

IA PUCP - Diplomado de Desarrollo de Aplicaciones de Inteligencia Artificial
Python para Ciencia de Datos



Creación de Dashboards con Streamlit

Contenido

- Instalación de Anaconda Python
- `Streamlit` y su instalación
- Algunos componentes UI en `Streamlit`
- Git/Github
 - Crear Cuenta de Github
 - Crear repositorio y agregar colaboradores
 - Instalar Git
 - Clonar el Repositorio
 - Crear archivo de prueba y subir al repositorio
- Streamlit Cloud
 - Crear cuenta en Streamlit community cloud
 - Desplegar aplicación
- Caché y otras funcionalidades avanzadas

Anaconda Python

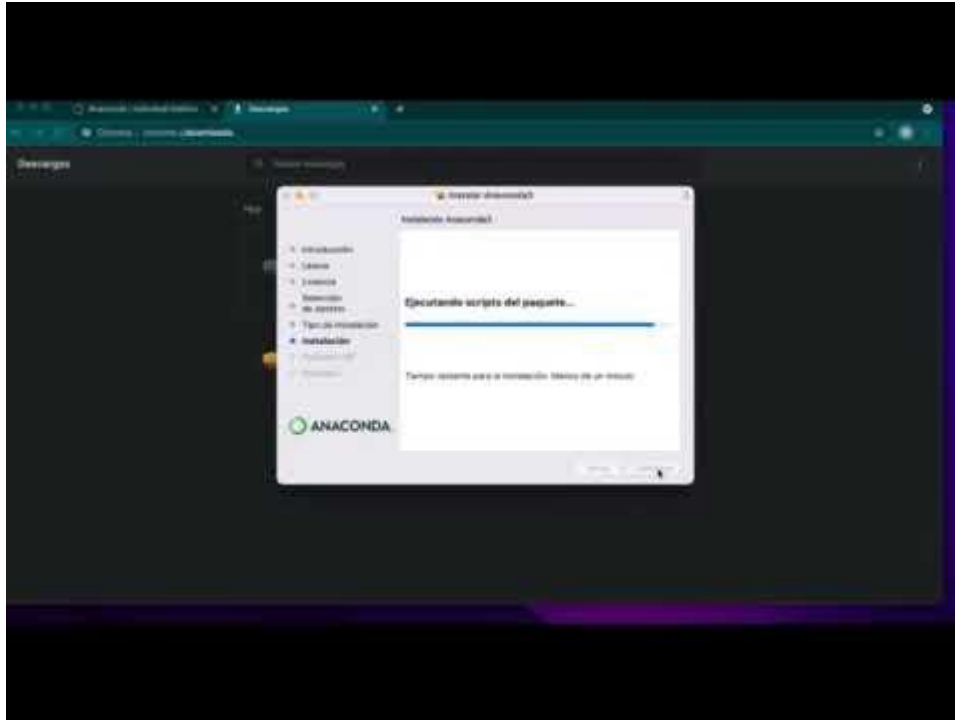
Anaconda es una distribución libre y abierta de los lenguajes Python y R, utilizada en ciencia de datos, y aprendizaje automático. Esto incluye procesamiento de grandes volúmenes de información, análisis predictivo y cómputos científicos.

[https://es.wikipedia.org/wiki/Anaconda_\(distribuci%C3%B3n_de_Python\)](https://es.wikipedia.org/wiki/Anaconda_(distribuci%C3%B3n_de_Python))

Anaconda Python: Instalación en Windows



Anaconda Python: Instalación en Mac OS X

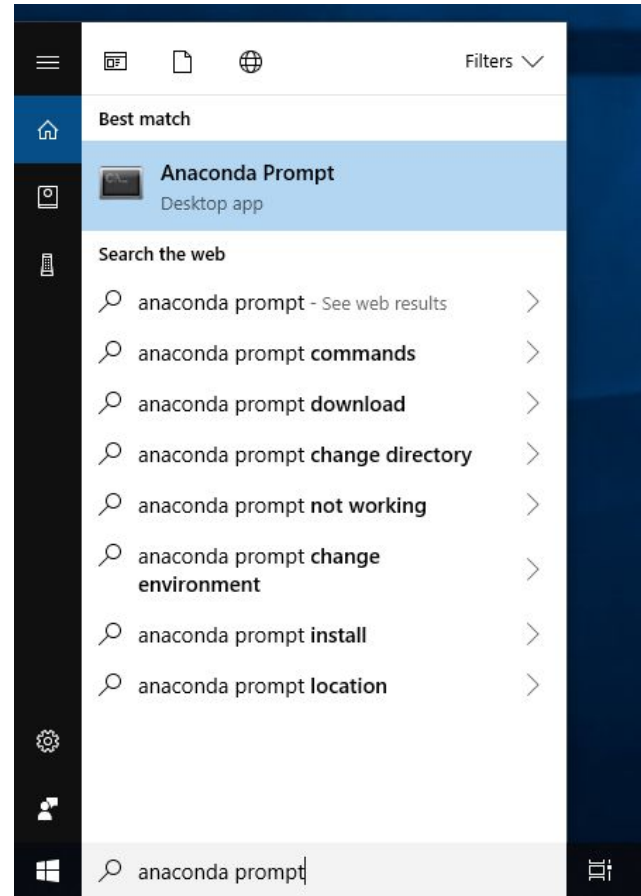


<https://www.youtube.com/watch?v=dTIW4geBti0>

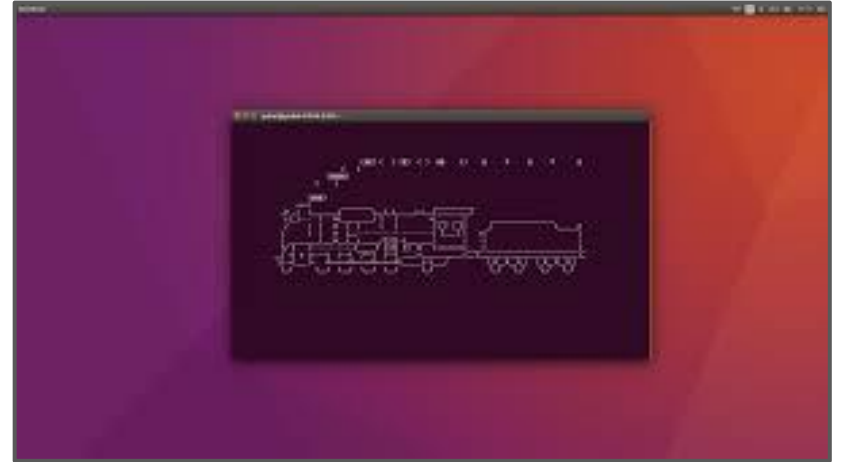
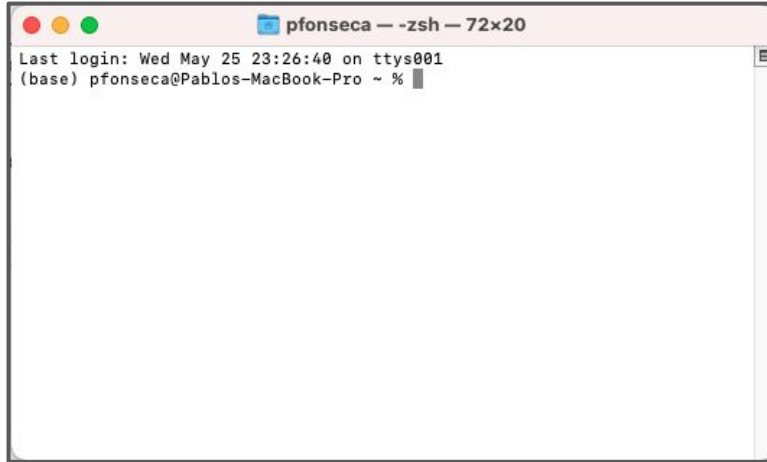
Anaconda Prompt: Windows



<https://docs.anaconda.com/anaconda/user-guide/getting-started/>



Anaconda Prompt: Linux & MacOSX (solo abrir el terminal)



<https://docs.anaconda.com/anaconda/user-guide/getting-started/>

Streamlit: Instalación

```
pip install streamlit
```

Realizar esto en Anaconda Prompt

Streamlit: Instalación

```
streamlit hello
```

Verificar la instalación

Streamlit: Instalación

```
streamlit hello
```

Podemos ingresar
nuestro correo

```
(base) pfonseca@Pablos-MacBook-Pro ~ % streamlit hello
```



Welcome to Streamlit!

If you're one of our development partners or you're interested in getting personal technical support or Streamlit updates, please enter your email address below. Otherwise, you may leave the field blank.

Streamlit: Instalación

streamlit hello

Se iniciará el servidor web

```
(base) pfonseca@Pablos-MacBook-Pro ~ % streamlit hello
```

👋 **Welcome to Streamlit!**

If you're one of our development partners or you're interested in getting personal technical support or Streamlit updates, please enter your email address below. Otherwise, you may leave the field blank.

Privacy Policy:

As an open source project, we collect usage statistics. We cannot see and do not store information contained in Streamlit apps. You can find out more by reading our privacy policy at: <https://streamlit.io/privacy-policy>

If you'd like to opt out of usage statistics, add the following to `~/.streamlit/config.toml`, creating that file if necessary:

```
[browser]
gatherUsageStats = false
```

Welcome to Streamlit. Check out our demo in your browser.

Local URL: <http://localhost:8501>

Network URL: <http://192.168.100.101:8501>

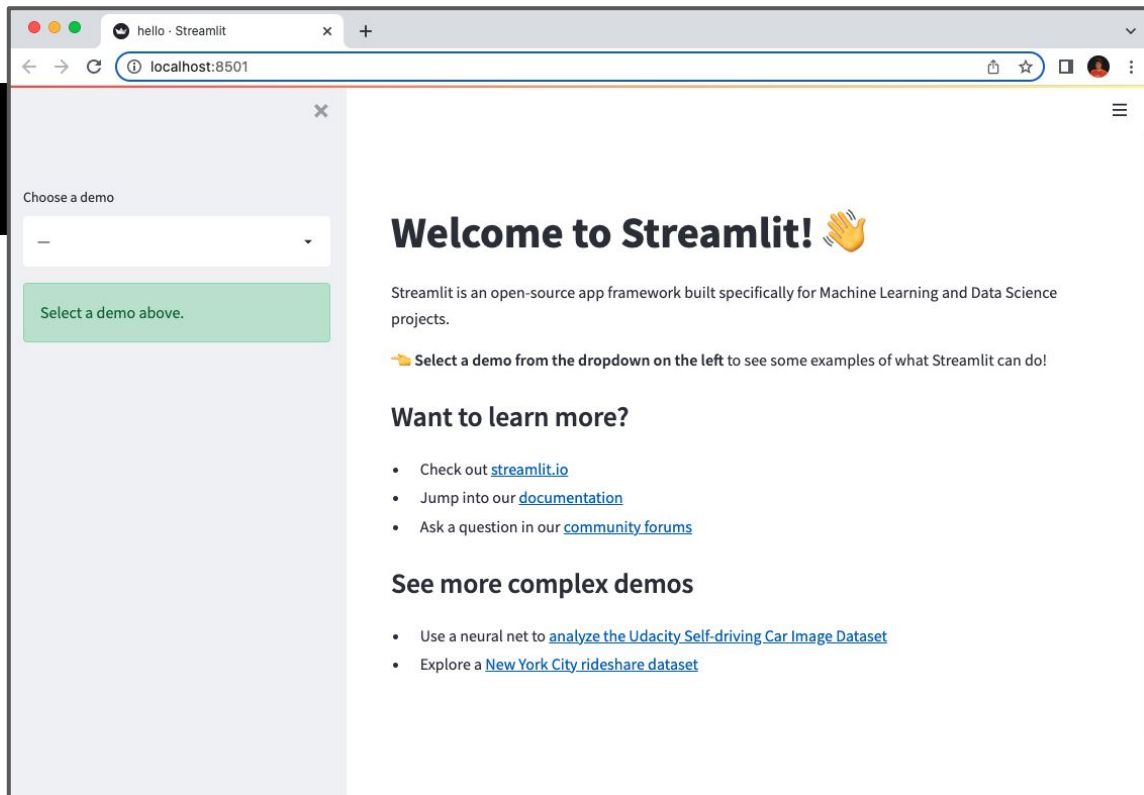
Ready to create your own Python apps super quickly?
Head over to <https://docs.streamlit.io>

May you create awesome apps!

