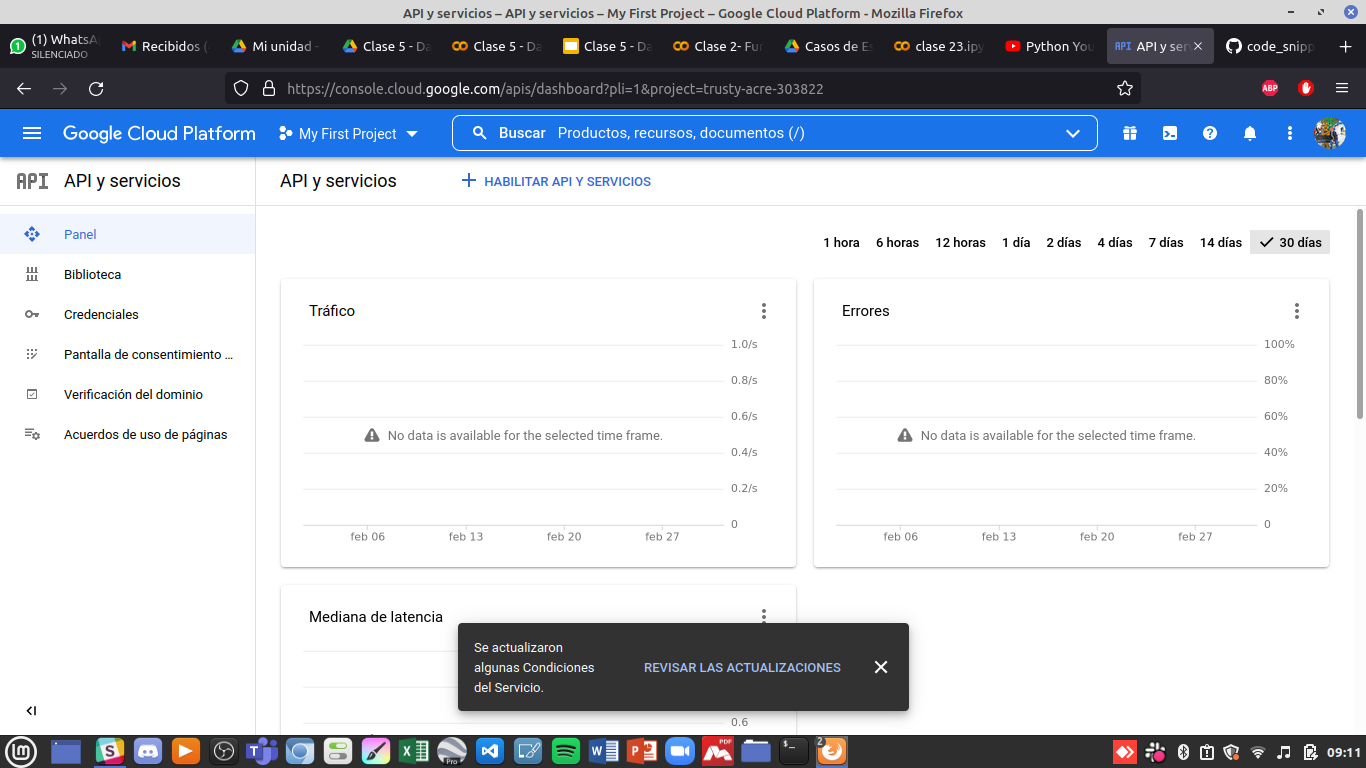
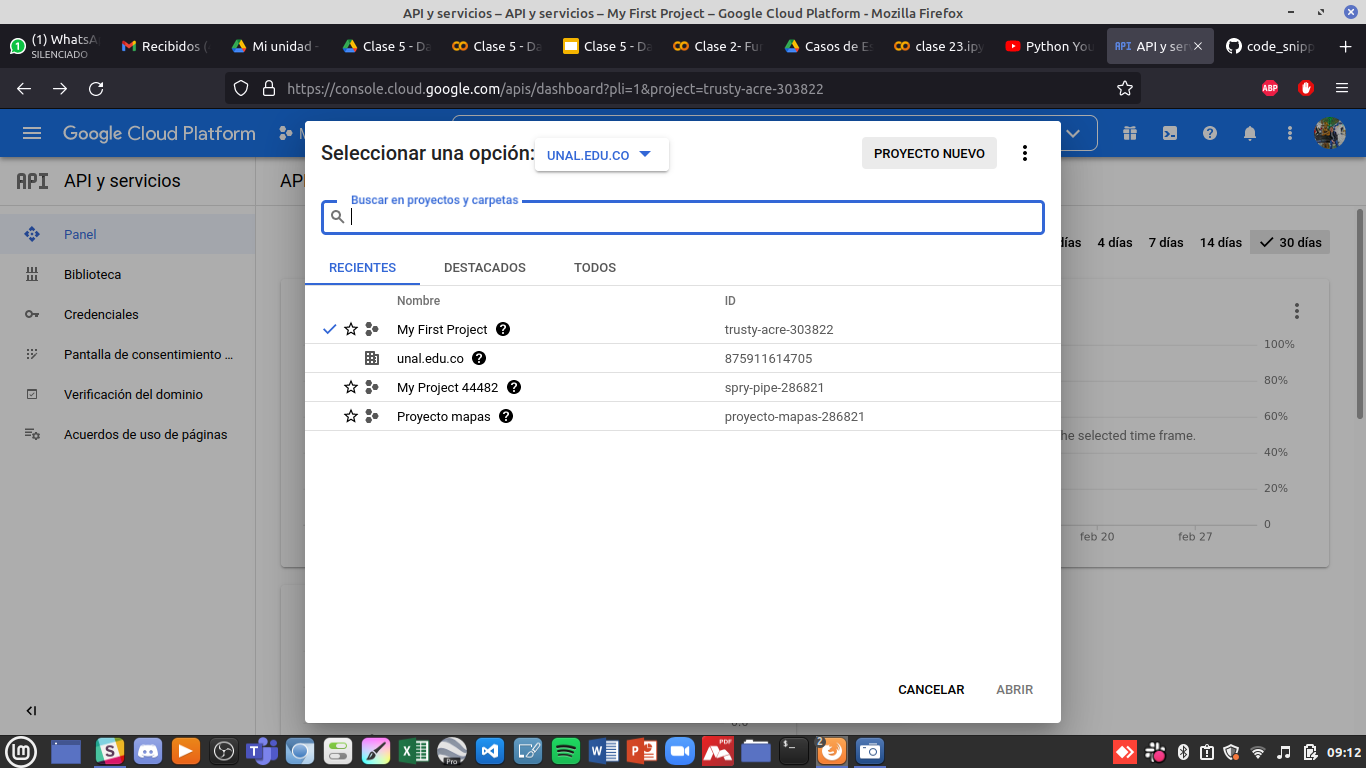
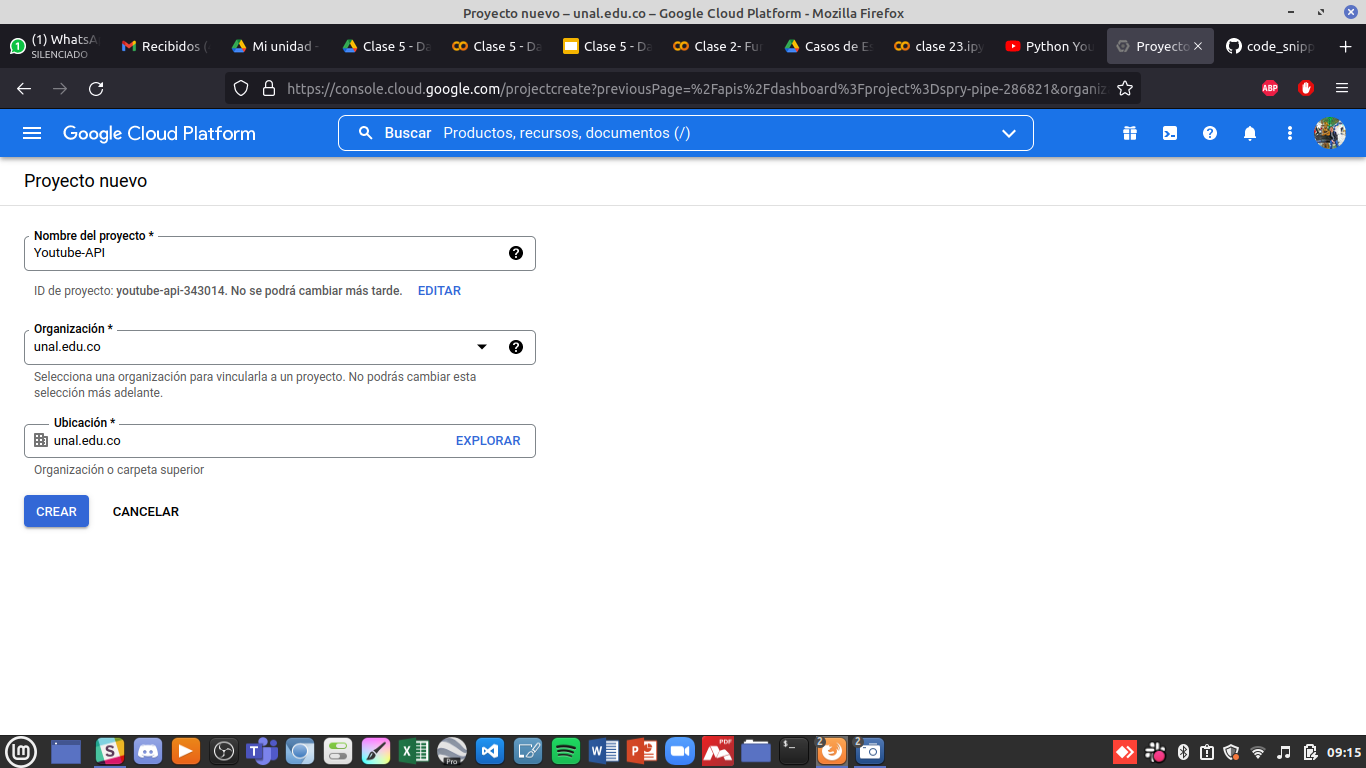
**Cómo obtener API Key para la API de Youtube:**

