Imagen que contiene Logotipo

Descripción generada automáticamente

Jesus Orlando Cueva Villela

21741211

Ing. Igor Mejía

Redes: CodeBook

12/3/2021

# Contenido

[Contenido 2](#_Toc66482414)

[Problema 3](#_Toc66482415)

[Implementación 4](#_Toc66482416)

[Documentación del código 5](#_Toc66482417)

[Conclusión 9](#_Toc66482418)

[Bibliografía 10](#_Toc66482419)

# Problema

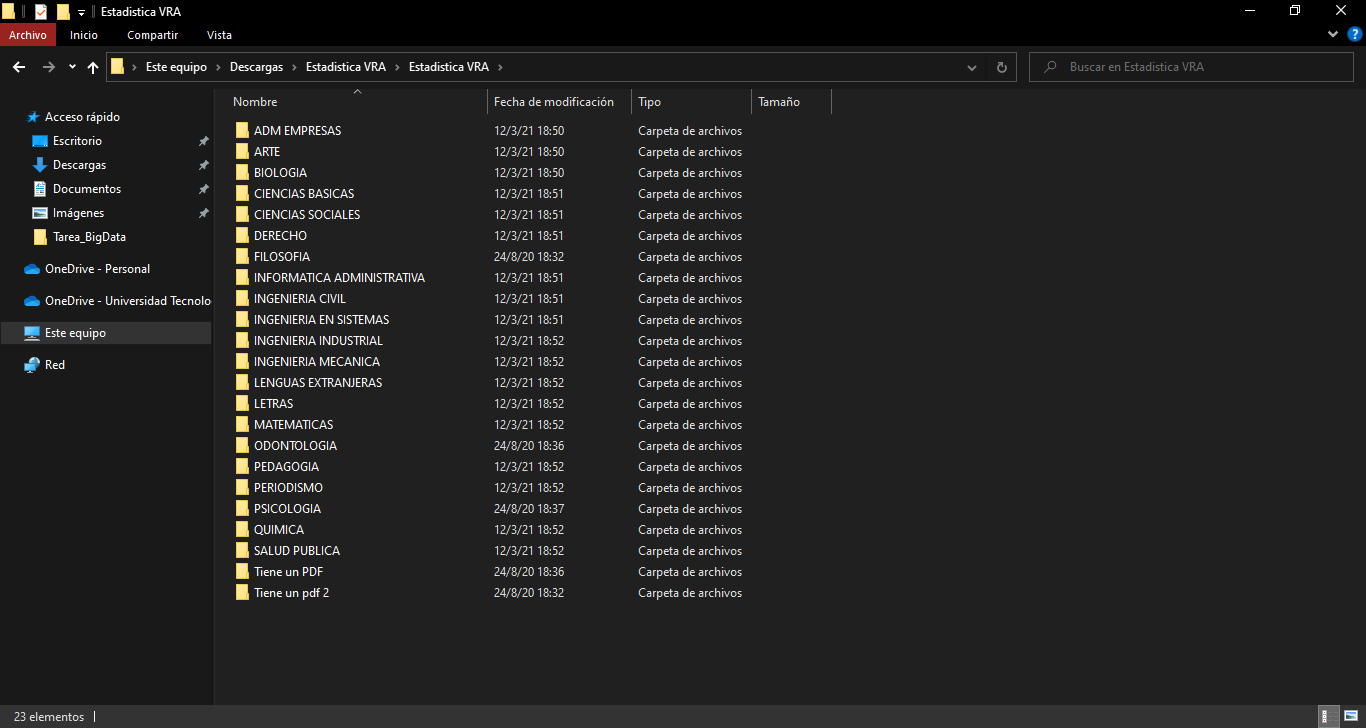
El problema dado trata en que a través de acciones como

1. Extracción
2. Transformación
3. Carga

De datos de archivos de Excel utilicemos herramientas de Python tales como Panda / Matplotib / Scipy / Numpy y logremos determinar en qué fecha termina el periodo.

