Contenido

[CREACIÓN DE ENTORNO VIRTUAL 3](#_Toc203045391)

[INSTALACIÓN DE STREAMLIT 11](#_Toc203045392)

[ACTUALIZAR REQUIREMENTS.TXT 15](#_Toc203045393)

[AUTOMATIZAR CON ARCHIVOS BAT 15](#_Toc203045394)

[ACTIVAR ENTORNO VIRTUAL 15](#_Toc203045395)

[ACTUALIZAR REQUIREMENTS.TXT 15](#_Toc203045396)

[CREANDO MI PRIMER HOLA MUNDO 16](#_Toc203045397)

[MANEJO DE DATAFRAMES 20](#_Toc203045398)

[LEER DATOS DESDE API DATYOS ABIERTOS COLOMBIA 21](#_Toc203045399)

[MOSTRAR DATOS JSON 22](#_Toc203045400)

[MOSTRAR CODIGO DENTRO DE LA APP 23](#_Toc203045401)

[CAJA DE SELECCIÓN 24](#_Toc203045402)

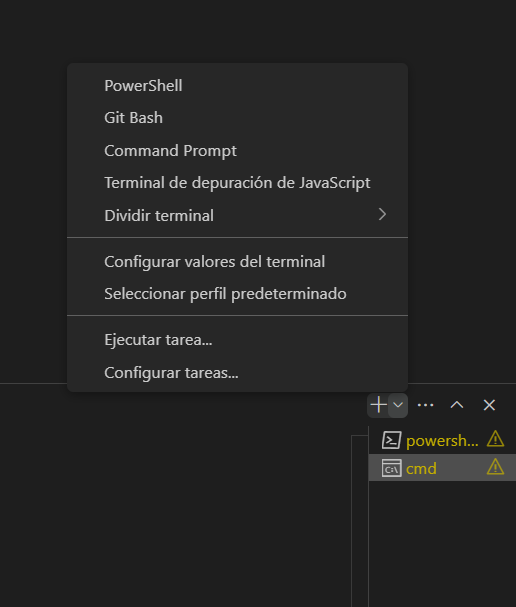
[SELECCIÓN MULTIPLE 24](#_Toc203045403)

[DESLIZADOR 25](#_Toc203045404)

[MANEJO DE ARCHIVOS MULTIMEDIA 25](#_Toc203045405)

# CREACIÓN DE ENTORNO VIRTUAL

Usar el comando python -m venv env esto creara el entorno dentro del proyecto, recomiendo usar la consola de command prompt en vs code



Después de ejecutar vera alcomo como esto

Texto

El contenido generado por IA puede ser incorrecto.

Y se creara una carpeta

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Hasta este punto se asume que previo a todo esto, tiene instalado Git en su comutador y tiene una cuenta en [www.github.com](http://www.github.com) donde debió crear un repositorio y debió haber clonado el mismo en su Pc, por ello mi ejemplo tiene un archivo que github crea y escoji que usara Python este es el archivo gitignore, este archivo evitara se se carguen cambios al repositorio productos de archivos de ejecución del proyecto, a lo que le llamaríamos archivos basura.

No se recomienda cargar el entorno virtual al repositorio por eso se recomienda agregar env dentro del archi gitignore



Cave resaltara que debe escribirlo tal cual y como creo su entorno virtual

Aquí dejare el contenido del git ignore:

# Byte-compiled / optimized / DLL files

\_\_pycache\_\_/

\*.py[codz]

