**Introducción a la programación en Python**

**Caso Práctico 3. Temas 10, 11, 12 y 13**

**Autor:** Antonio Villena Martín

**Alumno: Vicente Luis Cánovas Gil**

**Del 13 de septiembre al 25 de octubre de 2021**

**Contenido**

[1. Pregunta 1. Tema 10 - Entrada/Salida y Ficheros. 3](#_Toc80023387)

[2. Pregunta 2. Tema 11 - Expresiones regulares. 4](#_Toc80023388)

[3. Pregunta 3. Tema 12 - Sockets. 5](#_Toc80023389)

[4. Pregunta 4. Tema 13 - Interactuar con webs – web scraping. 6](#_Toc80023390)

### Pregunta 1. Tema 10 - Entrada/Salida y Ficheros.

Escribe el código de un script en el que se realicen los siguientes pasos:

* 1. Abra un fichero de texto en modo escritura.
  2. Pregunte al usuario los siguientes datos y los guarde en sus respectivas variables
     1. Nombre
     2. Primer apellido
     3. Edad
  3. Escribir en el fichero creado en el punto 1 los datos del usuario con el siguiente formato:

Nombre: Antonio

1er Apellido: Villena

Edad: 24 años

* 1. Cerrar el fichero

**CÓDIGO PROPUESTO POR EL ALUMNO:**

from bs4 import BeautifulSoup

import requests

url = requests.get( "https://www.ign.es/web/ign/portal/ultimos-terremotos/-/ultimos-terremotos/get30dias")

b\_soup= BeautifulSoup(url.content,"lxml")

busca\_tb= b\_soup.find('table')

## print(busca\_tb)

busca\_filas= busca\_tb.find\_all('tr')

## print(busca\_filas)

for linea in busca\_filas:

busca\_col= linea.find\_all('td')

##print(busca\_col)

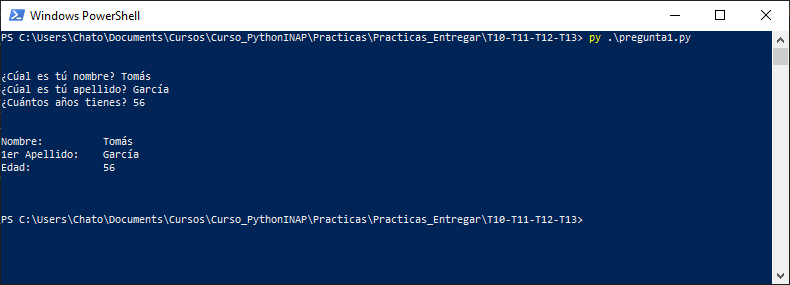
if len(busca\_col) >=2:

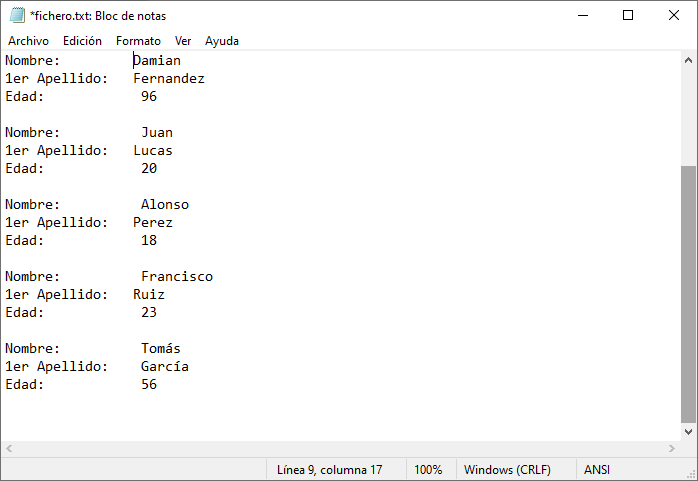
col\_1= busca\_col[0].get\_text()

col\_6= busca\_col[6].get\_text()

print(f"{col\_1:15} {col\_6:15}")

**CAPTURA DE PANTALLA DEL RESULTADO OBTENIDO:**





### Pregunta 2. Tema 11 - Expresiones regulares.

Mediante expresiones regulares, se programará un script que leerá el fichero de texto creado en la anterior pregunta. Se le solicitará al usuario que indique el nombre de un usuario y el programa devolverá todos los datos disponibles a partir de esa cadena de texto introducida. El texto introducido podrá comenzar tanto en minúsculas como en mayúsculas. En caso de no encontrar ninguna coincidencia, se imprimirá un mensaje indicándolo.