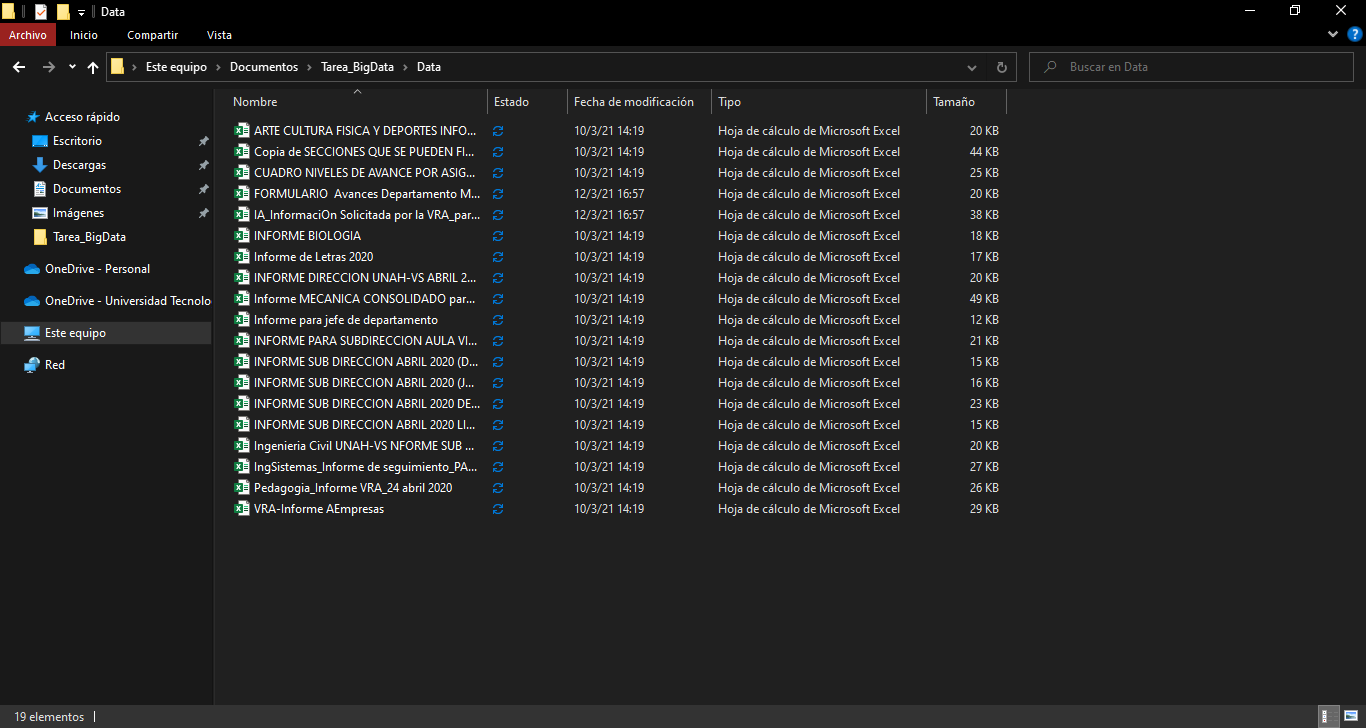
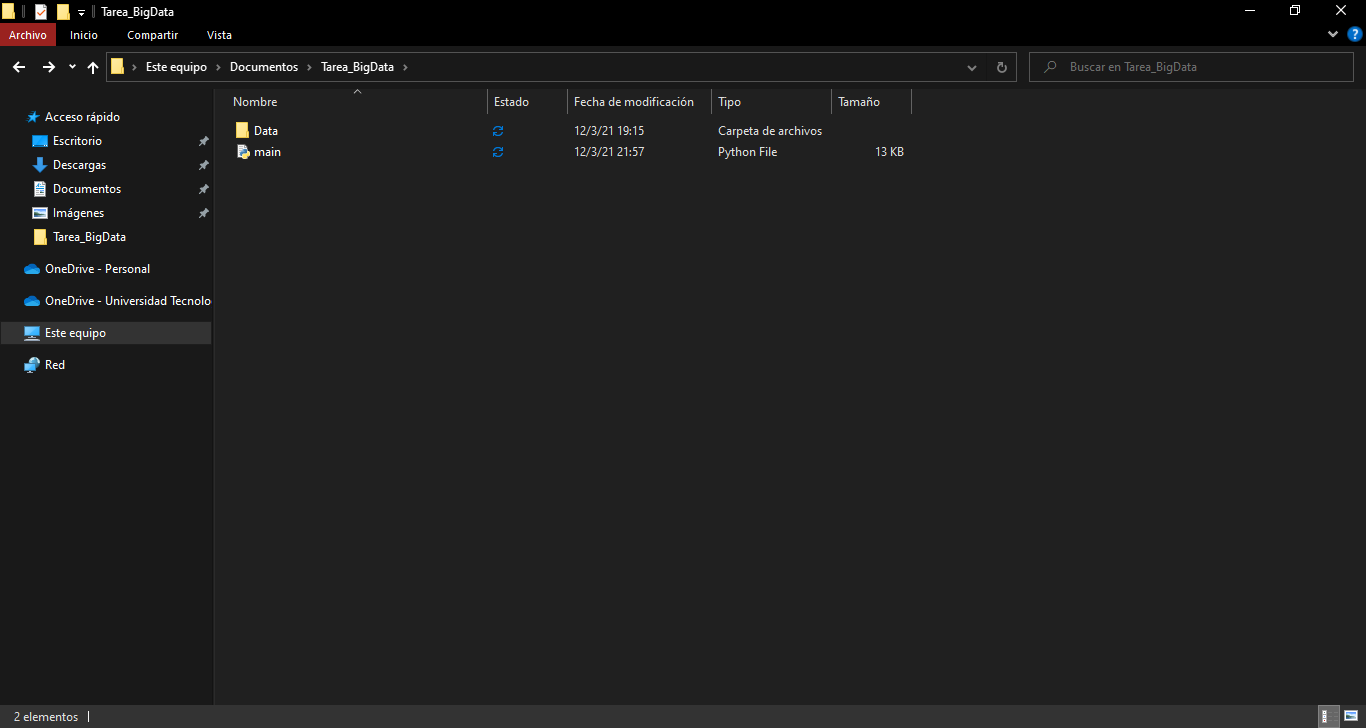
Adicionalmente se nos entre una carpeta comprimida que contiene múltiples archivos de Excel con datos y fechas en las cuales el periodo se prolonga.

