

# Despliegue en producción de un proyecto Streamlit

## Introducción al tema

¿Sabías que el proceso de despliegue o puesta en producción de un sitio web es uno de los procesos más interesantes y valorados en la industria?

Como **Data Scientist**, es importante conocer no solo el proceso de aplicaciones web de ciencias de datos y hacer pruebas locales, sino conocer el proceso de cómo llevarlo a una sitio web real que pueda ser consumido por clientes potenciales.

Al terminar este tema tendrás una perspectiva muy clara del proceso de despliegue en producción de un proyecto web. Entonces, ¿estás preparado para iniciar este apasionante viaje de uno de los temas interesantes en el proceso de desarrollo de un producto?

### SUBTEMA 1. Despliegue en producción de la aplicación Streamlit



www.shutterstock.com • 1920804761

Afortunadamente todo marcha muy bien hasta ahora, en el módulo anterior ya se realizó una aplicación completa que realiza todas las operaciones básicas sobre una base de datos CSV.

### Documento de Entrenamiento para despliegue de aplicaciones

Para iniciar con el proceso de entrenamiento en despliegue de aplicaciones en producción, debemos seguir los siguientes pasos:

- El primer paso será crear un repositorio en github, ingresar en <https://github.com>, loguearse con las credenciales creadas en el tema 1, posteriormente crear un repositorio de código. NOTA: el repositorio debe ser Público y agregar un archivo README, tal como se muestra a continuación.


## 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 \*

 waves-cloud ▾

/

deploy\_streamlit


✓

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

Description (optional)


---

☒

 **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: None ▾


**Choose a license**

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

License: None ▾

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

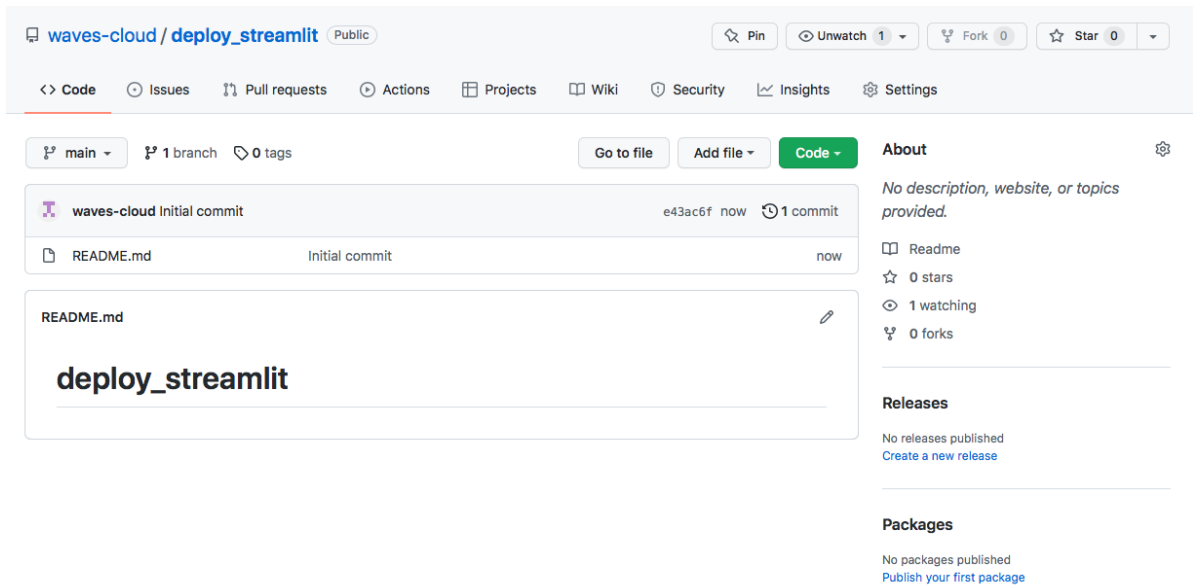
---

 You are creating a public repository in your personal account.

