#Leer el archivo "TodasLasNoticias.csv" y pasarlo a un archivo JSON

#El objeto contendrá una lista con un diccionario por cada posición, donde cada diccionario tendrá cinco llaves: #I Fecha #II Título #III url #IV Descripción #V Categoría

#Importamos modulos #

#==========================================================#

from flask import Flask, jsonify, request, Response

import json

import csv

#Leemos el archivo csv, con encoding utf 8

#==========================================================#

with open('.\TodasLasNoticias.csv','r', encoding = 'utf-8') as outfile1:

fileReader = csv.reader(outfile1, delimiter=',', quotechar = '"' )

# Mostramos, que no todas las noticias tienen el formato deseado

count = 0

count\_lineas = 0

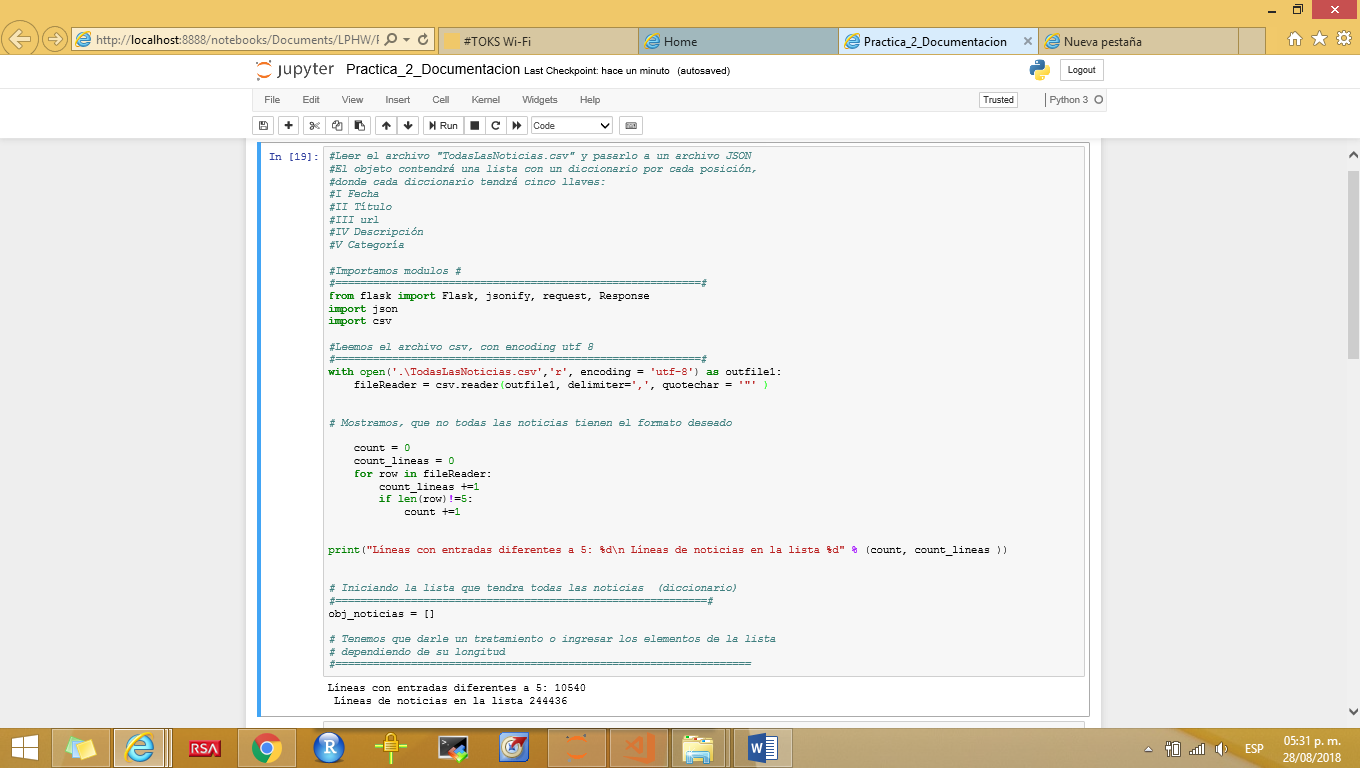
for row in fileReader:

count\_lineas +=1

if len(row)!=5:

count +=1

print("Líneas con entradas diferentes a 5: %d\n Líneas de noticias en la lista %d" % (count, count\_lineas ))