Esta es la vista que obtenemos al descargar los Excel’s originales.



# Implementación

En mi implementación comencé reuniendo todos los Excel y juntarlos en una sola carpeta para que sea mucho más fácil poder acceder y leer a ellos a través de Python. Obteniendo como resultado lo siguiente:



Mi carpeta de trabajo contiene estos dos archivos. Una carpeta como mencioné que es utilizada para almacenar los archivos de Excel, y un archivo **main.py** en el cual se encuentra el código fuente del proyecto.

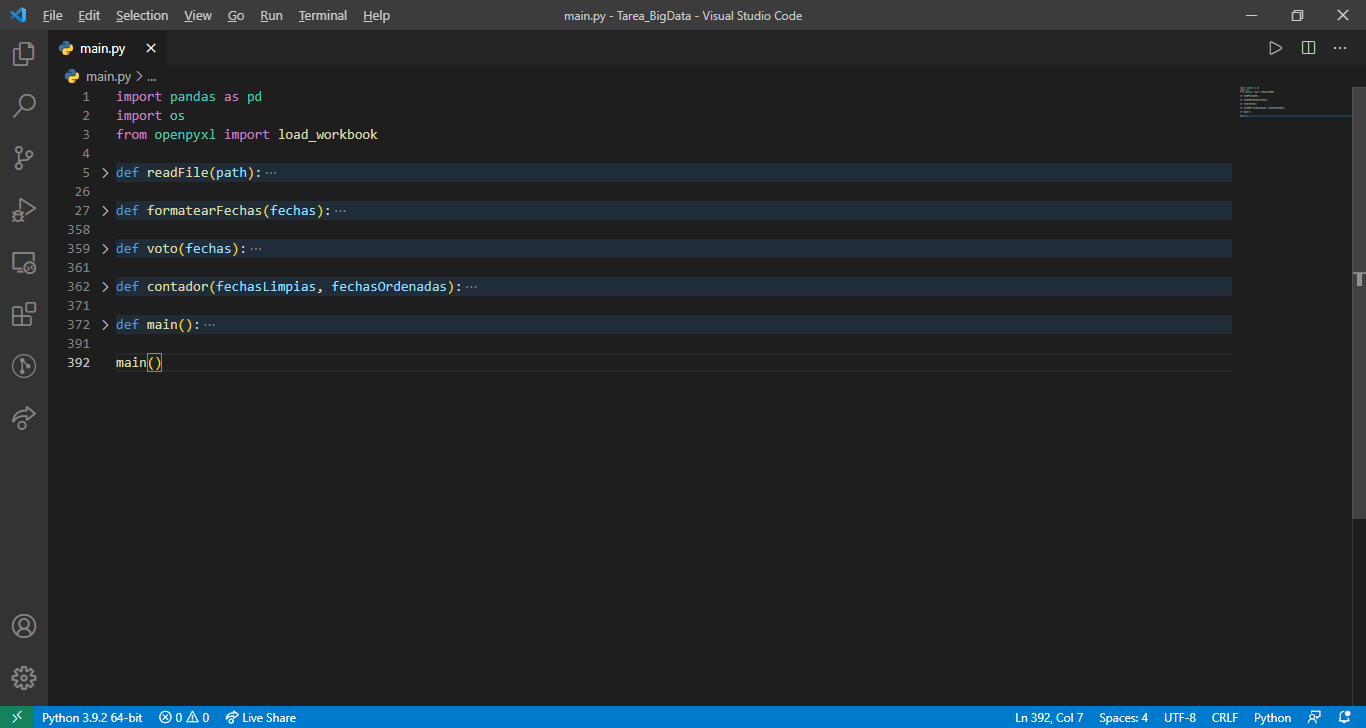
Esto debido a que mas adelante veremos que para leer todos los archivos de Excel, no podía haber ningún otro tipo de archivo.

También cabe mencionar que se tuvo que modificar los archivos de Excel en el sentido que la información no estaba normalizada. Es decir, algunos campos realmente no se necesitaban, o generaban conflicto entre ellos. Para esto, se le dio un formato especial de tabla a la lista y nombrar a la columna de con las fechas del periodo como “**Semana tentativa para finalizar”** para simplificar el proceso de búsqueda desde Python. También verificamos que la información este en la primera hoja por motivos de comodidad y no tener que estar trabajando con múltiples hojas de Excel.

