



Criação de aplicativos Data Web com linguagem Python e biblioteca Streamlit

Contextualização

As vezes, ao finalizarmos um projeto de Data Science e Machine Learning, podemos nos deparar com a seguinte questão: como podemos compartilhar nosso projeto com usuários? Disponibilizar somente o código-fonte puro pode parecer ser uma saída fácil, mas nem sempre é o ideal ou com melhor aplicação, principalmente na hora de criar um portfólio ou apresentar esse estudo de forma mais profissional.

Para este problema, já temos diversas ferramentas e aplicações prontas disponíveis e variadas soluções para uma boa apresentação. Publicar seu trabalho em uma aplicação web, ou melhor, em um *Data APP* - aplicativo de/sobre dados -, é um ótimo caminho, afinal, com apenas um URL você pode compartilhar e disponibilizar para diversos públicos!

Porém, criar uma aplicação nem sempre se mostra simples, principalmente para aqueles que não tem experiência em desenvolvimento de páginas web. Como uma forma de contornar o problema e fazer com que a criação de um

aplicativo seja possível apenas com a linguagem Python - muito difundida e uma das principais utilizadas na área de Dados -, surgiu o Streamlit.

O Streamlit é uma biblioteca Python de uso gratuito e tem como principal objetivo facilitar o desenvolvimento de aplicações *web* para projetos de *Data Science* e *Machine Learning*, ou seja, não é necessário conhecimentos extras para criar seu próprio aplicativo e compartilhar seu projeto!



Image by Streamlit

Agora que você já entende quando usar o Streamlit, iremos ver como ele pode agilizar a criação da sua aplicação!

Streamlit: rápido e eficiente!

Para demonstrar o quão prático é criar com o Streamlit, seguiremos com uma simples demonstração: uma interface para o usuário informar sua idade e ter como retorno seu ano de nascimento. Por preferência, não envolveremos diretamente um conjunto de dados neste primeiro contato - *por enquanto!*

Com apenas algumas linhas de código e em uns pouquíssimos minutos, já temos uma interface funcional e estilizada. Abaixo segue um exemplo de aplicação:

Calculadora de ano de nascimento

Qual idade você completará este ano?

21

☒ Já fiz aniversário esse ano

Clique aqui para calcular

Você nasceu no ano de: **2000**

Mas e o código, como ficou?

Para você que já está imaginando algo complexo, difícil de entender e que vai lhe render horas e horas de trabalho para fazer algo mínimo, diríamos: *relaxe, é extremamente simples!* Se não acredita, veja o código utilizado para este exemplo.

```
import streamlit as st

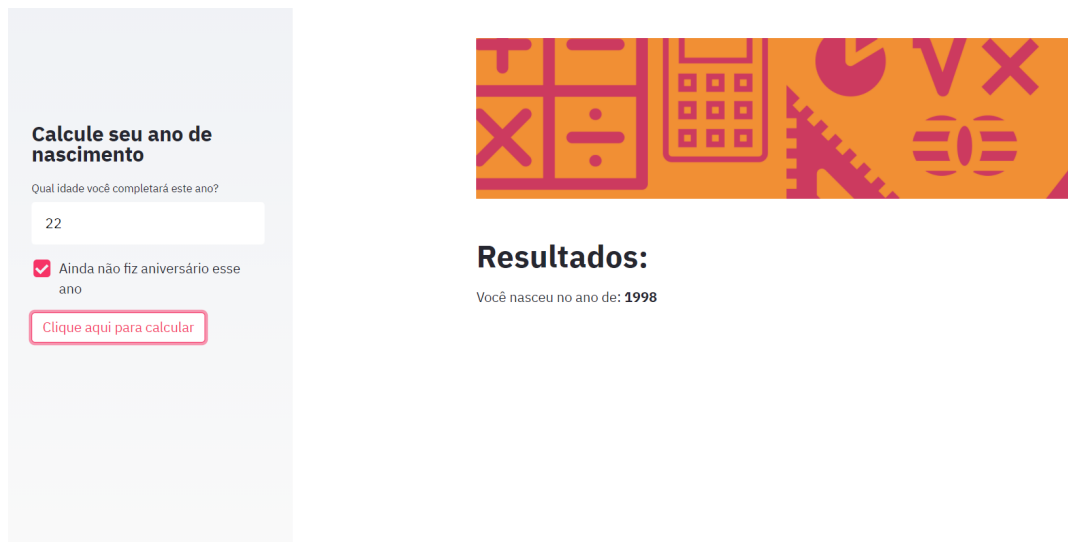
st.title("Calculadora de ano de nascimento")
idade = st.text_input("Qual idade você completará este ano?")
opcao = st.checkbox('Já fiz aniversário esse ano', value=False)
botao = st.button("Clique aqui para calcular")

if botao:
    if opcao: ano = 2021-int(idade)
    else: ano = 2021-int(idade)-1
    st.write(f"Você nasceu no ano de: **{ano}**")
```

Pouquíssimas linhas, não é mesmo?

Os elementos utilizados no Streamlit são bem intuitivos e fáceis de implementar, com uma rápida consulta na [documentação](#) disponibilizada, você consegue criar um layout sem se preocupar com a parte visual - cores, tamanhos, formas -, afinal, por padrão, as interfaces Streamlit já tem uma aparência minimalista e bem organizada.