# Iniciando la lista que tendra todas las noticias (diccionario)

#===========================================================#

obj\_noticias = []

# Tenemos que darle un tratamiento o ingresar los elementos de la lista

# dependiendo de su longitud

#==================================================================

#Leemos el archivo csv, con encoding utf 8

#========================================================================================#

with open('.\TodasLasNoticias.csv','r', encoding = 'utf-8') as outfile1:

fileReader = csv.reader(outfile1, delimiter=',', quotechar = '"' )

# Damos un tratamiento a algunos registros de noticias

#========================================================================================#

for linea in fileReader:

if len(linea) ==5:

obj\_noticias.append({

'Fecha' : str(linea[0]),

'Titulo' : str(linea[1]),

'url' : str(linea[2]),

'Descripcion' : str(linea[3]),

'Categoria' : str(linea[4])

})

elif len(linea) == 4:

pedazo = len(linea[3].split("\",")) ## Patrón observado en conjunto de noticias, en vez de tener 'a\*', '\*' se tenía ",

if pedazo == 2:

linea.append(linea[3].split("\",")[1])

### Hay un patron, en el que a la direccion le falta una coma y por tanto se le como el mismo elemento de la lista

elif pedazo ==1 and linea[1].rfind("\",http") != -1 :

swap = tuple(linea)

tupla = swap[1].split("\",")

linea[0] = linea[0]

linea[1] = tupla[0]

linea[2] = tupla[1]

linea[3] = swap[2]

linea.append(swap[3])

obj\_noticias.append({

'Fecha' : str(linea[0]),

'Titulo' : str(linea[1]),

'url' : str(linea[2]),

'Descripcion': str(linea[3:-1]),

'Categoria' : str(linea[len(linea)-1])

})

elif len(linea)>5:

obj\_noticias.append({

'Fecha' : str(linea[0]),

'Titulo' : str(linea[1]),

'url' : str(linea[2]),

'Descripcion': str(linea[3:-1]),

'Categoria' : str(linea[len(linea) - 1])

})

elif len(linea) ==2:

if linea[(len(linea)-1)] in {'Coahuila', 'Durango', 'Nacional'}:

obj\_noticias.append({

'Fecha' : str(""),

'Titulo' : str(""),

'url' : str(""),

'Descripcion': str(linea[0]),

'Categoria' : str(linea[len(linea) - 1])

})

## Hay poemas y textos incompletos, no los metemos a la lista de noticias

print(len(obj\_noticias))

## Verificamos que guardamos una lista de diccionarios

print(obj\_noticias[350])

{'Fecha': '09/16/16', 'Titulo': 'Camioneta choca contra mototaxi en Motul y deja tres lesionados', 'url': 'http://yucatan.com.mx/yucatan/motul/camioneta-choca-mototaxi-motul-deja-tres-lesionados', 'Descripcion': 'MOTUL.- Por no respetar la señal de alto de disco, una camioneta colisionó a un mototaxi con saldo de tres personas lesionadas en céntrica calle de la ciudad de Motul. Camioneta choca contra mototaxi en Motul <https://t.co/b8pKPgOS9k> #Motul #Accidentes pic.twitter.com/yRjTAxI6cV &#8212; Diario de Yucatán (@DiariodeYucatan) September 14, 2016 La camioneta Toyota Hilux que circulaba sobre [&#8230;]', 'Categoria': 'Yucatan'}