Puntos clave del proyecto

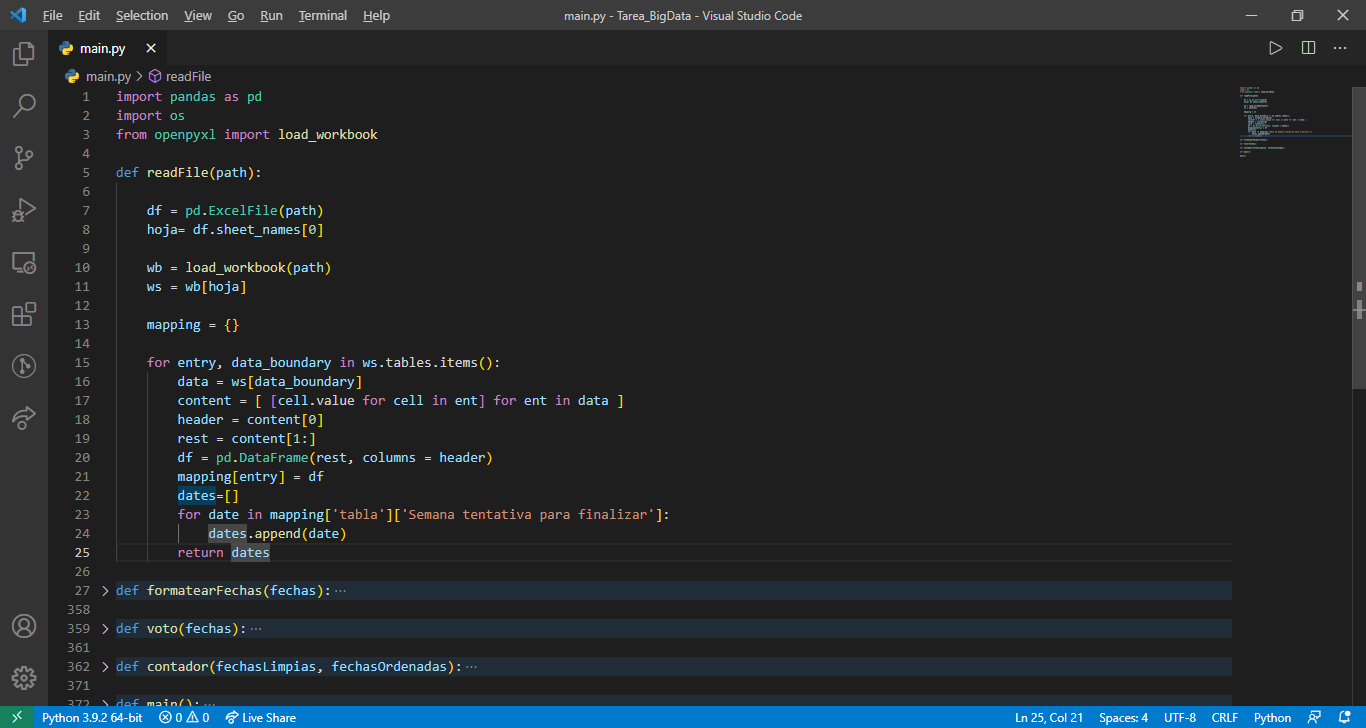
1. Leer la información de los archivos de Excel
2. Obtener los datos y separarlos por categoría
3. Hacer el recuento de la categoría

# Documentación del código

El proyecto consta de 5 funciones en las cuales se dividen los procesos. Primero en las líneas 1-3 podemos observar ciertas librerías que se deben importar para poder trabajar con los archivos de Excel.

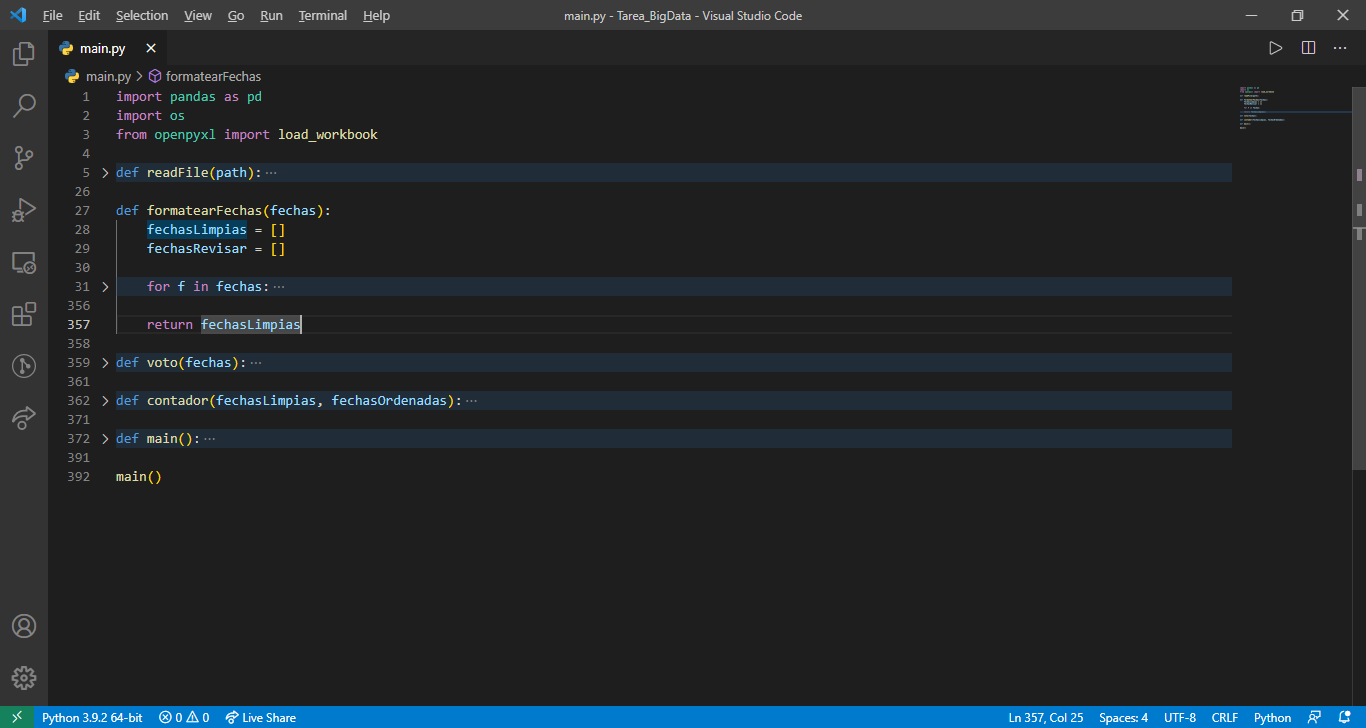
La librería panda nos ayuda a leer los archivos, la librería os, nos ayuda a trabajar con el directorio en donde se encuentran los archivos, y la librería openpyxl es requerida para poder manipular la data obtenida de los archivos.