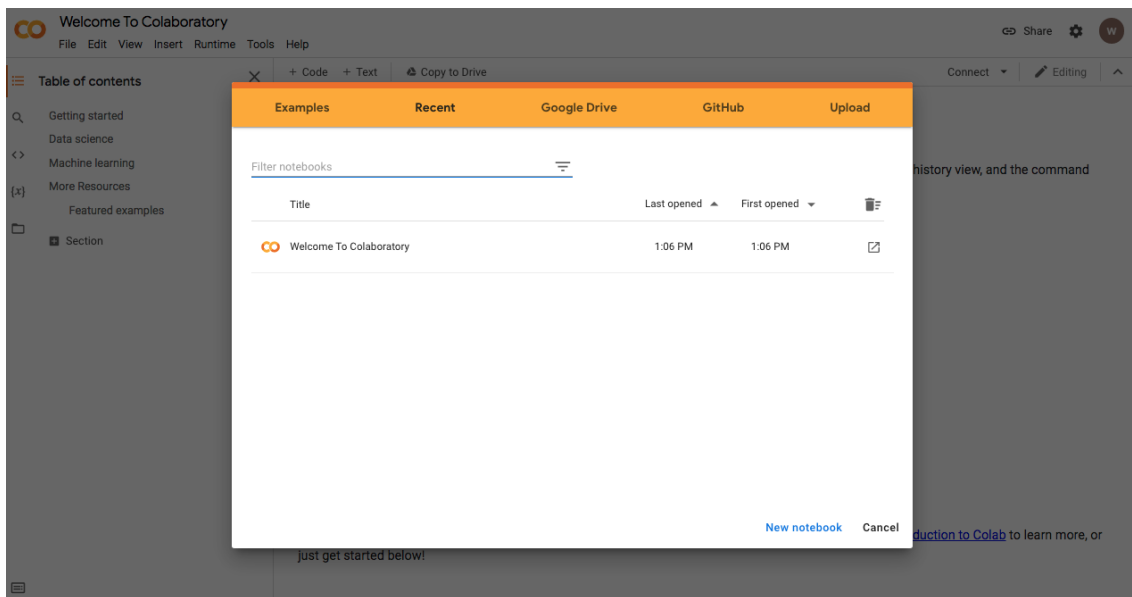
---

Create repository

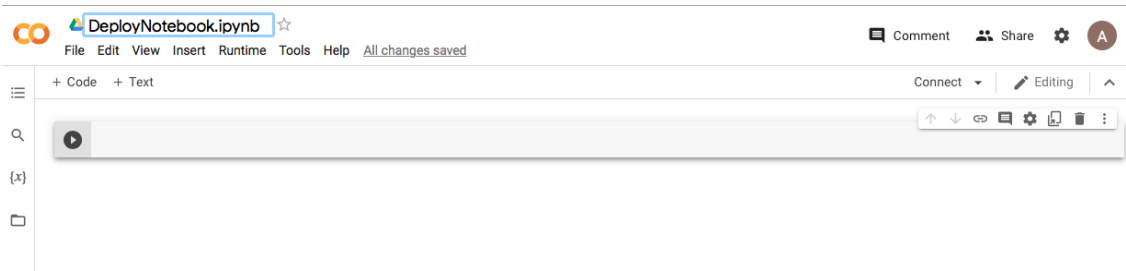
- Una vez creado, nuestro repositorio nuevo, luciria asi:



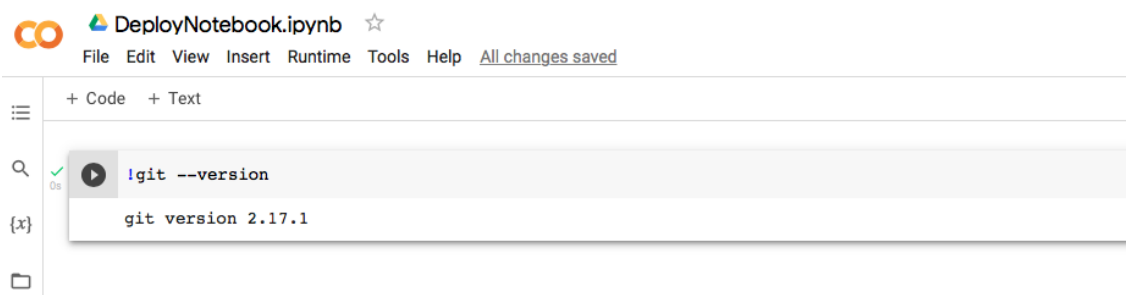
- El siguiente paso es crear un Proyecto Colab nuevo (llamado DeployNotebook) en Google Colab, ingresamos a <https://colab.research.google.com/> y presionamos New notebook.  
NOTA : Si estás usando Visual Studio Code solo debes crear una carpeta y colocar tu **programa.py** y tu **dataset.csv**



- De inmediato se crea nuestro proyecto y podemos asignarle un nombre, y está listo para ingresar código en python en las celdas de código.