Contudo, iremos melhorar este exemplo adicionando mais elementos, inclusive, mostrando como é fácil inserir uma *sidebar* - um menu/barra lateral - e uma imagem para deixar a aplicação visualmente mais agradável. Com algumas poucas alterações, chegamos a isto:



E nosso código foi este:

```
import streamlit as st

st.image("calc-guide.png", use_column_width=True)
st.title("Resultados:")
st.sidebar.title("Calcule seu ano de nascimento")
idade = st.sidebar.text_input("Qual idade você completará este ano?")
opcao = st.sidebar.checkbox('Ainda não fiz aniversário esse ano', value=False)
botao = st.sidebar.button("Clique aqui para calcular")

if botao:
    if opcao: ano = 2021-int(idade)-1
    else: ano = 2021-int(idade)
    st.write(f"Você nasceu no ano de: **{ano}**")
```

Para os olhos mais atentos, verá que a única alteração necessária para adicionar nossa barra lateral foi acrescentar a palavra *sidebar* - marcada em vermelho no código - antes dos elementos que desejávamos inserir nela.

Exato, simples assim!

Passo-a-passo: criando um Data APP

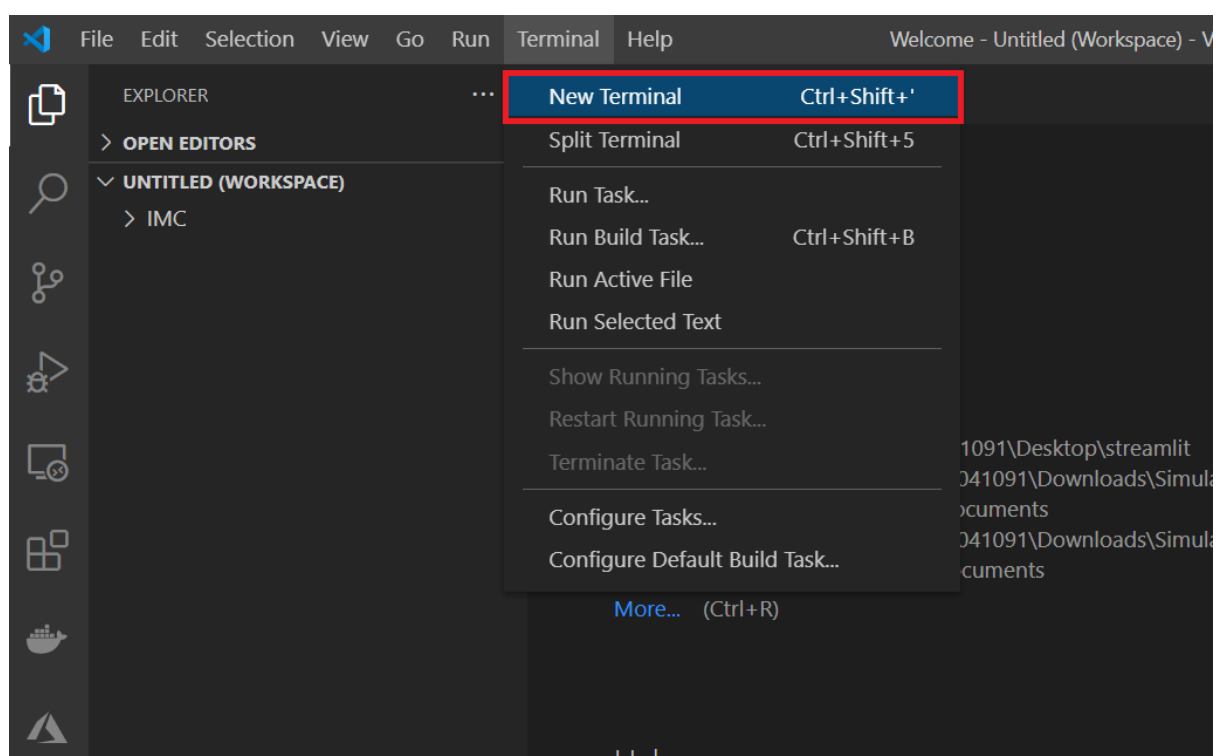
Agora iremos sair das demonstrações e ir direto ao ponto: como eu crio o aplicativo? Indo do ponto zero, começaremos pela instalação do ambiente. Você precisa ter instalado uma versão mais recente do Python e algum editor de sua preferência - nos exemplos usaremos o Visual Studio Code, mas também temos o PyCharm e vários outros. Escolha ou utilize um ambiente que já esteja familiarizado!

