

Aula 9 - Streamlit I



Streamlit é um framework em python (pacote) para criar aplicativos interativos (<https://streamlit.io/>).

9.1 - Pacote

A biblioteca que usaremos para criar os aplicativos é o *Streamlit* (<https://docs.streamlit.io/>)

```
>> pip install streamlit
```

O streamlit é um aplicativo que usamos para executar arquivos fonte .py, de forma que *não usamos Streamlit junto com o Notebook*. Todos os códigos aqui devem ser transferidos para um arquivo .py. Para executar os arquivos usaremos o comando:

```
python -m streamlit run <caminho_do_arquivo>
```

Com isso, o Streamlit cria um aplicativo e mostra no explorador. Quando esse comando é executado, o terminal fica "travado" na aplicação Streamlit. Para fechar, não feche a aplicação, mas digite CTRL + C no terminal, que ele será interrompido. Após isso a aplicação pode ser fechada. Usaremos a convenção de importação do Streamlit como:

```
import streamlit as st
```

9.2 Textos

Podemos escrever textos nas aplicações usando a função `st.text()`, com o argumento sendo uma string que desejamos escrever. O Streamlit

escreve os textos obedecendo a sintaxe de Markdown (a mesma usada nos Notebooks!). Considere o código abaixo: (lembre de copiar e colar o código em um arquivo .py, e executar o streamlit como descrito na seção acima, nada acontece executando diretamente no Notebook).

```
In [ ]: import streamlit as st

st.write("""
# Primeira aplicação criada!
### Notem que o texto é em linguagem markdown
Vamos escrever uma fórmula latex :  $f(x) = x^2$ .
""")
```

Por conveniência, sempre que o streamlit encontra uma string no código, ele interpreta e a incapsula dentro da função `st.write()`, de forma que o código acima é equivalente a:

```
In [ ]: import streamlit as st

"""
# Não precisa estar em um write
### Qualquer string é interpretada na função
write.
"""
```

A função `st.text()` é um "canivete suíço", como o próprio site menciona. Qualquer coisa que seja passada é formatada e impressa de alguma forma. Considere a criação e impressão de um dataframe qualquer:

```
In [ ]: import pandas as pd
import streamlit as st

"""
# Não precisa estar em um write
### Podemos imprimir um dataframe.
"""

dt = pd.DataFrame({
    'coluna 1': [1, 2, 3, 4],
    'coluna 2': [10, 20, 30, 40]
})

dt
```

Podemos também usar a função própria para impressão de dataframes (ou estruturas como dataframes) do streamlit `st.dataframe()`

```
In [ ]: import streamlit as st
import numpy as np

dataframe = np.random.randn(10, 20)
st.dataframe(dataframe)
```

9.3 Gráficos e mapas

O streamlit suport gráficos do Matplotlib, Altair, deck e mais.

9.3.1 Gráfico de linhas

Para plotar um gráfico de linhas, basta usar a função `st.line_chart()`. No código abaixo criamos um dataframe com uma coluna e valores aleatórios da normal padrão, plotando em um gráfico de linhas interativo:

```
In [ ]: import streamlit as st
import numpy as np
import pandas as pd

dt = pd.DataFrame(
    np.random.randn(20),
    columns=['a'])

st.line_chart(dt)
```

Ao passarmos um dataframe com várias colunas, cada uma será uma série do gráfico de linhas. O código abaixo cria um dataframe com 3 colunas.

```
In [ ]: import streamlit as st
import numpy as np
import pandas as pd

dt = pd.DataFrame(
    np.random.randn(20,2),
    columns=['Coluna1', 'Coluna2'])

st.line_chart(dt)
```

9.3.2 Mapas

Plotar pontos em um mapa é extremamente fácil no Streamlit, basta criar um dataframe com as latitudes e longitudes dos pontos, e chamar a função `st.map()`. O código abaixo marca a UFPR centro em um mapa interativo:

```
In [ ]: import streamlit as st
import numpy as np
import pandas as pd

map_data = pd.DataFrame( [[-25.427024374829, -49.26159413862929]],
    columns=['lat', 'lon'])

st.map(map_data)
```