\*$py.class

# C extensions

\*.so

#entorno virtual

env/

# Distribution / packaging

.Python

build/

develop-eggs/

dist/

downloads/

eggs/

.eggs/

lib/

lib64/

parts/

sdist/

var/

wheels/

share/python-wheels/

\*.egg-info/

.installed.cfg

\*.egg

MANIFEST

# PyInstaller

#  Usually these files are written by a python script from a template

#  before PyInstaller builds the exe, so as to inject date/other infos into it.

\*.manifest

\*.spec

# Installer logs

pip-log.txt

pip-delete-this-directory.txt

# Unit test / coverage reports

htmlcov/

.tox/

.nox/

.coverage

.coverage.\*

.cache

nosetests.xml

coverage.xml

\*.cover

\*.py.cover

.hypothesis/

.pytest\_cache/

cover/

# Translations

\*.mo

\*.pot

# Django stuff:

\*.log

local\_settings.py

db.sqlite3

db.sqlite3-journal

# Flask stuff:

instance/

.webassets-cache

# Scrapy stuff:

.scrapy

# Sphinx documentation

docs/\_build/

# PyBuilder

.pybuilder/

target/

# Jupyter Notebook

.ipynb\_checkpoints

# IPython

profile\_default/

ipython\_config.py

# pyenv

#   For a library or package, you might want to ignore these files since the code is

#   intended to run in multiple environments; otherwise, check them in:

# .python-version

# pipenv

#   According to pypa/pipenv#598, it is recommended to include Pipfile.lock in version control.

#   However, in case of collaboration, if having platform-specific dependencies or dependencies

#   having no cross-platform support, pipenv may install dependencies that don't work, or not

#   install all needed dependencies.

#Pipfile.lock

# UV

#   Similar to Pipfile.lock, it is generally recommended to include uv.lock in version control.

#   This is especially recommended for binary packages to ensure reproducibility, and is more

#   commonly ignored for libraries.

#uv.lock

# poetry

#   Similar to Pipfile.lock, it is generally recommended to include poetry.lock in version control.

#   This is especially recommended for binary packages to ensure reproducibility, and is more

#   commonly ignored for libraries.

#   https://python-poetry.org/docs/basic-usage/#commit-your-poetrylock-file-to-version-control

#poetry.lock

#poetry.toml

# pdm

#   Similar to Pipfile.lock, it is generally recommended to include pdm.lock in version control.

#   pdm recommends including project-wide configuration in pdm.toml, but excluding .pdm-python.

#   https://pdm-project.org/en/latest/usage/project/#working-with-version-control

#pdm.lock

#pdm.toml

.pdm-python

.pdm-build/

# pixi

#   Similar to Pipfile.lock, it is generally recommended to include pixi.lock in version control.

#pixi.lock

#   Pixi creates a virtual environment in the .pixi directory, just like venv module creates one

#   in the .venv directory. It is recommended not to include this directory in version control.

.pixi

# PEP 582; used by e.g. github.com/David-OConnor/pyflow and github.com/pdm-project/pdm

\_\_pypackages\_\_/

# Celery stuff

celerybeat-schedule

celerybeat.pid

# SageMath parsed files

\*.sage.py

# Environments

.env

.envrc

.venv

env/

venv/

ENV/

env.bak/

venv.bak/

# Spyder project settings

.spyderproject

.spyproject

# Rope project settings

.ropeproject

# mkdocs documentation

/site

# mypy

.mypy\_cache/

.dmypy.json

dmypy.json

# Pyre type checker

.pyre/

# pytype static type analyzer

.pytype/

# Cython debug symbols

cython\_debug/

# PyCharm

#  JetBrains specific template is maintained in a separate JetBrains.gitignore that can

#  be found at https://github.com/github/gitignore/blob/main/Global/JetBrains.gitignore

#  and can be added to the global gitignore or merged into this file.  For a more nuclear

#  option (not recommended) you can uncomment the following to ignore the entire idea folder.

#.idea/

# Abstra

# Abstra is an AI-powered process automation framework.

# Ignore directories containing user credentials, local state, and settings.

# Learn more at https://abstra.io/docs

.abstra/

# Visual Studio Code

#  Visual Studio Code specific template is maintained in a separate VisualStudioCode.gitignore

#  that can be found at https://github.com/github/gitignore/blob/main/Global/VisualStudioCode.gitignore

#  and can be added to the global gitignore or merged into this file. However, if you prefer,

#  you could uncomment the following to ignore the entire vscode folder

# .vscode/

# Ruff stuff:

.ruff\_cache/

# PyPI configuration file

.pypirc

# Cursor

#  Cursor is an AI-powered code editor. `.cursorignore` specifies files/directories to

#  exclude from AI features like autocomplete and code analysis. Recommended for sensitive data

#  refer to https://docs.cursor.com/context/ignore-files

.cursorignore

.cursorindexingignore

# Marimo

marimo/\_static/

marimo/\_lsp/

\_\_marimo\_\_/

Si todo ha salido bien, entonces la próxima vez que hagas un push en git no se cargara ningún archivo o extensión relacionada en el gitignore.