Ir a la siguiente página: [https://console.developers.google.com/](https://colab.research.google.com/corgiredirector?site=https%3A%2F%2Fconsole.developers.google.com%2F) Elegir el botón My First Project ****

Elegir la opcion Proyecto nuevo: ****

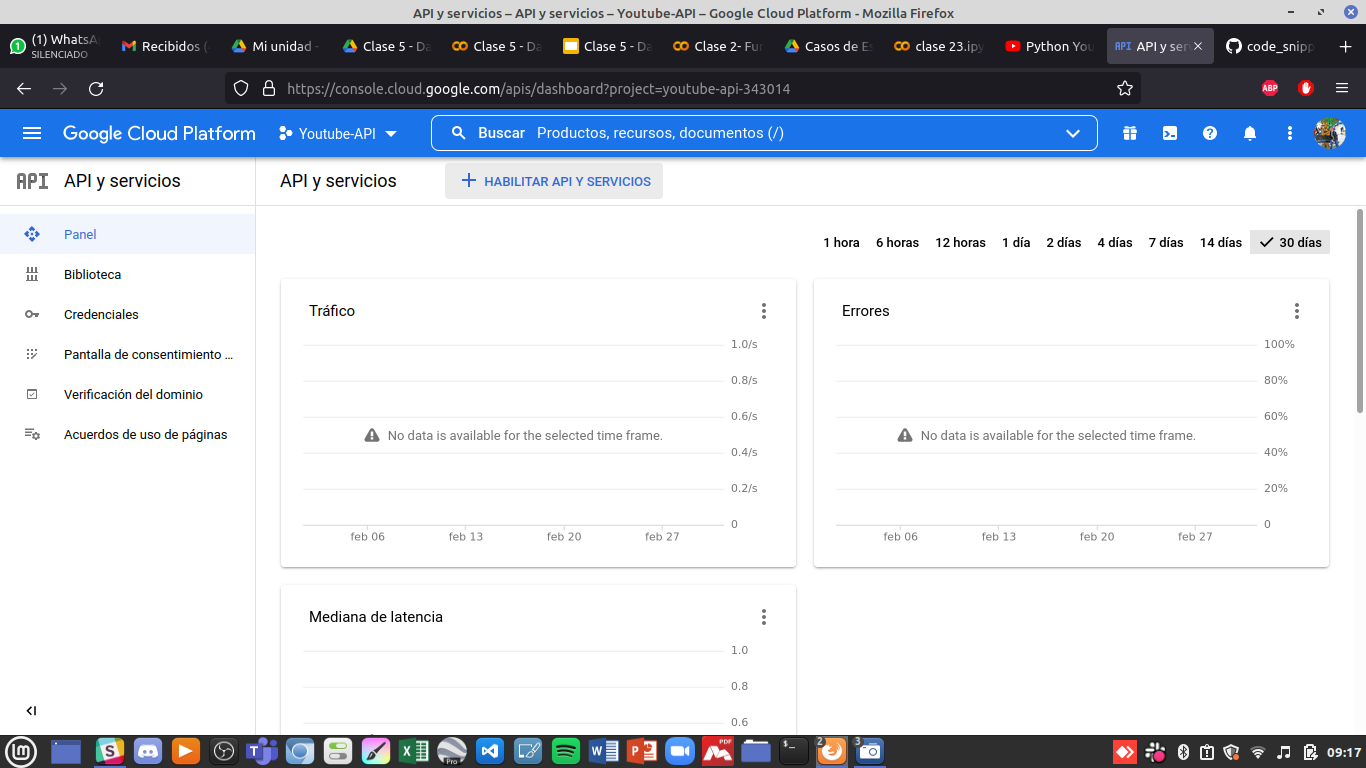
Colocar el nombre del proyecto: Youtube-API

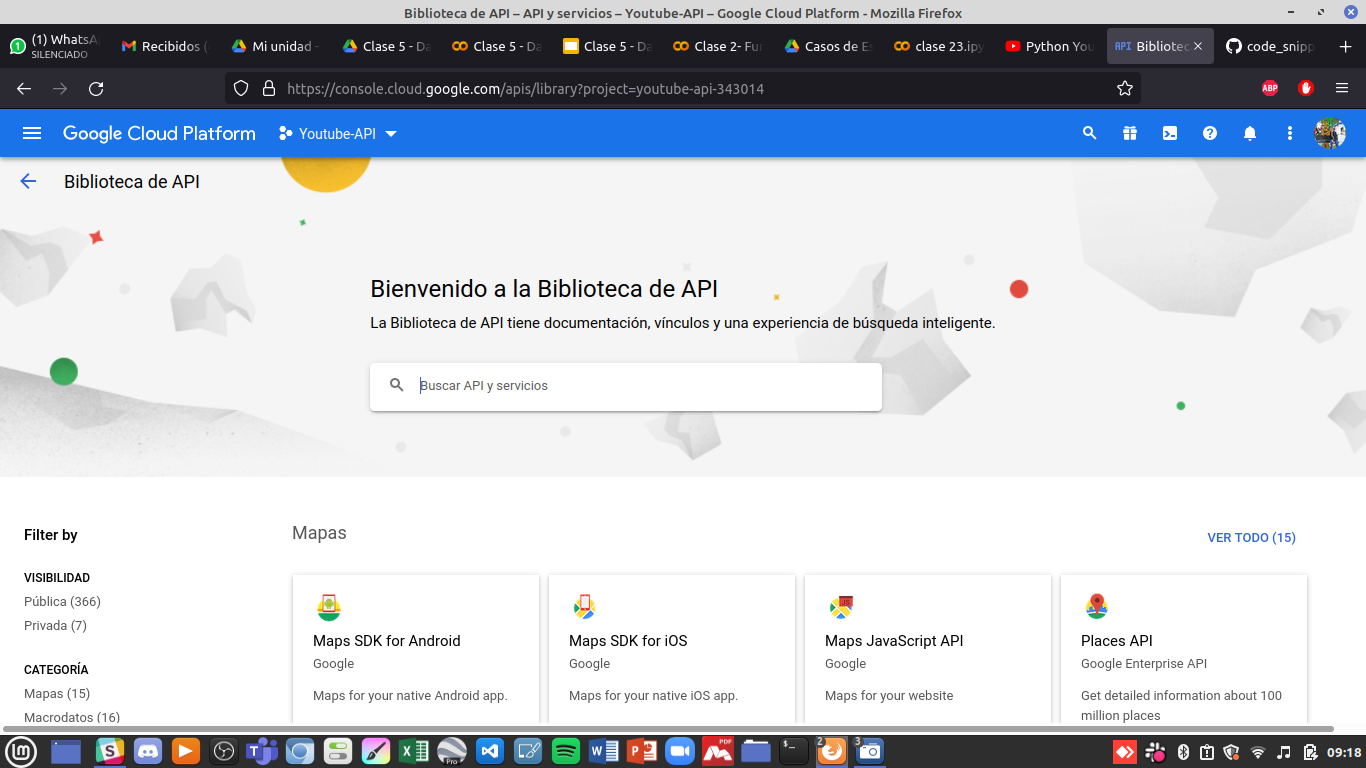
Pueden elegir la opcion Organización: Sin organización