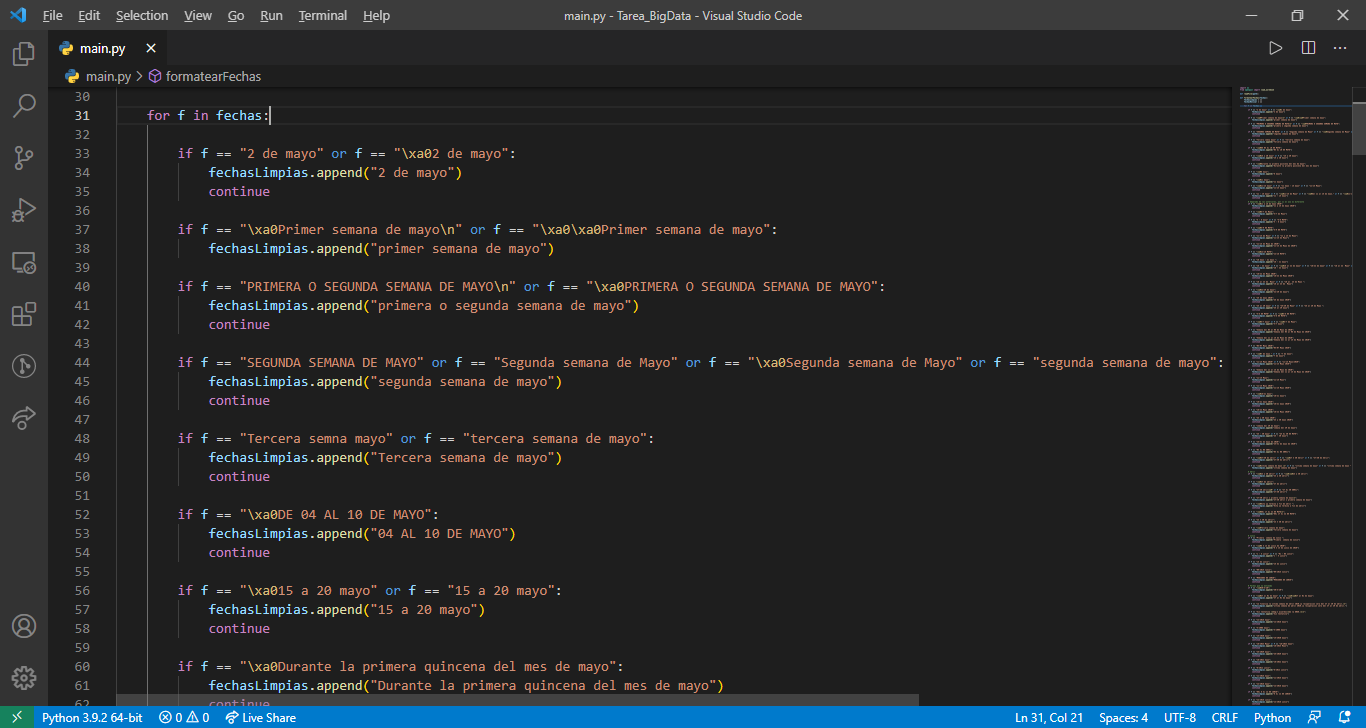
Función readFile ()

La función readFile se utiliza para leer los archivos de Excel que se encuentran en la carpeta Data. En el ciclo for realizamos la acción de recorrer la tabla que tenemos y al finalizar nos fijamos en la columna de **Semana tentativa para finalizar** dado que es la única información que nos interesa.

Almacenamos estos datos en la lista de fechas y la retornamos para su posterior trabajo.