9.3.3 Gráfico do matplotlib

Podemos plotar qualquer gráfico do matplotlib usando a função `st.pyplot()`:

```
In [ ]: import streamlit as st
import numpy as np
```

```
import matplotlib.pyplot as plt

fig, ax = plt.subplots(1,1)
ax.plot([x for x in np.random.randint(5,10,20)])

st.pyplot(fig)
```

9.4 Widgets

Widgets são todos os componentes de uma interface gráfica, que possibilitam interação com o usuário: botões, *sliders*, caixas de input, etc. Com o streamlit também é extremamente intuitivo criar e receber o valor de widgets. Em streamlit os *widgets* são tratados como variáveis. A lista com todos os widgets pode ser consultada na documentação, ao lado esquerdo->Streamlit library-> API Reference-> Input widgets. Alguns são mostrados abaixo:

9.4.1 Slider

O *slider* é o componente que quando arrastado, gera um número entre um determinado intervalo. Toda vez que o slider é alterado, a página é atualizada.

```
In [ ]: import streamlit as st
x = st.slider('x') # Esse é o slider

...

# Alterando o valor com um slider
...

st.write("O valor do slider é de ", x)
st.write("Aplicando a função  $x^2$  temos :", x**2)
```

9.4.2 Botão

O botão recebe um argumento obrigatório com o seu nome. Verificamos o status do botão como `True` ou `False`, se ele foi pressionado ou não. Considere o código abaixo:

```
In [ ]: import streamlit as st
btn_ativar = st.button("Ativar")

if btn_ativar == True:
    st.write("Botão pressionado!")
```

9.4.3 File uploader

Esse componente permite que o usuário escolha um arquivo e carregue-o no aplicativo.

```
In [ ]: import streamlit as st
        uploaded_file = st.file_uploader("Choose a file")
```

Com o arquivo carregado, podemos usá-lo normalmente (lendo um .csv no pandas, por exemplo):

```
In [ ]: import streamlit as st
        import pandas as pd

        uploaded_file = st.file_uploader("Carregue um arquivo...")
        if uploaded_file is not None:
            dt = pd.read_csv(uploaded_file)
            dt
```

9.4.4 Radiobutton

O radiobutton dispõe possibilidades de escolha que são únicas. Seu valor é igual ao nome do botão selecionado:

```
In [ ]: import streamlit as st
        rdb_disc = st.radio(
            "Qual a sua disciplina favorita na faculdade?",
            ('PO I', 'Qualidade', 'Cálculo', 'Nenhuma'))

        if rdb_disc == 'PO I':
            st.write('Muito bom!')
        elif rdb_disc == 'Qualidade':
            st.write('Bem aplicado')
        elif rdb_disc == 'Cálculo':
            st.write('Difícil mas gratificante!')
        else:
            st.write("Está no seu direito!")
```

9.4.5 Input de texto

Simplemente recebe o input de texto do usuário.

```
In [ ]: import streamlit as st

        diario = st.text_input('Diário')
        st.write('Atualmente seu diário contém a informação :')
        st.write(diario)
```

Exercícios

1. Crie um aplicativo que receba dois valores do usuário, um valor de latitude e um de longitude. Em seguida, quando um botão for pressionado o ponto digitado pelo usuário é mostrado em um mapa.
2. Crie um aplicativo em que o usuário pode adicionar um texto. Quando um botão for clicado, uma tabela deve ser criada com a frequência de cada palavra do texto. A palavra com a maior frequência deve ser

informada ao usuário, bem como um gráfico de barras com as frequências por palavras criado.

3. Crie uma aplicação que leia um arquivo de dados do usuário, em seguida plote um gráfico (de linhas ou colunas), para cada coluna do conjunto de dados carregado.
4. Busque na documentação do streamlit 4 componentes dentre *Input widgets* e *Media elements*, e crie uma aplicação própria usando esses elementos.