****

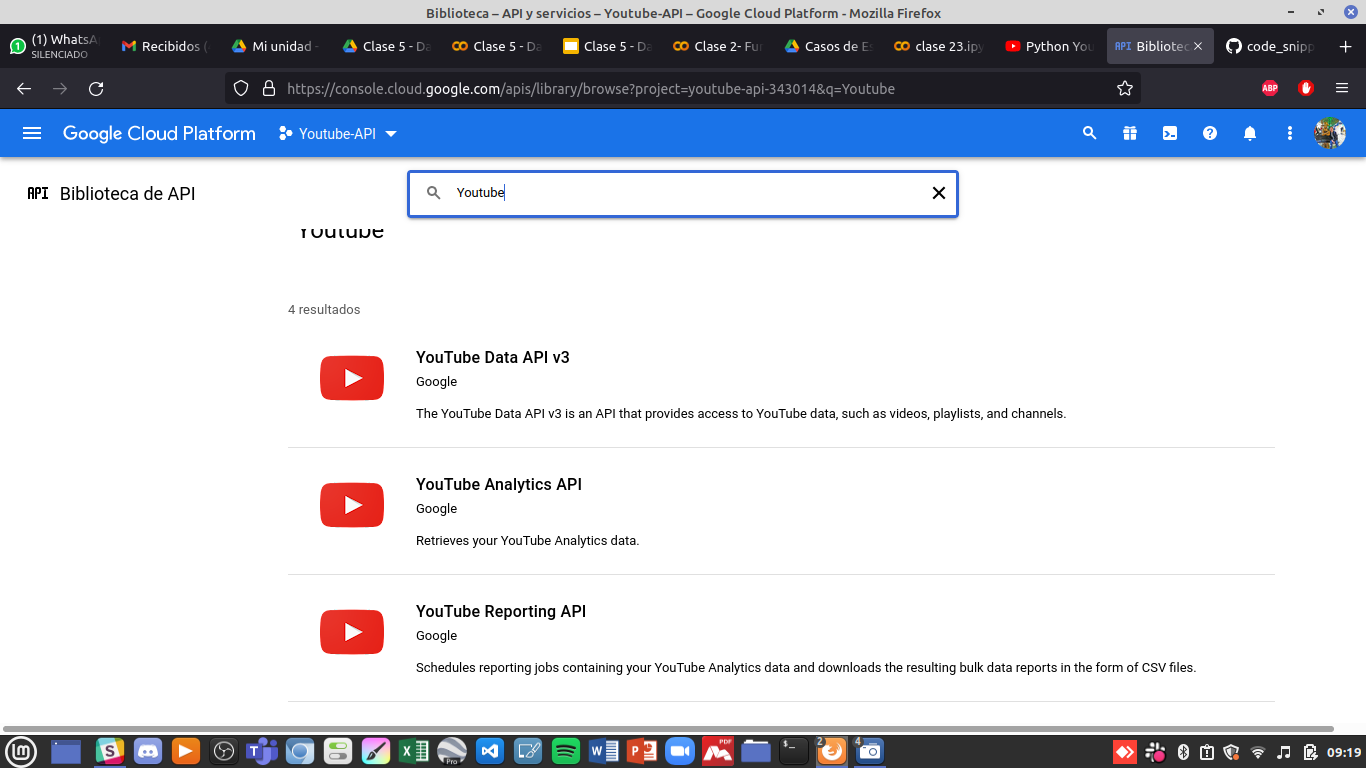
Luego dar click en Crear

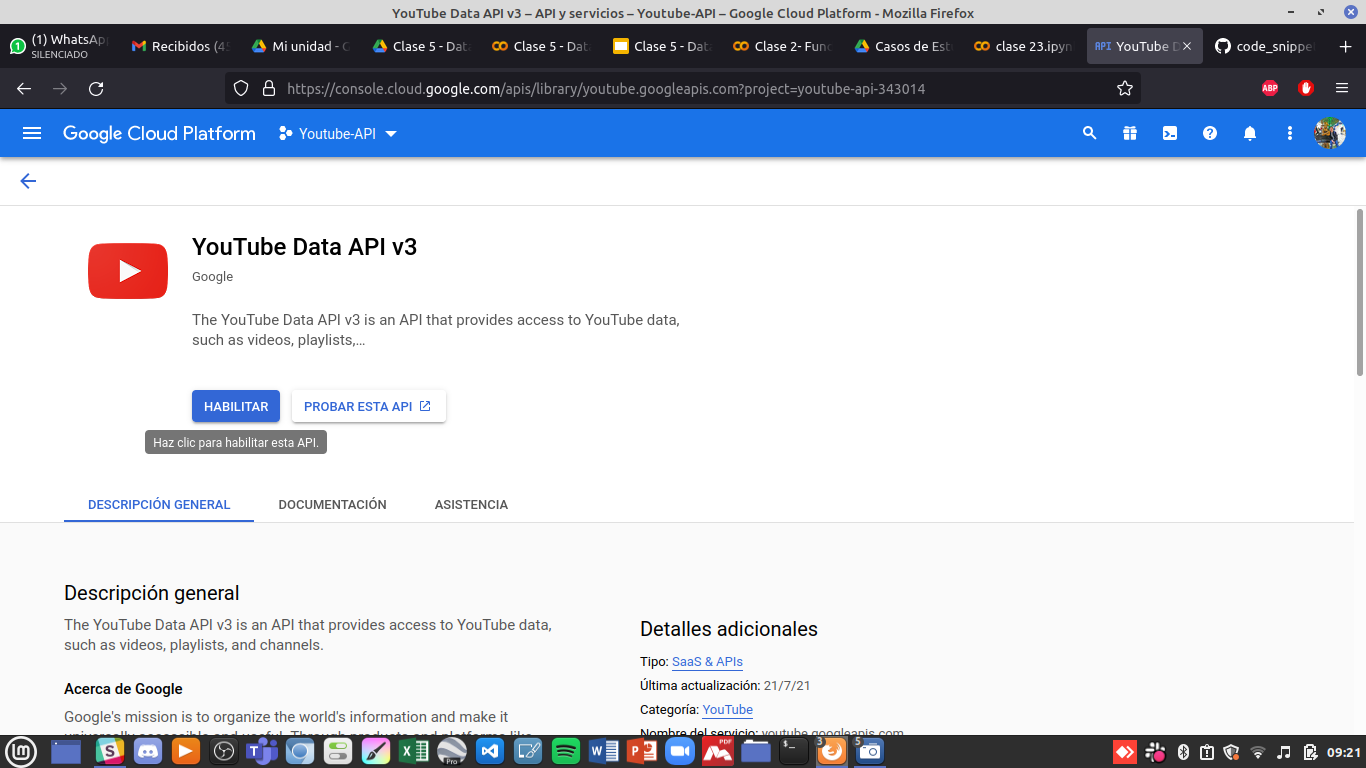
Es bueno recordar que Google tiene multiples API pero en este caso particular estaremos viendo el ejemplo con la API de Youtube

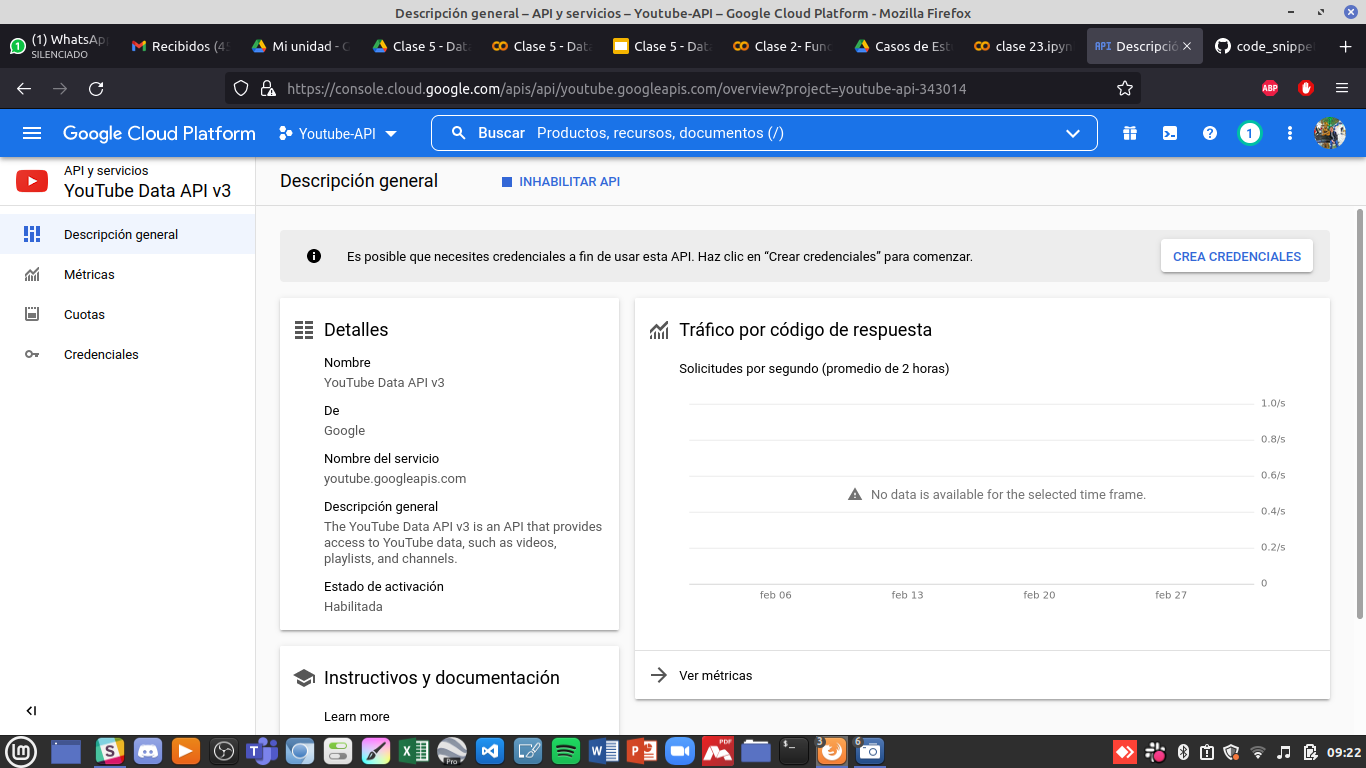
Al momento de crear y seleccionar el proyecto deberan ver un mensaje que dice + habilitar API y Servicios ****