- Entonces, nuestro primer paso es verificar que nuestro Notebook tenga el software de git instalado, para eso usamos el comando: **!git --version**



- Entonces una vez confirmado el comando git, explicas que el primer paso es ejecutar un comando que nos permita descargar el repositorio vacío creado previamente, por lo que debemos copiar el url o dirección del repositorio y usar el comando **git clone** con nuestro usuario, TOKEN y dirección de repositorio.



- Por lo tanto nuestro repositorio descargado estará en **/content/deploy\_streamlit** y podemos movernos a nuestro proyecto, ver su contenido y verificar la dirección url del repositorio al que está direccionado, con los siguientes comandos:
  - **cd deploy\_streamlit**
  - **!ls -la**

- `!git remote -v`

The screenshot shows a Jupyter Notebook interface. On the left, the 'Files' sidebar displays a directory structure with 'deploy\_streamlit' and 'sample\_data' folders, and a 'README.md' file. The main area shows a terminal with the following commands and output:

```
[9] cd deploy_streamlit
/content/deploy_streamlit

ls -la
total 16
drwxr-xr-x 3 root root 4096 Apr 25 21:46 .
drwxr-xr-x 1 root root 4096 Apr 25 21:46 ..
drwxr-xr-x 8 root root 4096 Apr 25 21:46 .git
-rw-r--r-- 1 root root 18 Apr 25 21:46 README.md

!git remote -v
origin https://waves-cloud:ghp_gWJAz10bh25p2ZWzIrUZzAYz4MCdqlw99EC@github.com:waves-cloud/deploy_streamlit
origin https://waves-cloud:ghp_gWJAz10bh25p2ZWzIrUZzAYz4MCdqlw99EC@github.com:waves-cloud/deploy_streamlit
```

El último comando deberá regresarnos la dirección de github de nuestro repositorio.

Ahora debemos asegurarnos de tener el código completo del proyecto listo para ser desplegado en producción, deberá ser ejecutado desde una celda de código en Colab o bien estar guardado en con el nombre: `streamlit_app.py`

```
%%writefile /content/deploy_streamlit/netflix.py
```


```
...
...
```

Con el paso anterior deberá de crear el archivo `/content/deploy_streamlit/streamlit_app.py` que posteriormente usaremos para el despliegue en producción.

Nuestra tarea ahora es crear un archivo llamado ***requirements.txt*** con los módulos de python que serán necesarios para que nuestro programa funcione sin necesidad de Google Colab.

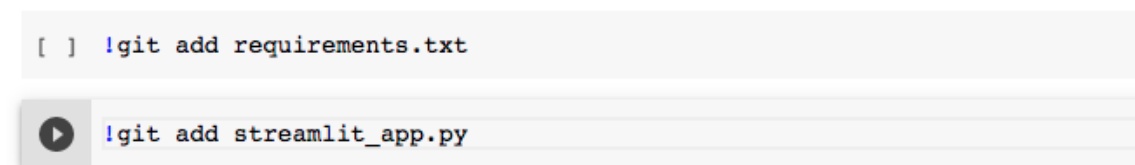
```
%%writefile requirements.txt
firebase-admin
streamlit
altair
pandas
numpy
```

Para que nuestro proyecto pueda ser puesto en producción el archivo ***streamli\_app.py*** y ***requirements.txt*** deben ser subidos al repositorio de github. Posteriormente otro proceso los tomara de esa ubicación y los desplegara en la nube y nos otorgará una dirección de internet única, por tanto explicas que debemos ejecutar el comando ***!ls -la*** para verificar que existan los archivos requirements.txt, streamlit\_app.py.



