlit as st

*# criar uma caixa de seleção*

build\_histogram = st.checkbox('Criar um histograma')

if build\_histogram: *# se a caixa de seleção for selecionada*

st.write('Criando um histograma para a coluna odometer')

...

1. Certifique-se de atualizar o arquivo README quando terminar. Ele deve incluir uma breve descrição do projeto, explicando para que serve o aplicativo web e quais funcionalidades ele oferece.
2. Para tornar o Streamlit compatível com o Render, adicione um arquivo de configuração do Streamlit ao repositório do seu projeto em streamlit/config.toml com o seguinte conteúdo:
3. [server]
4. headless = true
5. port = 10000
7. [browser]
8. serverAddress = "0.0.0.0"
9. serverPort = 10000

Ele dirá ao Render para procurar no lugar certo para ouvir seu aplicativo Streamlit ao colocá-lo em seus servidores.

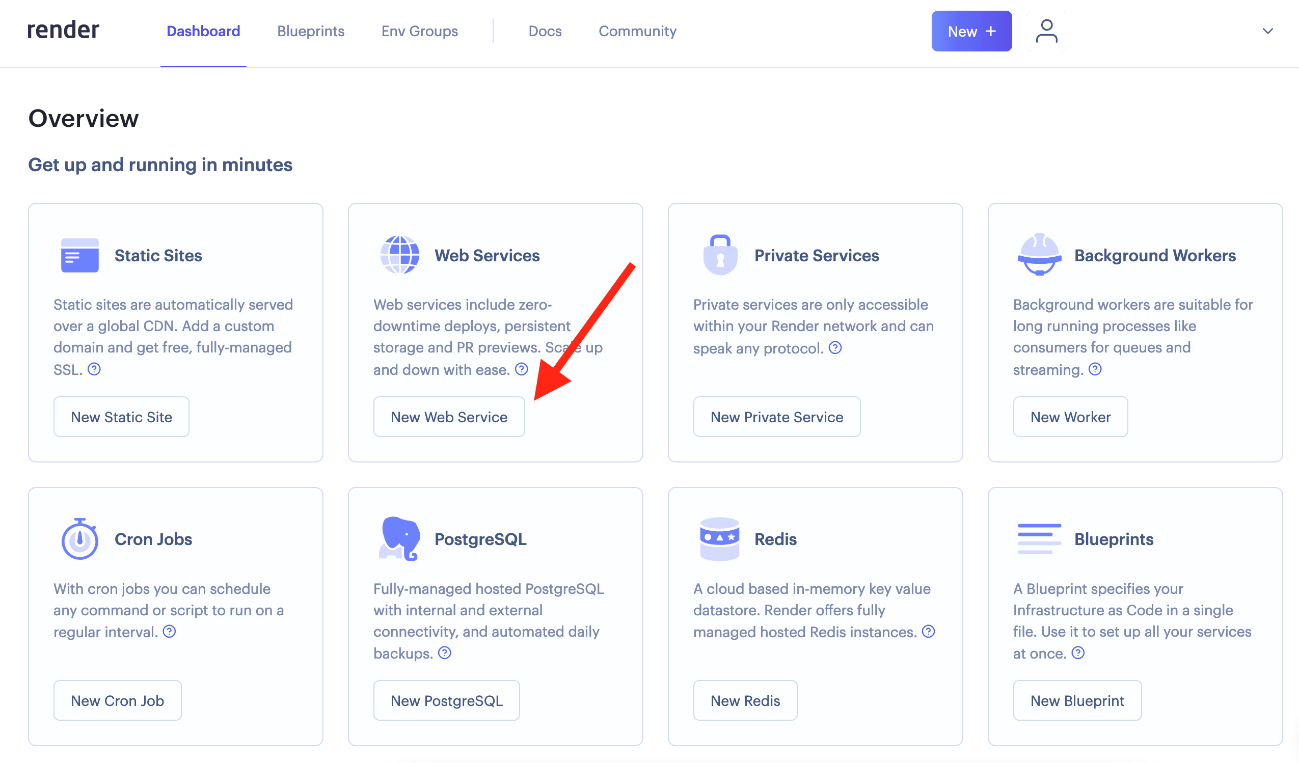
1. Não se esqueça de confirmar e enviar todas as alterações de volta para o seu repositório após terminar o trabalho. Se você não fizer isso, nada vai funcionar direito.