Función formatearFechas ()

Esta función tiene como parámetro las fechas brutas que obtenemos de los archivos de Excel. El propósito de esta función es darle sentido a la data que recopilamos. Consta de un solo ciclo for y al finalizar, retornamos una lista que creamos localmente donde guardamos las fechas de manera ordenada para poder trabajar.



Notamos que es la función mas extensa del proyecto, sin embargo, la idea de esta función es sencilla. Luego de haber obtenido toda la data de los archivos de Excel, vamos a ir comparándola con la data que ya tenemos. Básicamente agarramos la data en bruto del Excel y la convertimos en un formato legible para una persona. Por ejemplo, podemos observar como ejemplo:

if f == "SEGUNDA SEMANA DE MAYO" or f == "Segunda semana de Mayo" or f == "\xa0Segunda semana de Mayo" or f == "segunda semana de mayo":

En donde podemos observar que en la data de Excel se repite múltiples veces “segunda semana de mayo” sin embargo esta escrito de forma distinta, o en distinto formato al tener un espacio tabulador al principio o al final. Así que, el programa al encontrar esta secuencia en el archivo de Excel convierte esta cadena a:

fechasLimpias.append("segunda semana de mayo")

En donde se convierte a una sola cadena como **segunda semana de mayo** y la almacenamos en nueva lista llamada fechasLimpias. Cabe mencionar que cada vez que se encuentre una cadena, esta se guardara las n veces que se encuentre. Al utilizar esta implementación podemos reducir considerablemente la cantidad de veces que se hace referencia a la misma fecha en los diferentes archivos de Excel.

También hay que aclarar que el proyecto se basa en casos, es decir:

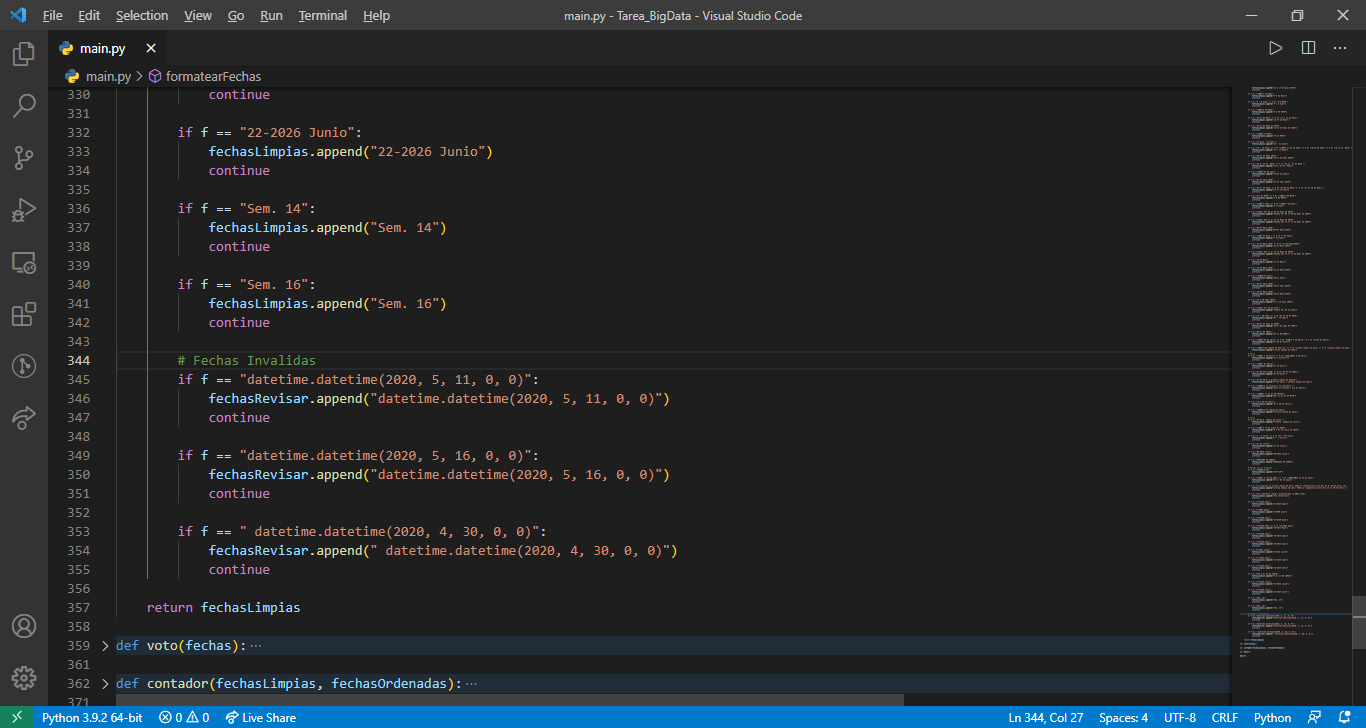
if f == "11 - 15 mayo" or f == "\xa011-15 de mayo" or f == "\xa0Del 11 al 15 de mayo." or f == "\xa011-15 de mayo" or f == "11-15 de mayo" or f == "11 al 15 mayo" or f == "11 al 15 de mayo":

