

DESARROLLO DE PROYECTOS I

MAESTRÍA EN CIENCIAS DE LOS DATOS

CENTRO UNIVERSITARIO DE CIENCIAS ECONÓMICO-ADMINISTRATIVAS

**BARENCA SOTELO CRISTIAN ULISES
KINIL CERVERA EYDER URIEL**

STREAMLIT

MAESTRO

MTRO. VÍCTOR CUSPINERA

13 DE NOVIEMBRE DE 2024



Streamlit

Agenda

- Introducción
- Características
- Ventajas y desventajas
- Configuración básica
- Despliegue
- Demostración práctica.
- Casos de uso
- Integración con otras herramientas
- Conclusión y futuro
- Preguntas y respuestas



Streamlit

Introducción

Streamlit es un framework open-source para Python que permite crear aplicaciones e interfaces web interactivas. Permite convertir scripts en aplicaciones web sin necesidad de conocimientos avanzados en desarrollo frontend.

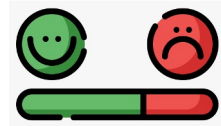
Características clave:

- **Simplicidad:** Puedes crear widgets interactivos con una sola línea de código.
- **Python-Céntrico:** Todo se programa en Python, desde la interfaz hasta la lógica.
- **Framework reactivo:** Ejecuta automáticamente el script cada vez que cambian los inputs.
- **Widgets integrados:** Incluye widgets esenciales como controles deslizantes, campos de texto, selectores de fecha y cargadores de archivos.
- **Visualización de datos:** Permite mostrar tablas, data frames, gráficos, mapas de manera sencilla.
- **Diseños personalizables:** Usa columnas y contenedores de elementos para crear diseños flexibles.



Streamlit

Ventajas y desventajas



Fácil de usar	Limitaciones en diseño y personalización avanzada
API intuitiva y centrada en Python	Escalabilidad limitada
Interactividad automática	Soporte limitado para usuarios concurrentes
Visualización de datos	Falta de soporte en funciones frontend complejas
Despliegue fácil y rápido	Dependencia de streamlit cloud para despliegue
Código limpio y reproducible	



Streamlit

Ventajas y desventajas

¿Cuándo utilizar Streamlit?

- Prototipos rápidos y aplicaciones de datos (Dashboard o Modelo ML de análisis rápido)
- Proyectos de pequeña escala (Manejo pequeño de tráfico y usuarios)

¿Cuándo considerar otras herramientas?

- Aplicaciones de producción a gran escala (Django)
- Diseños y personalizaciones específicas (HTML/CSS/JavaScript)



Streamlit

Configuración básica

1. Instalar Streamlit (Terminal)

```
pip install streamlit
```

1. Crear una app en Streamlit (crear script app.py)

```
import streamlit as st

st.title("¡Hola, Streamlit!")

st.write("Esta es tu primera aplicación en Streamlit.")
```

1. Ejecutar la aplicación

```
streamlit run app.py
```

Esto abrirá un servidor web local donde podrás ver tu aplicación en el navegador.

Profundizar en tutorial: <https://www.datacamp.com/es/tutorial/streamlit>

Despliegue



Una vez que tu aplicación esté lista, Streamlit ofrece Streamlit Community Cloud para despliegues gratuitos. También puedes usar otras plataformas como Heroku, AWS o GCP.

- **Streamlit Community Cloud:** Ideal para despliegues rápidos y gratuitos.
- **Heroku:** Buena opción para despliegues de prototipos en una nube pública con poca configuración.
- **AWS/GCP/Azure:** Adecuado para proyectos con necesidades avanzadas, escalabilidad y personalización.
- **Docker:** Asegura portabilidad y facilidad para despliegues en cualquier entorno compatible con Docker.

1. Despliegue con Streamlit Community Cloud (gratuito)

Sube tu código a GitHub:

- Asegúrate de que tu aplicación esté en un repositorio público de GitHub. También deberás incluir un archivo `requirements.txt` con las dependencias necesarias.