Com isso pronto, você pode instalar a biblioteca Streamlit com um simples código no terminal - do Windows ou direto no editor que você escolheu:

```
pip install streamlit
```

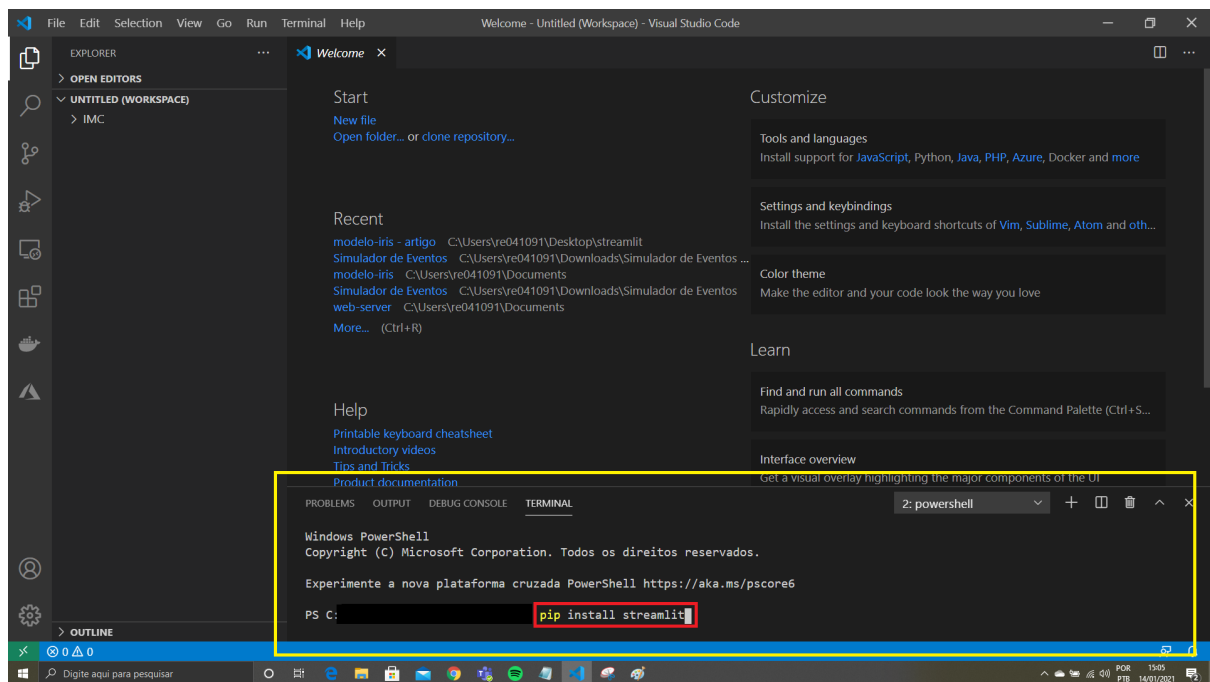
Instalando o Streamlit passo-a-passo no Visual Studio Code

Caso esteja utilizando o Visual Studio Code, inicie o editor e, na barra superior, encontre o menu *Terminal*. Dentro dele, localize a opção *New Terminal* e selecione-o, como mostrado abaixo:

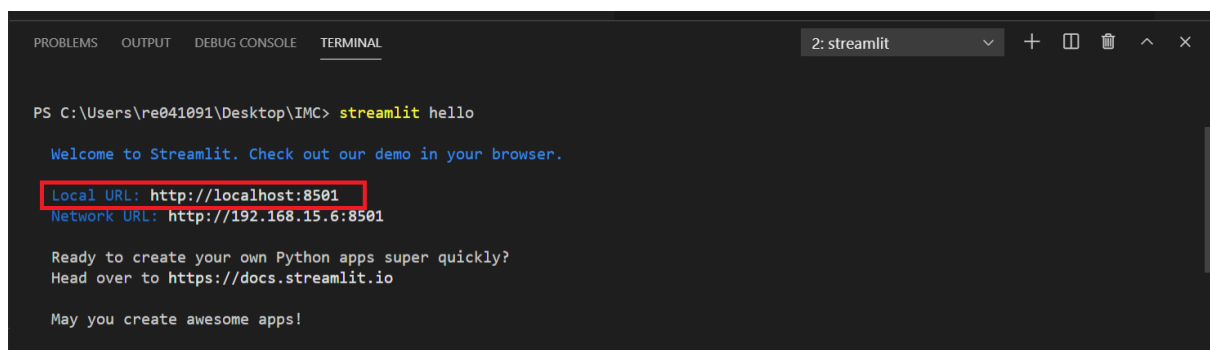


Após isso, uma janela abrirá na parte inferior da janela, esse é nosso *Terminal*, onde podemos executar códigos, baixar e atualizar pacotes e bibliotecas. Lembre-se disso, iremos utilizá-lo mais vezes.

Dentro do *terminal*, escreva `pip install streamlit` e aperte *enter* no teclado.