Interfaz de usuario gráfica, Texto

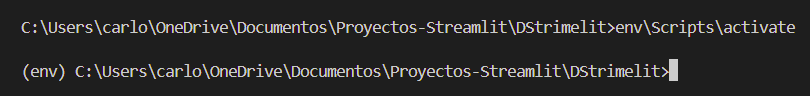
El contenido generado por IA puede ser incorrecto.

Lo sabaras porque en git las carpetas o archivos nuevos se visualizan con color verde y los archivos modificados con amarillo, pero los ignorados salen con letras grises.

# INSTALACIÓN DE STREAMLIT

Debe primero activar el entorno virtual , puede usar el comando : env\Scripts\activate

Sabrá que esta activo porque aparece el nombre del entorno en paréntesis (env)



Primero revisemos que dependencias tenemos instaladas

Con el entorno activo ejecuta: pip list

Texto

El contenido generado por IA puede ser incorrecto.

En mi caso tengo la librería de pip y setuptools. Normalmente solo sale pip

Ahora si instalaremos la librería de Streamli

Ejecutar en la consola de comandos pip install streamlit

Para ello asegurece de estar dentro de la carpeta Scripts de su entorno virtual, esta es la forma correcta de instalar dependencias y librerias



Se cargara muchas cosas relacionadas a la libreria

Texto

El contenido generado por IA puede ser incorrecto.

Al finalizar puee hacer de nuevo : pip list y vera las nuevas dependencias que se intalaron con base en el comando

En mi caso esta es un vista de algo del contenido, puede observar que streamlit esta ahora instalado

Texto

El contenido generado por IA puede ser incorrecto.

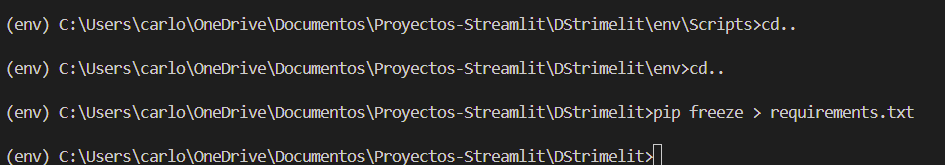
Okay es importante aclarar qwue estas dependencias estarán instaladas en el entorno virtual para evitar instalarlas en nuetro pc, pro como el entorno virtual no lo cargamos a github es importante crear el archivo requirements.txt

Creando Archivo requirements.txt

Para que sirve?

sirve para **guardar la lista de librerías que usa tu proyecto**, con sus versiones exactas, para que cualquier persona (o tú mismo en otra computadora) pueda **instalarlas rápido y sin errores** usando un solo comando (pip install -r requirements.txt), asegurando que el proyecto funcione igual en cualquier entorno.

Antes de ejecutar el comando en la consola debe salir de la carpeta Scrips y env. Asegurarse de que se encuentra en la carpeta principal de su proyecto y allí ejecutar el comando



Comando : pip freeze > requirements.txt

Si no sabe como salir de las carpetas usa cd..

Una vez ejecutado se crear un archivo llamado requirements.txt, si no aparece es posible que no haya salido de Scripts y dentro del entorno virtual lo haya creado por error, es mejor buscarlos manualmente y borrarlo, segurarse como en la imagen de arriba que esta en la carpeta principal y ejecutar el comando

Captura de pantalla de un teléfono celular

El contenido generado por IA puede ser incorrecto.

De esta forma debe verse y si abre el archivo puede ver que están todas las librerías

Pantalla de computadora con letras

El contenido generado por IA puede ser incorrecto.

