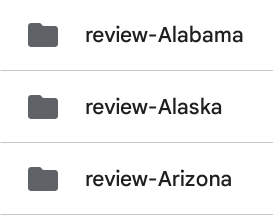
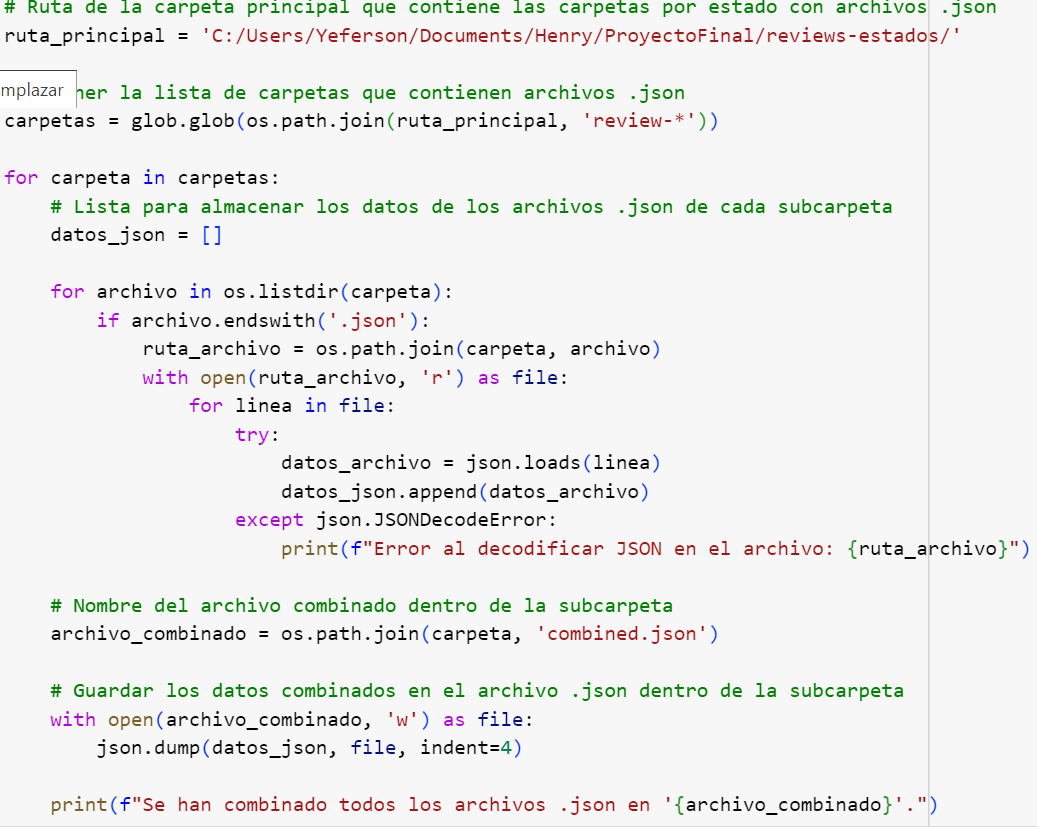
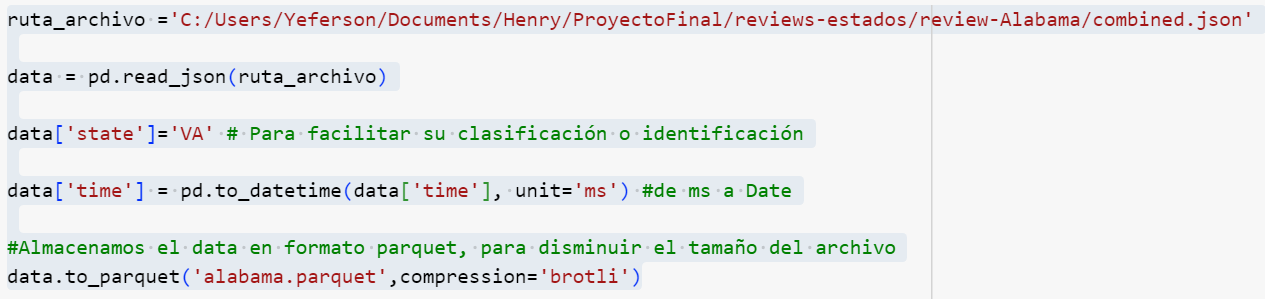
Este documento es una breve reseña de la implementación de dos script de python para generar archivos tabulares a partir de diferentes fuentes.