Vincula tu repositorio en Streamlit Community Cloud:

- Ve a [Streamlit Community Cloud](#) e inicia sesión con tu cuenta de GitHub.
- Selecciona el repositorio de tu aplicación y sigue las instrucciones para desplegarla.

Configura y despliega:

- Streamlit Community Cloud se encargará del resto del proceso. Después de unos momentos, tu aplicación estará disponible en una URL pública que podrás compartir.



Streamlit

Despliegue

2. Despliegue en un Servidor Privado o en una Nube Pública (Heroku, AWS, GCP, Azure)

Si necesitas más control o acceso privado, puedes desplegar tu aplicación en servidores como Heroku, Amazon Web Services (AWS), Google Cloud Platform (GCP) o Microsoft Azure.

Despliegue en Heroku

- 1. Instala el CLI de Heroku:**
 - Si no lo tienes, instala el CLI de Heroku y autentícate.
- 2. Configura tu aplicación:**
 - Asegúrate de tener un archivo **Procfile** en tu proyecto, el cual indique cómo Heroku debe ejecutar la app. Un ejemplo de **Procfile** para Streamlit:
web: streamlit run app.py --server.port=\$PORT --server.address=0.0.0.0
- 3. Sube el proyecto a Heroku:**
 - En la terminal, navega a tu proyecto y ejecuta: `heroku create nombre-de-tu-app; git push heroku main`
- 4. Abre la aplicación:**
 - Después de que Heroku termine el despliegue, abre la aplicación ejecutando **heroku open**.

Despliegue en AWS

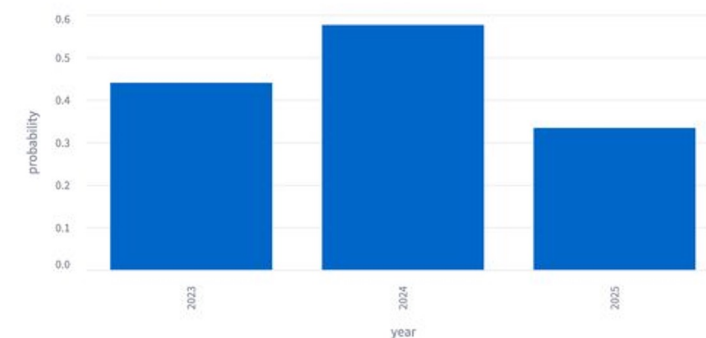
- 1. Configura una instancia EC2:**
 - En AWS EC2, crea una instancia (por ejemplo, Ubuntu) y configura la seguridad para permitir el puerto 8501 (puerto por defecto de Streamlit).
- 2. Configura tu instancia:**
 - Conéctate a la instancia, instala Python, pip, y las dependencias de tu aplicación (**requirements.txt**).
`streamlit run app.py --server.port=8501 --server.address=0.0.0.0`
- 3. Accede a la app:**
 - Accede a la IP pública de tu instancia en el puerto 8501 para ver la aplicación.

Demostración Práctica

```
1 import streamlit as st
2 import altair as alt
3 import pandas as pd
4 import numpy as np
5
6 st.title("Adrien Treuille: The Unofficial Singularity Predictor")
7
8 st.write("""
9 Welcome to the Unofficial Singularity Predictor! Here,
10 we're putting Adrien Treuille's LLM genius to the test by
11 predicting the rise of generative AI in the next 5 years.
12 Will we reach the singularity? That's what we're here to
13 find out!
14 """)
15
16 # Input for the bar chart
17 years_ahead_bar = st.sidebar.slider('Years Ahead for Bar Chart', 1, 5, 3)
18
19 # Generate some data
20 bar_data = pd.DataFrame({
21     'year': range(2023, 2023+years_ahead_bar),
22     'probability': np.random.rand(years_ahead_bar)
23     *years_ahead_bar/5
24 })
25
26 # Draw the chart
27 bar_chart = alt.Chart(bar_data).mark_bar().encode(
28     x='year:Q',
29     y='probability:Q',
30 )
31
32 st.altair_chart(bar_chart, use_container_width=True)
```

Adrien Treuille: The Unofficial Singularity Predictor

Welcome to the Unofficial Singularity Predictor! Here, we're putting Adrien Treuille's LLM genius to the test by predicting the rise of generative AI in the next 5 years. Will we reach the singularity? That's what we're here to find out!