Streamlit: Instalación

```
streamlit hello
```

Y podremos visualizar la app en un navegador y probar los ejemplos!



Nuevo Proyecto

Streamlit: Nuevo Proyecto

myfile.py

```
import streamlit as st
import pandas as pd
import numpy as np

st.title('Titulo del Proyecto')
```

Streamlit: Nuevo Proyecto

```
streamlit run myfile.py
```

Ejecutar un nuevo proyecto (previa creación del archivo `myfile.py`)

Streamlit: Nuevo Proyecto

```
streamlit run
```

Ejecutar un nuevo proyecto



Streamlit: Cambios en el código fuente

myfile.py

```
import streamlit as st
import pandas as pd
import numpy as np

st.title('Titulo')
```



Source file changed.

Rerun

Always rerun



Streamlit notificará que el código ha cambiado. No será necesario reiniciar el servidor.

Elementos de UI

Streamlit: Texto

myfile.py

```
import streamlit as st
import pandas as pd
import numpy as np

st.title('Titulo del Proyecto')
st.write('Hola **como** estas')
```

Streamlit: Texto

```
import streamlit as st
import pandas as pd
import numpy as np

st.title('Titulo del
st.write('Hola **como
```

st.write usa markdown para el formato



Streamlit: Texto

myfile.py

```
import streamlit as st
import pandas as pd
import numpy as np

st.title('Titulo del Pr
st.write('Hola **como**')
```

- st.markdown
- st.title
- st.header
- st.subheader
- st.caption
- st.code
- st.text
- st.latex

Hay otros elementos de texto

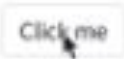











<https://docs.streamlit.io/library/api-reference/text>





Streamlit: Entrada (Input)

Los elementos de entrada en `streamlit` se declaran como “variables” que al cambiar su valor hacen que se re-ejecute el código

<https://docs.streamlit.io/library/api-reference/widgets>

Streamlit: Entrada (Input)

 <p>Button Display a button widget.</p> <pre>clicked = st.button("Click me")</pre>	 <p>Download button Display a download button widget.</p> <pre>st.download_button("Download")</pre>	 <p>Checkbox Display a checkbox widget.</p> <pre>selected = st.checkbox("I agree")</pre>	 <p>Slider Display a slider widget.</p> <pre>number = st.slider("Pick a number")</pre>	 <p>Select-slider Display a slider widget to select items from a list.</p> <pre>size = st.select_slider("Pick a size")</pre>	 <p>Text input Display a single-line text input widget.</p> <pre>name = st.text_input("First name")</pre>
 <p>Radio Display a radio button widget.</p> <pre>choice = st.radio("Pick one", ["Dog", "Cat", "Goldfish"])</pre>	 <p>Selectbox Display a select widget.</p> <pre>choice = st.selectbox("Pick a choice")</pre>	 <p>Multiselect Display a multiselect widget. The multiselect widget starts as empty.</p> <pre>choices = st.multiselect("Pick choices")</pre>	 <p>Number input Display a numeric input widget.</p> <pre>choice = st.number_input("Pick a number")</pre>	 <p>Text-area Display a multi-line text input widget.</p> <pre>text = st.text_area("Text to enter")</pre>	 <p>Date input Display a date input widget.</p> <pre>date = st.date_input("Your birthday")</pre>

 <p>Color picker Display a color picker widget.</p> <pre>color = st.color_picker("Pick a color")</pre>	 <p>Time input Display a time input widget.</p> <pre>time = st.time_input("Pick a time")</pre>	 <p>File Uploader Display a file uploader widget.</p> <pre>data = st.file_uploader("Upload a file")</pre>	 <p>Camera input Display a widget that allows users to upload images directly from a camera.</p> <pre>image = st.camera_input("Take a photo")</pre>
---	---	--	--

Streamlit: Slider



Streamlit Shorts

How to make a slider



<https://youtu.be/tzAdd-MuWPw>

Streamlit: Slider

myfile.py

```
import streamlit as st
import pandas as pd
import numpy as np

st.title("Titulo")
num = st.slider("num", 0, 100, step=1)
st.write("El numero ingresado es {}".format(num))
```

Streamlit: Slider

```
import
```

```
import
```

```
import
```

```
st.title
```

```
num =
```

```
st.write
```

Function signature

```
st.slider(label, min_value=None, max_value=None, value=None, step=None,
format=None, key=None, help=None, on_change=None, args=None, kwargs=None, *,
disabled=False)
```

Parameters

label (str)	
	A short label explaining to the user what this slider is for.

Streamlit: Slider

myfile.py

```
import streamlit as st
import pandas as pd
import numpy as np

st.title("Titulo")
num = st.slider("num", 0, 10)
st.write("El numero ingresado es: " + str(num))
```

- int
- float
- date
- time
- datetime

Soporta varios tipos de datos

Streamlit: Slider (con inicio y final!)



Streamlit Shorts

How to make a double-ended slider



<https://youtu.be/sCvdt79asrE>

Streamlit: Slider de horario

myfile.py

```
import streamlit as st
import pandas as pd
import numpy as np
from datetime import time

appointment = st.slider(
    "Programe la asesoria:",
    value=(time(11, 30), time(12, 45)))
st.write("Esta agendado para:", appointment)
```

Streamlit: Slider de horario

myfile.py

```
import streamlit as st
import pandas as pd
import numpy as np
from datetime import datetime

def main():
    title = "Programa de Asesoría"
    st.write(title)

    # Slider de horario
    start, end = st.slider(
        "Programe la asesoría:",
        0, 24,
        (11.5, 15.5),
        format="%H:%M"
    )

    # Mostrar el horario seleccionado
    st.write(f"Esta agendado para: {datetime.time(11, 30), datetime.time(15, 15)}")
```



Streamlit: Slider de fecha

myfile.py

```
import streamlit as st
import pandas as pd
import numpy as np
from datetime import datetime
start_time = st.slider(
    "Ver casos ocurridos en",
    value=datetime(2020, 1, 1, 9, 30),
    format="DD/MM/YY - hh:mm")
st.write("Fecha seleccionada:", start_time)
```

Streamlit: Slider de fecha



Streamlit: Ingreso de fecha con calendario

myfile.py

```
import streamlit as st
import pandas as pd
import numpy as np
import datetime
d = st.date_input(
    "Fecha de cumpleaños",
    datetime.date(2019, 7, 6))
st.write('Tu cumpleaños es:', d)
```

Streamlit: Ingreso de fecha con calendario

```
import streamlit as st
import pandas as pd
import numpy as np
import datetime
d = st.date_input(
    "Fecha de cumpleaños"
    datetime.date(2019
st.write('Tu cumpleaños
```

myfile · Streamlit

localhost:8501

Fecha de cumpleaños

2019/07/06

← July 2019 →

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

py

Streamlit: Lista de selección

myfile.py

```
import streamlit as st

option = st.selectbox(
    '¿Cómo desearía ser contactado/a?',
    ('Email', 'Teléfono', 'Whatsapp'))

st.write('Seleccionó:', option)
```

Streamlit: Lista de selección

myfile.py

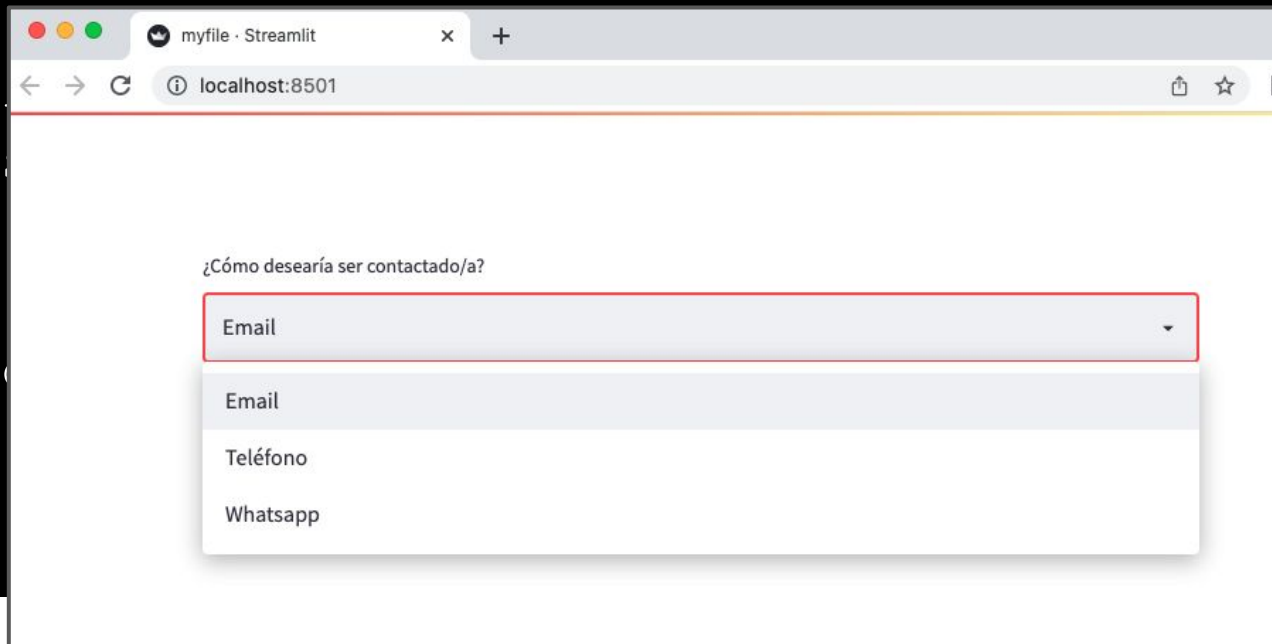
```
import streamlit as st
```

```
option = st.selectbox
```

```
    '¿Cómo de
```

```
    ('Email',
```

```
st.write('Sele
```



Function signature

```
st.selectbox(label, options, index=0, format_func=special_internal_function,  
key=None, help=None, on_change=None, args=None, kwargs=None, *,  
disabled=False)
```

Parameters

label (*str*)

A short label explaining to the user what this select widget is for.

options (*Sequence,
numpy.ndarray,
pandas.Series,
pandas.DataFrame,
or pandas.Index*)

Labels for the select options. This will be cast to str internally by default. For pandas.DataFrame, the first column is selected.

index (*int*)

The index of the preselected option on first render.