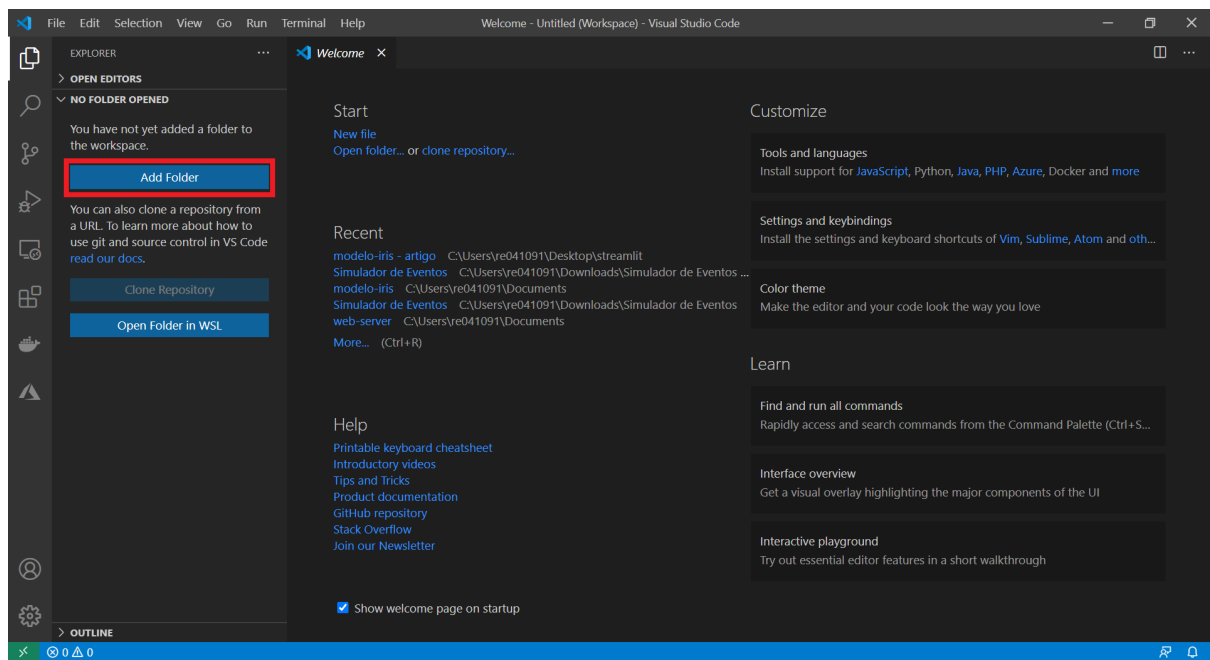
Espere a instalação finalizar. Caso queria testar, escreva `streamlit hello` no *terminal*, ele irá gerar uma mensagem e lhe fornecer uma *local URL*, você pode acessar e verificar uma página com algumas instruções iniciais do Streamlit. Caso queria ler, sintá-se a vontade para testar - fica como uma tarefa extra 😊!



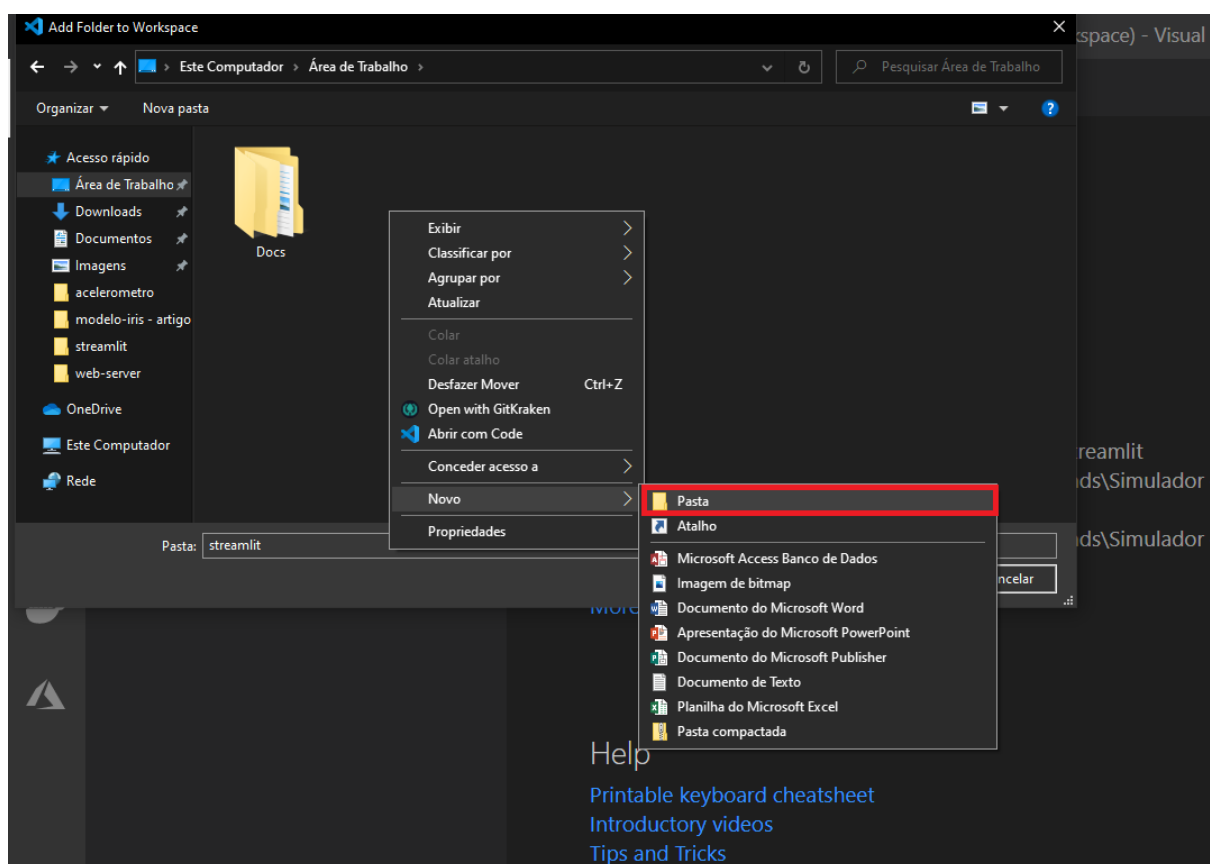
E vamos começar a programar!

