Clase 08 - Manejo de archivos y datos

**Persistencia**

Cuando uno piensa en la palabra persistencia, lo relaciona a la capacidad humana de “aguantar”, “perdurar” o “de no dejar de hacer una acción”, en este curso no hablaremos de esas cosas obviamente, pero sí de la persistencia de los datos.

Hasta el momento todos los programas que realizamos ya tenían datos; por ejemplo: las edades de las personas, nombres, apellidos, lista de números, etc. Pero todos esos datos había que generarlos cada vez que usábamos nuestro programa y los datos se perdían de un día al otro.

La persistencia es lo que nos va a permitir guardar y recuperar los datos que se generaron en algún programa (ya sea nuestro o ajeno).

En programación, la persistencia es la acción de preservar la información de un objeto de forma permanente (guardado), pero a su vez también se refiere a poder recuperar la información del mismo (leerlo) para que pueda ser nuevamente utilizado.

De forma sencilla, puede entenderse que los datos tienen una duración efímera; desde el momento en que estos cambian de valor se considera que no hay persistencia de los mismos.

## **Tipos de persistencia**

El guardado de datos se puede hacer en dos grandes estructuras:

* En base de datos (o almacenes de datos y sus variantes)
* En archivos

## **Base de datos**

Las bases de datos son sin duda la mejor alternativa para almacenar y explorar los datos. Estas son complejas y por lo general necesitan de la instalación de algún motor o programa que nos permita utilizarlas.

Trabajaremos con ellas desde la clase 19 en adelante; así que por ahora las ignoraremos.

## **Archivos**

Los archivos son la forma más antigua, primitiva y simple de almacenar datos.

Pero créanlo o no, aún se sigue utilizando este mecanismo en más de un aplicativo; incluso en programas bastante sofisticados.

En este apartado del curso aprenderemos a guardar datos en archivos y recuperarlos.

## **¿Qué son los archivos?**

Un archivo o fichero informático es una secuencia de bytes que son almacenados en un dispositivo. Un archivo es identificado por un nombre y la descripción de la carpeta o directorio que lo contiene. A los archivos informáticos se les llama así porque son los equivalentes digitales de los archivos escritos en expedientes, tarjetas, libretas, papel o microfichas del entorno de oficina tradicional.

## **Tipos de archivos**

Hay dos grandes tipos de archivos:

* Los binarios son aquellos archivos que mejoran su eficiencia, pero los datos están guardados bajo agrupaciones de bytes; lo que hace que solo la PC pueda decodificarlos.
* Los de texto son lo que uno imagina, texto que guarda el dato en particular de una forma bastante descriptiva e intuitiva.

## 

## **Tipos de archivos**

Por simplicidad en el curso trabajaremos solo con archivos de texto.

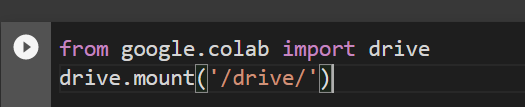
A su vez, los archivos de texto pueden ser de muchas extensiones, las más clásicas son: .txt, .doc, .docx, .xml, .csv, .json, etc.

****

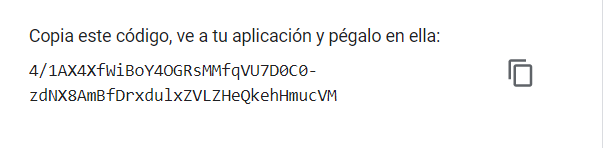
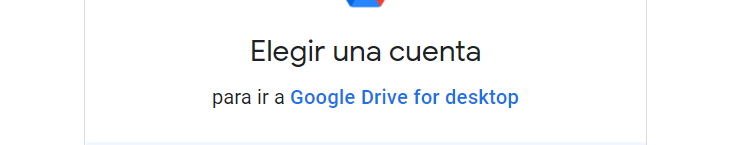
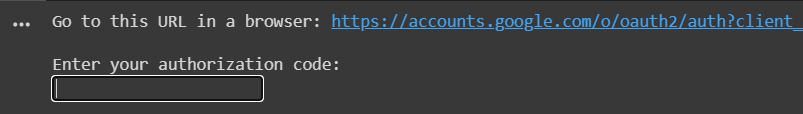
## **Escritura de archivos**

SOLO porque estamos trabajando con Colabs, debemos realizar lo siguiente:

Para dar acceso a nuestro drive.