Observações importantes:

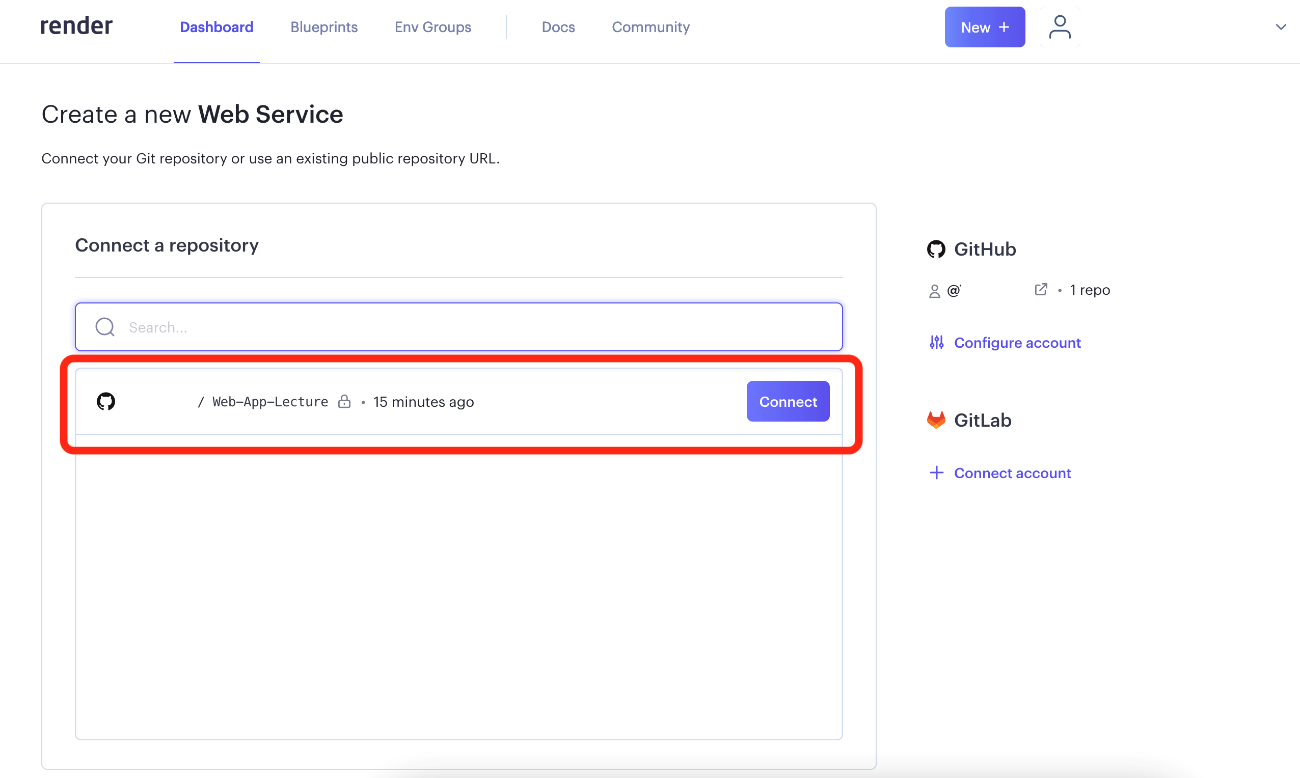
* À medida que você desenvolve o aplicativo adicionando um novo componente Streamlit, você pode executar o comando streamlit run app.py do terminal para ver como está o resultado.
* À medida que você atinge alguns marcos no desenvolvimento do aplicativo (por exemplo, você adiciona um componente de trabalho e o aplicativo é executado sem erros), é uma boa prática é confirmar e enviar seu trabalho para um repositório remoto no GitHub. Portanto, não se esqueça de escrever uma mensagem de commit significativa!