Damos click en ese boton y debera aparecer una ventana como la siguiente: ****

Buscamos la API de Youtube escribiendo Youtube en la barra de navegacion y damos enter:

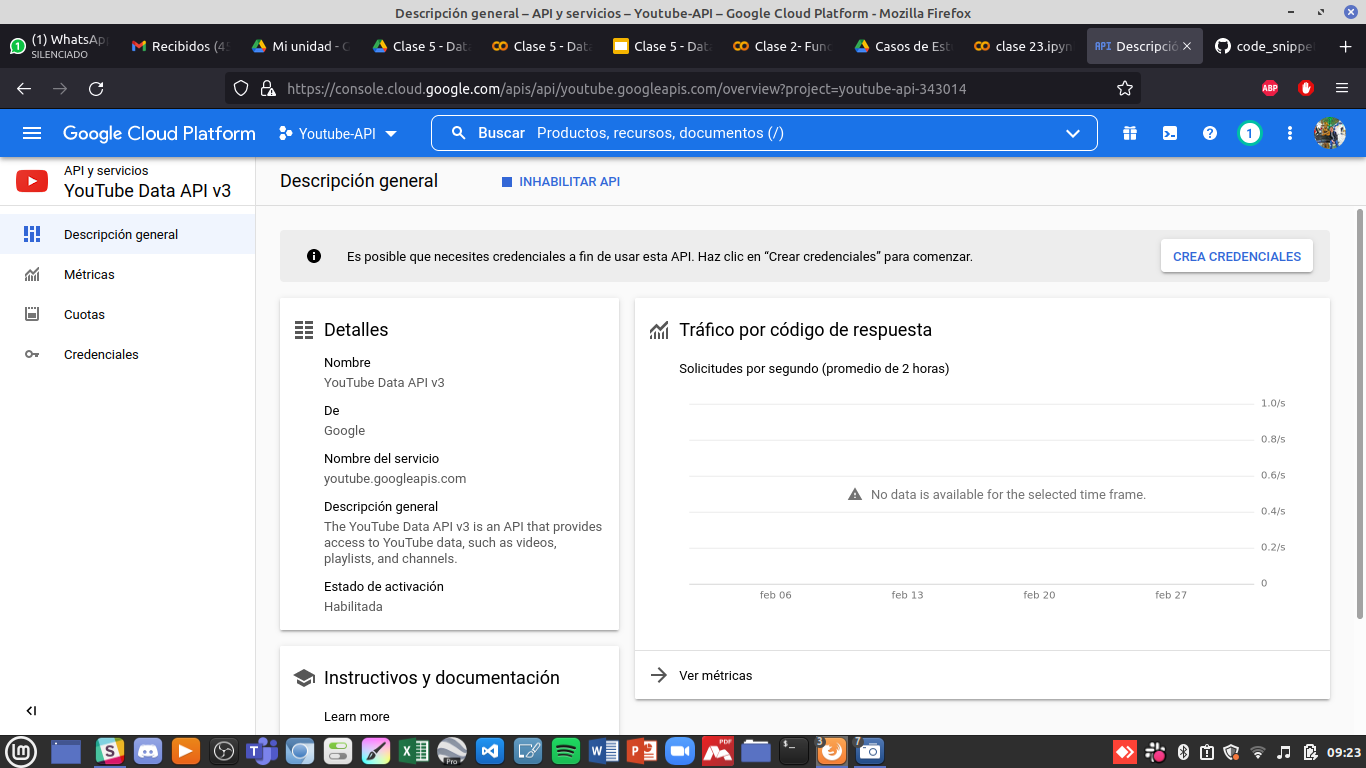
Apareceran diversas APIS: ****

Para este caso particular vamos a usar la API: Youtube Data API v3 la seleccionamos, luego damos clieck en el botón Habilitar ****

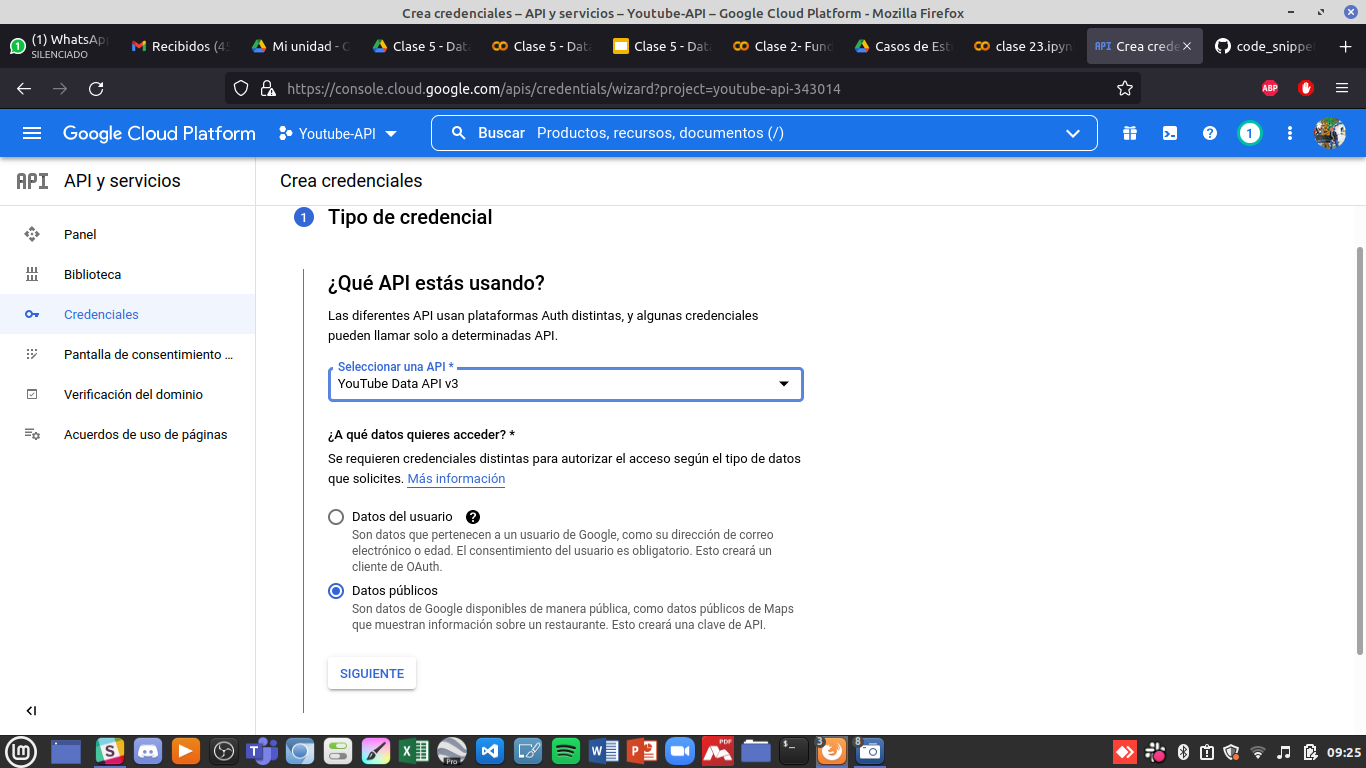
Aparecera una ventana como la siguiente: ****