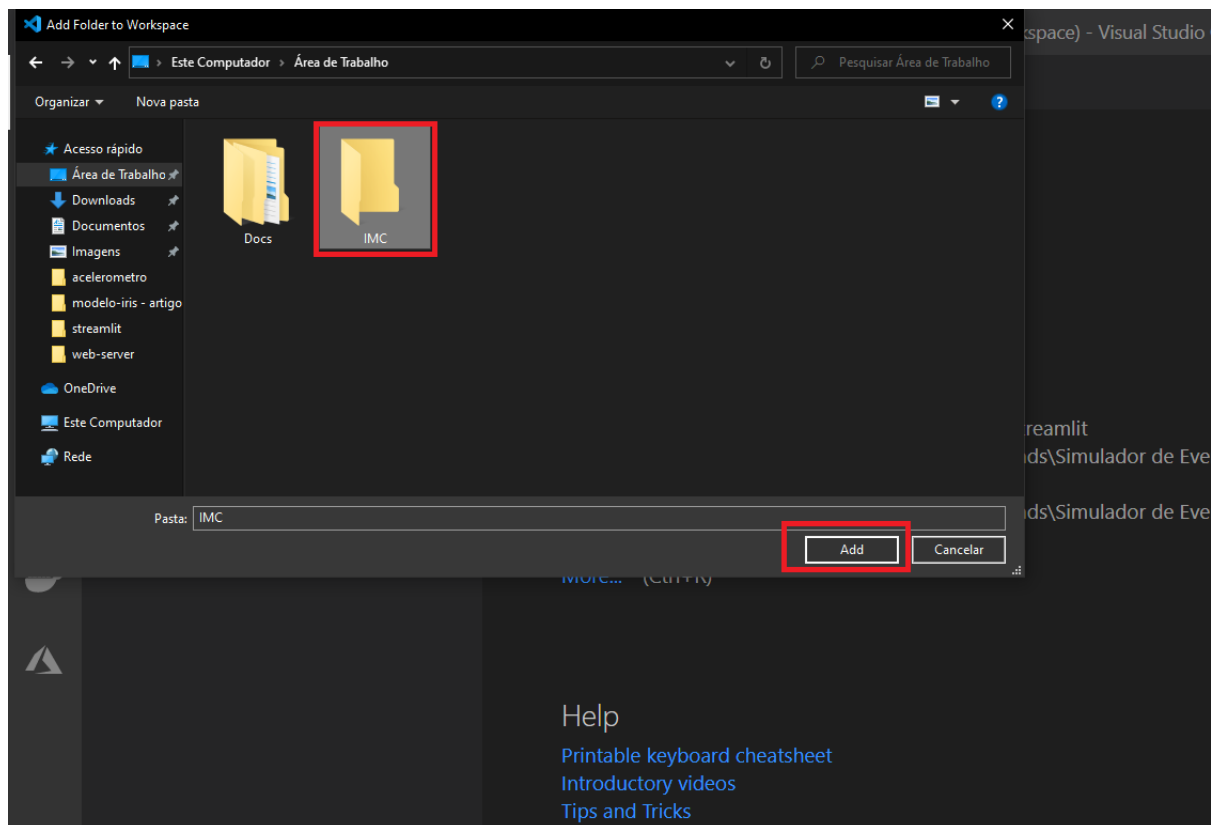
Finalmente com o Streamlit instalado, iremos seguir com um pequeno projeto: criar uma calculadora de IMC! Os códigos estarão todos disponibilizados, porém, recomendo abrir seu editor e ir escrevendo cada linha junto com as etapas.

Para começar, iremos criar uma nova pasta para armazenar nosso projeto. Se for a primeira vez que você está usando o editor ou não tenha nenhuma outra pasta aberta, poderá criar diretamente na opção *Add Folder*, como mostrado abaixo:

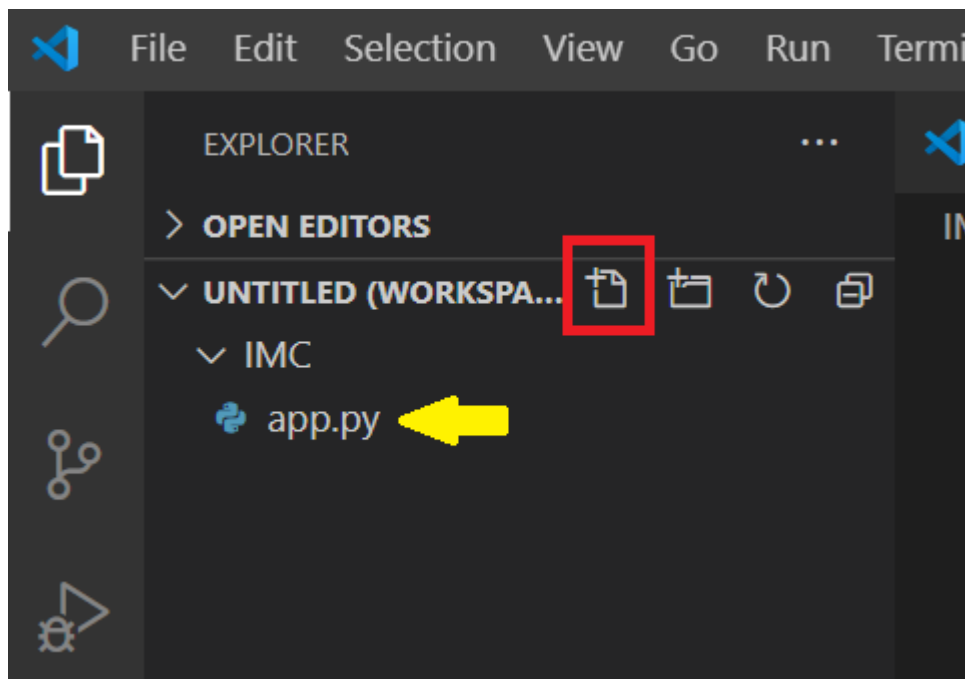


Em seguida, abrirá uma nova janela. Escolha um local de preferência e crie uma nova pasta com o nome de IMC, selecione esta pasta e clique em *Add*. É nela que iremos salvar nossos códigos e aplicações.

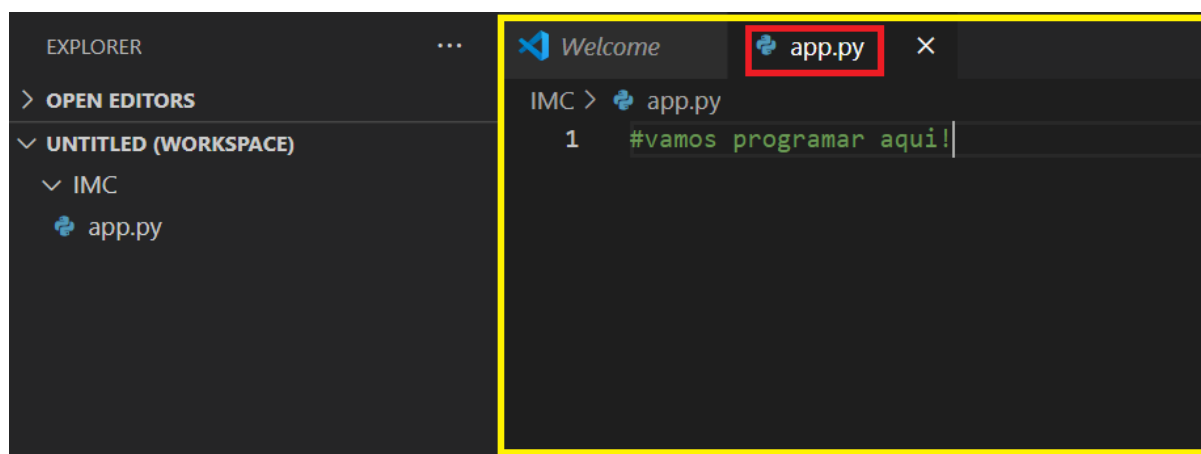




Após a pasta criada, iremos agora abrir um novo arquivo Python, onde iremos programar nossa calculadora de IMC. Para este exemplo, nomeei o arquivo de `app.py`.



Abra o arquivo app.py. Para quem não está familiarizado, uma janela lateral será aberta como na imagem a seguir:



Agora sim, podemos começar a programar! Primeiro, precisaremos importar as bibliotecas que serão utilizadas no projeto, como será apenas uma calculadora de IMC, apenas a própria biblioteca Streamlit é suficiente. Sendo assim, digite esse código no seu editor:

```
import streamlit as st
```

Com o Streamlit importado, precisamos pensar em quais dados iremos precisar do usuário, para assim calcular seu IMC, e quais elementos serão mais intuitivos para que ele insira os dados sem dificuldade.

Sendo o IMC a divisão do peso (em kg) pela altura ao quadrado (em metros), são estas as informações essenciais para nossa calculadora. Podemos então testar algumas estruturas e verificar qual se encaixará melhor no projeto, começando com o elemento *slider*.

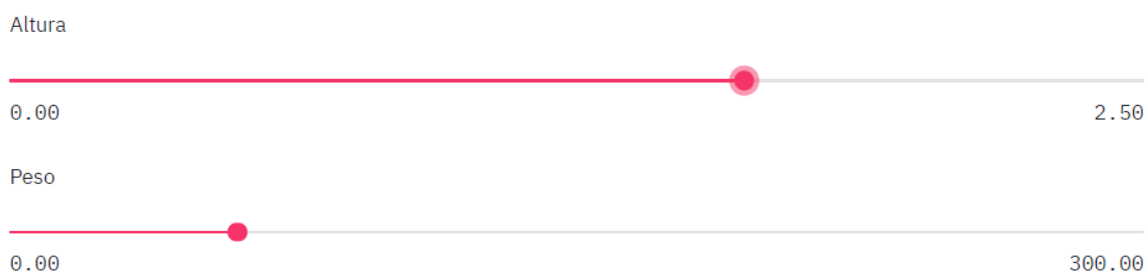
Slider nada mais é que uma barra de valores - podemos assim dizer -, pela qual você pode percorrer com um indicador. Como por exemplo:



```
st.slider("Altura", 0.00, 2.50, value = 1.70)
```

Com apenas uma linha de código, já criamos um elemento formatado e funcional na aplicação! Outros elementos podem ser utilizados, a [documentação do Streamlit](#) traz vários exemplos e alguns são bem interessantes para quem quer utilizar [gráficos e tabelas de dados](#).