Podemos notar que en estos casos solo se trabaja con los días entre los rangos, sin embargo, pudimos observar que también existía el caso de:

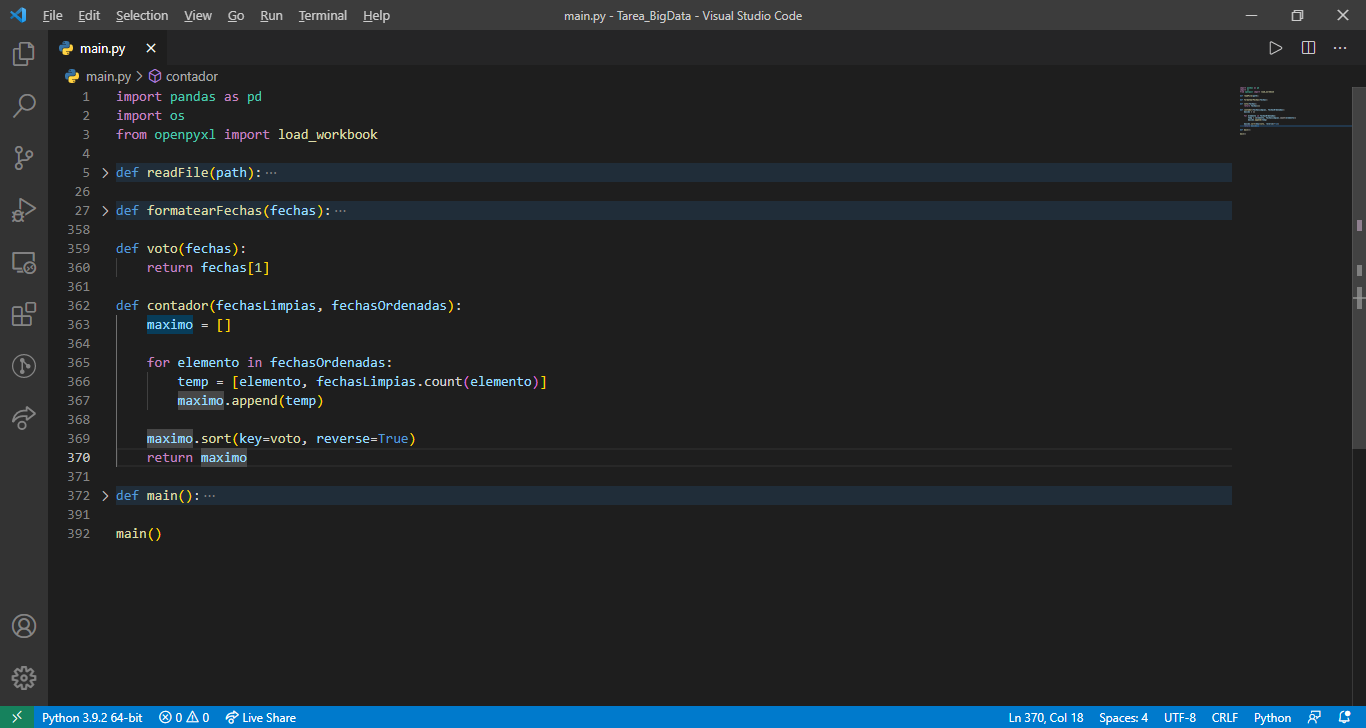
if f == "\xa011 a 15 de mayo 2020":

Mi implementación al encontrar este caso toma la data como diferentes fechas dado que el primer caso no sabemos con certeza si se trata del 11-15 de mayo del año 2020, o si hace referencia a otro año. Por lo tanto, solo toma en consideración 11-15 de mayo como un caso y 11-15 de mayo de 2020 como otro caso.

Y así sucesivamente con todas las fechas que estaban en los archivos Excel.