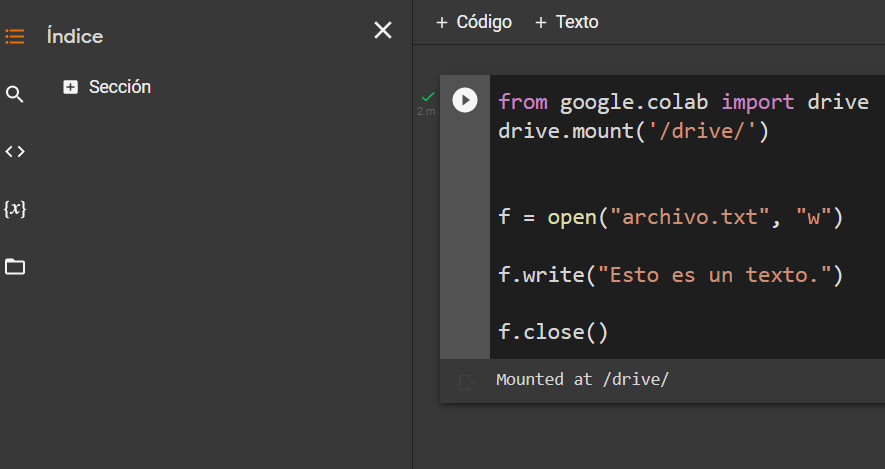
****

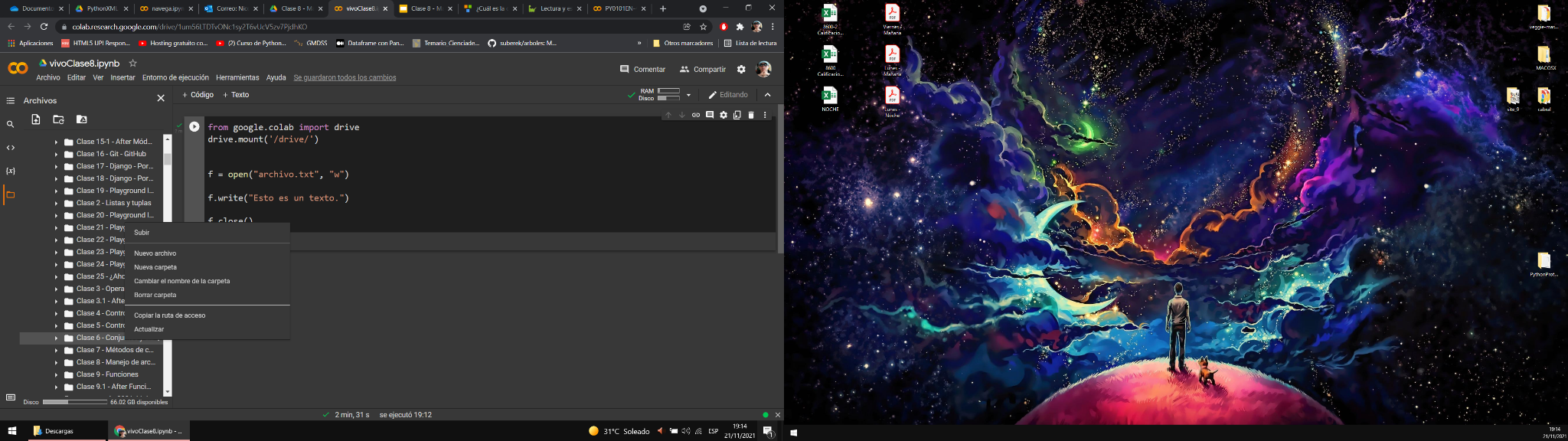
**Verificar nuestra identidad:**

****

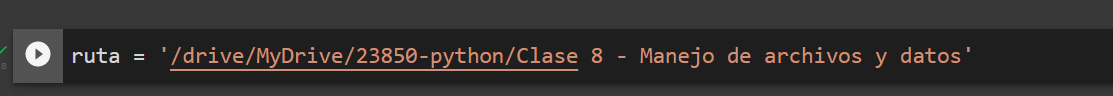
## **Escritura de archivos**

**Seleccionar la carpeta donde queremos escribir nuestro archivo:**

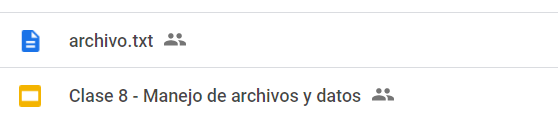
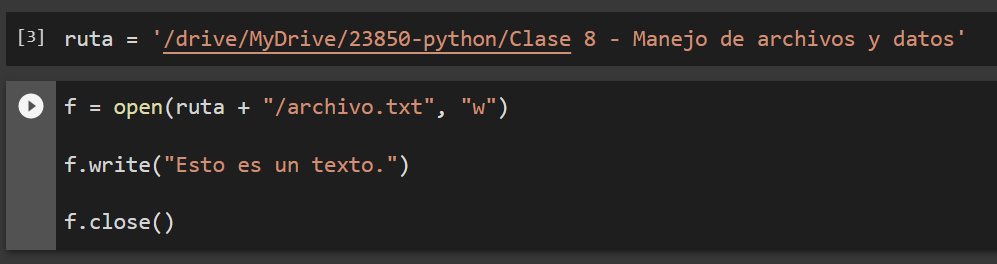
****

****

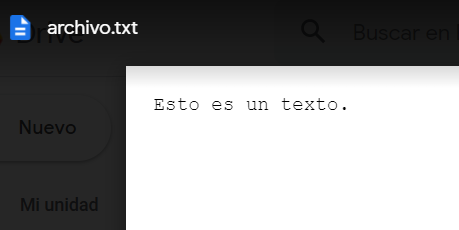