Seguindo com os *sliders* para cada informação, teremos:



Como queremos receber estes dados, uma saída é atribuir os *sliders* diretamente à variáveis, que irão manter o valor indicado pelo usuário - e que utilizaremos para o alimentar o modelo!

```
import streamlit as st

altura = st.slider("Altura", 0.00, 2.50, value = 1.70)
peso = st.slider("Peso", 0.0, 300.0, value = 90.0)
```

Okay, mas como eu visualizo essa aplicação?

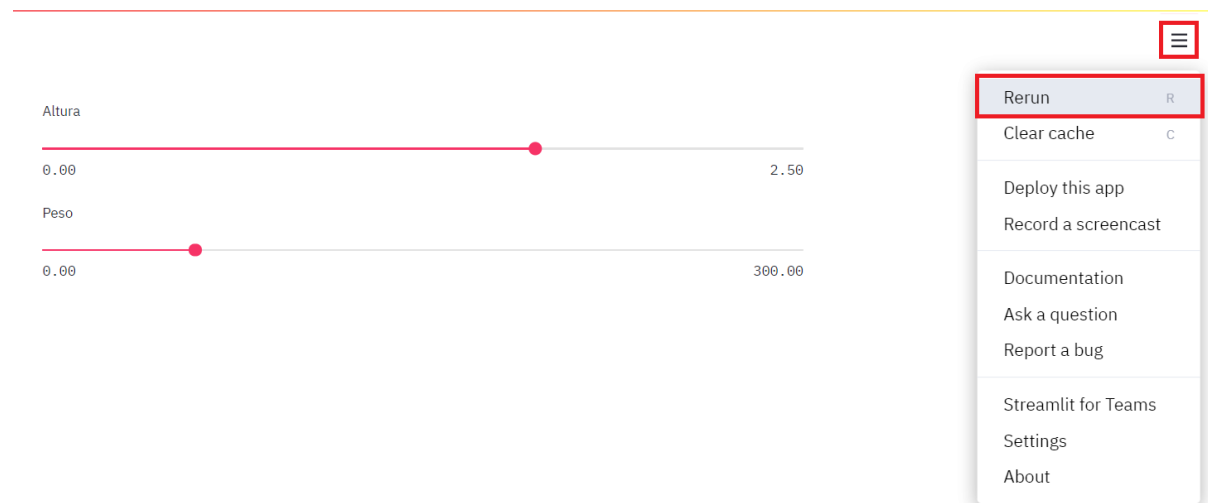
Abra seu *terminal* novamente e digite `streamlit run app.py`. Ao executar o comando, um *Local URL* será fornecido - como no teste que fizemos - e, ao acessar esse *URL*, uma página será aberta no seu navegador com sua aplicação!

```
PS C:\Desktop\IMC> streamlit run app.py

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://192.168.15.6:8501
```

Agora que você já consegue visualizar, qualquer alteração salva no código será automaticamente válida, sendo assim, você pode atualizar a página do navegador ou utilizar a opção *Rerun* da própria aplicação *Streamlit*, que suas alterações surgiram na aplicação também.



Para deixar a aplicação mais completa, adicionaremos alguns títulos e descrições. Aqui, utilizaremos os elementos *title* e *write*, desta forma:

```
st.title("Calculadora de IMC") #escreva seu título aqui
st.write("Informe abaixo seu peso e altura: ") #escreva seu texto aqui
```

Visualmente, já teremos:

Calculadora de IMC

Informe abaixo seu peso e altura:

Altura



Peso



Perceba o quanto a interface já ganhou forma e traz uma sensação de profissionalismo, em comparação a simplesmente subir e enviar um código puro, que nem sempre é simples de manipular e testar.

Para fechar nossa aplicação, iremos adicionar um botão para ativar o cálculo de IMC com os dados recebidos. Para inserir este mecanismo, iremos atribuir também uma variável ao botão, que irá receber *True* - se o botão for acionado - ou *False* - estado em que o usuário não interage com o botão.

```
botao = st.button("Calcular")
```

Calcular

Pronto, nossa interface já está pronta!

Calculadora de IMC

Informe abaixo seu peso e altura:

Altura

0.00 2.50

Peso

0.00 300.00

Calcular

Agora iremos programar o cálculo de IMC e ativá-lo/desativá-lo conforme a ação do botão. Para isto, o primeiro passo é verificar se o botão está sendo acionado:

```
if botao:
    #será verificado se a variavel botao é True
```

Depois, precisamos calcular o IMC do usuário com os dados que ele informou:

```
if botao:
    IMC = peso/(altura**2)
```

Com o resultado atribuído na variável *IMC*, utilizaremos o elemento *markdown* para retornar esse resultado e exibi-lo.

```
if botao:
    IMC = peso/(altura**2)
    st.markdown(f"De acordo com os valores, seu IMC é: **{IMC}**")
```

Pronto, temos uma aplicação completa! Agora iremos testar:

Calculadora de IMC

Informe abaixo seu peso e altura:

Altura

0.00

2.50

Peso

0.00

300.00

Calcular

De acordo com os valores, seu IMC é: **23.003353147386065**

Nossa aplicação retornou um IMC de, aproximadamente, 23! Daqui, podemos evoluir o projeto: arredondar o resultado - que tem muitas casa decimais -, retornar qual escala esse IMC representa - abaixo do peso, peso normal, sobrepeso, etc -, coletar o nome, idade, colocar o preenchimento de informações em uma *sidebar*.