Streamlit: Lista de selección

myfile.py

```
import streamlit as st
import pandas as pd
import numpy as np

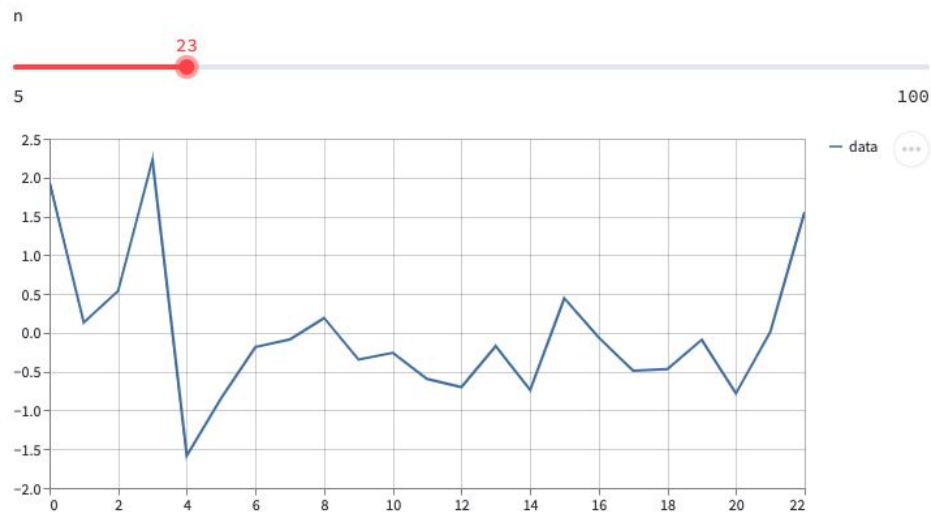
n = st.slider("n", 5, 100, step=1)
chart_data = pd.DataFrame(np.random.randn(n), columns=['data'])
st.line_chart(chart_data)
```

Streamlit: Lista de selección

myfile.py

```
import streamlit as st
import pandas as pd
import numpy as np
```

```
n = st.slider("n", 5, 100)
chart_data = pd.DataFrame(
    st.line_chart(chart_data
```



Streamlit: Lista de selección

myfile.py

```
import streamlit as st
import pandas as pd
import numpy as np

df = pd.DataFrame(
    np.random.randn(1000, 2) / [50, 50] + [37.76, -122.4],
    columns=['lat', 'lon'])

st.map(df)
```

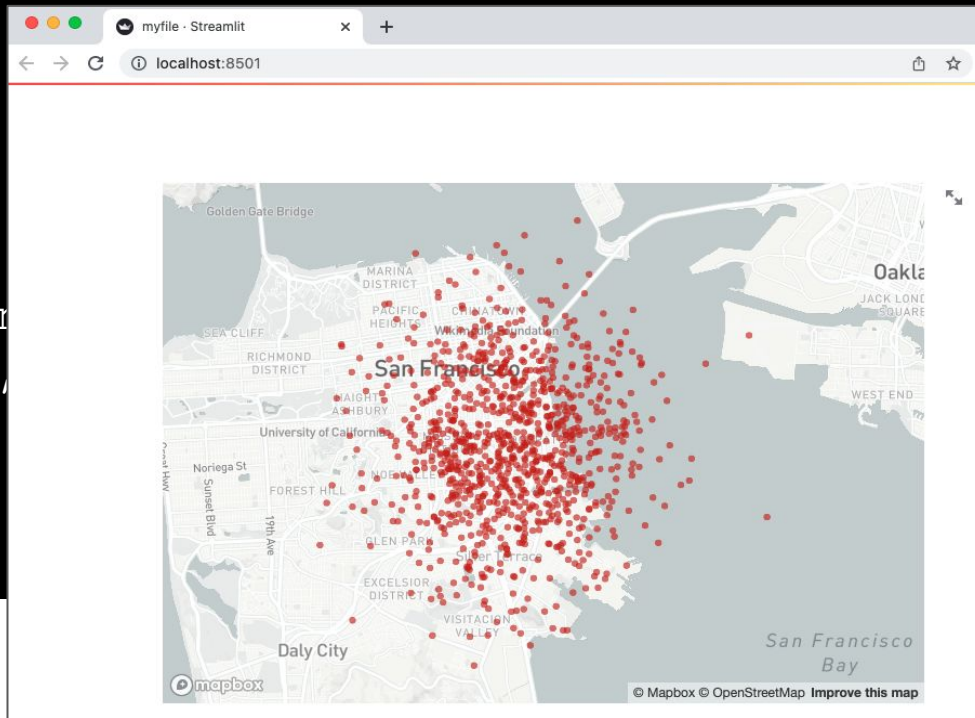

Streamlit: Lista de selección

myfile.py

```
import streamlit as st
import pandas as pd
import numpy as np
```

```
df = pd.DataFrame(
    np.random.randn(
        1000, 2),
    columns=['lat', 'lon'])
```

```
st.map(df)
```



22.4],

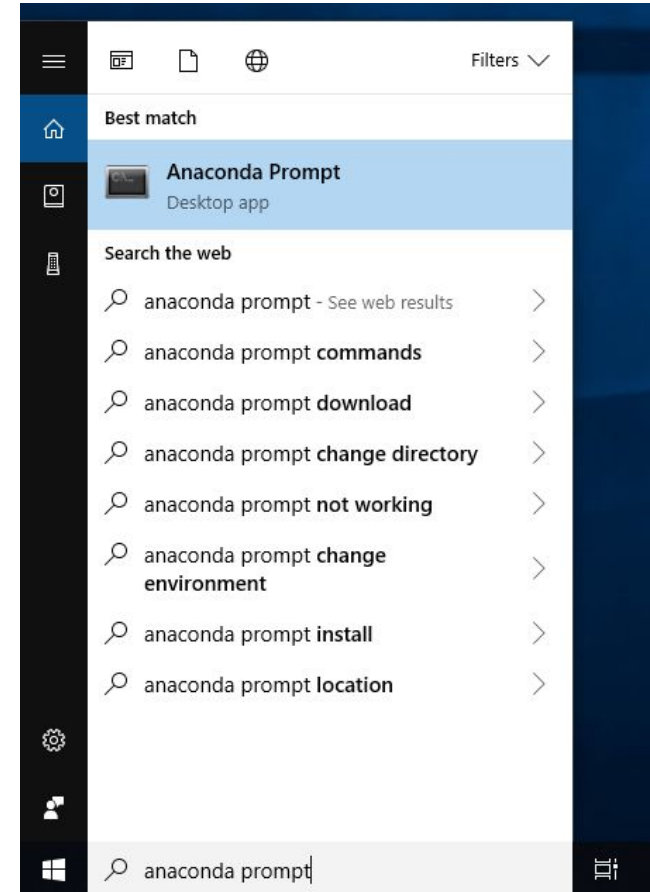
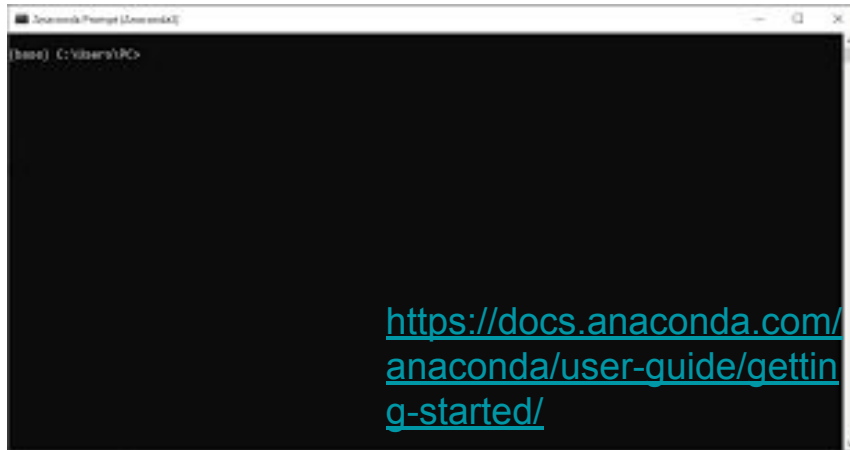
Git: Instalación con conda

```
conda install -c anaconda git
```

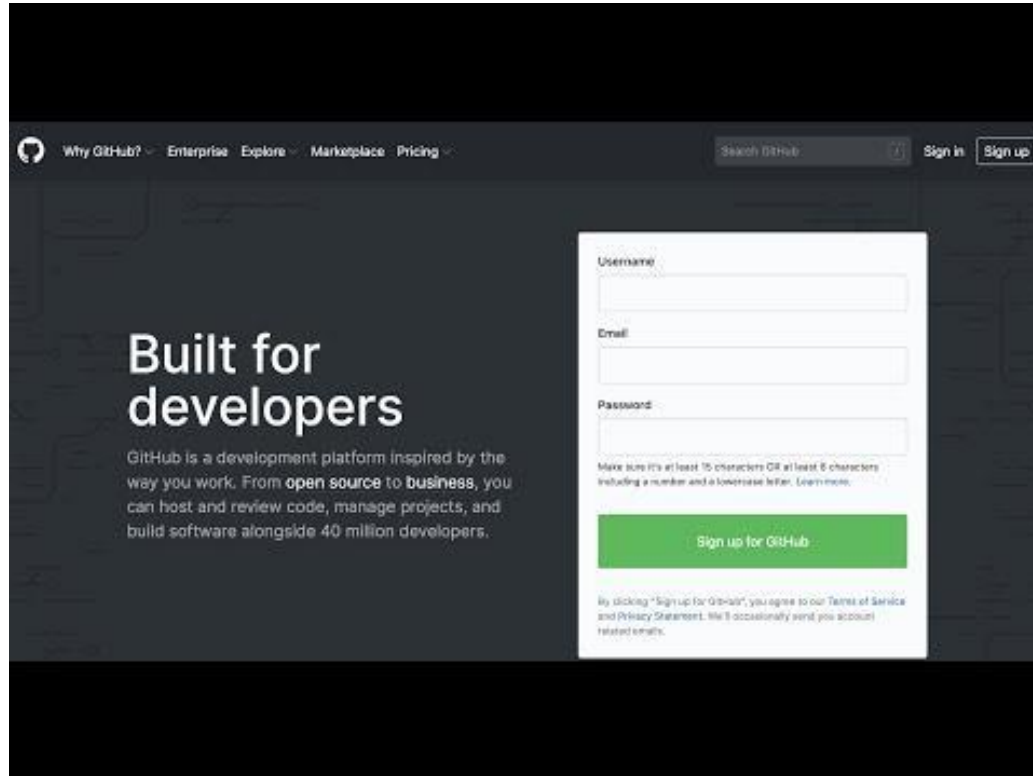
<https://anaconda.org/anaconda/git>

Pre-requisito

- Tenemos Streamlit funcionando localmente con una instalación de Anaconda
- Tenemos acceso a Anaconda prompt en Windows o al terminal en Mac OSX o Linux



Crear Cuenta de <https://github.com/>

A screenshot of the GitHub website's sign-up page. The background is dark with a faint circuit-like pattern. On the left, the text 'Built for developers' is prominently displayed in white, followed by a paragraph describing GitHub as a development platform for open source and business. On the right, a white sign-up form is centered. The form includes input fields for 'Username', 'Email', and 'Password', each with a small eye icon for toggling visibility. Below the password field is a note about password requirements: 'Make sure it's at least 10 characters OR at least 8 characters including a number and a lowercase letter. Learn more.' A green 'Sign up for GitHub' button is positioned below the form. At the bottom of the form, a small disclaimer states: 'By clicking "Sign up for GitHub", you agree to our Terms of Service and Privacy Statement. We'll occasionally send you account related emails.'

Crear nuevo repositorio


- Que sea un repositorio público
- Opcionalmente se puede agregar:
 - Archivo README
 - Template de .gitignore
 - Licencia

github.com/new

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner * / Repository name *

 palefo / streamlit_app ✓

Great repository names are short and memorable. Need inspiration? How about [glowing-bassoon?](#)

Description (optional)

App para el tutorial de Streamlit

☒ Public
Anyone on the internet can see this repository. You choose who can commit.

☐ Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.