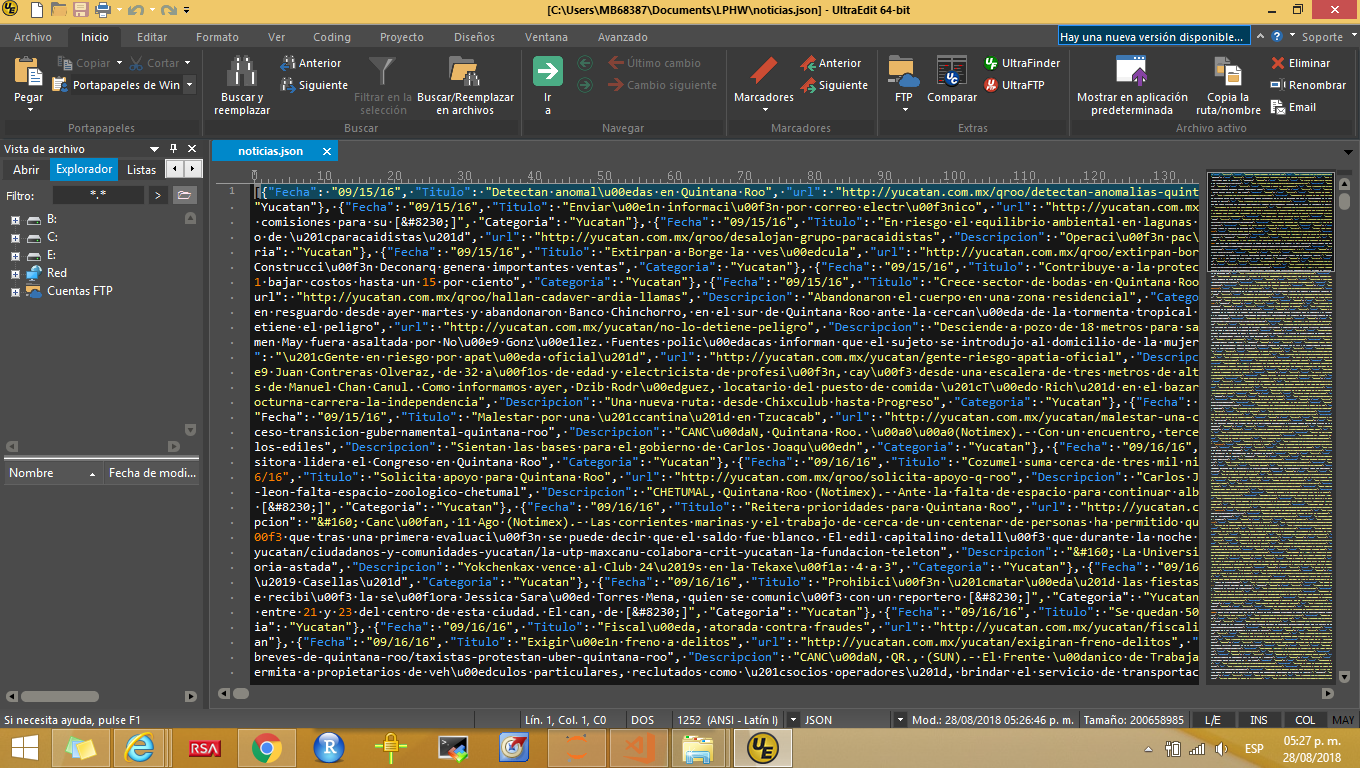
2. Guardar el archivo JSON con el nombre "noticias.json"

# Ya que tenemos nuestra lista de diccionarios, guardamos el objeto como un objeto JSON

with open('noticias.json', 'w') as outfile:

json.dump(obj\_noticias, outfile)

Abrimos el archivo en un editor especial, vemos el formato de diccionario



3. Hacer un endpoint que reciba una petición de tipo GET con el parámetro "categoría" para que retorne las noticias correspondientes al parámetro dado.

app = Flask(\_\_name\_\_)