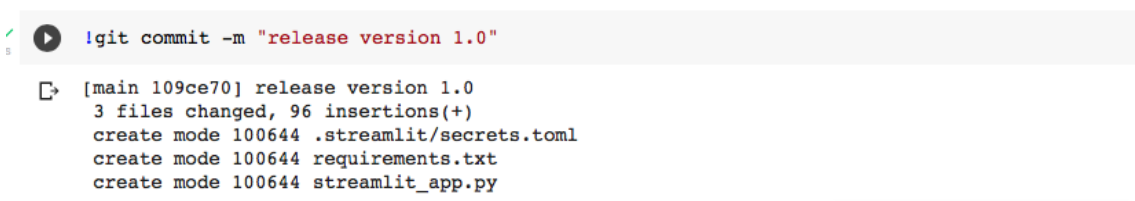
```
!ls -la
total 28
drwxr-xr-x 4 root root 4096 May  8 04:31 .
drwxr-xr-x 1 root root 4096 May  8 04:31 ..
drwxr-xr-x 8 root root 4096 May  8 04:31 .git
-rw-r--r-- 1 root root  18 May  8 04:31 README.md
-rw-r--r-- 1 root root  45 May  8 04:31 requirements.txt
drwxr-xr-x 2 root root 4096 May  8 04:31 .streamlit
-rw-r--r-- 1 root root 2253 May  8 04:31 streamlit_app.py
```

En este punto comentas que solo necesitamos subir 2 archivos (***requirements.txt*** y ***streamlit\_app.py***), entonces debemos agregar estos cambios a la nueva versión de nuestro repositorio con el comando git add



```
[ ] !git add requirements.txt
!git add streamlit_app.py
```

Crear el commit para esta versión del proyecto que subiremos



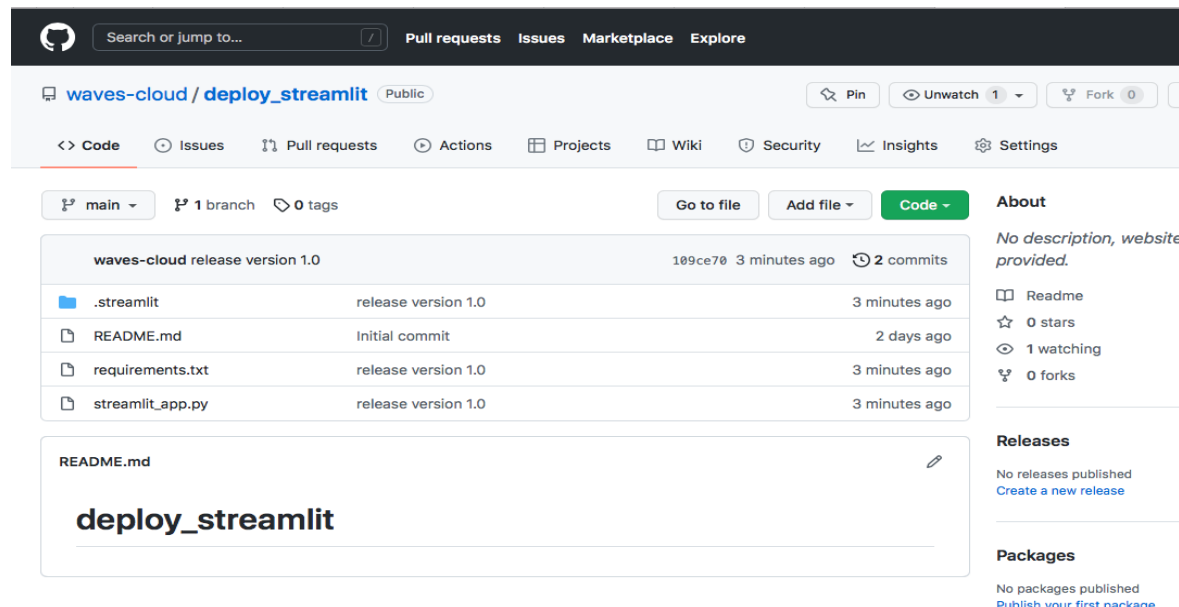
```
!git commit -m "release version 1.0"
[main 109ce70] release version 1.0
3 files changed, 96 insertions(+)
create mode 100644 .streamlit/secrets.toml
create mode 100644 requirements.txt
create mode 100644 streamlit_app.py
```