**Crear una variable con esa ruta:**



Y luego solo decidir cómo quieren que se llame el archivo y dar permisos para escribir en él “w”.

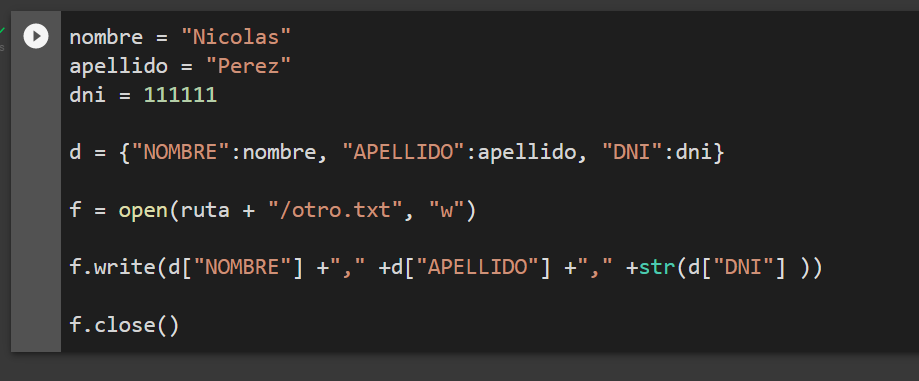
****

Dicho archivo quedará así:



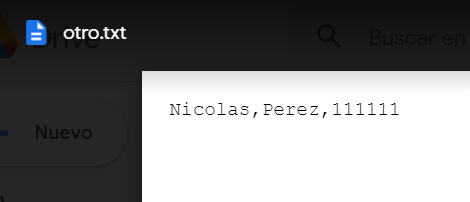
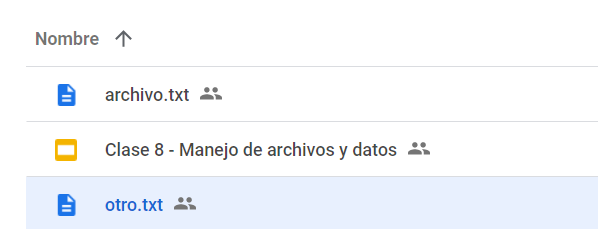
## 

Veamos un ejemplo similar pero un poco más completo:



## 

¿Qué creen que se guardó?