@app.route('/<categoria>')

def imprime(categoria):

with open('C:/Users/MB68387/Documents/LPHW/noticias.json', 'r') as jfile1 :

data\_news = json.load(jfile1)

responde = Response(json.dumps([noti for noti in data\_news if noti['Categoria']==str(categoria)]), content\_type = "application/json; charset=utf-8")

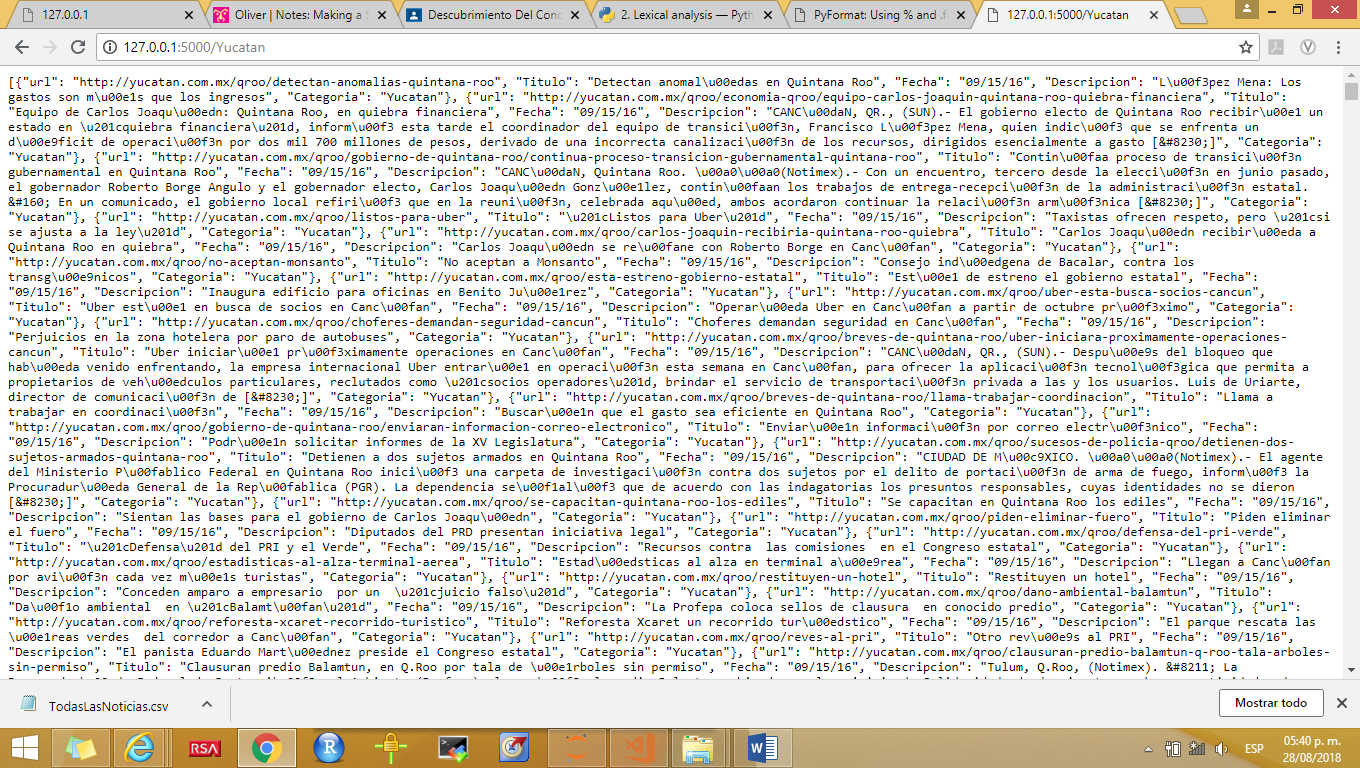
return responde

if \_\_name\_\_ == '\_\_main\_\_':

app.run(host = '127.0.0.1', debug = True )

########################################################################

Con categoría igual a Yucatan



Con Categoría igual a Durango



5- Ejecutar la aplicación en su maquina virtual Centos.