No se asuste si no ve los mismos nombres suelen variar según sus versiones

# ACTUALIZAR REQUIREMENTS.TXT

Básicamente es ejecutar en la carpte aprincipal de nuevo: pip freeze > requirements.txt  
esto reemplazara el archivo anterior

INSTALAR REQUIREMENTS. TEXT DE OTROS PROYECTOS CLONADOS

Opcional- Solo en caso de clonar proyectos

Ejecutas el comando : pip install -r requirements.txt

Pero debes tener el entorno virtual activo

# AUTOMATIZAR CON ARCHIVOS BAT

## ACTIVAR ENTORNO VIRTUAL

Se puede automatizar la activación del entorno virtual

Cree un archivo llamado inicio.bat y dentro agregue:

Texto

El contenido generado por IA puede ser incorrecto.

En cmd solo ejecute inicio.bat y automáticamente se ejecutará el proyecto

## ACTUALIZAR REQUIREMENTS.TXT

Se puede automatizar la actualización del archivo requirements.txt

Cree un archivo llamado dependencias.bat y agregue el siguiente codigo

Texto

El contenido generado por IA puede ser incorrecto.

Al igual que el de inicio debe ejecutar en CMD de visual studio y esto hará la actualización.

# CREANDO MI PRIMER HOLA MUNDO

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.

Crea una archivo llamado app.py

Dentro del archivo escribe

Texto

El contenido generado por IA puede ser incorrecto.

Luego con el entorno virtual abierto ejecuta en la consola: streamlit run app.py

Texto

El contenido generado por IA puede ser incorrecto.

Cuando es primera vez pregunta si quieres recibir correos informativos acerca del framework yo en este caso puse mi correo y di enter

Aparece Texto

El contenido generado por IA puede ser incorrecto.

Y en el navegador se ejecuta localmente la app

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Okay hasta este punto ya ejecutamos un proyecto básico

TIPOS DE TEXTOS

Depende de la necesidad el framework ofrece distintos textos

Texto

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

USANDO VALORES DE VARIABLES EN EL TEXTO

Texto, Chat o mensaje de texto

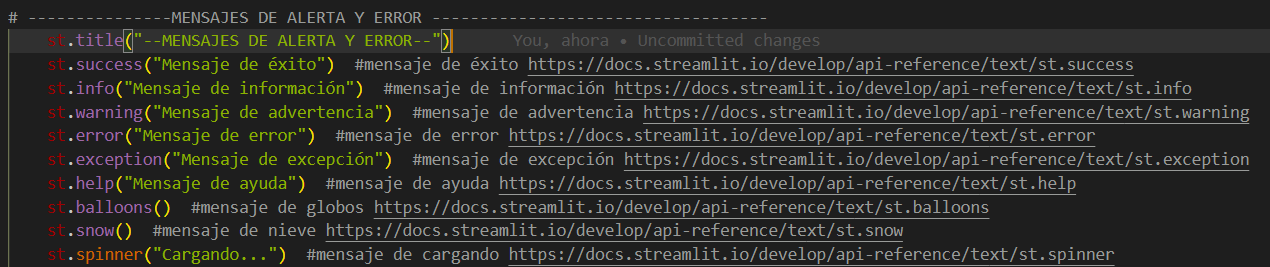
El contenido generado por IA puede ser incorrecto.

Tabla

El contenido generado por IA puede ser incorrecto.

MENSAJES DE ALERTA Y OTROS

No solo se pueden agregar mensajes simples también con una personalización por defecto



Escala de tiempo

El contenido generado por IA puede ser incorrecto.

Los cuales sirven para personalizar bien el website o dashboard

# MANEJO DE DATAFRAMES

Debe primero en la carpeta principal almacenar una base de datos puede ser csv, xlsx, etc

Luego crear el dataframe como normalmente se hace en Python

Para terminar en la función agregar st.dataframe(df), esto mostrara en dataframe completo en el proyecto

Texto

El contenido generado por IA puede ser incorrecto.