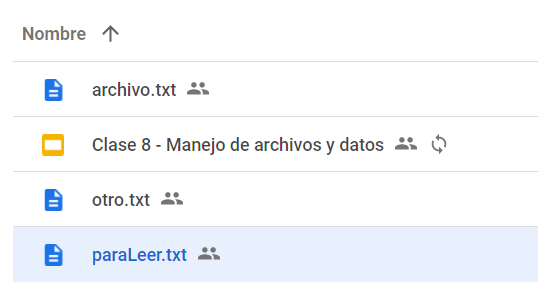
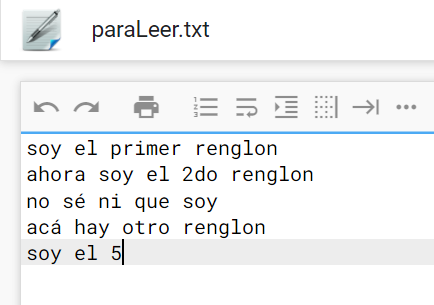
Esta es una forma muy útil de guardar información utilizando lo que se llaman delimitadores

## 

## **Lectura de archivos**

Entonces… ya sabemos inmortalizar nuestros datos en un archivo, pero... ¿Cómo haríamos para recuperar esos datos?

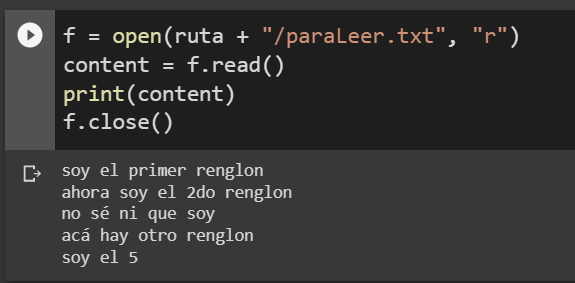
Supongamos que queremos recuperar los datos de un archivo cualquiera, paraLeer.txt



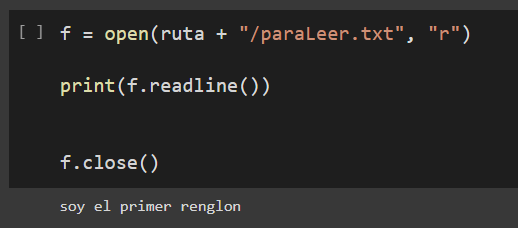
Es muy sencillo y similar a lo que hicimos con la escritura; solo debemos dar permisos de lectura (read - r).

Además, Python nos da tres formas de realizarlo:

* Read
* Readline
* Readlines



Es lo mismo… ¿Me estas cargando?

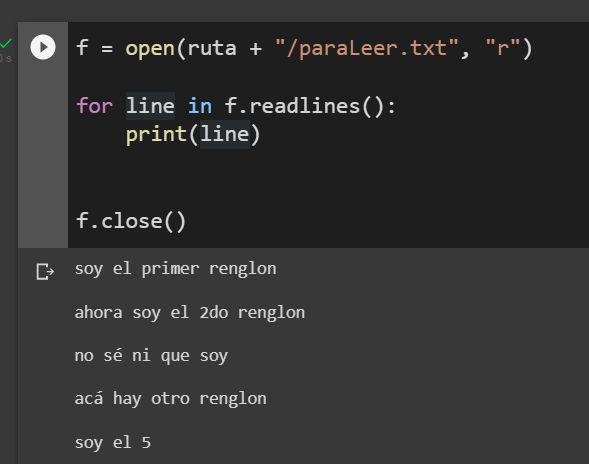


Solo nos deja ver el primer renglón.