☒ Add a README file
This is where you can write a long description for your project. [Learn more.](#)


Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: Python ▼

Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

License: Apache License 2.0 ▼

This will set  main as the default branch. Change the default name in your [settings](#).

 You are creating a public repository in your personal account.

Create repository

Crear nuevo repositorio

The screenshot shows a new GitHub repository page for 'palefo / streamlit_app'. The repository is public and has 1 branch (main) and 0 tags. The commit history shows an initial commit by 'palefo' 5 hours ago, with files .gitignore, LICENSE, and README.md. The README.md file is displayed, showing the repository name 'streamlit_app' and a description 'App para el tutorial de Streamlit'. The right sidebar contains sections for 'About', 'Releases', and 'Packages', all indicating no published content yet.

Search or jump to... Pull requests Issues Marketplace Explore

palefo / streamlit_app Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags

Go to file Add file Code

palefo Initial commit c2cfd77 5 hours ago 1 commit

.gitignore	Initial commit	5 hours ago
LICENSE	Initial commit	5 hours ago
README.md	Initial commit	5 hours ago

README.md

streamlit_app

App para el tutorial de Streamlit

About

App para el tutorial de Streamlit

- Readme
- 0 stars
- 1 watching
- 0 forks

Releases

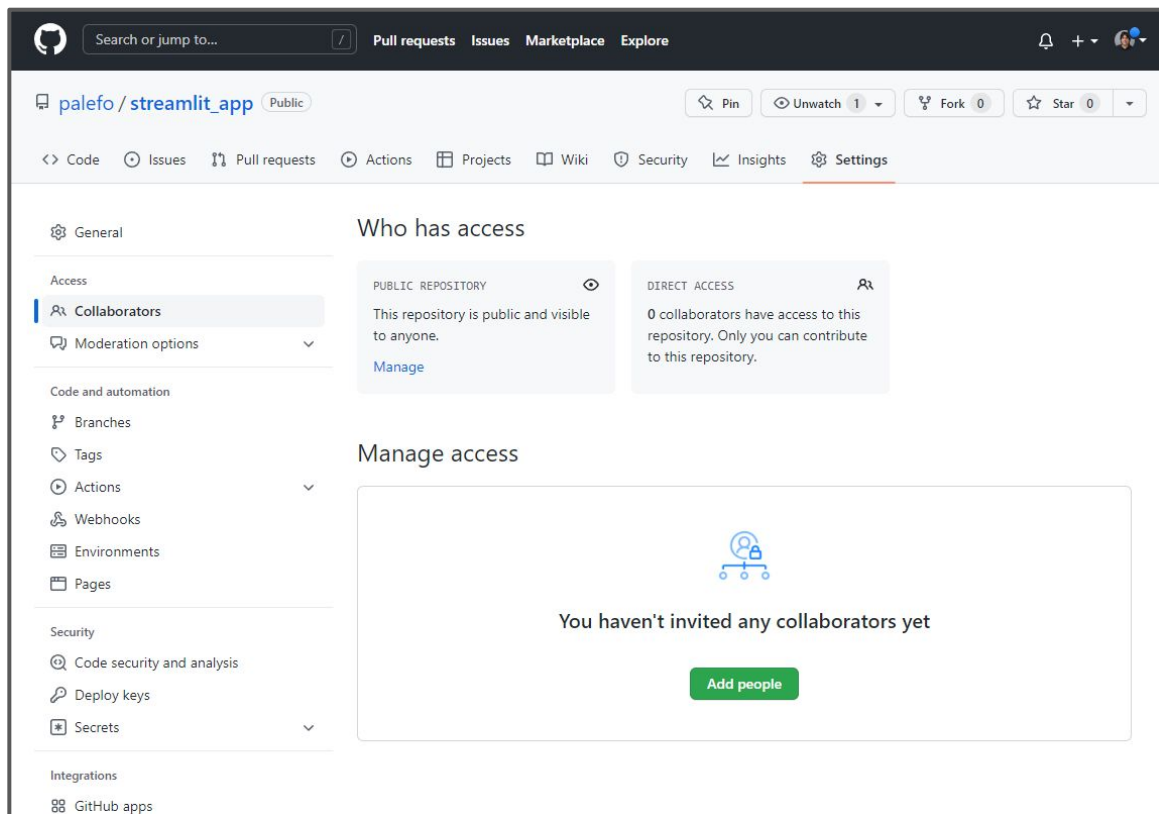
No releases published
[Create a new release](#)

Packages

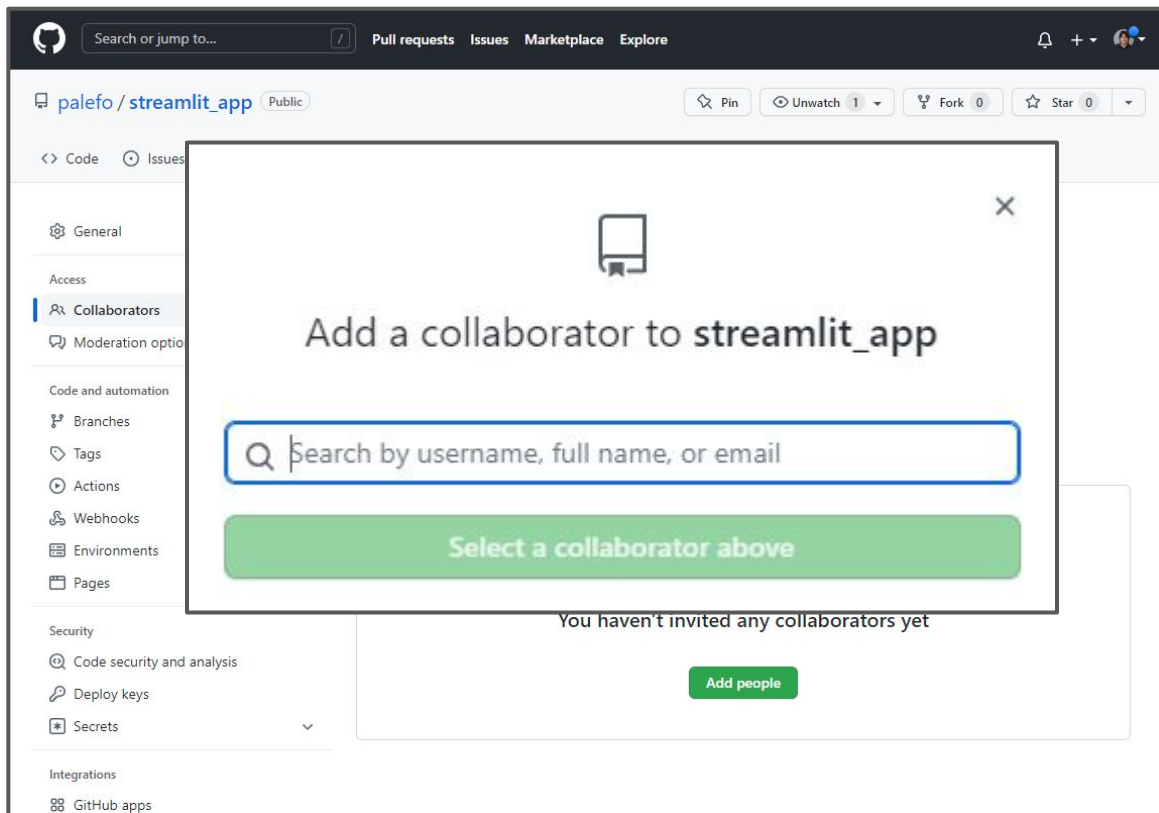
No packages published
[Publish your first package](#)

© 2022 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

Agregar colaboradores: Settings → Collaborators → Add People



Agregar colaboradores: Settings → Collaborators → Add People



Git: Instalación con conda

```
conda install -c anaconda git
```

```
Anaconda Prompt (anaconda3)

-----
certifi-2021.10.8      | py39haa95532_2      | 156 KB | anaconda
conda-4.12.0           | py39haa95532_0      | 17.0 MB | anaconda
git-2.34.1             | haa95532_0          | 95.5 MB | anaconda
openssl-1.1.1n         | h2bbff1b_0          | 5.8 MB | anaconda
-----
Total: 118.4 MB

The following NEW packages will be INSTALLED:

git                                anaconda/win-64::git-2.34.1-haa95532_0

The following packages will be SUPERSEDED by a higher-priority channel:

certifi                            pkgs/main --> anaconda
conda                             pkgs/main --> anaconda
openssl                           pkgs/main --> anaconda

Proceed ([y]/n)? y

Downloading and Extracting Packages
openssl-1.1.1n      | 5.8 MB | ##### | 100%
git-2.34.1         | 95.5 MB | ##### | 100%
conda-4.12.0       | 17.0 MB | ##### | 100%
certifi-2021.10.8  | 156 KB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(base) C:\Users\Usuario>
```

<https://anaconda.org/anaconda/git>

Clonar repositorio localmente

The screenshot shows the GitHub interface for the repository 'palefo/streamlit_app'. The repository is public and has 1 branch and 0 tags. The 'Clone' dropdown menu is open, displaying three options: HTTPS, SSH, and GitHub CLI. The HTTPS URL, 'https://github.com/palefo/streamlit_app.g', is highlighted with a yellow box. Below the URL, there is a note: 'Use Git or checkout with SVN using the web URL.' Other options in the dropdown include 'Open with GitHub Desktop' and 'Download ZIP'.

Repository: palefo / streamlit_app (Public)

Actions: Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, Settings

Branches: main (1 branch), 0 tags

Files:

- palefo Initial commit
- .gitignore Initial commit
- LICENSE Initial commit
- README.md Initial commit

Clone options:

- HTTPS: https://github.com/palefo/streamlit_app.g (highlighted)
- SSH
- GitHub CLI

Use Git or checkout with SVN using the web URL.

Other options:

- Open with GitHub Desktop
- Download ZIP

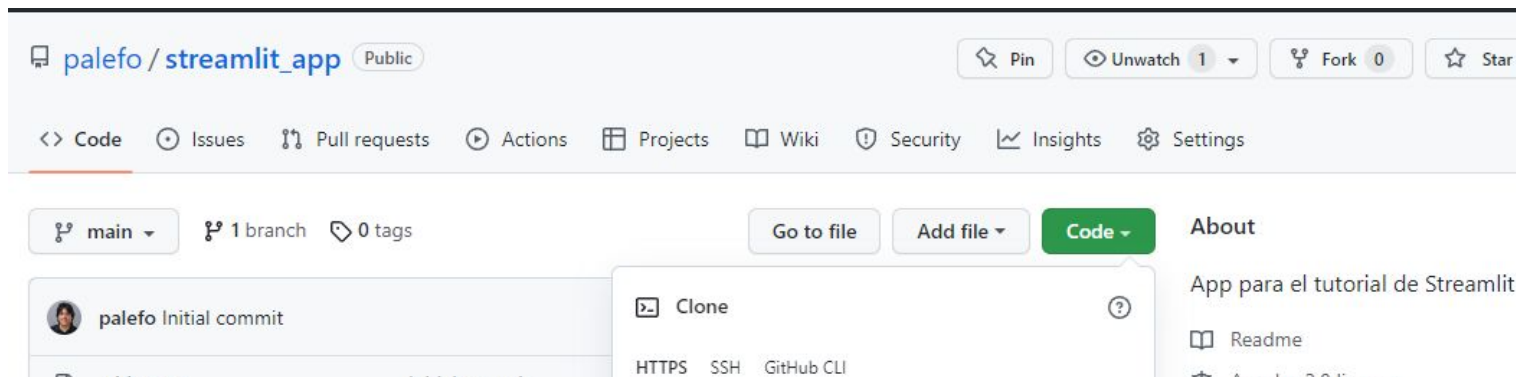
About:

- App para el tutorial de Streamlit
- Readme
- Apache-2.0 license
- 0 stars
- 1 watching
- 0 forks

Releases:

- No releases published
- [Create a new release](#)

Clonar repositorio localmente

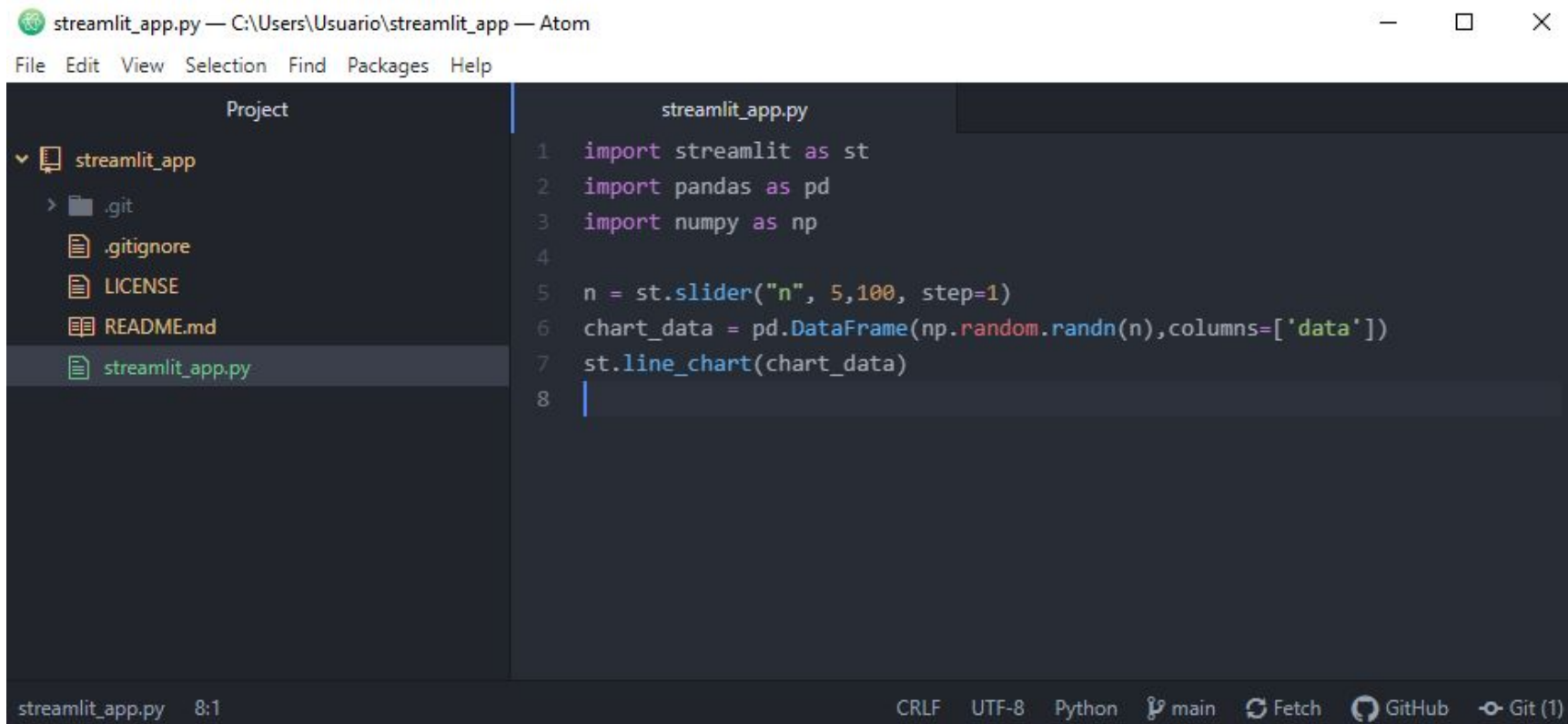


```
Anaconda Prompt (anaconda3)

(base) C:\Users\Usuario>git clone https://github.com/palefo/streamlit_app.git
Cloning into 'streamlit_app'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), 5.50 KiB | 5.50 MiB/s, done.

(base) C:\Users\Usuario>
```

Crear nuevo archivo



streamlit_app.py — C:\Users\Usuario\streamlit_app — Atom

File Edit View Selection Find Packages Help

Project

- ▼ streamlit_app
 - > .git
 - .gitignore
 - LICENSE
 - README.md
 - streamlit_app.py

```
1 import streamlit as st
2 import pandas as pd
3 import numpy as np
4
5 n = st.slider("n", 5, 100, step=1)
6 chart_data = pd.DataFrame(np.random.randn(n), columns=['data'])
7 st.line_chart(chart_data)
8
```

streamlit_app.py 8:1

CRLF UTF-8 Python main Fetch GitHub Git (1)

Crear nuevo archivo

streamlit_app.py — C:\Users\Usuario\streamlit_app — Atom

File Edit View Selection Find Packages Help

```
(base) C:\Users\Usuario\streamlit_app>streamlit run streamlit_app.py
2022-06-05 18:20:56.658 INFO      numexpr.utils: Note: NumExpr detected 16 cores but "NUMEXPR_MAX_THREADS" not set, so enforcing safe limit of 8.
2022-06-05 18:20:56.658 INFO      numexpr.utils: NumExpr defaulting to 8 threads.

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://192.168.5.192:8501
```

streamlit_app.py 8:1

CRLF UTF-8 Python main Fetch GitHub Git (1)

Crear nuevo archivo

streamlit_app.py — C:\Users\Usuario\streamlit_app

File Edit View Selection Find Packages Help

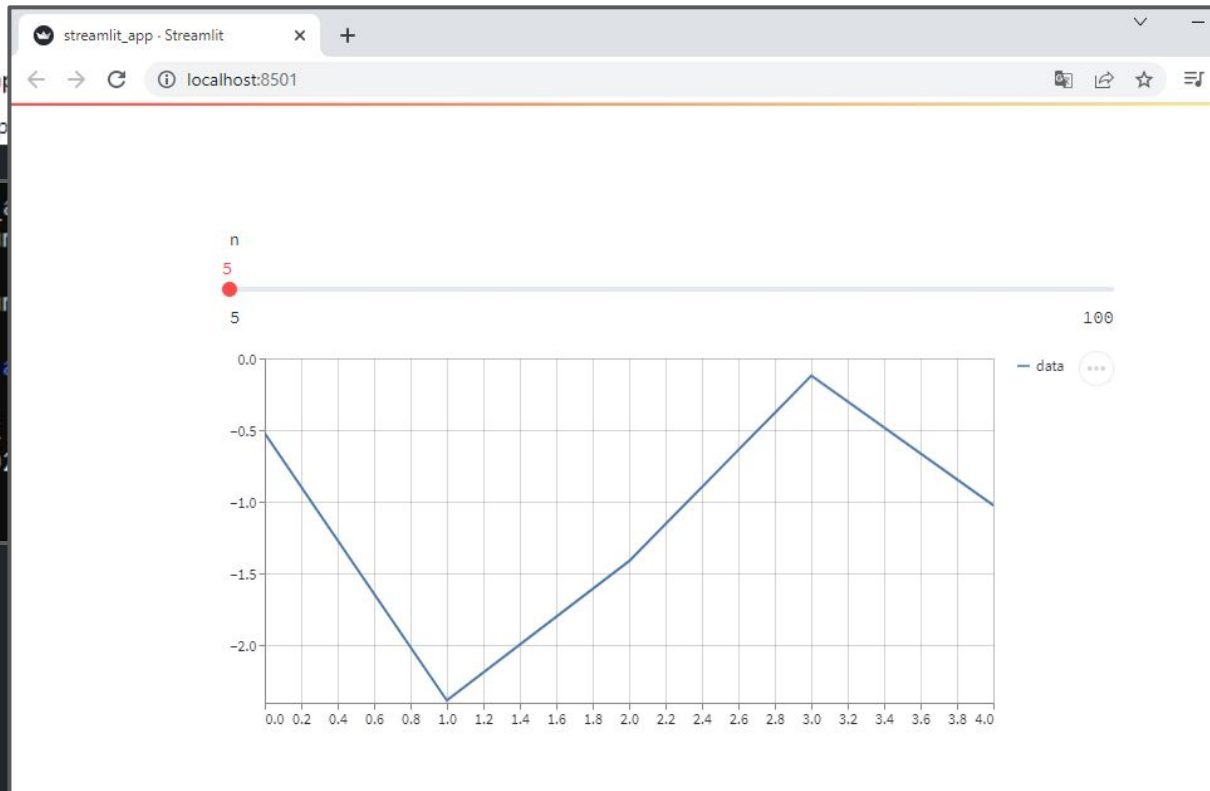
Project

```
(base) C:\Users\Usuario\streamlit_app>
2022-06-05 18:20:56.658 INFO      nur
t, so enforcing safe limit of 8.
2022-06-05 18:20:56.658 INFO      nur
```

You can now view your Streamlit app

Local URL: <http://localhost:8501>

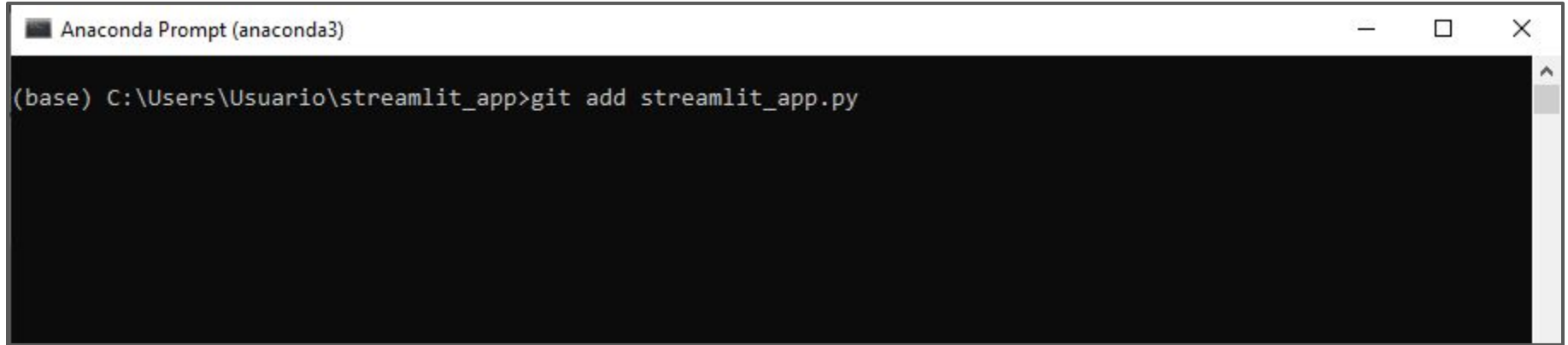
Network URL: <http://192.168.5.192:8501>



streamlit_app.py 8:1

CRLF UTF-8 Python main Fetch GitHub Git (1)

Agregar archivo `streamlit_app.py` al repositorio



```
Anaconda Prompt (anaconda3)
(base) C:\Users\Usuario\streamlit_app>git add streamlit_app.py
```

The image shows a terminal window titled "Anaconda Prompt (anaconda3)". The prompt is "(base) C:\Users\Usuario\streamlit_app>". The command entered is "git add streamlit_app.py". The terminal has a black background with white text. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

Hacer un commit del repositorio

```
Anaconda Prompt (anaconda3)
(base) C:\Users\Usuario\streamlit_app>git commit . -m "Primera version de la app"
Author identity unknown

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.
```