Finalmente subimos los cambios a github usando el comando git push

```
✓ 2s !git push -u origin main

Counting objects: 6, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 2.92 KiB | 2.92 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0)
To https://github.com/waves-cloud/deploy_streamlit.git
 e43ac6f..109ce70  main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

El repositorio debería lucir así, solo con el ARCHIVO **README**, el archivo de **requirements.txt** y **streamlit\_app.py**



Una vez que tengamos nuestro proyecto cargado en github, usaremos una herramienta en la nube de **streamlit** que permitirá poner en producción nuestro proyecto. Para empezar esta seccion ir a <https://share.streamlit.io/>





### Sign in

Continue with Google

Continue with GitHub

OR


Your email...

Continue with email

You can also [continue with SSO](#)  
New to Streamlit? [Sign up, it's free!](#)

Seleccionar continuar con el logueo tipo github, donde usaremos nuestras credenciales ya creadas con anterioridad para la plataforma github.com. Esto es muy importante ya que debemos tener los permisos para acceder a nuestros repositorios de github desde la plataforma de **streamlit** en la nube.





Sign in to **GitHub**  
to continue to **Streamlit**

---

Username or email address

Password [Forgot password?](#)

**Sign in**

New to GitHub? [Create an account.](#)

Posteriormente la plataforma te pedirá que plan desear usar para el despliegue de nuestro proyecto, seleccionar el plan Free

STEP 1 / 2

### Choose a plan

<b>Community</b>  Build and share public apps with the world.  Free	<b>Teams</b>  A workspace for your team to build and share apps with your stakeholders.  Includes free trial
---	--

[Learn more about our plans →](#)

Continue

- En esta sección solo debemos completar nuestro registro

First Name required

Adolfo

Last Name

Centeno

Primary Email required

waves-cloud-erp@gmail.com

How long have you been building apps with Streamlit?

Under 3 months

How are you planning on using Streamlit Cloud?

For myself

What's your functional area?

Data Science

Company name

waves lab

Country required

Mexico

Are you using Snowflake in any capacity?

No