## 

**Es lo mismo… ¿Me estas cargando?**

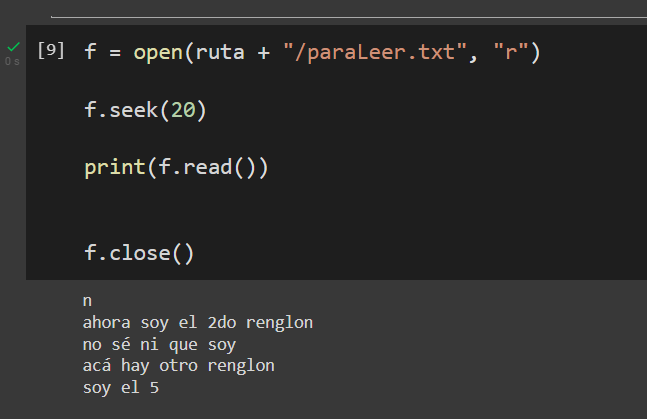
Parece idéntico, pero esto nos permite acceder renglón por renglón al txt.



## **Funcionalidad útil - SEEK**

Acceder a una ubicación en particular, es decir, empezar la lectura desde la posición indicada.

¿Qué pasó entonces ahí?, ¿Por qué la n?

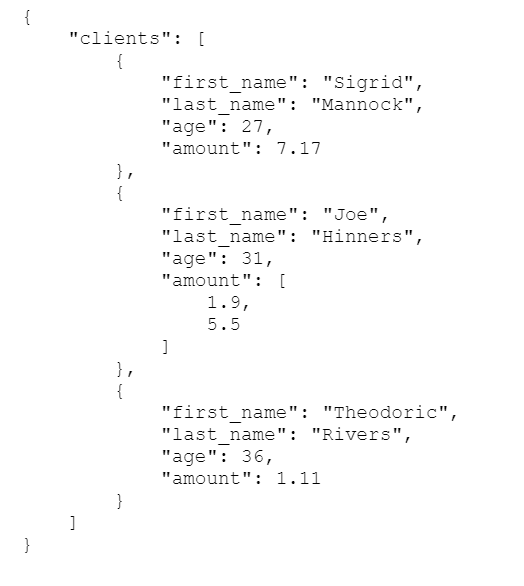
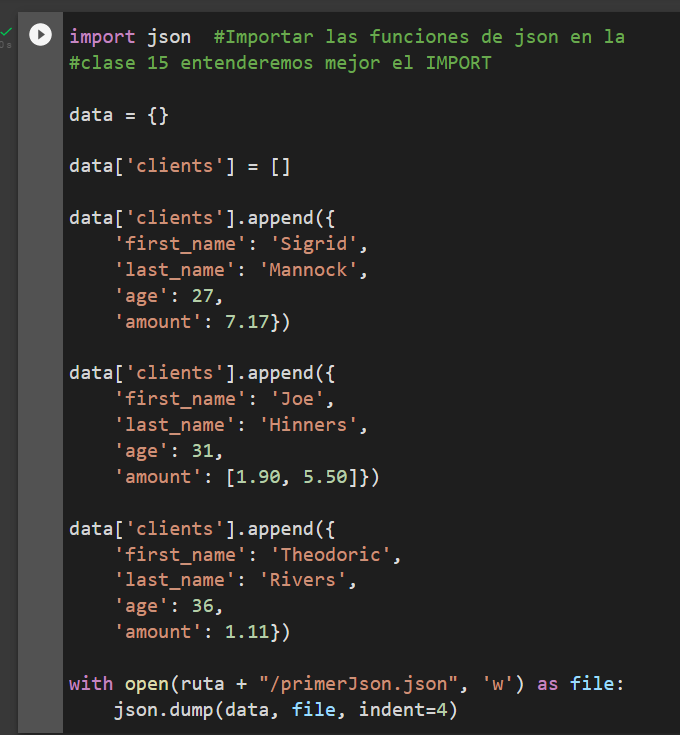
****