Ahora que ya tenemos habilitada la API solo debemos crear una API Key

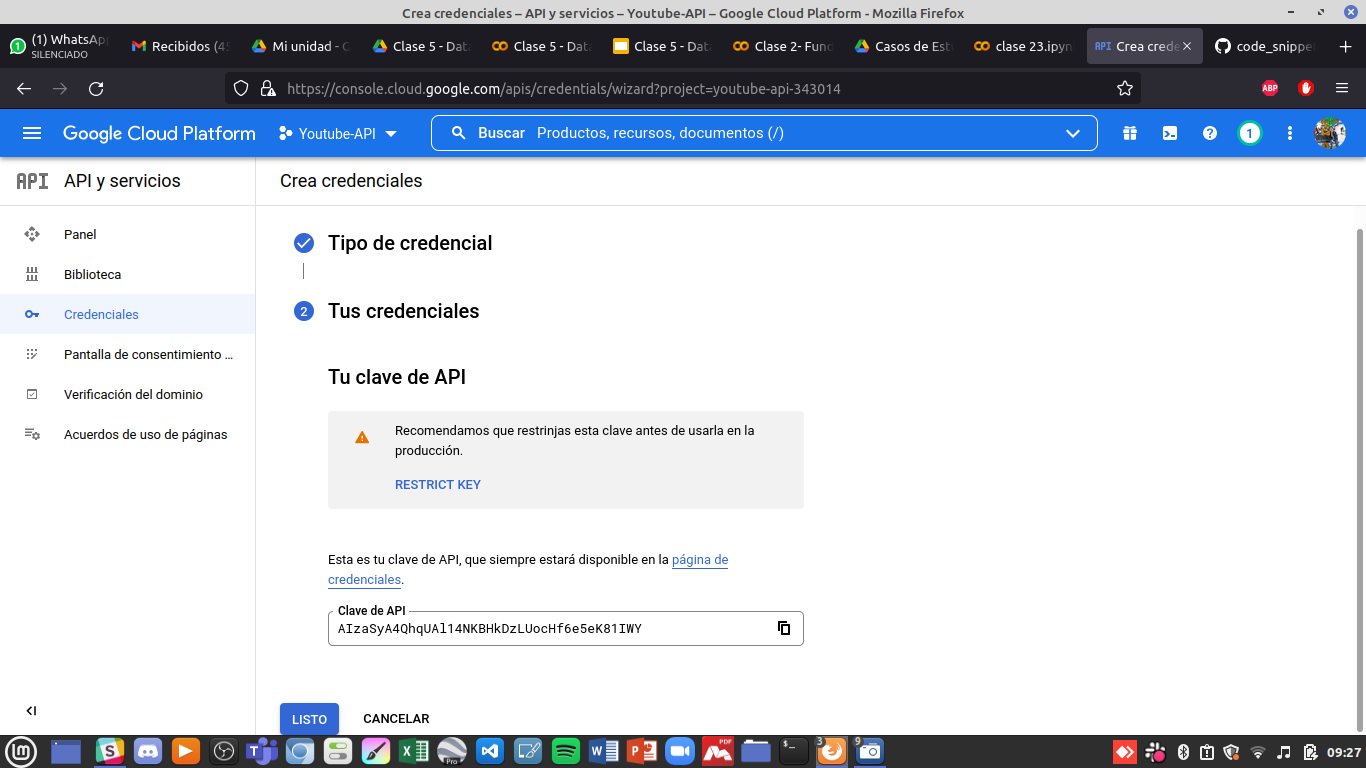
Damos click en el boton Crea Credenciales

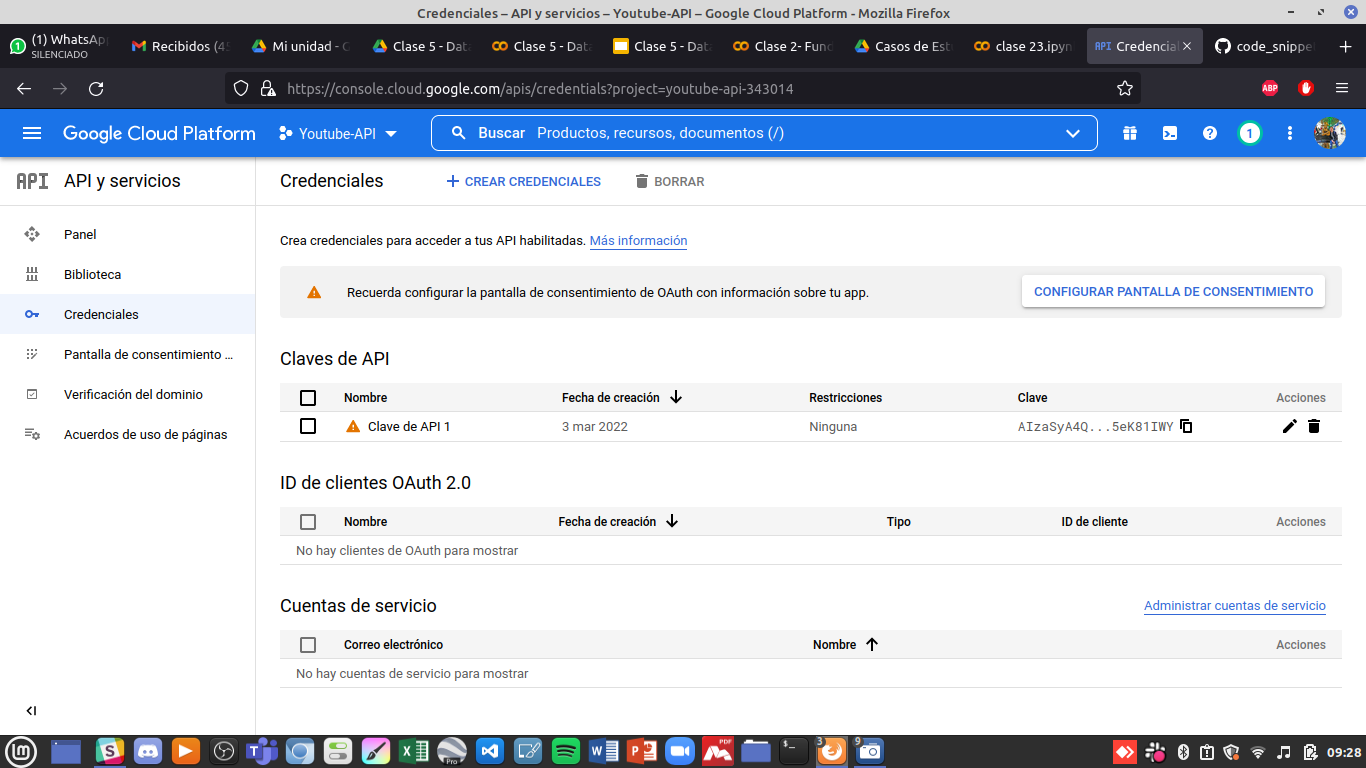
****

Deberemos hacer la siguiente seleccion:

****

Damos click en Siguiente y debera aparecer la siguiente ventana con la respectiva API key:

****

Copiamos la API key y damos click en Done (Listo) y aparecera la siguiente ventana: ****

Guargemos la API en una variable string

[ ]

api\_key='AIzaSyAVxBti356P\_n0VOT8hup1Ggj5Pv36JATM' # La borrare despues asi que no se preocupen  
# Pueden guardar esta key en un archivo encriptado o en un archivo de configuration para que no se puede extraer de ningun lado

Para obtener documentacion de la API de google pueden ir a: <https://github.com/googleapis/google-api-python-client>

[ ]

# Instalando libreria  
!pip install google-api-python-client

Collecting google-api-python-client

Downloading google\_api\_python\_client-2.90.0-py2.py3-none-any.whl (11.4 MB)

---------------------------------------- 11.4/11.4 MB 9.2 MB/s eta 0:00:00

Collecting httplib2<1.dev0,>=0.15.0 (from google-api-python-client)

Downloading httplib2-0.22.0-py3-none-any.whl (96 kB)

---------------------------------------- 96.9/96.9 kB ? eta 0:00:00

Collecting google-auth<3.0.0.dev0,>=1.19.0 (from google-api-python-client)

Downloading google\_auth-2.21.0-py2.py3-none-any.whl (182 kB)

-------------------------------------- 182.1/182.1 kB 5.4 MB/s eta 0:00:00

Collecting google-auth-httplib2>=0.1.0 (from google-api-python-client)

Downloading google\_auth\_httplib2-0.1.0-py2.py3-none-any.whl (9.3 kB)

Collecting google-api-core!=2.0.\*,!=2.1.\*,!=2.2.\*,!=2.3.0,<3.0.0.dev0,>=1.31.5 (from google-api-python-client)

Downloading google\_api\_core-2.11.1-py3-none-any.whl (120 kB)

-------------------------------------- 120.5/120.5 kB 7.4 MB/s eta 0:00:00

Collecting uritemplate<5,>=3.0.1 (from google-api-python-client)

Downloading uritemplate-4.1.1-py2.py3-none-any.whl (10 kB)

Collecting googleapis-common-protos<2.0.dev0,>=1.56.2 (from google-api-core!=2.0.\*,!=2.1.\*,!=2.2.\*,!=2.3.0,<3.0.0.dev0,>=1.31.5->google-api-python-client)

Downloading googleapis\_common\_protos-1.59.1-py2.py3-none-any.whl (224 kB)

------------------------------------- 224.5/224.5 kB 13.4 MB/s eta 0:00:00

Collecting protobuf!=3.20.0,!=3.20.1,!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<5.0.0.dev0,>=3.19.5 (from google-api-core!=2.0.\*,!=2.1.\*,!=2.2.\*,!=2.3.0,<3.0.0.dev0,>=1.31.5->google-api-python-client)

Downloading protobuf-4.23.3-cp38-cp38-win\_amd64.whl (422 kB)

-------------------------------------- 422.4/422.4 kB 6.5 MB/s eta 0:00:00

Requirement already satisfied: requests<3.0.0.dev0,>=2.18.0 in c:\users\kapla\anaconda3\lib\site-packages (from google-api-core!=2.0.\*,!=2.1.\*,!=2.2.\*,!=2.3.0,<3.0.0.dev0,>=1.31.5->google-api-python-client) (2.28.2)

Collecting cachetools<6.0,>=2.0.0 (from google-auth<3.0.0.dev0,>=1.19.0->google-api-python-client)

Downloading cachetools-5.3.1-py3-none-any.whl (9.3 kB)

Collecting pyasn1-modules>=0.2.1 (from google-auth<3.0.0.dev0,>=1.19.0->google-api-python-client)

Downloading pyasn1\_modules-0.3.0-py2.py3-none-any.whl (181 kB)