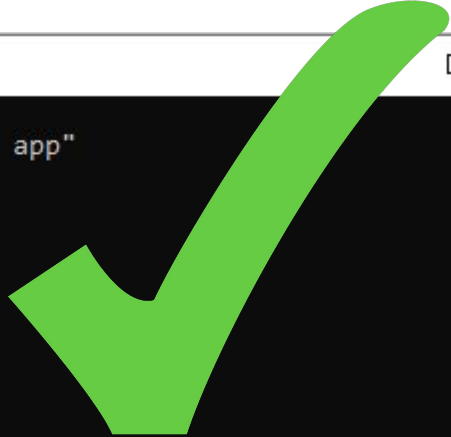
Es probable que nos muestre el siguiente error (la primera vez)

Agregar usuario (solo por única vez)

```
Anaconda Prompt (anaconda3)

(base) C:\Users\Usuario\streamlit_app>git commit . -m "Primera version de la app"
[main cacd69f] Primera version de la app
1 file changed, 7 insertions(+)
create mode 100644 streamlit_app.py

(base) C:\Users\Usuario\streamlit_app>
```



Actualizar repositorio

```
Anaconda Prompt (anaconda3)
(base) C:\Users\Usuario\streamlit_app>git commit . -m "Primera version de la app"
[main cacd69f] Primera version de la app
1 file changed, 7 insertions(+)
create mode 100644 streamlit_app.py

(base) C:\Users\Usuario\streamlit_app>ls
LICENSE  README.md  streamlit_app.py

(base) C:\Users\Usuario\streamlit_app>git push
```

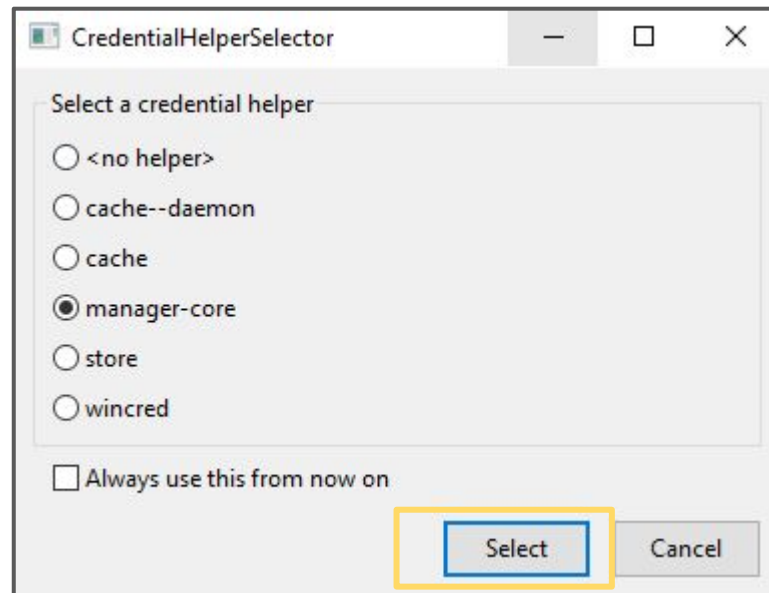
Actualizar repositorio

```
Anaconda Prompt (anaconda3)

(base) C:\Users\Usuario\streamlit_app>git commit .
[main cacd69f] Primera version de la app
1 file changed, 7 insertions(+)
create mode 100644 streamlit_app.py

(base) C:\Users\Usuario\streamlit_app>ls
LICENSE  README.md  streamlit_app.py

(base) C:\Users\Usuario\streamlit_app>git push
```



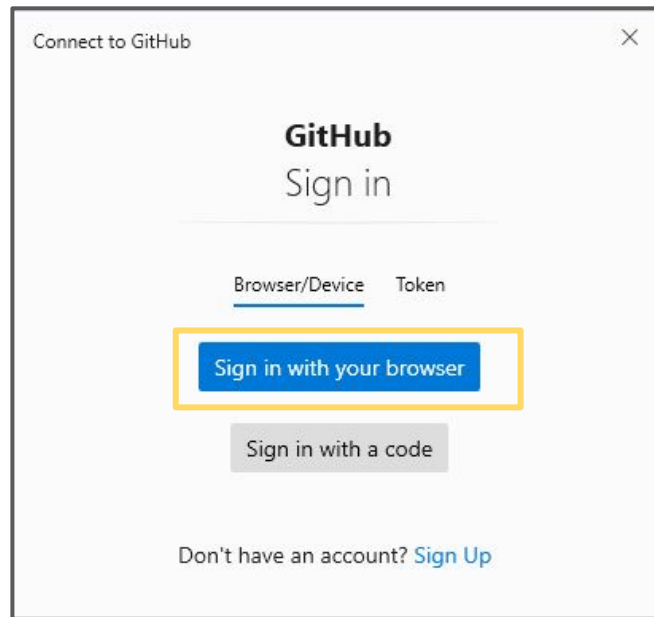
Actualizar repositorio

```
Anaconda Prompt (anaconda3)

(base) C:\Users\Usuario\streamlit_app>git commit .
[main cacd69f] Primera version de la app
1 file changed, 7 insertions(+)
create mode 100644 streamlit_app.py

(base) C:\Users\Usuario\streamlit_app>ls
LICENSE  README.md  streamlit_app.py

(base) C:\Users\Usuario\streamlit_app>git push
```



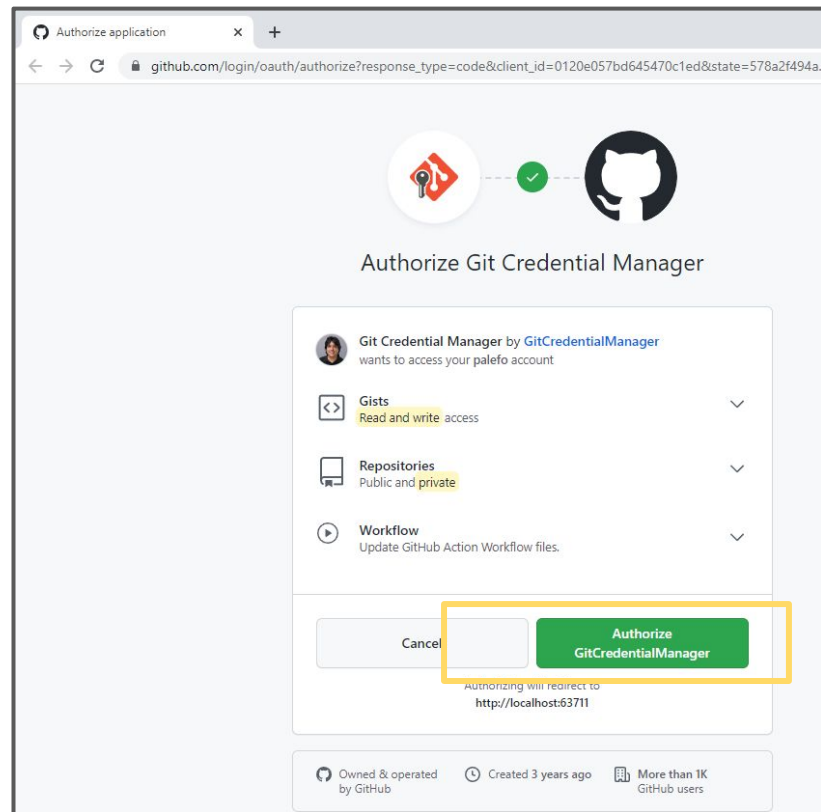
Actualizar repositorio

```
Anaconda Prompt (anaconda3)

(base) C:\Users\Usuario\streamlit_app>git commit .
[main cacd69f] Primera version de la app
1 file changed, 7 insertions(+)
create mode 100644 streamlit_app.py

(base) C:\Users\Usuario\streamlit_app>ls
LICENSE  README.md  streamlit_app.py

(base) C:\Users\Usuario\streamlit_app>git push
```



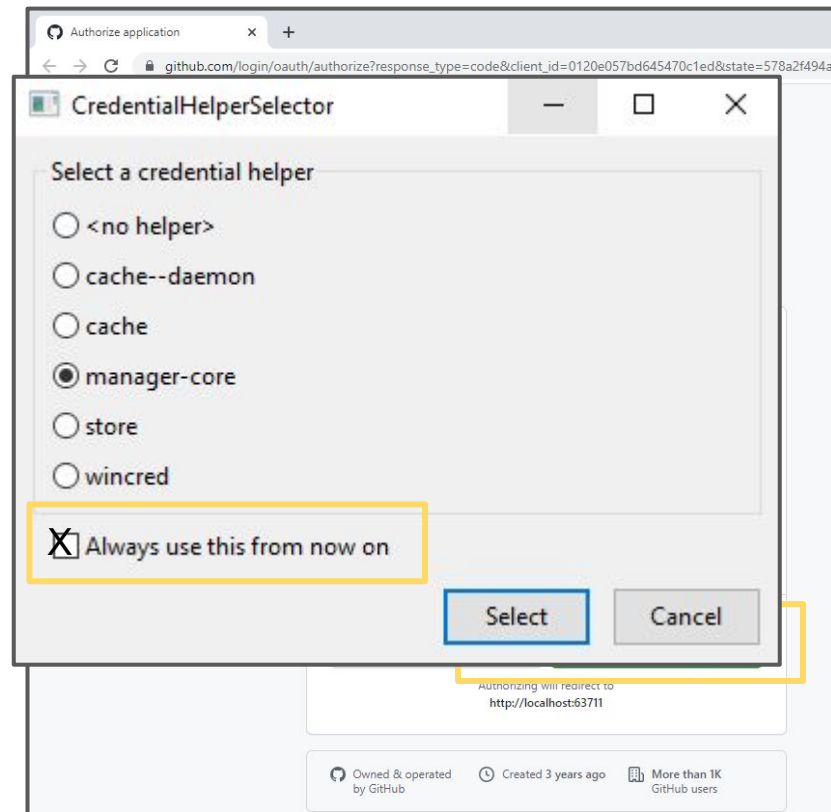
Actualizar repositorio

```
Anaconda Prompt (anaconda3)

(base) C:\Users\Usuario\streamlit_app>git commit .
[main cacd69f] Primera version de la app
1 file changed, 7 insertions(+)
create mode 100644 streamlit_app.py

(base) C:\Users\Usuario\streamlit_app>ls
LICENSE  README.md  streamlit_app.py

(base) C:\Users\Usuario\streamlit_app>git push
```



Actualizar repositorio

```
Anaconda Prompt (anaconda3)

(base) C:\Users\Usuario\streamlit_app>git commit . -m "Primera version de la app"
[main cacd69f] Primera version de la app
1 file changed, 7 insertions(+)
create mode 100644 streamlit_app.py

(base) C:\Users\Usuario\streamlit_app>ls
LICENSE  README.md  streamlit_app.py

(base) C:\Users\Usuario\streamlit_app>git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 423 bytes | 423.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/palefo/streamlit_app.git
c2cfd77..cacd69f  main -> main

(base) C:\Users\Usuario\streamlit_app>
```