# **Archivo JSON**

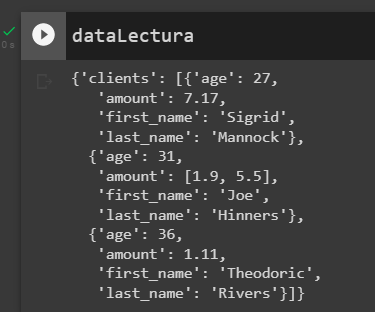
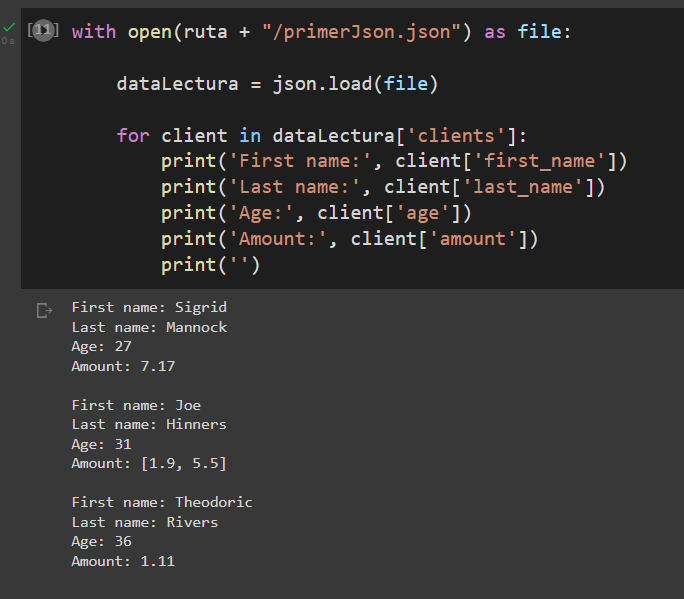
Por ahora solo trabajamos con extensiones .txt, pero para el guardado de datos suele ser más útil otro tipo de formato, como el csv, xml o el json.

Usar para slides de texto e imagen. Si no alcanza, no sobrecargar, usar otra con el mismo título para indicar que continúa el mismo módulo.

## **Archivo JSON - Escritura**



## **Archivo JSON - Lectura**



# **Trabajo con datos reales**

## Trabajo con datos reales -CSV

Para trabajar con datos reales, los mismos pueden estar en cualquier formato, ya sea txt, json o cualquier otro.

Por completitud del curso nosotros supondremos que vienen en csv, así tenemos otro formato más analizado.

En este apartado vamos a leer datos de alguna fuente oficial para poderlos trabajar desde Python.

Para esta clase descargamos datos oficiales de la Nación y de CABA (Argentina)