Posiblemente si su archivo es xlsx necesite instalar pip install openpyxl

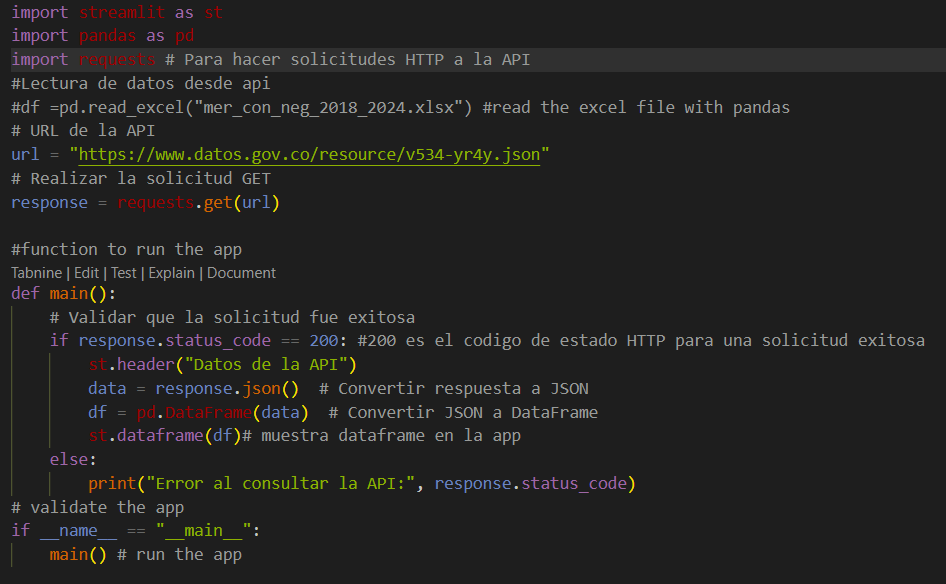
Si lo instala no olvide actualizar requirements.txt

Interfaz de usuario gráfica, Tabla

El contenido generado por IA puede ser incorrecto.

# LEER DATOS DESDE API DATYOS ABIERTOS COLOMBIA

La única diferencia entre el anterior caso y este es que en lugar de almacenar un base de datos usamos una api externa para traer datos json y convertirlos a dataframe para mostrarlo



Si desea ver los 5 primeros registros como en colab o jupyther basta con agregar Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

# MOSTRAR DATOS JSON

Con estas líneas dentro de la función pero fuera del IF puede agregar visualización de JS en la app

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

# MOSTRAR CODIGO DENTRO DE LA APP

Exactamente lo mismo que lo anterior, agregar el código dentro de la función, pero por fuera del if

Texto

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

# CAJA DE SELECCIÓN

Texto

El contenido generado por IA puede ser incorrecto.

Patrón de fondo

El contenido generado por IA puede ser incorrecto.

## SELECCIÓN MULTIPLE

Texto

El contenido generado por IA puede ser incorrecto.

Imagen que contiene Texto

El contenido generado por IA puede ser incorrecto.

## DESLIZADOR

Texto

El contenido generado por IA puede ser incorrecto.

Imagen que contiene Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

SELECT SLIDER

Texto

El contenido generado por IA puede ser incorrecto.

Forma

El contenido generado por IA puede ser incorrecto.

# MANEJO DE ARCHIVOS MULTIMEDIA

Imágenes

Copie y pegue dentro del proyecto múltiples archivos multimedía, preferiblemente dentro de una carpeta en mi caso la llamé mmultimedia.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

En el archivo app.py en la función principal agregue el siguiente código

Texto

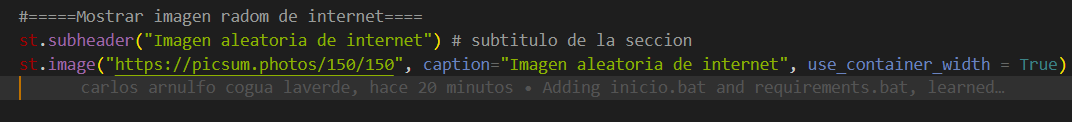
El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

IMAGEN ALEATORIA DE INTERNET

Usando la siguiente página web podemos obtener imágenes aleatorias



Un atardecer de colores

El contenido generado por IA puede ser incorrecto.