Segue abaixo como ficaria se fizemos as alterações sugeridas:

Calculadora de IMC

Informe abaixo seu peso e altura:

Nome

Idade


Altura

0.00 2.50

Peso

0.00 300.00

Calcular



Olá Anne!

Idade: **21** anos

Peso: **62.05** Kg

Altura: **1.62** m

De acordo com os valores informados:

Seu IMC é: **23.64**

Você está na faixa de: **Peso normal**

Uau!

Vou deixar aqui o código atualizado, como um desafio para quem quiser testar e tentar replicar esse projeto completo.

```
import streamlit as st

st.sidebar.title("Calculadora de IMC") #escreva seu título aqui
st.sidebar.write("Informe abaixo seu peso e altura: ") #escreva seu texto aqui

st.image("calc-guide.png", use_column_width=True) #adiciona imagem

nome = st.sidebar.text_input('Nome', 'Escreva seu nome') #adiciona uma caixa de texto
idade = st.sidebar.number_input('Idade', value = 0) #adiciona uma caixa p/ números

altura = st.sidebar.slider("Altura", 0.00, 2.50, value = 1.70)
peso = st.sidebar.slider("Peso", 0.0, 300.0, value = 90.0)
botao = st.sidebar.button("Calcular")

if botao:
    IMC = peso/(altura**2)

    if IMC <= 18.5: classificacao = "Abaixo do peso"
    elif IMC <= 24.9: classificacao = "Peso normal"
    elif IMC <= 29.9: classificacao = "Sobrepeso"
    elif IMC <= 34.9: classificacao = "Obesidade (Grau I)"
    elif IMC <= 39.9: classificacao = "Obesidade Severa (Grau II)"
    else: classificacao = "Obesidade Mórbida (Grau III)"

    st.title(f"Olá **{nome}**!")
    st.markdown(f"Idade: **{idade}** anos")
    st.markdown(f"Peso: **{peso}** Kg")
    st.markdown(f"Altura: **{altura}** m")
    st.markdown("De acordo com os valores informados:")
    st.markdown(f"Seu IMC é: **{round(IMC,2)}**")
    st.markdown(f"Você está na faixa de: **{classificacao}**")
```

O que foi alterado está marcado em vermelho, uma boa estratégia é olhar o nome dos elementos que você não conhece e pesquisar - pode ser na própria [documentação](#). Uma outra dica é baixar uma imagem para substituir a que aparece no exemplo, para isto, coloque-a dentro da pasta IMC e troque "calc_guide.png" pelo nome da sua imagem.

Por fim, está tudo funcionando e já podemos fazer a implementação dessa aplicação, para isto, temos um recurso disponibilizado como *Streamlit sharing*, este se conecta diretamente com os projetos hospedados em um repositório público no *GitHub*, e os deixa prontos para serem compartilhados.

Atualmente para utilizar este recurso, ainda é necessário um convite para cadastrar uma conta, contudo já há pretensão de aumentar a proporção e o

suporte para que possa ser implementado em grandes volumes. Você encontra mais informações na documentação do próprio [Streamlit sharing](#).

Porém, caso queira hospedar sua aplicação com outro provedor de hospedagem, o Streamlit é extremamente versátil e suas aplicações funcionam em qualquer ambiente com suporte para Python.

Conclusão

Como mostrado, o Streamlit pode ser uma ferramenta muito útil para cientistas de dados e outros profissionais de dados, facilitando e agilizando - e muito! - o processo de criação de um *Data APP*, com zero custos e implementação extremamente simples.

Outros artigos para leitura:

1. Tiago Dias, **Modelos em produção com Streamlit**. Link: <https://dadosaocubo.com/modelos-em-producao-com-streamlit/>.
2. Café com bug, **Streamlit: Programe em Python e crie Aplicativos WEB**. Link: <https://www.youtube.com/watch?v=OWxlbvL2lo>.
3. Josue L. Gebrim, **Streamlit: A maneira mais rápida de criar e compartilhar aplicativos de dados!** Link: <https://jlgjosue.medium.com/streamlit-a-maneira-mais-rápida-de-criar-e-compartilhar-aplicativos-de-dados-f8e9f2bc98f9>.
4. OtwoO, **Ciência de Dados (Data Science), Deploy de Modelos e Streamlit.app**. Link: <https://www.otwoo.com.br/?p=1377>.
5. Charles Alves, **Streamlit**. Link: <https://charlesalves.com.br/2020/04/26/streamlit/#:~:text=O streamlit é um framework,de elaborar um data app>.

Documentações:

Documentação StreamLit: <https://docs.streamlit.io/en/stable/>

Criando uma aplicação de dados com Streamlit:

https://docs.streamlit.io/en/stable/tutorial/create_a_data_explorer_app.html

Implementando uma aplicação com Streamlit:

https://docs.streamlit.io/en/stable/deploy_streamlit_app.html

Questions:

- **Library ou framework?**

Streamlit is an open-source Python library **(Streamlit)**.

- **Goal:**

Makes it easy to create and share beautiful, custom web apps for machine learning and data science **(Streamlit)**.

- **Function:**

Deploy powerful data apps! **(Streamlit)**.