Want to learn more about using Streamlit to speed up your company?

Continue

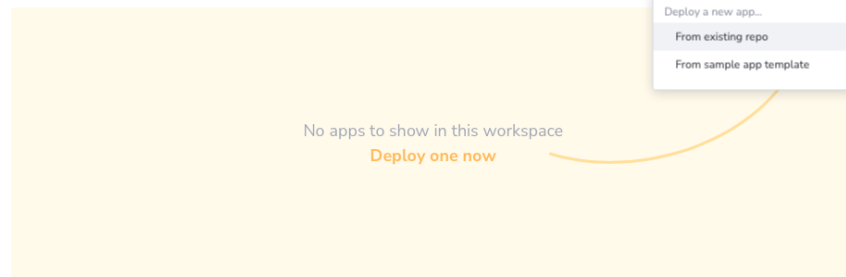
Una vez dentro de la plataforma **Streamlit**, seleccionaremos crear una nueva app, desde un repositorio existente



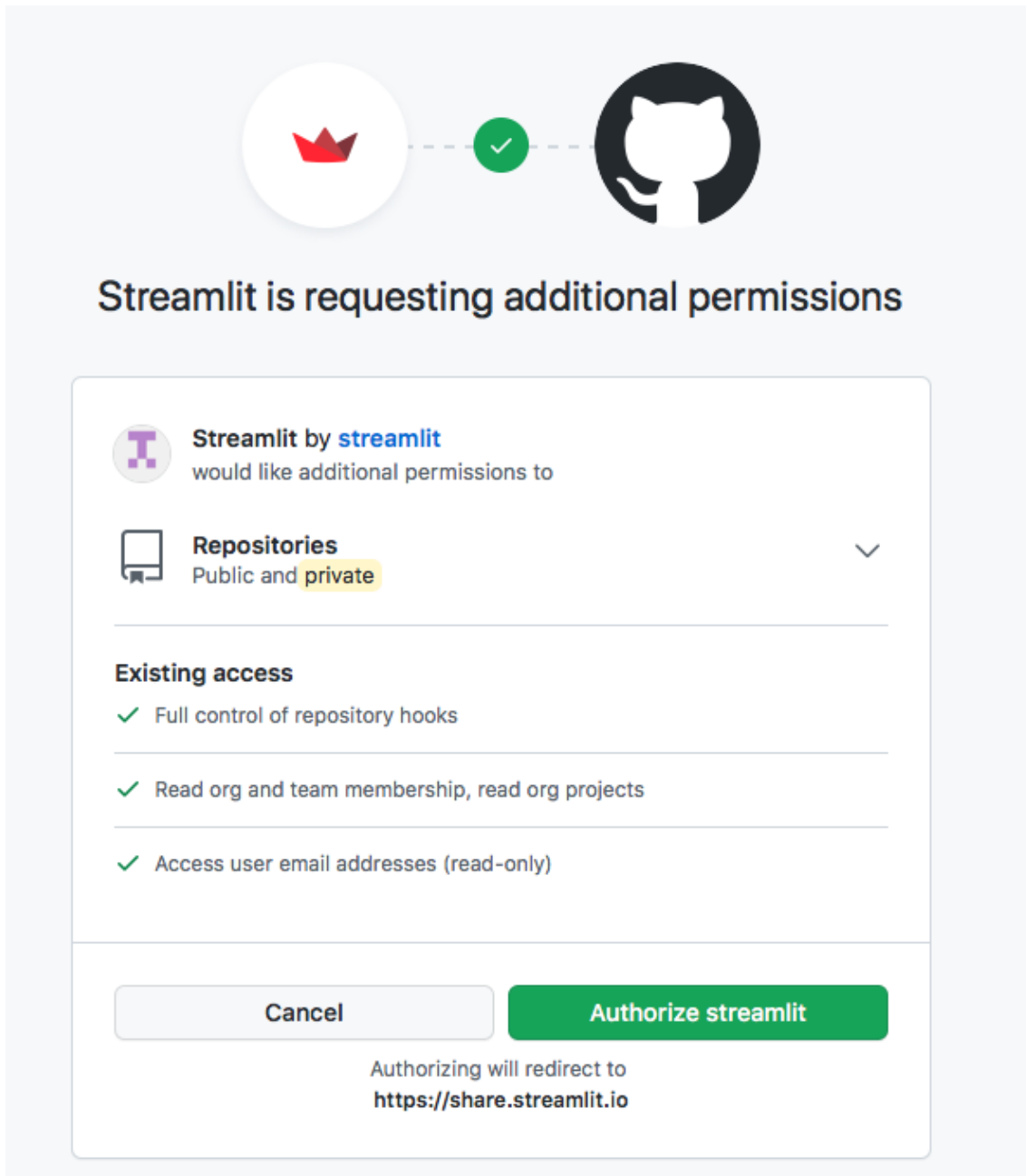
Upgrade! ⚠ Settings

waves-cloud

### Your apps



Debemos autorizar a **Streamlit** con los permisos a los repositorios de github, donde tomaremos el proyecto a desplegar.



Completar la configuración con el nombre del repositorio con el formato ***usuario/nombre\_repositorio***, el **branch**(regularmente llamado main o master) y el nombre del programa de python (***streamlit\_app.py***), una vez ingresado y revisado los 3 campos, presionar deploy para lanzar el proceso de Despliegue de la aplicación, que tomara el código del repositorio de github y lo montara en una instancia de cómputo de la nube de ***Streamlit***.

[← Back](#)

### Deploy an app

Repository

[Paste GitHub URL](#)

waves-cloud/deploy\_streamlit

Branch

main

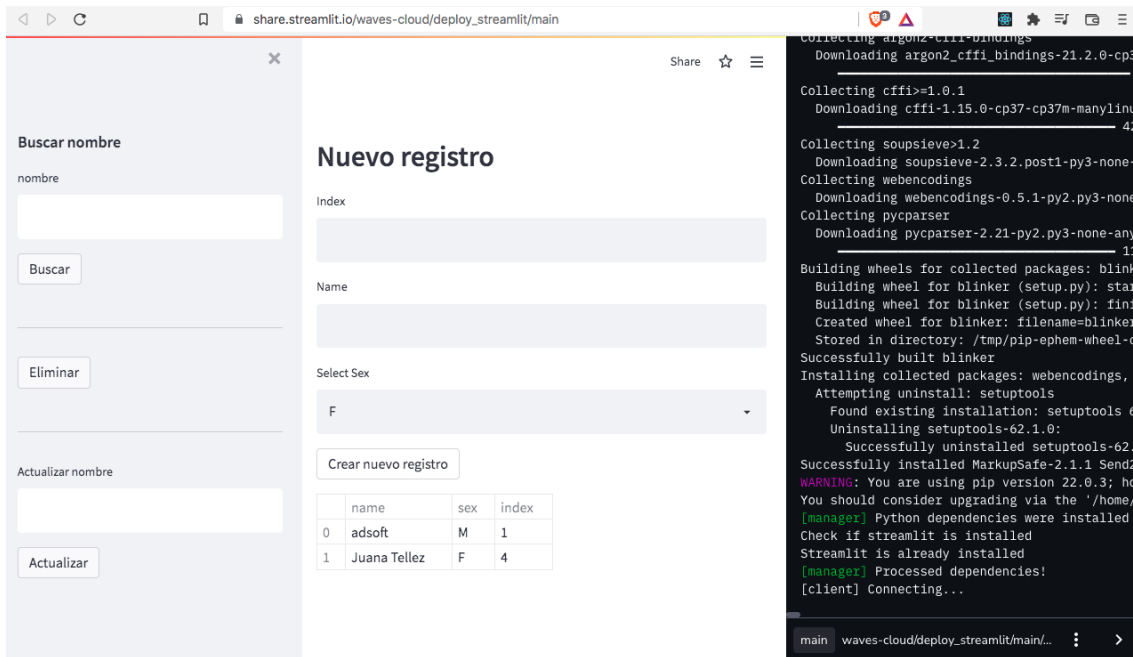
Main file path

streamlit\_app.py

[Advanced settings...](#)

Deploy!

NOTA: el paso anterior creará nuestra instancia de cómputo en la nube de **Streamlit** donde se hospedará nuestro proyecto, además se instalarán las dependencias de python y se generará nuestra url única.



-Finalmente, nuestra aplicación en producción estará en la url:

[https://share.streamlit.io/waves-cloud/deploy\\_streamlit/main](https://share.streamlit.io/waves-cloud/deploy_streamlit/main)

El proceso para el despliegue de aplicación ha sido un gran éxito, todos en el equipo de desarrollo quedaron satisfechos y están satisfechos con el proceso de integración desde construir el prototipo en Google Colab, subirlo a **Github** y finalmente desplegarlo desde github a la plataforma **Streamlit**.

Piensa en los proyectos de tecnología que has realizado hasta este punto de tu carrera como Científico de datos

- o ¿Cuántos de tus proyectos se quedaron solo como un prototipo y no se han podido compartir públicamente a clientes o amigos?
- o ¿Qué idea o proyecto actual te gustaría publicar en internet para ser consumido por una gran cantidad de usuarios de cualquier parte del mundo?