------------------------------------- 181.3/181.3 kB 10.7 MB/s eta 0:00:00

Collecting rsa<5,>=3.1.4 (from google-auth<3.0.0.dev0,>=1.19.0->google-api-python-client)

Downloading rsa-4.9-py3-none-any.whl (34 kB)

Requirement already satisfied: six>=1.9.0 in c:\users\kapla\anaconda3\lib\site-packages (from google-auth<3.0.0.dev0,>=1.19.0->google-api-python-client) (1.15.0)

Requirement already satisfied: urllib3<2.0 in c:\users\kapla\anaconda3\lib\site-packages (from google-auth<3.0.0.dev0,>=1.19.0->google-api-python-client) (1.26.4)

Requirement already satisfied: pyparsing!=3.0.0,!=3.0.1,!=3.0.2,!=3.0.3,<4,>=2.4.2 in c:\users\kapla\anaconda3\lib\site-packages (from httplib2<1.dev0,>=0.15.0->google-api-python-client) (2.4.7)

Collecting pyasn1<0.6.0,>=0.4.6 (from pyasn1-modules>=0.2.1->google-auth<3.0.0.dev0,>=1.19.0->google-api-python-client)

Downloading pyasn1-0.5.0-py2.py3-none-any.whl (83 kB)

---------------------------------------- 83.9/83.9 kB ? eta 0:00:00

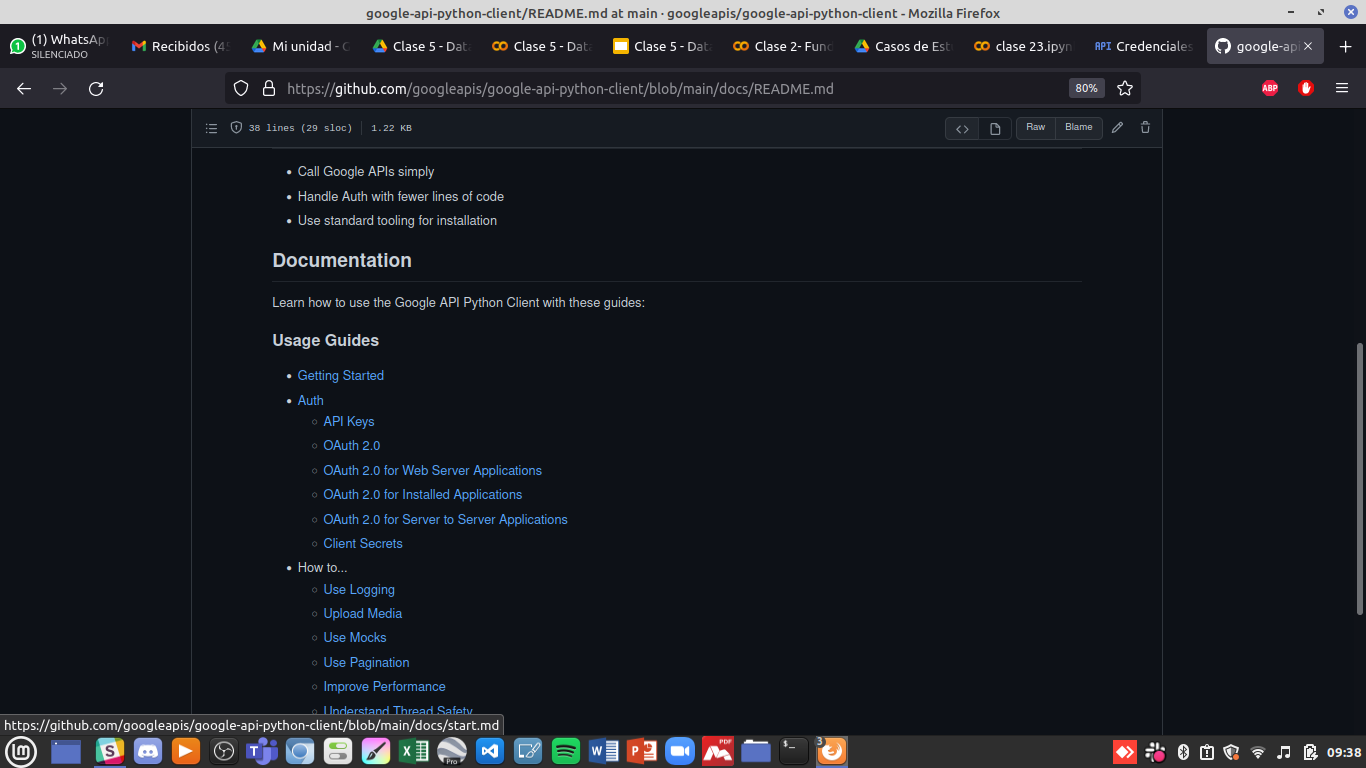
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\kapla\anaconda3\lib\site-packages (from requests<3.0.0.dev0,>=2.18.0->google-api-core!=2.0.\*,!=2.1.\*,!=2.2.\*,!=2.3.0,<3.0.0.dev0,>=1.31.5->google-api-python-client) (3.0.1)

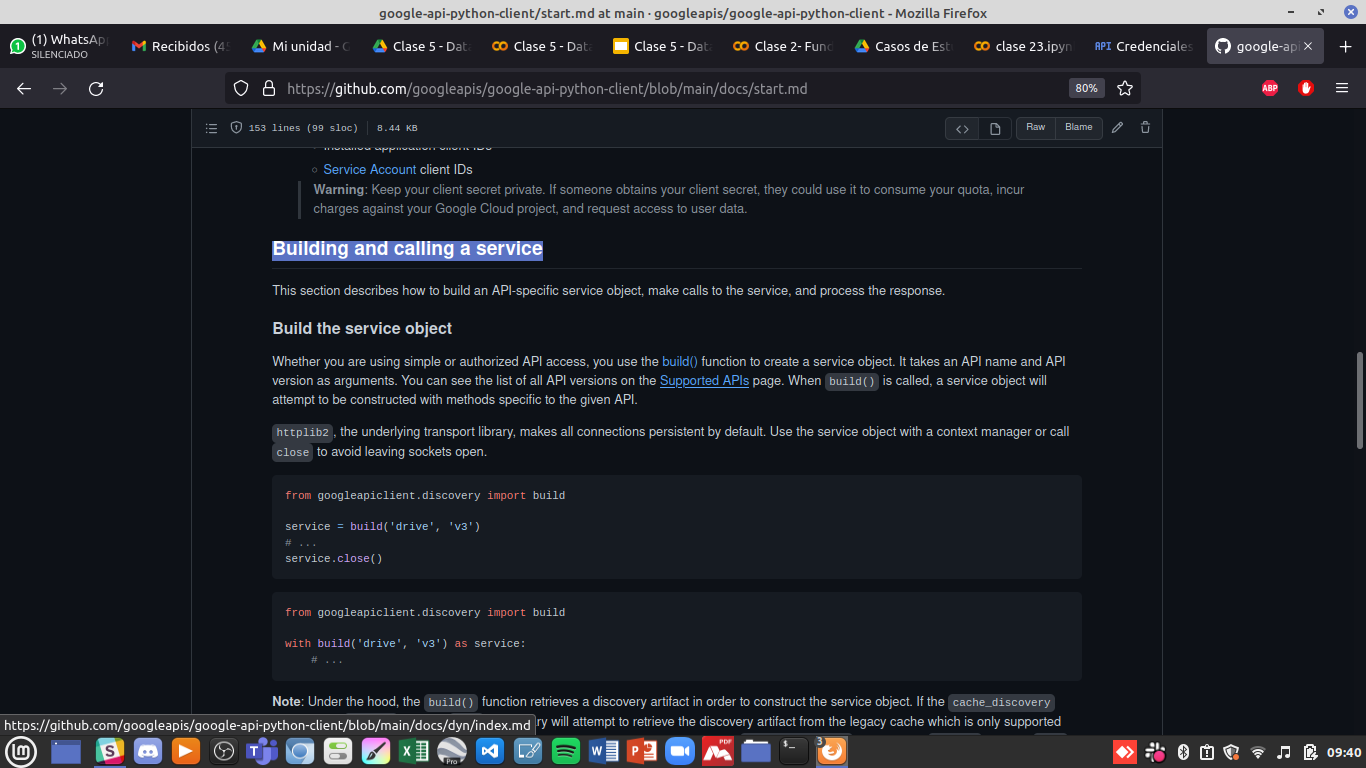
Requirement already satisfied: idna<4,>=2.5 in c:\users\kapla\anaconda3\lib\site-packages (from requests<3.0.0.dev0,>=2.18.0->google-api-core!=2.0.\*,!=2.1.\*,!=2.2.\*,!=2.3.0,<3.0.0.dev0,>=1.31.5->google-api-python-client) (2.10)