1. Reviews de Google por estado.

* Inicialmente nos entregan unos datos de reviews por estado

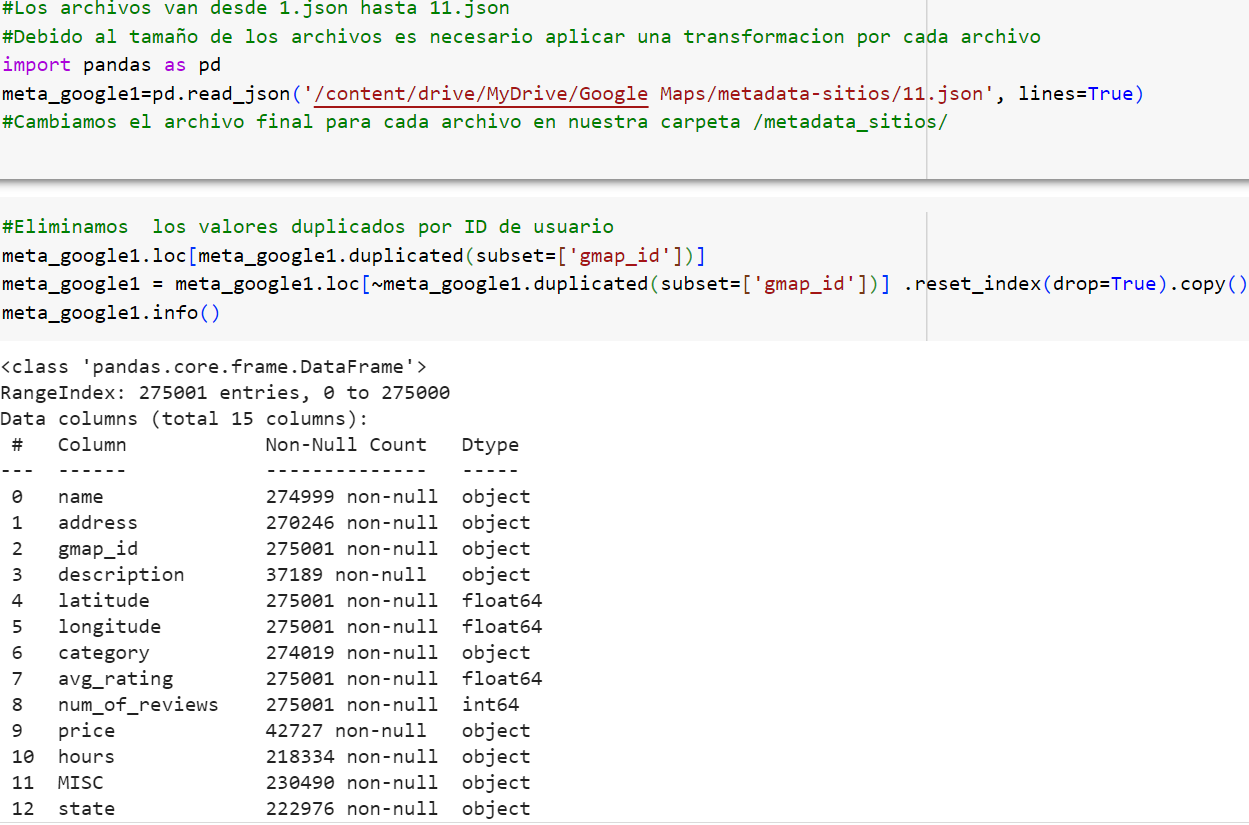


* En cada estado tenemos unos archivos JSON (No Tabulares)

1. En el script [**reviews\_state.ipynb**](https://colab.research.google.com/drive/1Q5ICcIJ2Aci3qj6zTKBZ49DYRLqe4bKn?usp=sharing)implementamos un algoritmo que permite seleccionar cada carpeta de estado y seleccionar cada archivo tipo JSON integrando todos en un solo archivo final por estado.
2. Generamos un Dataframe con la librería pandas de este archivo, agregamos un campo State, con la abreviatura del estado y cambiamos el formato de time en ms a date
3. Almacenamos el archivo definitivo en formato parquet para disminuir su tamaño (MB)
4. Metadata Sitios Google

* En esta data nos entregan archivos JSON con información por Comercio
* Cada archivo contiene Name, Id, Count\_reviews , Rating , Longitude, Latitude, Address y otros datos correspondientes a cada comercio.