Crear cuenta en Streamlit Community Cloud

- Usar opción con Google
- Usar opción conectar github

Crear cuenta en Streamlit Cloud

Set up your account

First Name

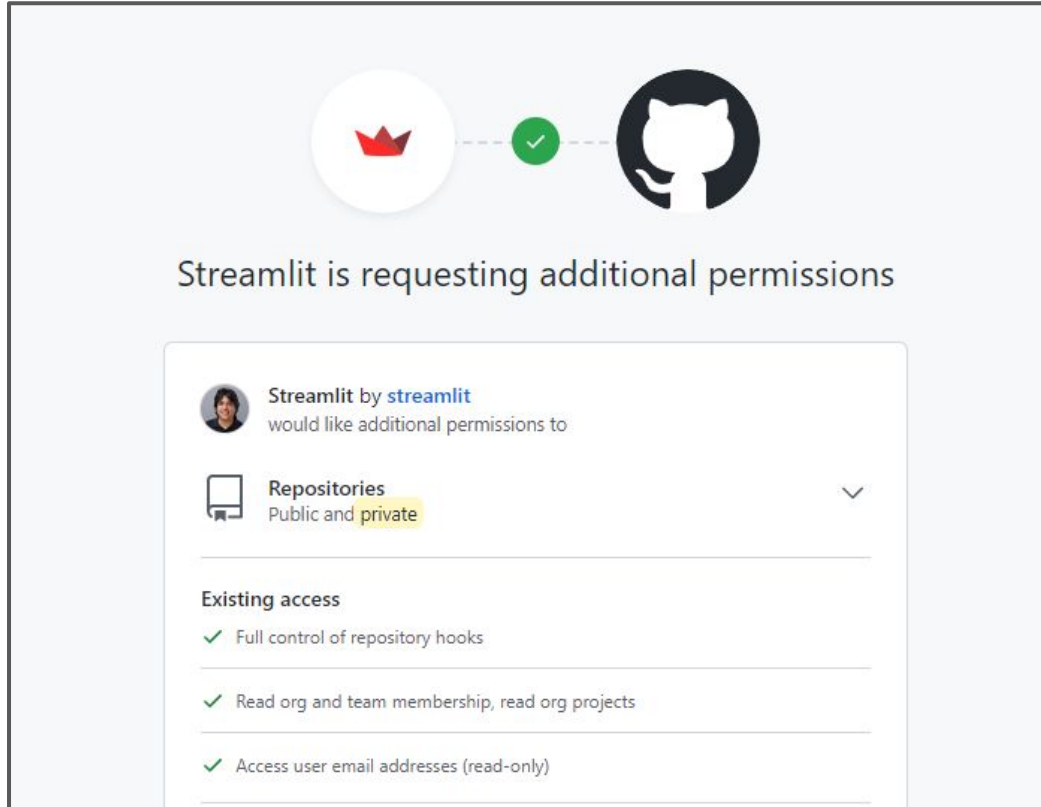
required

Pablo

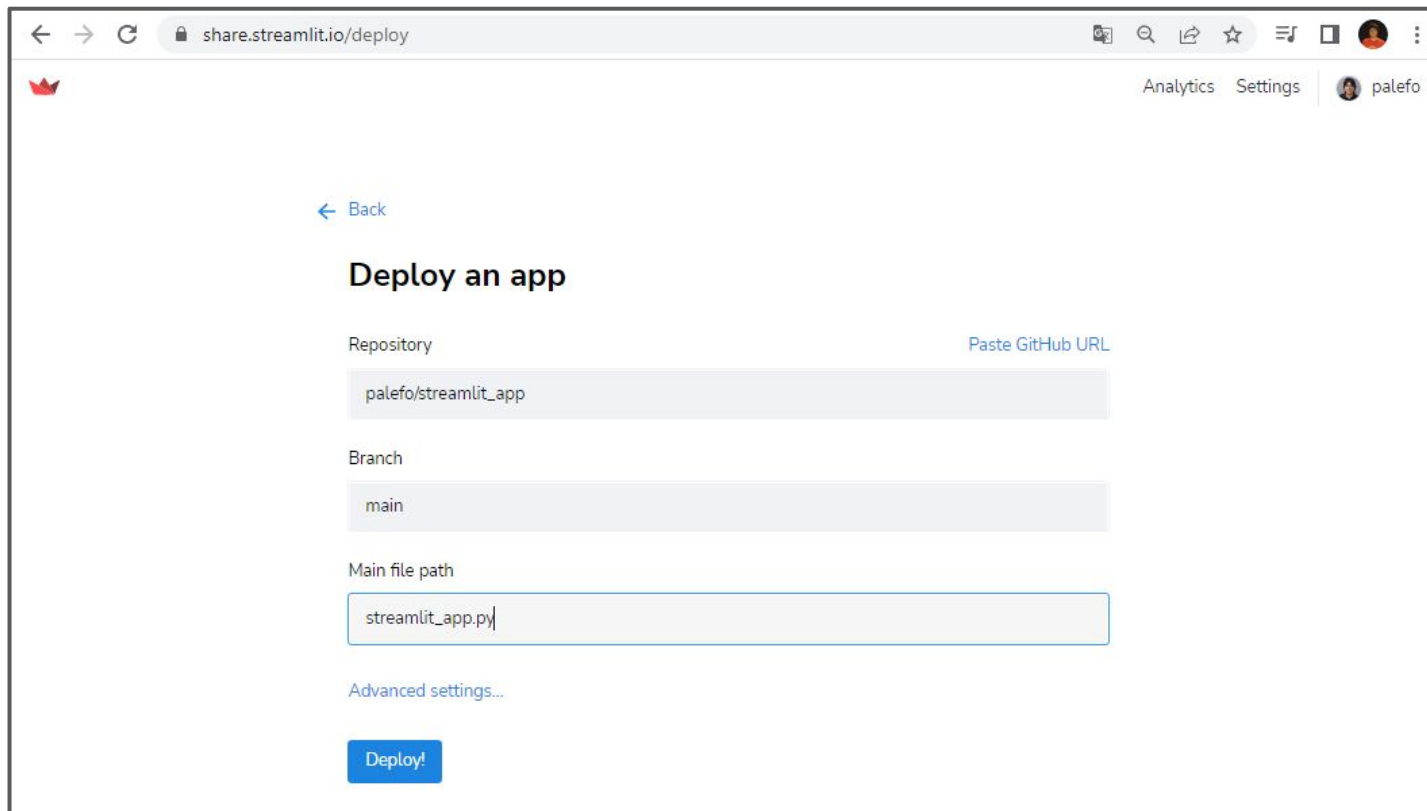
Last Name

Fonseca

Autorizar Acceso a Github



Seleccionar el repositorio creador... y dar click a Deploy!



The screenshot shows a web browser window at the URL `share.streamlit.io/deploy`. The page has a header with a red crown icon, navigation links for "Analytics" and "Settings", and a user profile for "palefo". The main content area is titled "Deploy an app" and includes a "Back" link. There are three input fields: "Repository" (containing "palefo/streamlit_app"), "Branch" (containing "main"), and "Main file path" (containing "streamlit_app.py"). A link "Paste GitHub URL" is next to the repository field. Below the inputs is a link for "Advanced settings...". At the bottom is a blue "Deploy!" button.

← Back

Deploy an app

Repository [Paste GitHub URL](#)