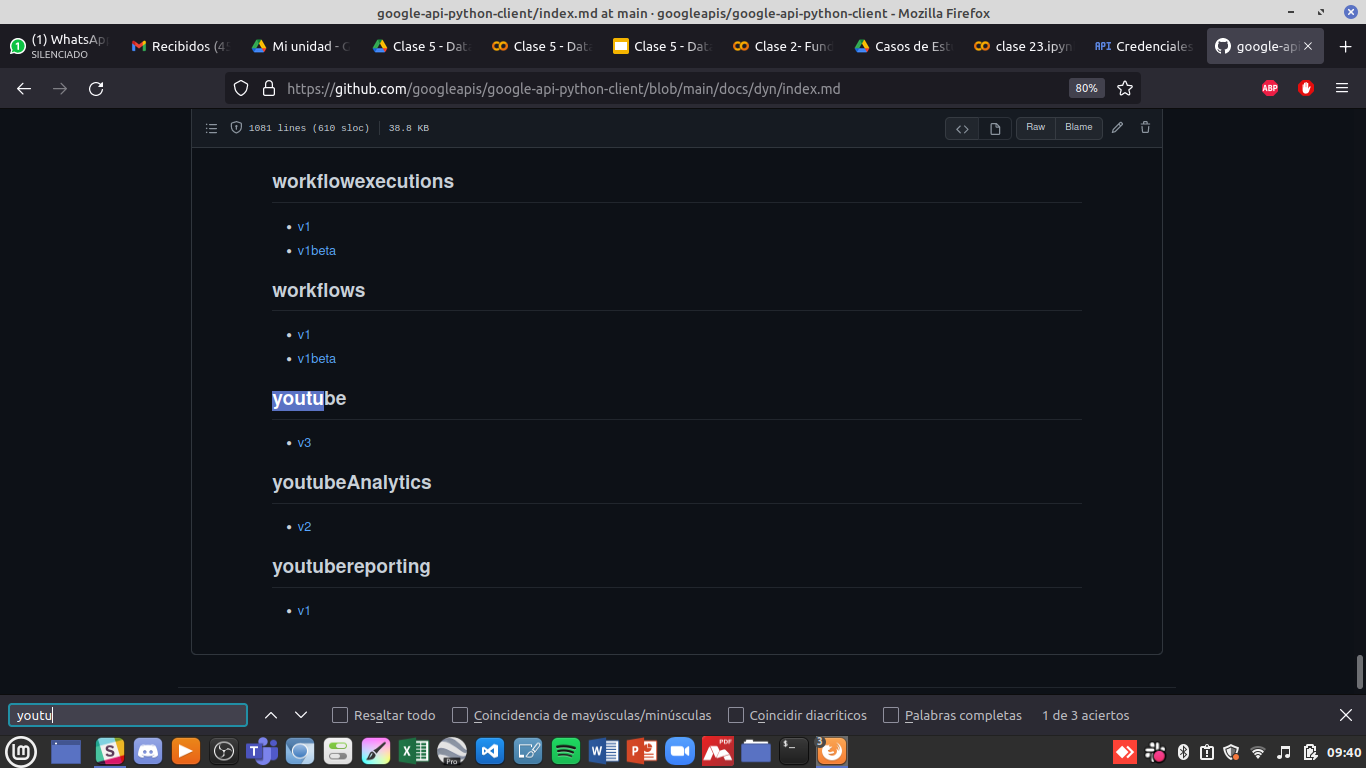
Requirement already satisfied: certifi>=2017.4.17 in c:\users\kapla\anaconda3\lib\site-packages (from requests<3.0.0.dev0,>=2.18.0->google-api-core!=2.0.\*,!=2.1.\*,!=2.2.\*,!=2.3.0,<3.0.0.dev0,>=1.31.5->google-api-python-client) (2020.12.5)

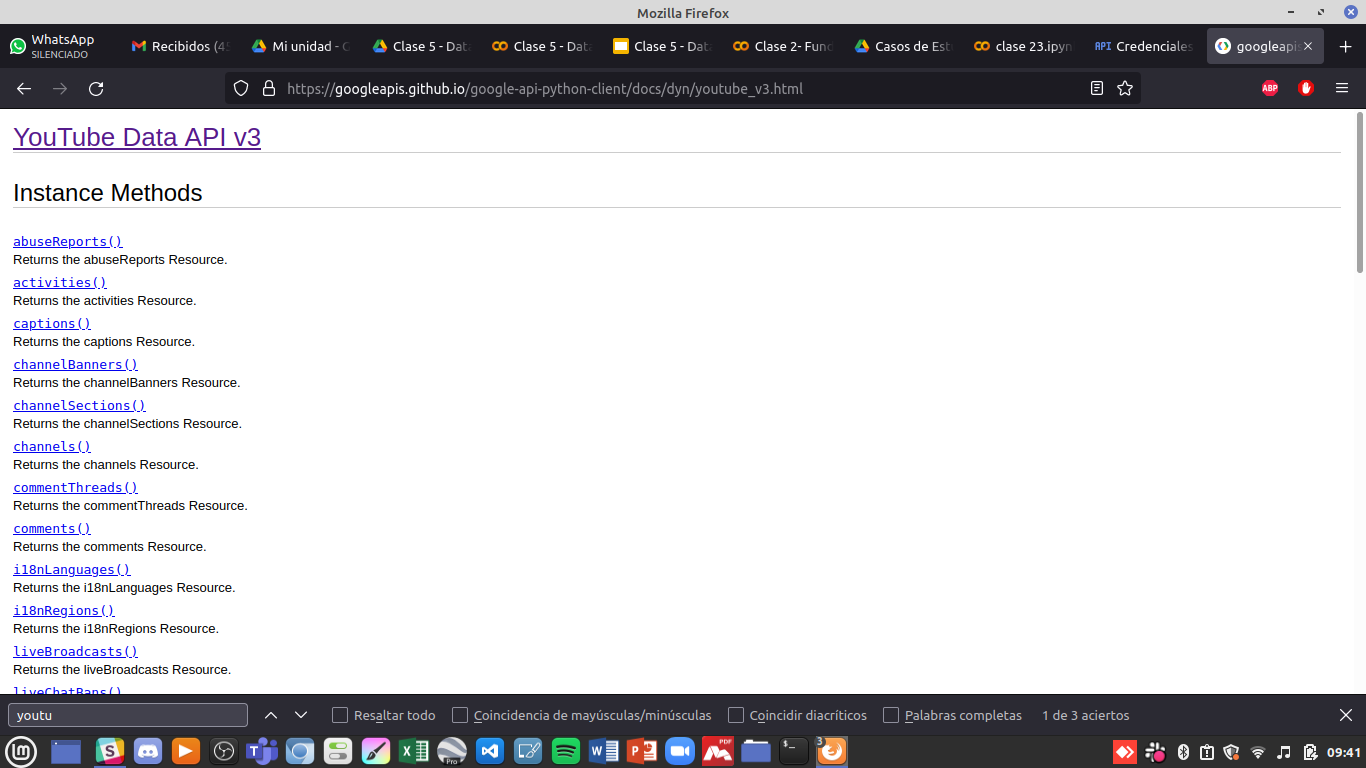
Installing collected packages: uritemplate, pyasn1, protobuf, httplib2, cachetools, rsa, pyasn1-modules, googleapis-common-protos, google-auth, google-auth-httplib2, google-api-core, google-api-python-client

Successfully installed cachetools-5.3.1 google-api-core-2.11.1 google-api-python-client-2.90.0 google-auth-2.21.0 google-auth-httplib2-0.1.0 googleapis-common-protos-1.59.1 httplib2-0.22.0 protobuf-4.23.3 pyasn1-0.5.0 pyasn1-modules-0.3.0 rsa-4.9 uritemplate-4.1.1

Vamos a la seccion docs en la pagina de Github Seleccionamos la Opcion Docs> Getting Started ****

Buscamos la seccion Building and calling a service y seleccionamos el boton Supported APIS ****