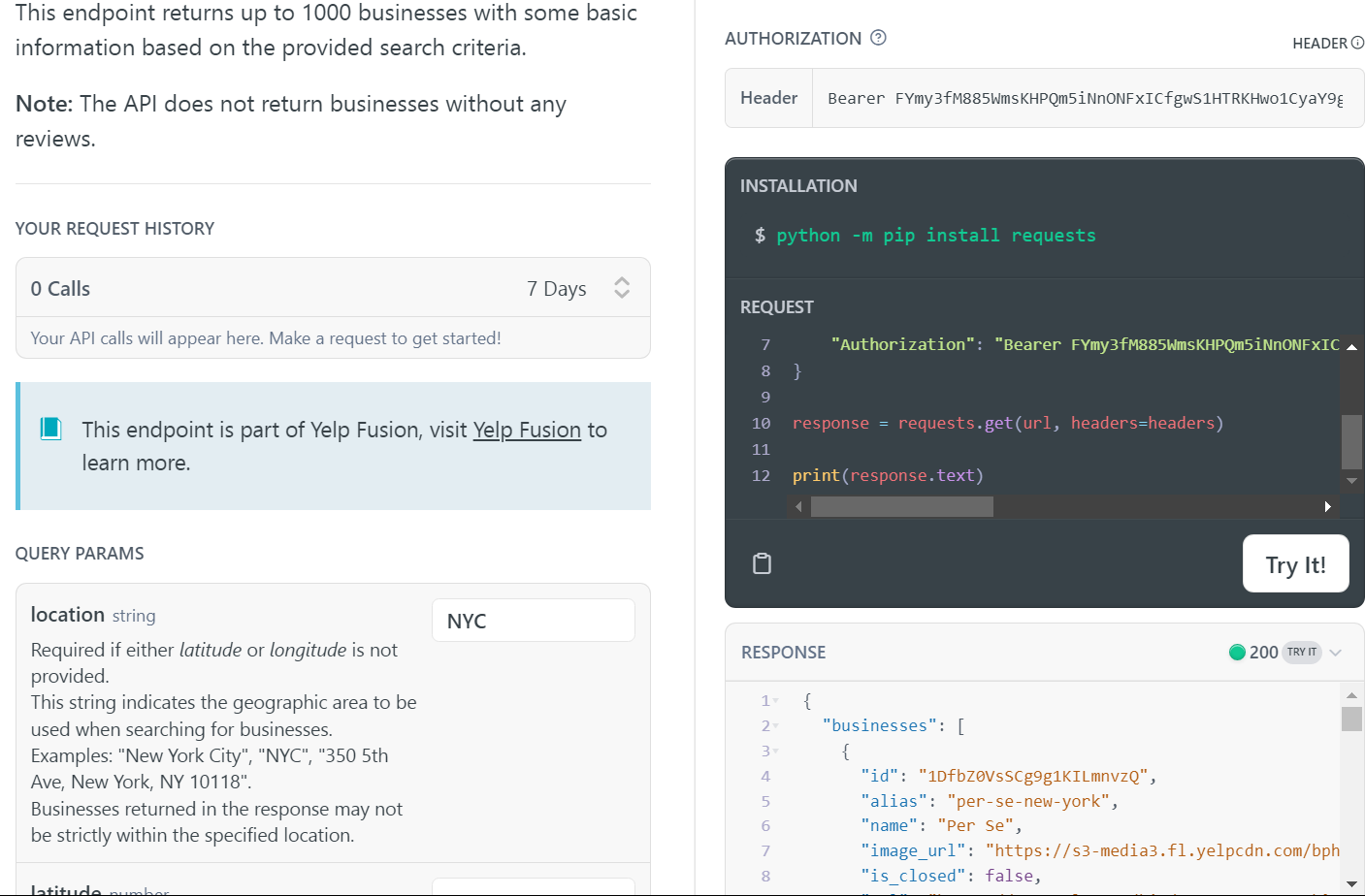


1. En el script [**estados\_meta\_google.ipynb**](https://colab.research.google.com/drive/1no9LJNWaWYup-CADZFsfv4tNO7ZlNVAR?usp=sharing) cargamos cada archivo en un DataFrame donde inicialmente realizamos la eliminación de datos duplicados
2. Creamos una columna estados y asignamos la abreviatura del estado al que pertenece el cual se identifica por una búsqueda en la dirección de los datos suministrados.
3. Como el DataFrame contiene información de todos los estados, generamos un filtro para seleccionar los comercios de los estados preseleccionados.
4. Creamos un nuevo DataFrame solo con los comercios por estados y por cada archivo JSON.
5. Finalmente creamos un DataFrame final, el cual contiene los comercios seleccionados por estados en cada archivo JSON
6. API de YELP