palefo/streamlit_app

Branch

main

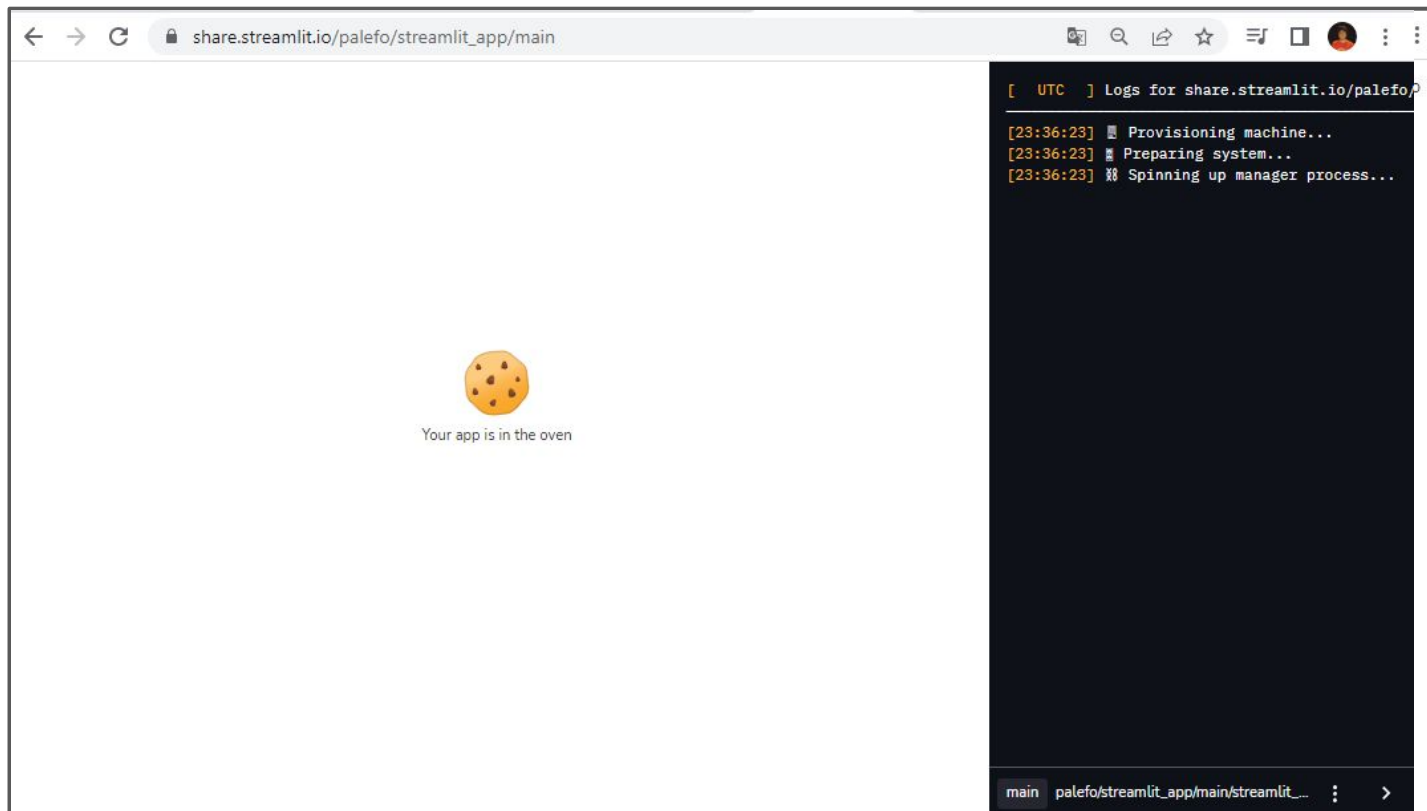
Main file path

streamlit_app.py

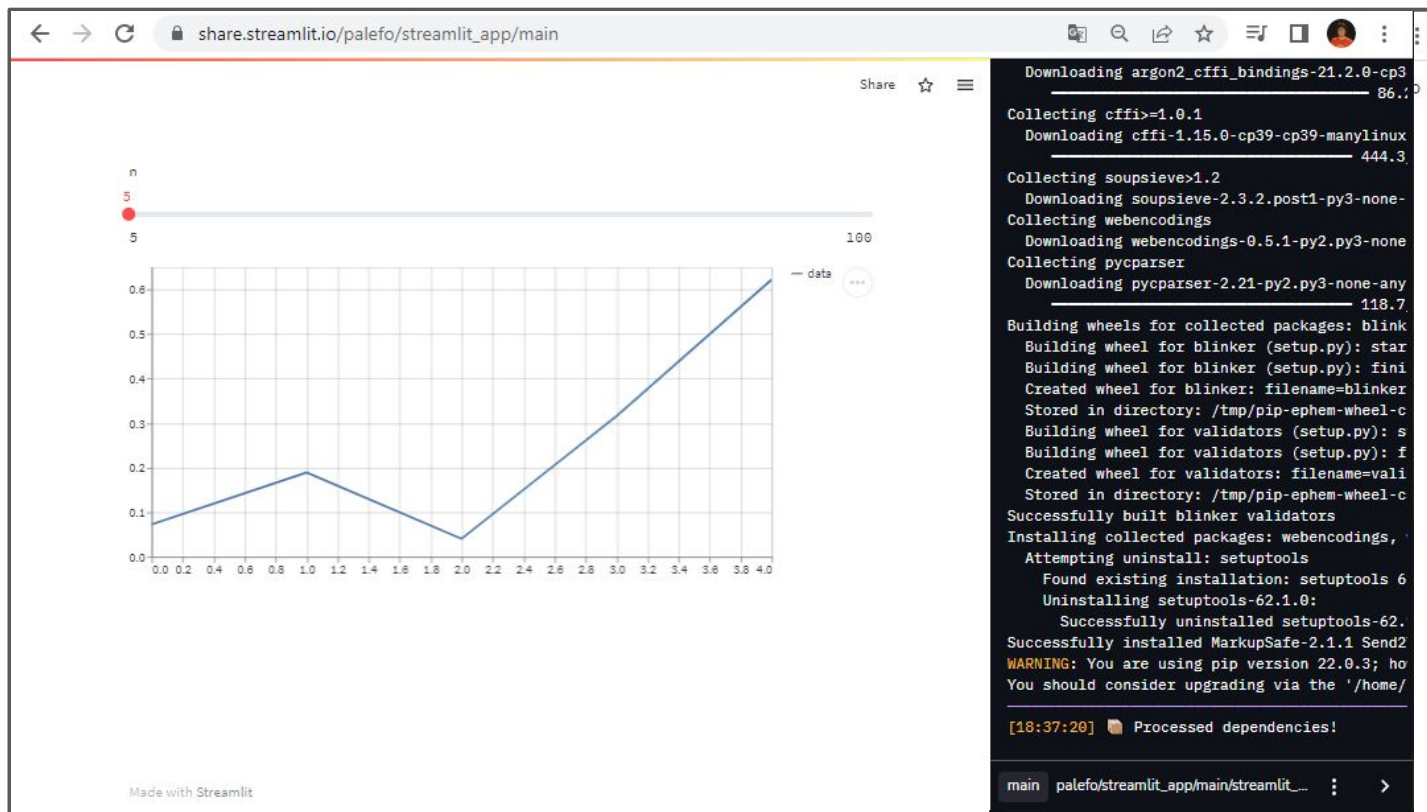
[Advanced settings...](#)

[Deploy!](#)

Construcción de la app

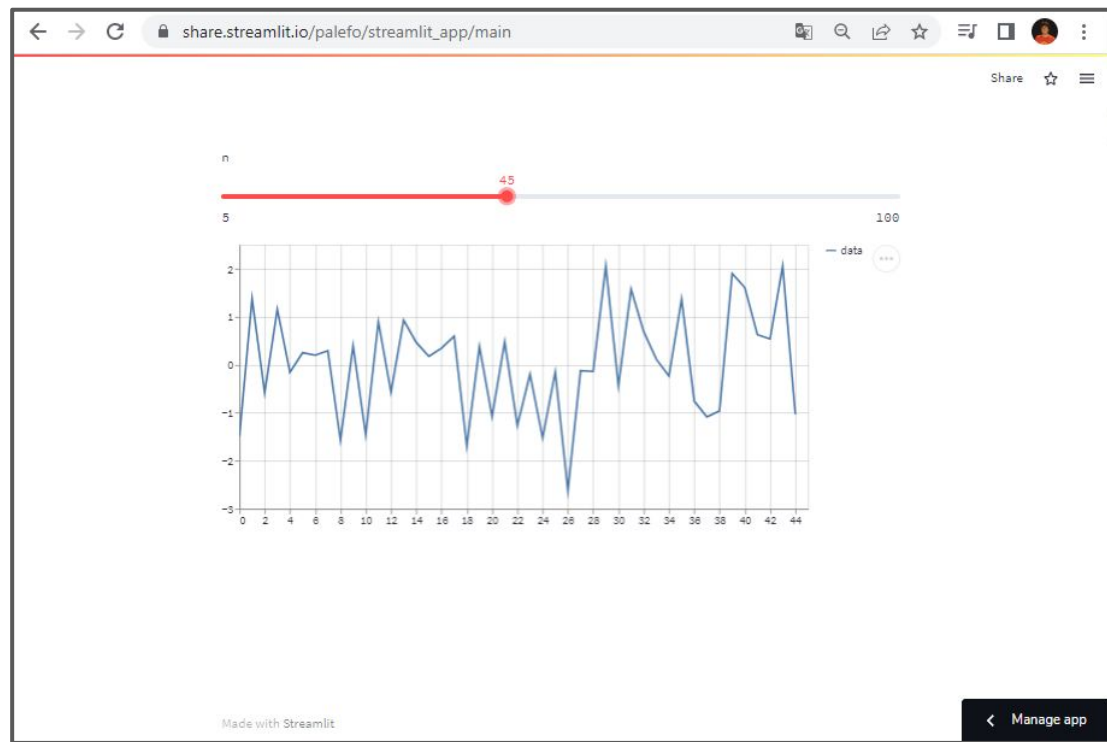


....finalmente, todo OK



App disponible en:

https://share.streamlit.io/palefo/streamlit_app/main



Otra opción: Usar la interfaz web de Github

Add file → Create new file

The screenshot shows the GitHub interface for the repository 'palefo/streamlit_app'. The top navigation bar includes the GitHub logo, a search bar, and links for Pull requests, Issues, Marketplace, and Explore. Below this, the repository name 'palefo/streamlit_app' is displayed with a 'Public' badge. A secondary navigation bar contains links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The main content area shows the 'main' branch with 1 branch and 0 tags. A dropdown menu is open under the 'Add file' button, highlighting the 'Create new file' option. The file list shows three files: .gitignore, LICENSE, and README.md, all from the initial commit. The README.md content is visible, showing the title 'streamlit_app' and the description 'App para el tutorial de Streamlit'. The right sidebar contains sections for 'About' (App para el tutorial de Streamlit), 'Releases' (No releases published), and 'Packages' (No packages published).

Search or jump to... / Pull requests Issues Marketplace Explore

palefo / streamlit_app Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags

Go to file Add file Code

Create new file
Upload files

palefo Initial commit

.gitignore	Initial commit	1 hour ago
LICENSE	Initial commit	1 hour ago
README.md	Initial commit	1 hour ago

README.md

streamlit_app

App para el tutorial de Streamlit

About

App para el tutorial de Streamlit

- Readme
- Apache-2.0 license
- 0 stars
- 1 watching
- 0 forks

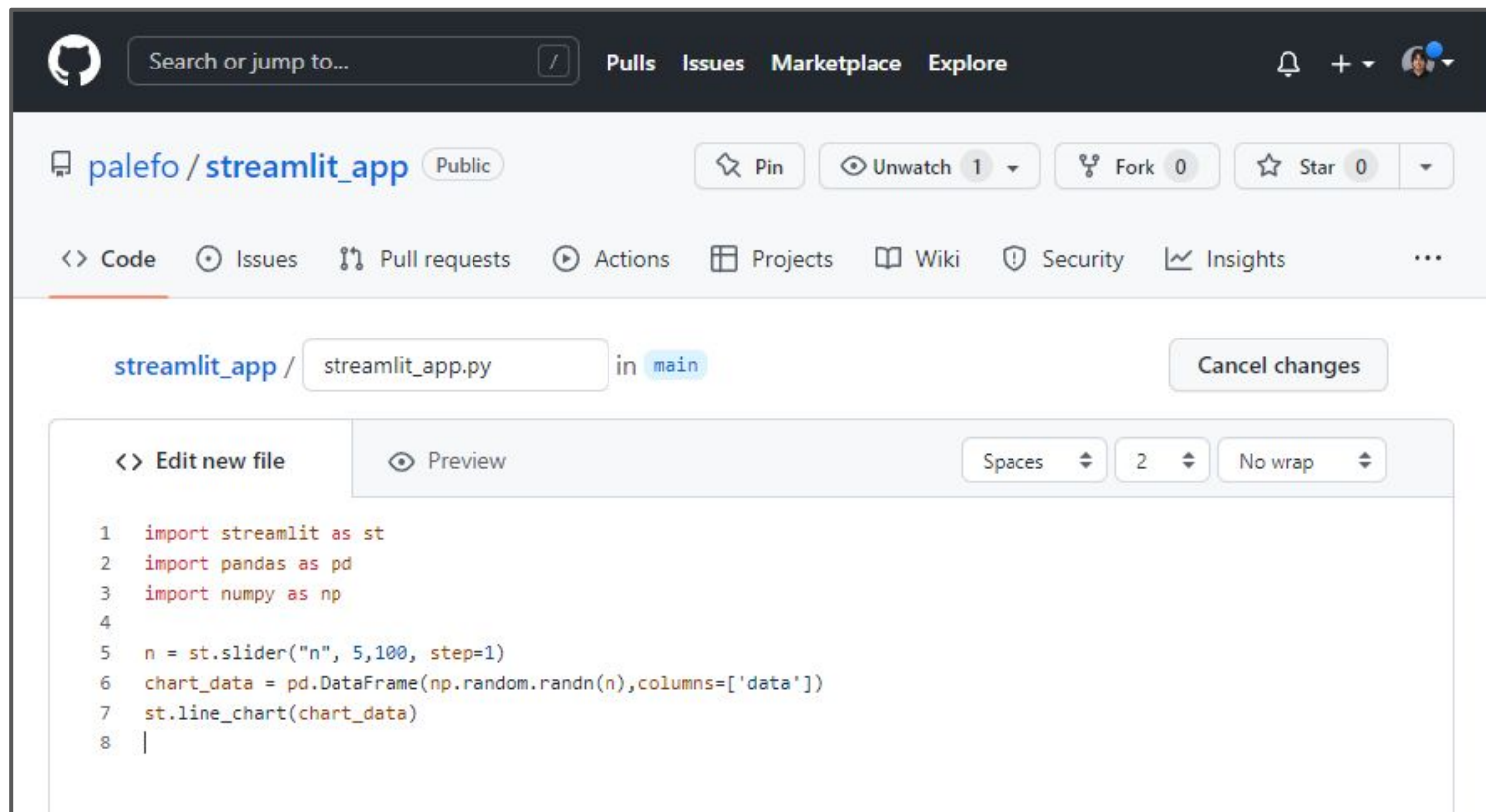
Releases

No releases published
[Create a new release](#)

Packages

No packages published

Otra opción: Usar la interfaz web de Github



The screenshot displays the GitHub web interface for a repository named 'palefo / streamlit_app'. The repository is public and has 1 unwatch, 0 forks, and 0 stars. The 'Code' tab is selected, showing the file 'streamlit_app.py' in the 'main' branch. The editor interface includes a 'Cancel changes' button and a 'Preview' tab. The code is as follows:

```
<> Edit new file    Preview    Spaces 2 No wrap

1 import streamlit as st
2 import pandas as pd
3 import numpy as np
4
5 n = st.slider("n", 5,100, step=1)
6 chart_data = pd.DataFrame(np.random.randn(n),columns=['data'])
7 st.line_chart(chart_data)
8 |
```

Decorador `st.cache_resource`

Este decorador sirve para recursos globales: por ej. modelos de ML, conexiones de bases de datos, etc. Es una instancia única para todos los threads del programa

```
@st.cache_resource
def foo(bar):
    # Do something expensive and return resource
    return resource
```

https://docs.streamlit.io/library/api-reference/performance/st.cache_resource

Decorador `st.cache_data`

En el tradeoff memoria-procesamiento-red , a veces es más conveniente almacenar elementos en memoria en lugar de recalcularlos o descargarlos nuevamente. El resultado de la función debe ser pickeable.

```
@st.cache_data
def foo(bar):
    # Do something expensive and return data
    return data
```

https://docs.streamlit.io/library/api-reference/performance/st.cache_data