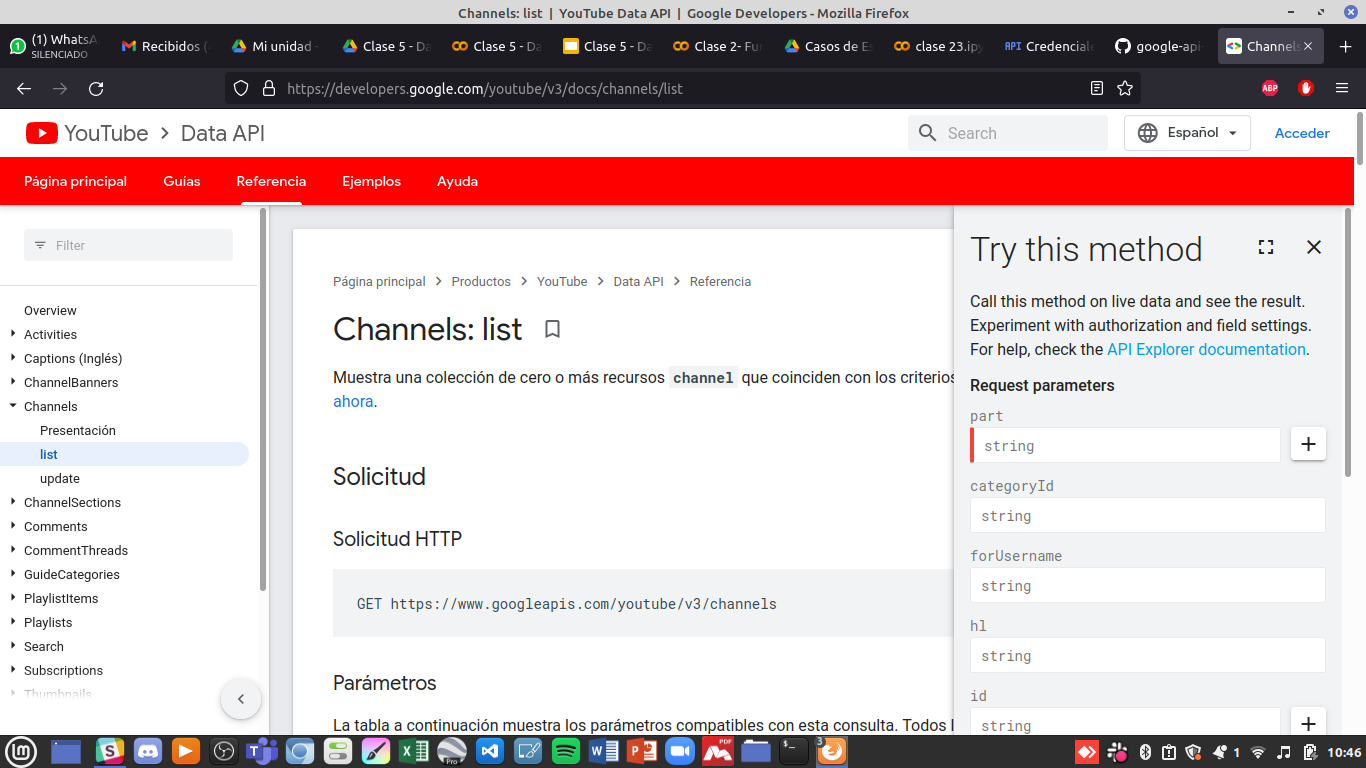
Si buscamos la opcion youtube veremos lo siguiente: ****

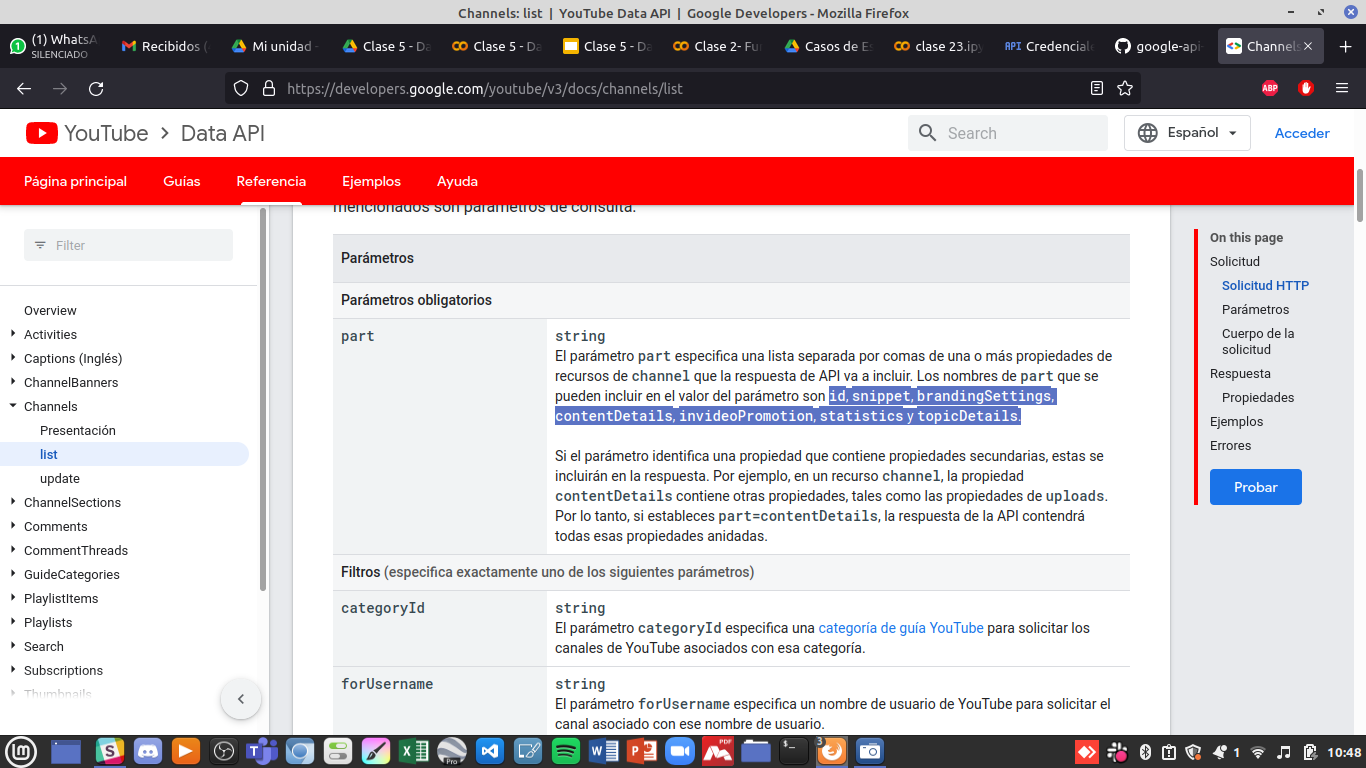
Si damos click en v3 podemos observar todos los metodos disponibles ****

Podemos explorar el metodo channels: [https://googleapis.github.io/google-api-python-client/docs/dyn/youtube\_v3.channels.html](https://colab.research.google.com/corgiredirector?site=https%3A%2F%2Fgoogleapis.github.io%2Fgoogle-api-python-client%2Fdocs%2Fdyn%2Fyoutube_v3.channels.html)

Si quieren encontrar mayor información sobre la google de api pueden ir a está página:

<https://developers.google.com/youtube/v3>

Podemos dar clieck en Reference > channels > list para mayor informacion: ****

Para la opcion Part podemos utilizar las siguientes opciones: ****

En este ejemplo particular usaremos statistics, por si requerimos obtener las estadisticas de algun canal particular

[ ]

from googleapiclient.discovery import build

api\_key='AIzaSyAVxBti356P\_n0VOT8hup1Ggj5Pv36JATM' # La borrare despues asi que no se preocupen

youtube\_david= build('youtube','v3',developerKey=api\_key)



[ ]

request=youtube\_david.channels().list(

    part='statistics',

    forUsername= 'Wimbledon' # puede reemplazarse por cualquiera

)



[ ]

response=request.execute()

print(response) # Obtenemos informacion



{'kind': 'youtube#channelListResponse', 'etag': 'z09lilOMoGCMkxJfqLIhF4Q\_g2A', 'pageInfo': {'totalResults': 1, 'resultsPerPage': 5}, 'items': [{'kind': 'youtube#channel', 'etag': 'W0TTaMgZ4DtzXn1gFt\_yTso4A48', 'id': 'UCNa8NxMgSm7m4Ii9d4QGk1Q', 'statistics': {'viewCount': '1297490426', 'subscriberCount': '1620000', 'hiddenSubscriberCount': False, 'videoCount': '4989'}}]}

CodeText

[ ]

response['pageInfo']



{'totalResults': 1, 'resultsPerPage': 5}

[ ]

response['items']



[{'kind': 'youtube#channel',

'etag': 'W0TTaMgZ4DtzXn1gFt\_yTso4A48',

'id': 'UCNa8NxMgSm7m4Ii9d4QGk1Q',

'statistics': {'viewCount': '1297490426',

'subscriberCount': '1620000',

'hiddenSubscriberCount': False,

'videoCount': '4989'}}]

[ ]

# Miremos en un canal diferente por ejemplo DW  
request2=youtube\_david.channels().list(  
    part='statistics',  
    forUsername= 'sentdex' # puede reemplazarse por cualquiera  
)  
response2=request2.execute()  
print(response2) # Obtenemos informacion  
print(response2['items'])  
print(response2['pageInfo'])

{'kind': 'youtube#channelListResponse', 'etag': 'cKOStr-Reul7TRThqrnYd8z5nvc', 'pageInfo': {'totalResults': 1, 'resultsPerPage': 5}, 'items': [{'kind': 'youtube#channel', 'etag': 'sALuGvcN6TOX6ccVkuHSY55B6Ak', 'id': 'UCfzlCWGWYyIQ0aLC5w48gBQ', 'statistics': {'viewCount': '110352309', 'subscriberCount': '1240000', 'hiddenSubscriberCount': False, 'videoCount': '1243'}}]}

[{'kind': 'youtube#channel', 'etag': 'sALuGvcN6TOX6ccVkuHSY55B6Ak', 'id': 'UCfzlCWGWYyIQ0aLC5w48gBQ', 'statistics': {'viewCount': '110352309', 'subscriberCount': '1240000', 'hiddenSubscriberCount': False, 'videoCount': '1243'}}]

{'totalResults': 1, 'resultsPerPage': 5}

**Bibliografía:**

**LINK:** [**https://colab.research.google.com/drive/1SJ93a3m8EReoZ8lA05HJ9\_9wUM2-gG6W#scrollTo=mQCkF07fHjM1**](https://colab.research.google.com/drive/1SJ93a3m8EReoZ8lA05HJ9_9wUM2-gG6W#scrollTo=mQCkF07fHjM1)