This highly scientific bar chart shows the predicted probability of reaching the singularity within the next 3 years, according to Adrien Treuille's estimations.





Streamlit

Casos de uso

Visualización de datos: Crear dashboards interactivos para explorar y visualizar conjuntos de datos. (Gráficos y tablas para representar datos de manera intuitiva).

Prototipos de Machine Learning: Desarrollar aplicaciones que permiten a los usuarios interactuar con modelos de machine learning, como la carga de datos, la selección de parámetros y la visualización de resultados.

Análisis de datos: Crear herramientas de análisis que permiten a los usuarios realizar filtrados, agrupamientos y otras operaciones en tiempo real sobre los datos.

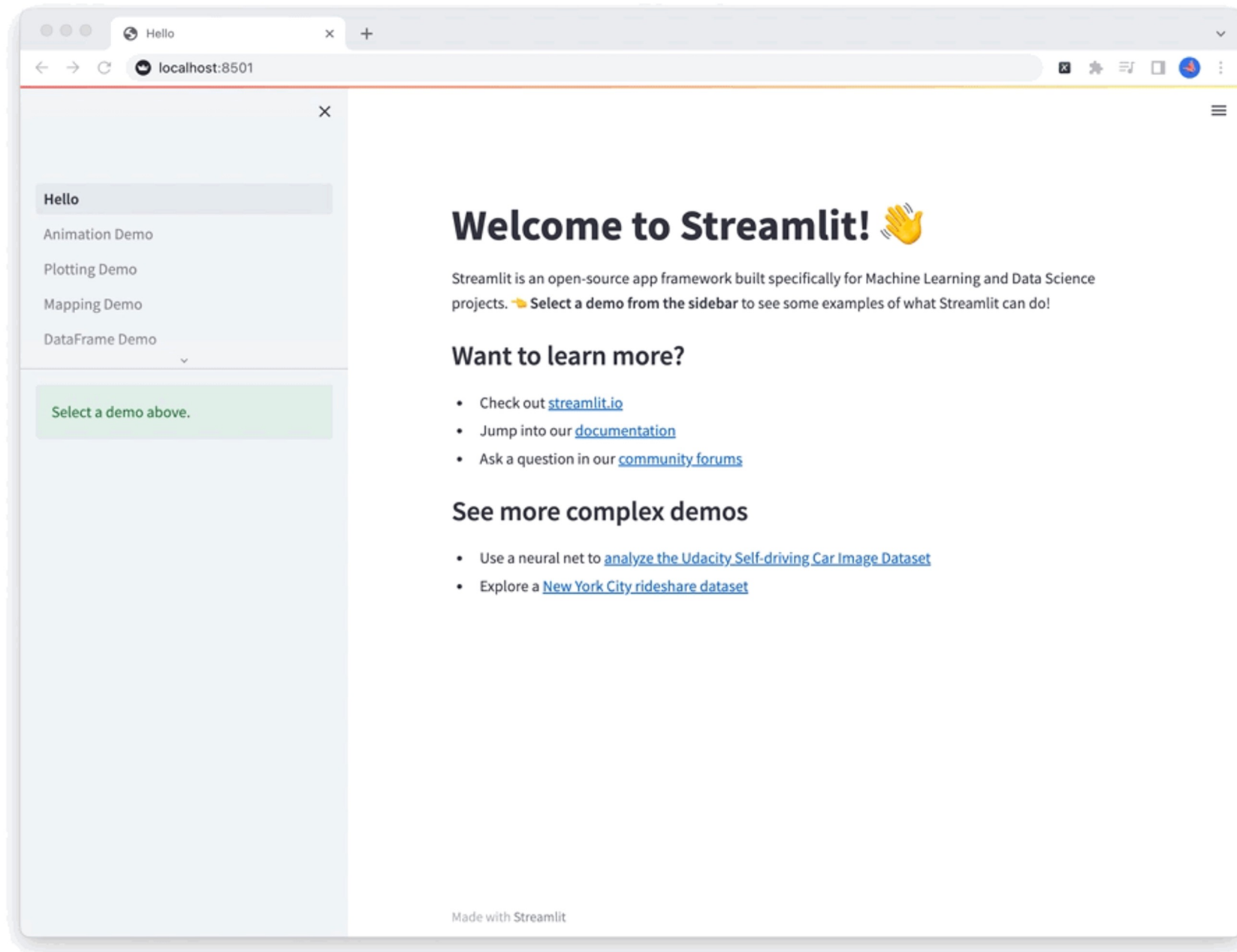


Streamlit

Casos de uso

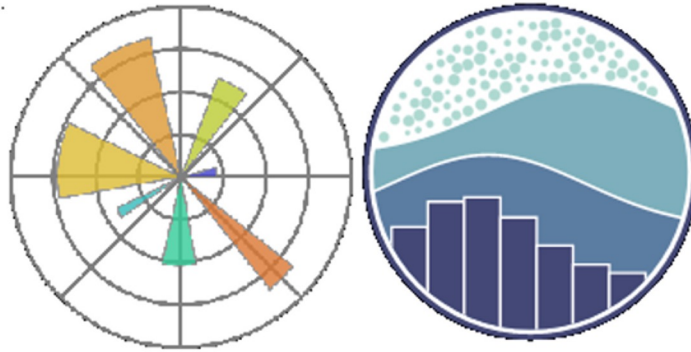
Informes interactivos: Generar informes que se pueden personalizar y actualizar dinámicamente según las entradas del usuario.

Formularios de entrada: Diseñar formularios para la recopilación de datos, donde los usuarios pueden ingresar información que se puede almacenar o procesar en tiempo real.

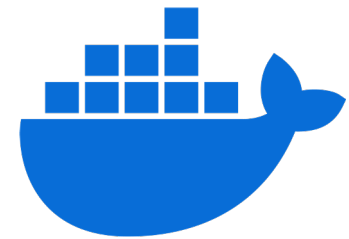


Streamlit

Integración con otras herramientas



MySQL™





Streamlit

Conclusión y futuro