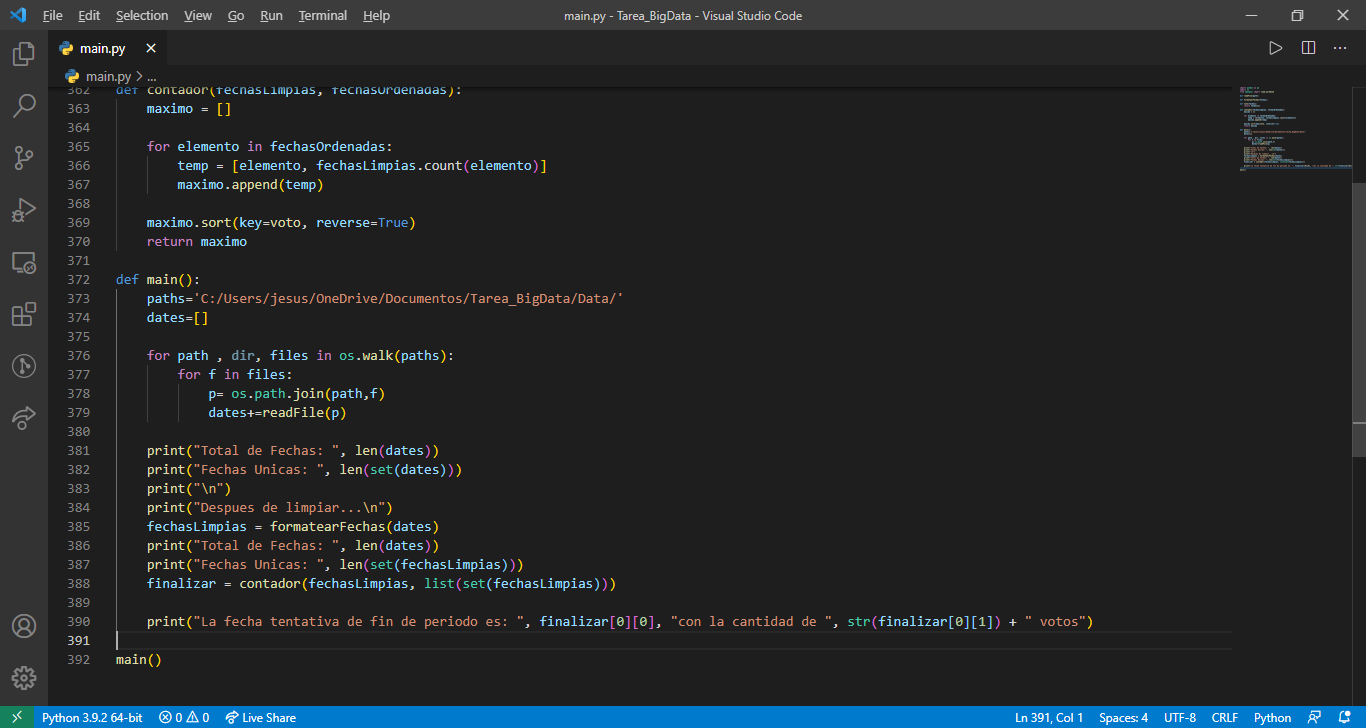


También nos encontramos con este tipo de formato, sin embargo, estos datos no se tomaron en consideración.



Siguiendo, las funciones de voto y contador participan de modo que voto obtiene como parámetro fechas, y regresa la primera posición de la lista. Ósea las fechas que hemos ordenado. Y contador obtiene como parámetros las fechas que hemos formateado y las fechas ordenadas de manera secuencial. Luego creamos una lista llamada **máximo** la cual almacenará la cantidad de votos que se obtienen por cada fecha.

Luego en el ciclo for recorremos los elementos que tenemos en las fechas ordenadas, y utilizamos una lista temporal para almacenar el valor. Este valor es el que agregamos a nuestra lista de máximo. Por último, clasificamos nuestra lista de máximos según su valor y la devolvemos para su posterior uso.



La función principal **main** en la cual nuestro proyecto se ejecuta consta de una variable paths donde almacenamos la ruta absoluta donde se encuentran los archivos de excel que debemos leer. Y luego tenemos una lista de fechas donde almacenaremos las fechas con las que trabajaremos a lo largo de nuestro proyecto.

Luego nos encontramos con un ciclo for que realiza la accion de recorrer nuestro directorio y agregar a la variable **paths** la direccion de cada uno de los archivos de manera dinamica.

Por ultimo mostramos en pantalla la salida de nuestro programa. Es simple y directa, primero mostramos la cantidad total de fechas que nos dan en los archivos de excel y luego mostramos la cantidad de veces que cada fecha se encuentra. Como resultado obtenemos 1002 fechas y 273 fechas respectivamente. Luego mostramos un tipo de divisor que dice **Despues de limpiar…** en el cual mostramos en pantalla el resultado obtenido por nuestro proyecto.

Este es, el total de fechas que no cambia en 1002 fechas totales, y las fechas unicas logramos reducirlas a 76 fechas. Recordemos que este numero se puede reducir aún mas, si hubieramos tomado fechas como 11-15 de Mayo todas en 2020. Por ultimo mostramos en pantalla nuestro resultado final en el cual mostramos al usuario que la fecha tentativa para finalizar el periodo es **11 - 15 mayo con la cantidad de 64 votos.**

# Conclusión

En conclusión, puedo decir que ha sido un proyecto muy interesante en el cual pude sentir lo que es trabajar con Big Data y tener el trabajo de Analizar la data en bruto y darle sentido para obtener un resultado esperado. También pude familiarizarme más con las herramientas que brinda Python como sus librerías y lo útiles que son para trabajar con proyectos de esta categoría. No logre utilizar matplotlib, sin embargo, luego de una rápida investigación, gracias a esta librería podemos visualizar la data de forma gráfica a través de gráficos.

# Bibliografía

<https://github.com/jcueva1109/SistemasInteligente>

<https://decodigo.com/python-leer-archivo-de-excel-con-pandas>

<https://towardsdatascience.com/how-to-work-with-excel-files-in-pandas-c584abb67bfb>

<https://www.programiz.com/python-programming/methods/list/remove>

<https://pythondiario.com/2016/11/eliminar-elementos-duplicados-de-una.html>