

Streamlit Avançado: Multipáginas, Arquivos e Layouts Interativos

Explore técnicas avançadas para construir aplicações web interativas e sofisticadas com Python



O que é Streamlit? Um rápido resumo básico

Streamlit é uma biblioteca Python revolucionária que transforma scripts em aplicações web interativas com apenas algumas linhas de código.

Perfeito para cientistas de dados e desenvolvedores que desejam prototipar rapidamente sem conhecimento profundo de desenvolvimento web.

A filosofia é simples: escreva Python puro e veja a mágica acontecer!

Exemplo Mínimo

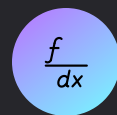
```
import streamlit as st

st.title("Olá, Streamlit!")
st.write("Meu primeiro app!")
```

Com estas duas linhas, você já tem uma aplicação web funcional rodando no navegador.



Conceitos Básicos na Prática



App de Cálculo de IMC

Utilize `st.number_input()` para capturar peso e altura, calcule o IMC e exiba resultados com `st.metric()` para visualização clara.



Lista de Tarefas

Combine `st.text_input()` para adicionar tarefas, `st.checkbox()` para marcar como concluído e `st.button()` para remover itens.



Visualização Instantânea

Use `st.line_chart()` e `st.bar_chart()` para transformar dados em gráficos interativos sem bibliotecas adicionais.

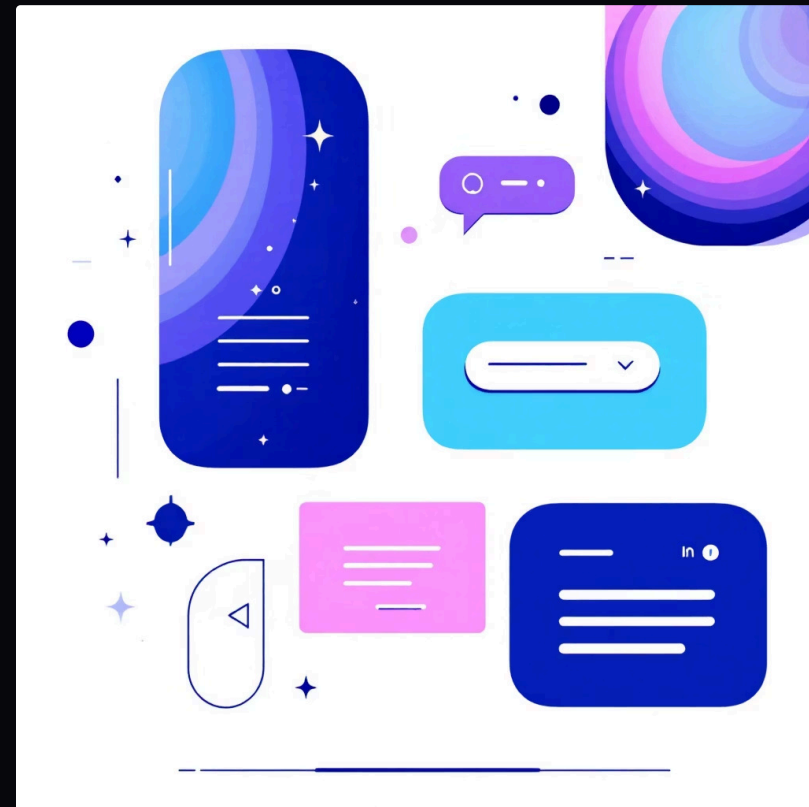
Expansão das Interfaces: Múltiplas Páginas

Aplicações reais exigem organização e separação de funcionalidades. O sistema de multipáginas do Streamlit permite criar apps complexos mantendo o código limpo e modular.

Cada página funciona como um módulo independente, facilitando manutenção e colaboração em equipe.

Vantagens das Multipáginas:

- Organização lógica do conteúdo
- Navegação intuitiva para usuários
- Facilita trabalho em equipe
- Código mais limpo e manutenível



Por que avançar além do básico?



Apps Simples têm Limites

Projetos reais demandam organização robusta e múltiplas funcionalidades integradas de forma coesa.



Multipáginas = Organização

Separar conteúdos e funcionalidades em páginas distintas facilita navegação e manutenção do código.



Dados Dinâmicos

Leitura de arquivos e visualização de dados são essenciais para aplicações que trabalham com informações reais e atualizadas.



Leitura de Arquivos no Streamlit

1

Upload Interativo

Use `st.file_uploader()` para permitir que usuários carreguem CSV, Excel, JSON ou imagens diretamente no navegador.

2

Processamento Imediato

Combine com `pandas` (`pd.read_csv()`) para processar dados instantaneamente e exibir resultados na interface.

3

Validação e Feedback

Implemente validação de formato e tamanho de arquivo, oferecendo mensagens claras ao usuário sobre o status do upload.

```
uploaded_file = st.file_uploader("Escolha um arquivo CSV", type="csv")
if uploaded_file is not None:
    df = pd.read_csv(uploaded_file)
    st.write(df)
```


Visualização Avançada de Tabelas

`st.dataframe()` – Interativa

Permite ordenação, busca e rolagem em grandes conjuntos de dados. Ideal para exploração.

`st.data_editor()` – Editável

Tabela com edição inline! Usuários podem modificar valores diretamente na interface.

`st.table()` – Estática

Renderização simples e rápida para tabelas pequenas que não precisam de interatividade.



Combine com formatação condicional e highlight de células para criar visualizações ainda mais impactantes usando a API do pandas styling.

Construção de Layouts Interativos

Colunas (st.columns)

Divida a tela horizontalmente para criar dashboards com múltiplos widgets lado a lado.

Expanders (st.expander)

Organize conteúdo em seções recolhíveis para manter a interface limpa e focada.

Sidebar (st.sidebar)

Barra lateral para controles e filtros, mantendo a área principal dedicada ao conteúdo.

Tabs (st.tabs)

Abas para alternar entre diferentes visualizações sem sair da página atual.

Mini-Sistema: App Multipágina Completo

Estrutura de Pastas

```
meu_app/  
├── app.py  
├── pages/  
└── 1_
```