El futuro de Streamlit se ve brillante, impulsado por el creciente interés en la ciencia de datos y machine learning, lo que llevará a una mayor adopción por parte de los desarrolladores que buscan crear aplicaciones interactivas.

La comunidad en constante crecimiento contribuirá a la incorporación de funcionalidades avanzadas, como soporte para autenticación y mejores herramientas de despliegue, mientras que la evolución hacia soluciones en la nube facilitará el acceso y la implementación en plataformas como AWS, GCP y Azure.

Search Results

[Advanced Search Tips](#)

[Ask Question](#)

Results for streamlit
Search options **not** deleted

5,332 results

Relevance Newest More

80 votes
✓ 11 answers
62k views
debugging intellij-idea pycharm ide streamlit
Ben 2,862 asked Feb 11, 2020 at 15:26

How to run/debug a streamlit application from an IDE

I really like **streamlit** as an environment for research. ... For this purpose, I was looking for a way to run or even debug a **streamlit** application, since the tutorials only show it being started via the command...

1 vote
✓ 3 answers
471 views
python streamlit python-3.12
Abdul Awal Nadim 97 asked Dec 4, 2023 at 5:37

'streamlit-card' FileNotFoundError when running Streamlit app

I'm having trouble using my **Streamlit** app because of a problem with the **streamlit-card** package. ...
streamlit-card --upgrade. ...