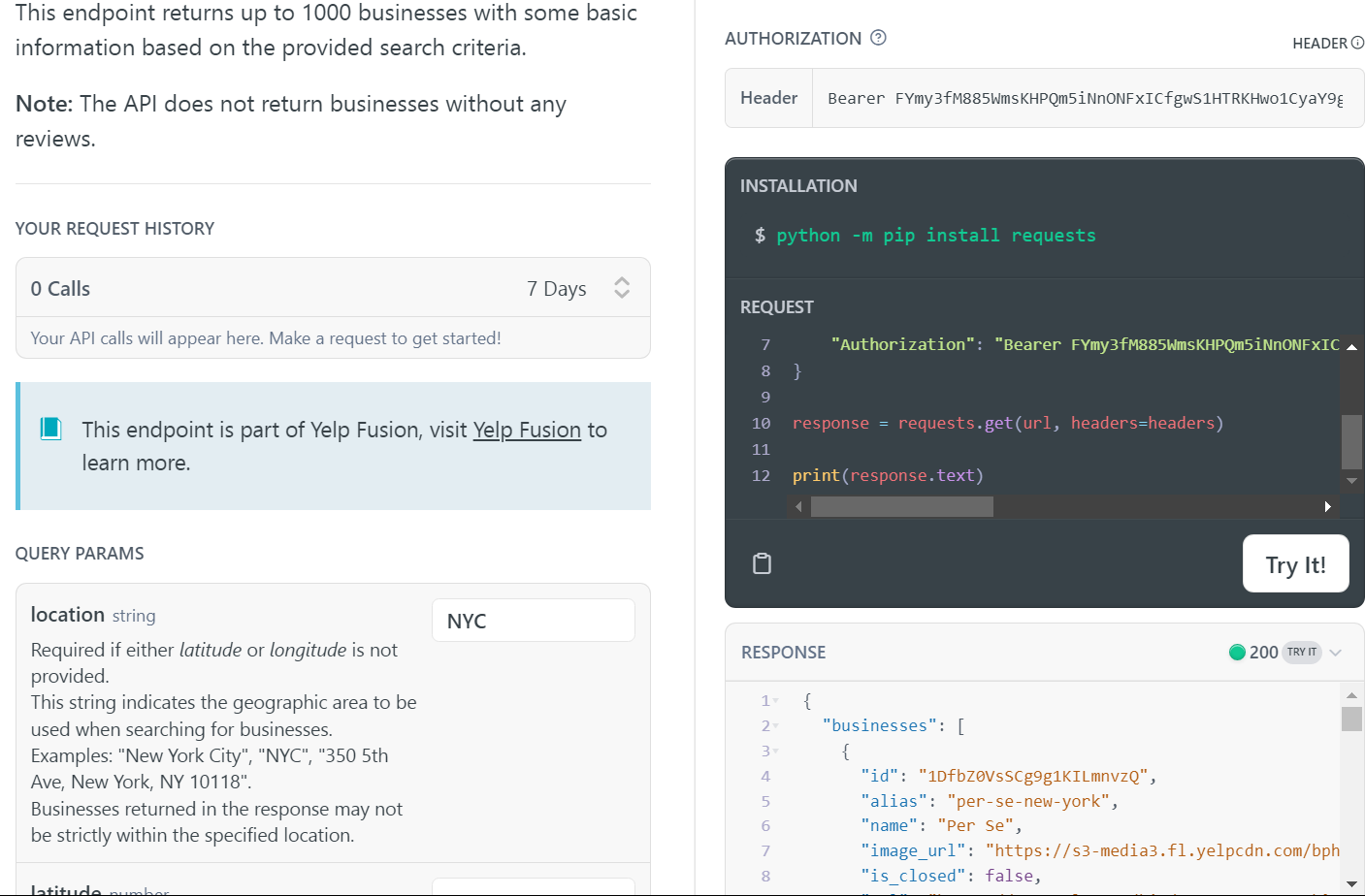
* En un análisis preliminar de datos de YELP observamos que el archivo que contiene la información de los diferentes negocios en el territorio nacional, solo presentaba resultados del estado de Florida de manera significativa.