### Etapa 5. Implementar a versão final do aplicativo no Render

1. Abra sua conta em [render.com](http://render.com/) e crie um novo serviço web:



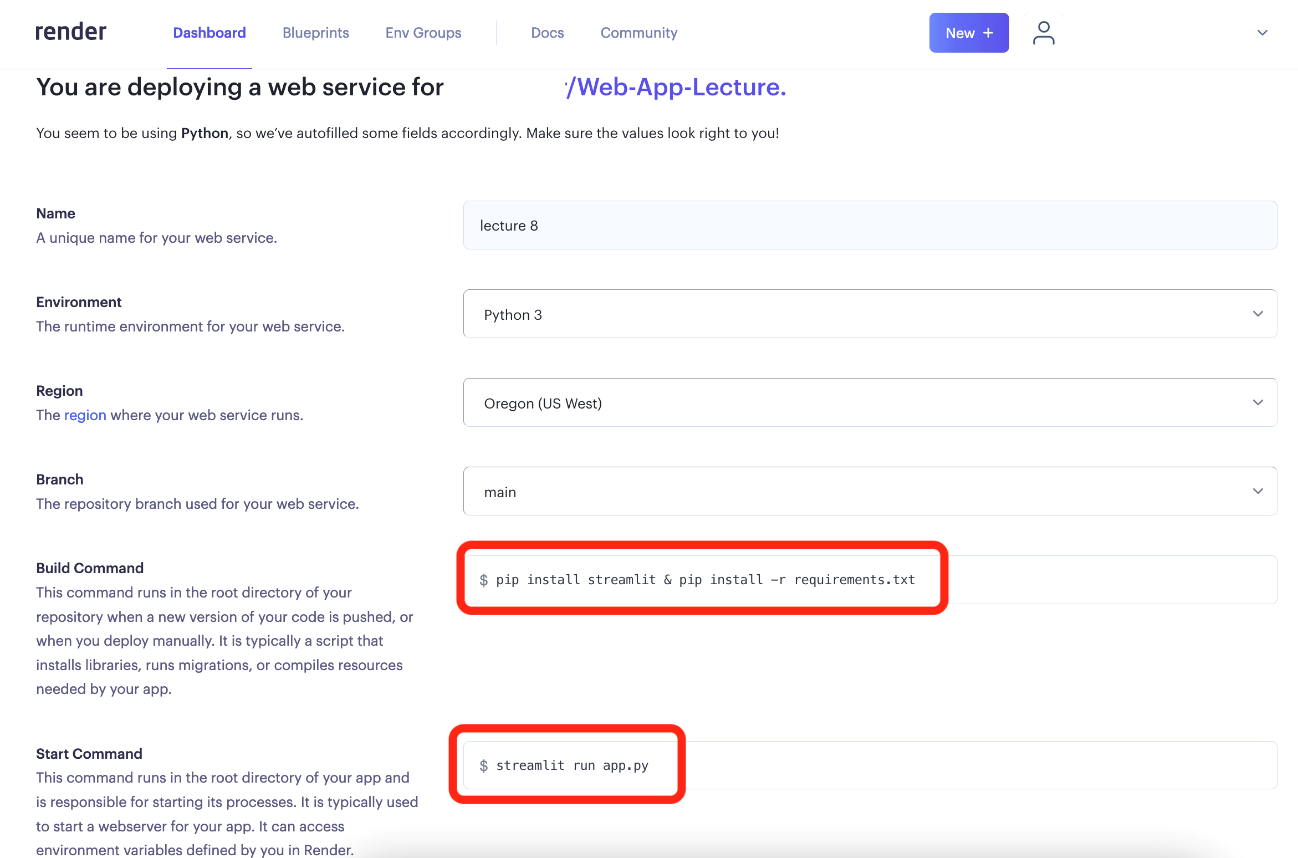
1. Crie um novo serviço web vinculado ao seu repositório Github:



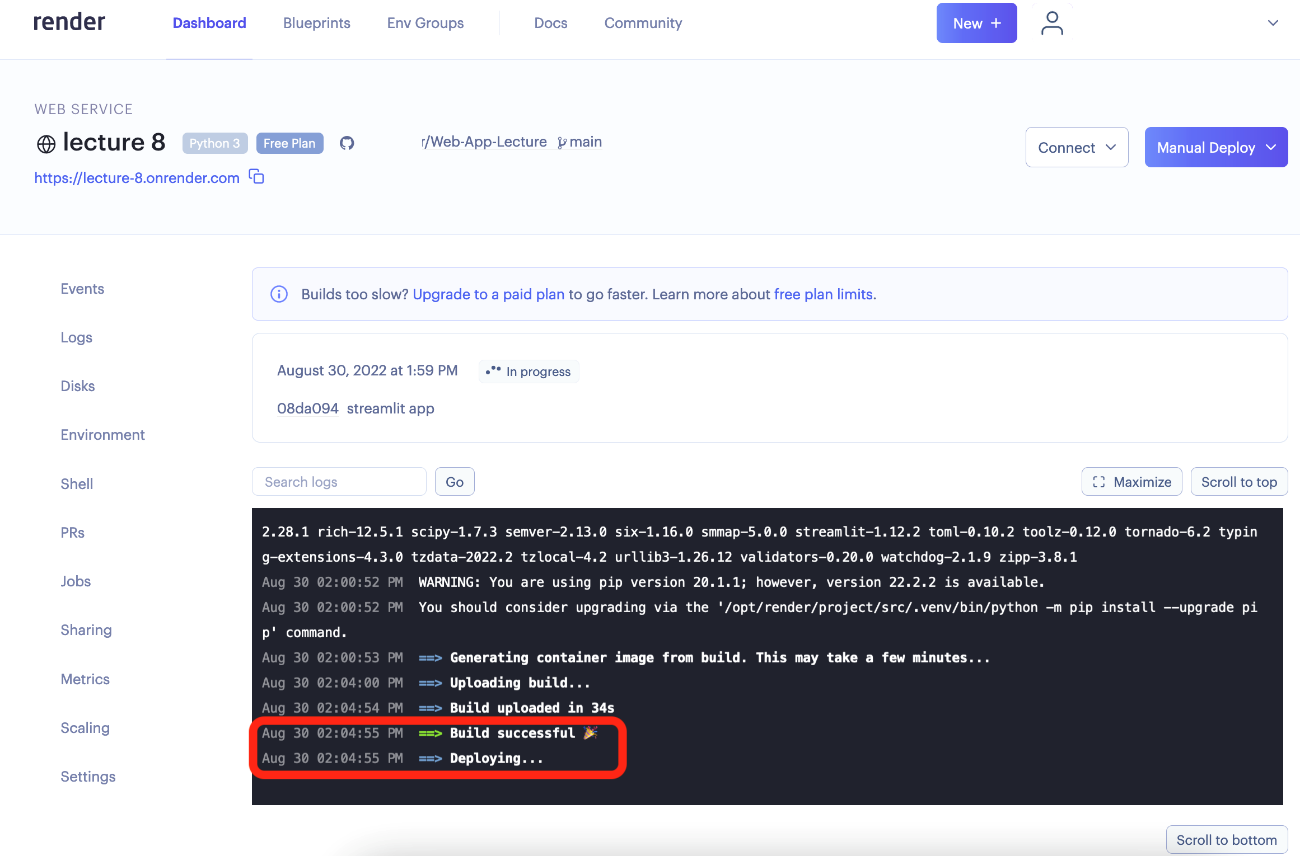
1. Configure o novo serviço web Render adicionando um comando de compilação Build Command que vai instalar tudo o que seja necessário para executar seu aplicativo, incluindo streamlit e todos os pacotes de requirements.txt. Use o seguinte comando:

pip install --upgrade pip && pip install -r requirements.txt

Adicione o seguinte ao seu Start Command: streamlit run app.py. Deve ficar assim:



1. Implemente o aplicativo no Render e aguarde até que a compilação seja bem-sucedida:



1. Verifique se seu aplicativo está acessível no seguinte URL: https://<APP\_NAME>.onrender.com/

Observação: pode levar vários minutos após uma implementação bem-sucedida para que o aplicativo fique disponível online em um nível gratuito. Observe também que os aplicativos ficam "adormecidos" após ficarem inativos por alguns minutos. Nesse caso, basta carregar e atualizar a página do aplicativo algumas vezes para que ele seja ativado.

Se você atualizar seu repositório GitHub, para implementar a versão mais recente no Render, clique em Manual Deploy → Latest Commit.

## Como enviar o projeto:

Você vai precisar enviar um link para seu repositório GitHub. Adicione também o URL do aplicativo no Render ao README.md do seu projeto

## Como meu projeto será avaliado?

Elaboramos alguns critérios de avaliação. Leia-os com atenção antes de fazer o envio.

Isto é o que os revisores procuram ao avaliar seu projeto:

* O repositório do projeto contém pelo menos os seguintes arquivos?
* $ tree
* .
* ├── README.md
* ├── app.py
* ├── vehicles\_us.csv
* ├── requirements.txt
* └── notebooks
* └── EDA.ipynb
* └── .streamlit
* └── config.toml