

# Introducción práctica a Streamlit

*(apps web ligeras para presentar resultados de datos)*

Dr. Marco A. Aquino-López

CIMAT Probabilidad y Estadística

Agosto 2025

# Agenda

- 1 Motivación
- 2 Primeros pasos
- 3 Componentes básicos
- 4 Despliegue
- 5 Buenas prácticas
- 6 Recursos

## ¿Por qué Streamlit?

- Permite construir **aplicaciones web interactivas** usando solo Python.
- Ideal para **prototipos rápidos** y **comunicar resultados** sin montar front-end.
- Integra fácil con pandas, numpy, matplotlib, plotly, altair, etc.
- Despliegue sencillo: local, *Streamlit Community Cloud* o servidores propios.

- **Dashboards** para reportes internos.
- **Exploración** de datos con controles (sliders, selectores).
- **Demos** de modelos estadísticos / ML.
- **Formularios** para recolectar parámetros y *qué-if* analyses.

# Instalación y ejecución

- Requiere Python 3.9+ (recomendado 3.10+ en entorno virtual).
- Instalar: `pip install streamlit`
- Ejecutar: `streamlit run app.py`
- Servidor local típico: `http://localhost:8501`

# Hola Mundo

```
# app.py
import streamlit as st

st.title("Hola, Streamlit ")
st.write("Esta es mi primera app web con Python.")

x = st.slider("Elige un número", 0, 100, 42)
st.write(f"Tu número al cuadrado es: {x**2}")
```

**Ejecuta:** streamlit run app.py

# Texto, layout y barra lateral

```
import streamlit as st

st.header("Resultados del análisis")
st.markdown("**Resumen:** estadísticas básicas del dataset.")

with st.sidebar:
    st.subheader("Controles")
    option = st.selectbox("Métrica", ["media", "mediana", "desv. est."])

col1, col2 = st.columns(2)
col1.metric("Observaciones", 500)
col2.metric("Variables", 12)
```

## Widgets frecuentes

```
st.number_input("Umbral", min_value=0.0, max_value=1.0, value=0.2)
st.date_input("Fecha de corte")
st.multiselect("Variables", ["x1","x2","x3"], default=["x1"])
if st.button("Recalcular"):
    st.success("Cálculo actualizado")
```



# Tablas y gráficos

```
import pandas as pd
import numpy as np
import altair as alt

# Datos de ejemplo
df = pd.DataFrame({"x": np.arange(100), "y": np.random.randn(100).cumsum()})

st.dataframe(df)

chart = alt.Chart(df).mark_line().encode(x="x", y="y")
st.altair_chart(chart, use_container_width=True)
```

## Carga y descarga de archivos

```
upload = st.file_uploader("Sube un CSV", type=["csv"])
if upload is not None:
    df = pd.read_csv(upload)
    st.dataframe(df.head())
    st.download_button("Descargar filtrado", df.to_csv(index=False), file_name="salida.csv"
    )
```

```
@st.cache_data
def cargar_datos(path):
    return pd.read_csv(path)

if "contador" not in st.session_state:
    st.session_state.contador = 0

if st.button("+1"):
    st.session_state.contador += 1

st.write("Clicks:", st.session_state.contador)
```

## Diseño: columnas, pestañas y expanders

```
col1, col2, col3 = st.columns([2,1,1])
with col1: st.write("Gráfico principal")
with col2: st.write("Controles")
with col3: st.write("KPIs")

p1, p2 = st.tabs(["Distribución", "Correlación"])
with p1: st.write("histograma...")
with p2: st.write("heatmap...")

with st.expander("Detalles técnicos"):
    st.code("Ecuaciones, supuestos, etc.")
```

- ❶ **Local:** compartir en la misma red (`-server.port` y `-server.address`).
- ❷ **Streamlit Community Cloud:** conectar un repo de GitHub y desplegar en minutos.
- ❸ **Servidores propios:** Docker + reverse proxy (nginx), o servicios como Fly.io, Render, etc.

# Estructura mínima del repositorio

- `app.py` (o varias apps)
- `requirements.txt` (streamlit, pandas, ...)
- `README.md` (cómo ejecutar)

- Separa lógica (`core.py`) de UI (`app.py`).
- Usa **validación** de entradas y maneja errores con mensajes claros.
- Diseña con layout **limpio** (columnas, tabs) y textos concisos.

- No fijar versiones en `requirements.txt`.
- Leer archivos grandes en cada interacción (sin caché).
- Suponer tipos de datos incorrectos (validar y castear columnas).



## Recursos recomendados

- Documentación oficial: *[docs.streamlit.io](https://docs.streamlit.io)*
- Galería de ejemplos: *[streamlit.io/gallery](https://streamlit.io/gallery)*

- Streamlit reduce la distancia entre **análisis** y **comunicación**.