Debido a esta necesidad fue necesario obtener información que nos permitiera desarrollar un posterior análisis.

Para ello fue necesario consumir la API de YELP para obtener dicha información.

← Esta API solo permite 1000 resultados por consulta

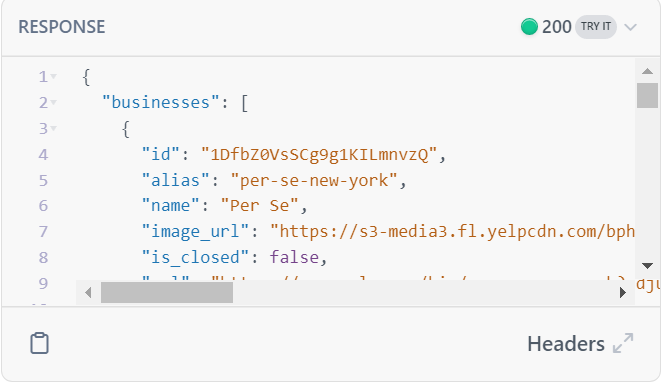
← Ingresamos Alguna Abreviatura o Ciudad para indicar la zona donde queremos que sea la consulta



\* Generamos un API\_KEY registrándose en YELP

← Debemos primero agregar Bearer antes de la API\_KEY

\* Aquí nos da el código inicial para generar nuestra consulta desde otro entorno



← 200 corresponde a una consulta exitosa

\*Observamos el formato de respuesta es de tipo JSON

1. Google Function - API YELP

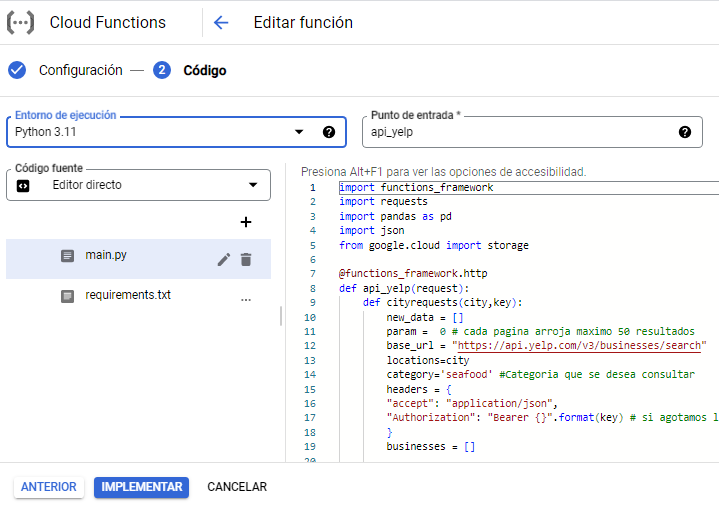
* Una vez consumida la API era necesario automatizar el proceso de búsqueda y almacenamiento de los datos. por medio de Google Cloud Platform.
* Google Function permite implementar nuestro script de Python para consumir la API y Almacenar la Data obtenida en Google Storage
* Desde Google Storage podemos disponer de la Data para generar nuestros análisis o modelos de recomendación.



°Google Functions es un servicio en la nube que permite implementar nuestros script.

° Se puede ejecutar estas funciones de forma automatizada y de manera independiente

← Nos permite asignar RAM y CPU de manera personalizada dependiendo de la Tarea a ejecutar



° Esta Herramienta nos permite diferentes entornos de ejecución, Python, Node, Go y otros más.

°En “main.py” se ejecuta nuestra Function con la tarea a realizar y en “requirements.txt” se declaran las librerías necesarias para ejecutar nuestra tarea

* En [FuncionYelp.ipynb](https://colab.research.google.com/drive/1StU7qL1VRs93Dh71BFetTkGeQJ-K9k3p?usp=sharing) se puede observar de manera más detallada cómo se implementó la Google Functions para obtener los datos de YELP y como fueron almacenados en un Bucket de Google Storage.