0 votes
3 answers
3k views
streamlit
Derrick Muduzie 1 asked Jun 12, 2023 at 8:54

Streamlit .streamlit/config.toml

I was trying to customize the theme of my **streamlit** app I tried the available method. creating a config.toml file inside a **.streamlit** folder in the directory of my streamlit_app but when i run the app ...



How do you feel about online communities?

Take our Knowledge Sharing Survey

[Take the survey](#)

[Report this ad](#)

JOIN TECH ENTHUSIASTS FROM AROUND THE WORLD

zoi

Zoi TechCon GmbH

Cloud Computing Private 201-500 people

vcuspiner/UDG_MCD_Project x Streamlit • A faster way to build x streamlit.pptx - Presentaciones x streamlit/streamlit: Streamlit— x

github.com/streamlit/streamlit

streamlit / streamlit

Code Issues 944 Pull requests 48 Actions Wiki Security 4 Insights

streamlit Public

Watch 320 Fork 3.1k Star 35.4k

develop 619 Branches 1871 Tags

Go to file Add file Code

sfc-gh-dmatthews Update docs for st.connection (#9661) 370b134 · 3 hours ago 6,755 Commits

.github	Enable jest test coverage and upload in CI workflow (#9448)	last month
.vscode	[chore] Add VSCode Launch file (#9497)	last month
component-lib	Bump micromatch from 4.0.5 to 4.0.8 in /component-lib (#9...	2 months ago
e2e_playwright	Revert the MPA Layout changes (#9757)	3 hours ago
frontend	Revert the MPA Layout changes (#9757)	3 hours ago
lib	Update docs for st.connection (#9661)	3 hours ago
proto/streamlit/proto	[Refactor] Metrics Provider Transition (#9729)	19 hours ago
scripts	Remove eval from frontend bundle (#9731)	5 days ago
.editorconfig	Specify max line length for Python file in .editorconfig to ma...	2 years ago
.gitignore	Re-introduce "Allow array parameters in CLI" (#9577)	28 days ago
.nvimrc	Fix JS Tests github actions (#7326)	last year
.pre-commit-config.yaml	Migrate iframe_resizer e2e test from Cypress to Playwright (...)	last month

About

Streamlit — A faster way to build and share data apps.

streamlit.io

python data-science machine-learning deep-learning data-visualization developer-tools data-analysis streamlit

Readme Apache-2.0 license Code of conduct Security policy Activity Custom properties 35.4k stars 320 watching 3.1k forks Report repository

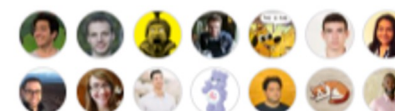
Releases 84

Used by 561k



+ 561,013

Contributors 243



+ 229 contributors

Deployments 201

nightly 20 hours ago

release 29 days ago

+ 199 deployments

Languages



Python 59.8% TypeScript 39.2%
JavaScript 0.5% Makefile 0.2%
HTML 0.2% SCSS 0.1%

Bibliografía



1. Streamlit Team. (n.d.). *Streamlit: A faster way to build and share data apps*. Streamlit.
<https://streamlit.io>
2. Streamlit Team. (n.d.). *Streamlit: The fastest way to build and share data apps*. GitHub.
<https://github.com/streamlit/streamlit>
3. Mendez, A. (2020). *Building web apps for data science with Streamlit*. Towards Data Science.
<https://towardsdatascience.com/building-web-apps-for-data-science-with-streamlit-8c82738d4b35>
4. Grinberg, M. (2021). *Streamlit for beginners: Build data apps in Python*. Real Python.
<https://realpython.com/streamlit/>
5. Laub, K. (2022). *Why Streamlit is the best tool for data scientists*. Medium.
<https://medium.com/@yourusername/why-streamlit-is-the-best-tool-for-data-scientists-123456>

Gracias

Para mayor información contacte a los correos

cristian.barenca1897@alumnos.udg.mx

eyder.kinil1047@alumnos.udg.mx



Streamlit