Piensa en las enormes posibilidades que tienes si ya puedes controlar todo el ciclo de vida de un proyecto de software, desde leer los requerimientos hacer un prototipo en Google Colab, conectarlo a una base de datos real CSV y finalmente ponerlo en producción.



Una vez que todos han experimentado el proceso de Despliegue y han comprobado la sencillez del mismo, los invito a responder las siguientes preguntas para reforzar los conceptos.

**Pregunta 1:** *El archivo requirements.txt con los módulos en python del proyecto, es **opcional** en el proceso de despliegue.*

Opciones de respuesta	Retroalimentación
<b>Opción a:</b> Falso	<b>CORRECTO:</b> El archivo requirements.txt es indispensable en el despliegue ya que contiene los módulos necesarios para que el proyecto funcione de forma autónoma.
<b>Opción b:</b> Verdadero	<b>INCORRECTO:</b> El archivo requirements.txt si es absolutamente necesario

**Pregunta 2:** *Para desplegar una app en Streamlit, primero se deben subir a un repositorio los archivos: requirements.txt y streamlit\_app.py*

Opciones de respuesta	Retroalimentación
<b>Opción a:</b> Verdadero	<b>CORRECTO:</b> Streamlit tomará estos 2 archivos del repositorio y lanzará un proceso de despliegue en una máquina virtual en la plataforma Streamlit
<b>Opción b:</b> Falso	<b>INCORRECTO:</b> Estos 2 archivos requirements.txt y streamlit_app.py son necesarios para el proceso de despliegue de la app en producción.

### Ideas para llevar

Como parte de tu formación como científico de datos ahora tienes más claro que:

- Es sencillo crear aplicaciones con base de datos hospedadas en la nube
- Se pueden crear y descargar llaves de acceso a una base de datos de firebase
- Para otorgar más seguridad a nuestros proyectos podemos convertir nuestras llaves a un formato .toml

- Puedes crear repositorios de código en Github para respaldarlos de forma segura y además ya en github pueden llevarse a producción de manera muy sencilla
- Es fácil poner en producción un proyecto de Ciencia de datos creado en Colab

### Referencias.

<https://github.com/>

<https://share.streamlit.io/>

<https://blog.streamlit.io/streamlit-firestore/>

<https://docs.streamlit.io/streamlit-cloud/get-started/deploy-an-app>