VIDEOS

Con el siguiente código se pueden insertar videos

Texto

El contenido generado por IA puede ser incorrecto.

Imagen que contiene reloj

El contenido generado por IA puede ser incorrecto.

AUDIO

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Rectángulo

El contenido generado por IA puede ser incorrecto.

INPUTS

Texto

El contenido generado por IA puede ser incorrecto.

Imagen que contiene Aplicación

El contenido generado por IA puede ser incorrecto.

TEXTO AREA O TEXTO LARGO

Texto

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

INPUTS DE NUMERICOS

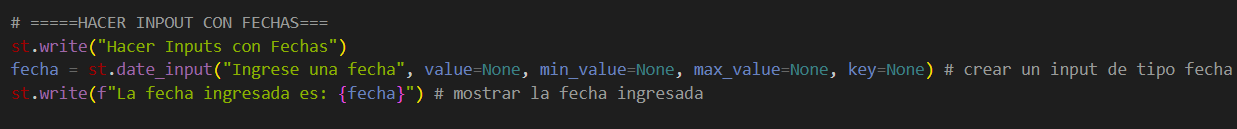
Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

INPUTS DE FECHAS



Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

INPUT CON HORA

Texto

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

INPUT CON COLORES

Texto

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación, Teams

El contenido generado por IA puede ser incorrecto.

PERSONALIZACIÓN DE LA APP

Configuración de nombre de la app



Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Icono de la pagina

Texto

El contenido generado por IA puede ser incorrecto.

Personalizar Icono

Primero debe copiar el archivo de icono dentro de la carpeta multimedia

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Luego al principio es decir fuera de la función principal importar la librería Pillow

Crear el objeto icono y poner la tura del archivo

Agregar la configuración de la pagina y dentro en el parámetro page\_icon agregar la variable llamada icono

Texto

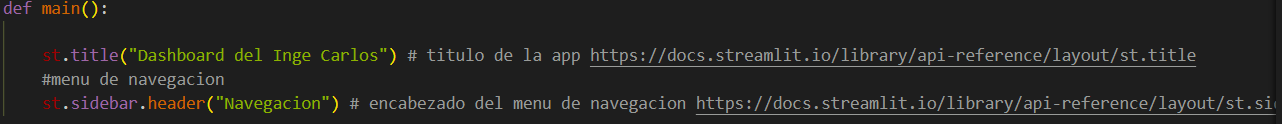
El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

AGREGANDO SIDEBAR MENU

Dentro de la función principal debe agregar la función .header



Esto dará como resultado

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Po defecto la función muestra el menú fijo si quiere que se oculte debe configurarlo en la función set\_page\_config

Texto

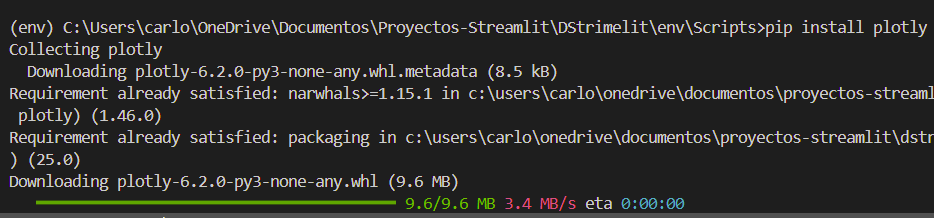
El contenido generado por IA puede ser incorrecto.

Agrega la propiedad initial\_sidebar\_state y dejarlo en modo colapsado

GRAFICAR CON PLOTLIB

Primero debe instalar la dependencia en el cdm de vs code con entorno virtual encendido y dentro de la carpeta scripts

Pip install plotly



Después debe en app.py llamar a la Liberia

En otro CMD ejecute dependencias.bat para actualizar el archivo requirements.txt

Después de esto en su archivo app.y en la parte superior debe importar las siguientes librerías



Dentro de la función principal debe agregar

Texto

El contenido generado por IA puede ser incorrecto.

Esto leerá el Excel que está en el proyecto como un dataframe

Tabla

El contenido generado por IA puede ser incorrecto.