Por ejemplo:

“Introduzca los tres primeros caracteres del nombre buscado: ”

>> ant

El sistema mostrará:

“Los resultados obtenidos son los siguientes:”

Nombre: Antonio

1er Apellido: Villena

Edad: 24 años

**CÓDIGO PROPUESTO POR EL ALUMNO:**

import sys

import re

caracteres\_a\_buscar = input("Introduzca los tres primeros caracteres del nombre buscado: ")

with open("fichero.txt", "r") as archivo:

contenido = archivo.read()

##

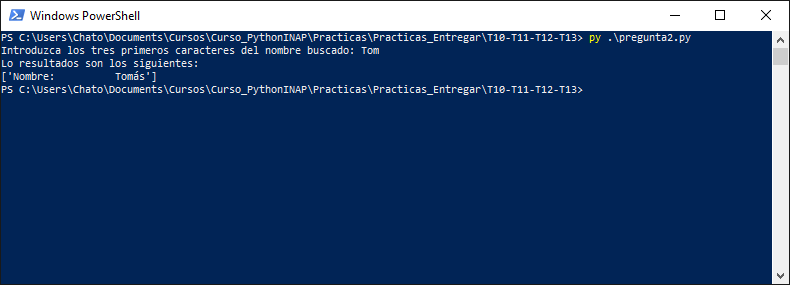
patron = r"Nombre:\W+\b"+ caracteres\_a\_buscar+r"\w+\b"

print("Lo resultados son los siguientes: " )

print( re.findall(patron,contenido, re.IGNORECASE))

## Faltaría mostrar las dos lineas siguientes, que no encuentro la forma.

**CAPTURA DE PANTALLA DEL RESULTADO OBTENIDO:**



### Pregunta 3. Tema 12 - Sockets.

Escribe el código en Python de un script que cree un cliente y otro script para un servidor que realicen la escucha a través del host ‘localhost’ y el puerto ‘8081’. La única funcionalidad que debe tener el servidor es mostrar por pantalla el mensaje del texto recibido y enviar el texto al cliente “Mensaje recibido”.

**CÓDIGO PROPUESTO POR EL ALUMNO:**

**Cliente.py**

import socket

def main():

host = 'localhost'

port = 8081

mySocket = socket.socket()

mySocket.connect((host, port))

message = input(" -> ")

while message != 'q':

mySocket.send(message.encode())

data = mySocket.recv(1024).decode()

print(data)

message = input(" -> ")

mySocket.close()

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

main()

**Servidor.py**

import socket

def main():

host = "localhost"

port = 8081

mySocket = socket.socket()

mySocket.bind((host, port))

mySocket.listen(1)

conn, addr = mySocket.accept()

while True:

data = conn.recv(1024).decode()

if not data:

break

print("Mesaje recibido: " + data)

mensaje= "Mensaje recibido"

conn.send(mensaje.encode())

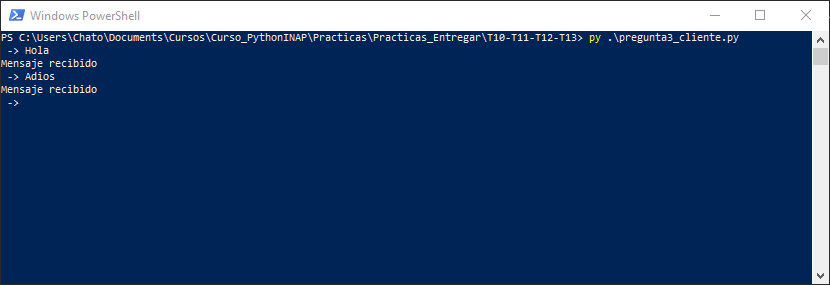
conn.close()

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

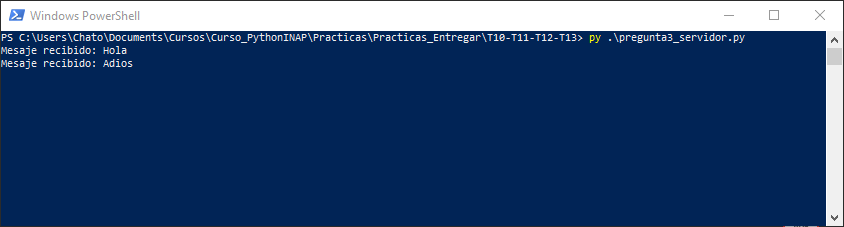
main()

**CAPTURA DE PANTALLA DEL RESULTADO OBTENIDO:**

**Cliente:**



**Servidor:**



### Pregunta 4. Tema 13 - Interactuar con webs – web scraping.

Escribe un script en Python que cargue la información de la web:

<https://www.ign.es/web/ign/portal/ultimos-terremotos/-/ultimos-terremotos/get30dias>

y que realice las siguientes funciones:

1. Que almacene el nombre de los terremotos próximos sentidos de los últimos 30 días (columna “Evento)”.
2. Que almacene el valor de la magnitud de cada evento sísmico (columna “Magnitud”).
3. Que imprima los datos de manera que el formato sea similar a lo siguiente:

EVENTO MAGNITUD

es2023dlesr 2.2

es2023della 2.2

… …

**CÓDIGO PROPUESTO POR EL ALUMNO:**

from bs4 import BeautifulSoup

import requests  
  
url = requests.get( "<https://www.ign.es/web/ign/portal/ultimos-terremotos/-/ultimos-terremotos/get30dias>")  
  
b\_soup=BeautifulSoup(url.content,"lxml")  
  
busca\_tb= b\_soup.find('table')  
  
## print(busca\_tb)  
  
busca\_filas= busca\_tb.find\_all('tr')  
  
## print(busca\_filas)

for linea in busca\_filas:

busca\_col= linea.find\_all('td')

##print(busca\_col)

if len(busca\_col) >=2:

col\_1= busca\_col[0].get\_text()

col\_6= busca\_col[6].get\_text()

print(f"{col\_1:15} {col\_6:15}")

**CAPTURA DE PANTALLA DEL RESULTADO OBTENIDO:**