**Datos ciudad de Buenos Aires:** <https://data.buenosaires.gob.ar/dataset/>

**Datos Nación:** <https://datos.gob.ar/>

## **Trabajo con datos reales - CSV**

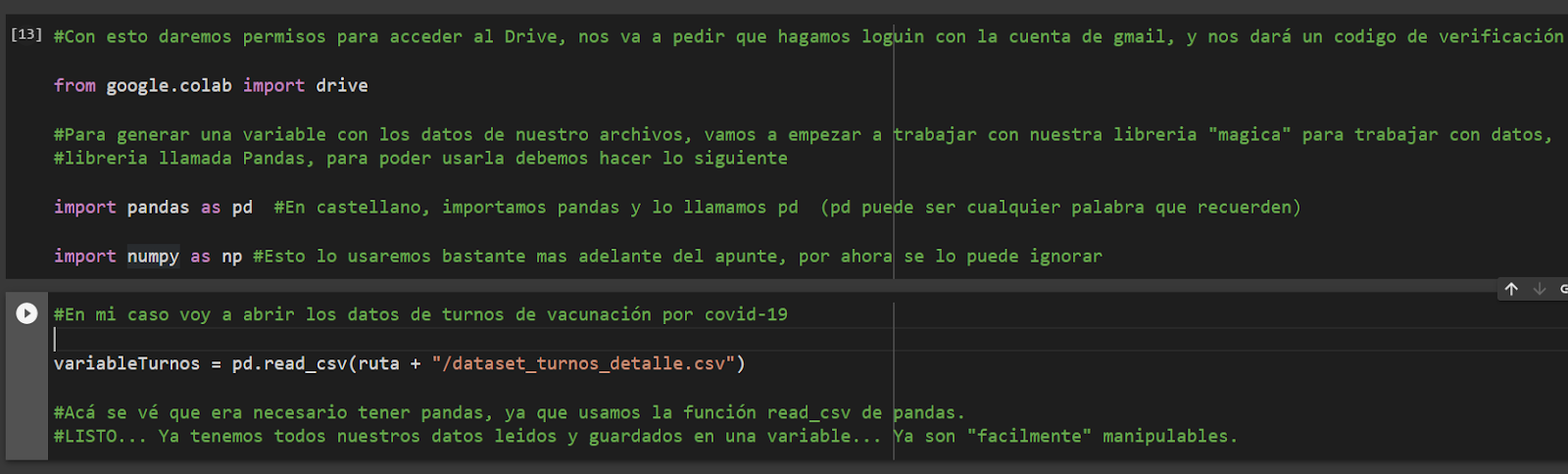
**Hagamos un pequeño Análisis Exploratorio de Datos**

**¿QUÉ ES UN EDA?**

El Análisis Exploratorio de Datos o también conocido como EDA de sus siglas en inglés: Exploratory Data Analysis, tiene como finalidad examinar nuestros datos para lograr un entendimiento holistico e integral de las relaciones existentes entre las variables analizadas.

El EDA, proporciona métodos sencillos para organizar y preparar los datos, detectar fallos en el diseño y recogida de datos, tratamiento y evaluación de datos ausentes, identificación de casos atípicos y mucho más.

**Acá hicimos la lectura de los datos que nos bajamos**

****

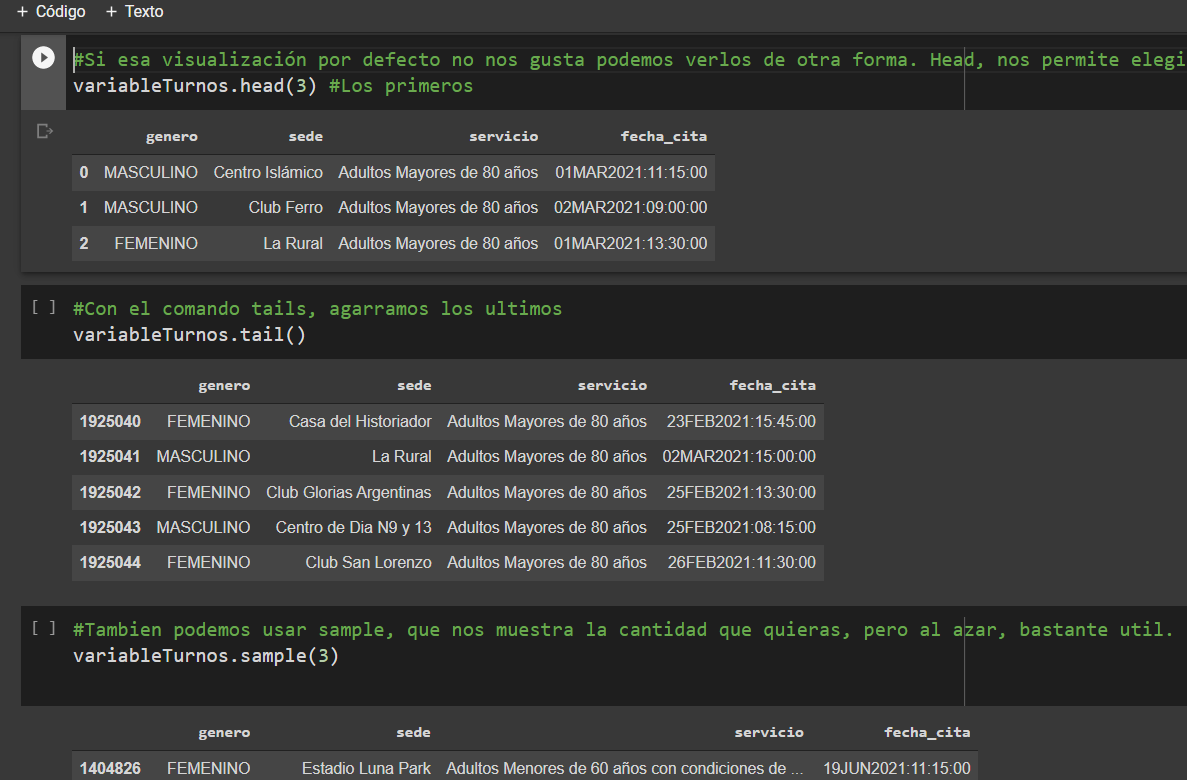
Admitimos que parece abrumador lo último, pero solo hicimos un read de un dato en otro formato csv. Pero veamos la magia; ya tenemos todos los datos guardados en una variable de Python como para trabajar sin problema:

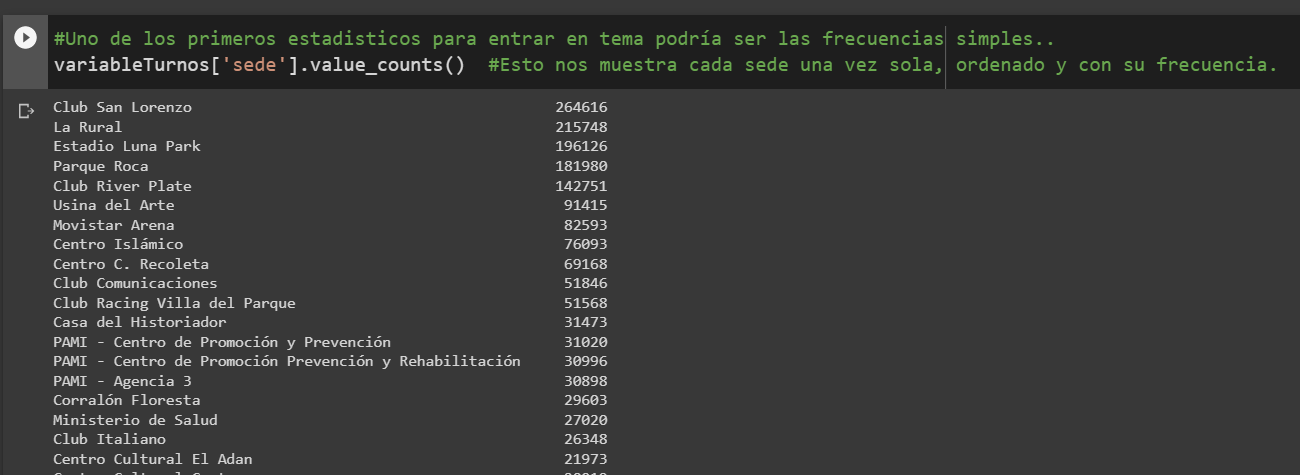
****

**Trabajo con datos reales - CSV**

Todo lo que podemos hacer ahora ya depende de otra rama de la multifuncionalidad de Python, que es el análisis de datos.

Pero veamos lo básico que les puede servir en este curso:

****

****

Bueno… Y así podemos seguir todo el día armando y agrupando los datos para manejarlos, cosa que se aleja del cometido del curso, lo importante es que ya saben recuperar datos de tres formatos distintos para reutilizarlos en cualquier proyecto de Python… ¿Por qué no, en su proyecto final?