Ahora esta información esta en crudo lo que podemos hacer para graficar es agrupar

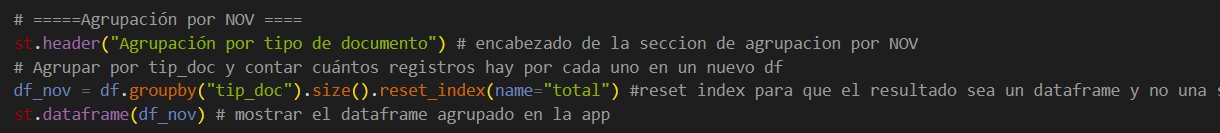
En este caso agruparemos por tipo de documento, queremos saber la cantidad por tipo de documento

Texto

El contenido generado por IA puede ser incorrecto.

AGRUPAMIENTO GROUPBY()

Usamos la función groupby ()



Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Básicamente lo anterior crea un nuevo dataframe y dentro almacena el valor de la agrupación

GRAFICAR CON PIE()

Para finalizar graficamos con diagrama de pastel usando la función pie(), llamamos el dataframe nuevo donde esta la agrupación y configuramos sus parametros

Texto

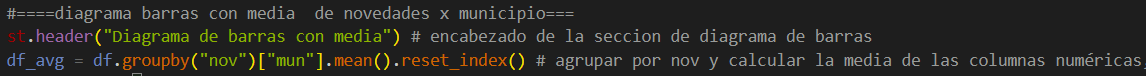
El contenido generado por IA puede ser incorrecto.

Gráfico, Gráfico circular

El contenido generado por IA puede ser incorrecto.

GRAFICO DE BARRAS BAR()

Crea un nuevo df llamado df\_avg y usa la función groupby() en este caso la columna obajetivo y en [] la columna numérica para promediar



Creación del diagrama

Texto

El contenido generado por IA puede ser incorrecto.

Gráfico, Gráfico de barras

El contenido generado por IA puede ser incorrecto.

MANEJO DE ARCHIVOS

Instalar las dependencias

pip install openpyxl docx2txt PyPDF2

ejecutar el bat de dependencias

Importar las librerías

Texto

El contenido generado por IA puede ser incorrecto.

CARGAR IMAGENES

Antes de la función main cree un decorador y seguido de este un función para cargar de archivo en este caso image\_file

Texto

El contenido generado por IA puede ser incorrecto.

CARGAR PDF

Ahora cree otra función aparte para leer archivos en pdf

Texto

El contenido generado por IA puede ser incorrecto.

CONFIGURAR LA CARGA

Dentro de la función principal main() agrege una validación para saber que tipo de archivo selecciono el usuario y así permitirle el cargue, Cree una variable que se llama archivo\_imagen y configurarla para aceptar 3 tipos de formatos JPGJ, JPEG y PNG dentro de esta validación cree otra validación anidada dónde sí esta variable archivo\_imagen está variable no está vacía, entonces cree una variable llamada detalle de archivo y dentro de ella creé un diccionario con (nombre de archivo tipo de archivo y tamaño de archivo) después muestre el detalle del archivo y después con la función image muestre el archivo de imagen

Imagen que contiene Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

CARGUE DE ARCHIVO DE EXCEL

Igual que en el ejercicio anterior dentro de la función principal y dentro de la validación agregue un elif donde si la selección del usuario es (Excel) entonces permita el cargue del archivo en tipo csv, xls, oxl, o cualquier otro archivo de Excel , cree otro if dentro del elif donde si el archivo Excel no está vacío entonces capture todos los detalles del archivo y luego muestre los detalles después de eso dentro de if, el sistema por sí solo no permite identificar o validar cualquier archivo de Excel por eso hay que hacer una validación dentro. C}Cree otro if donde valide el tipo de archivo\_Excel para poder validar si es separado por comas, xls y al finalizar si no se cumple ninguna de las condiciones crea un data frame en vacío para evitar un error y después muestra el data frame saliendo de la validación

Texto

El contenido generado por IA puede ser incorrecto.