

Algoritmo PIA_SCRIPT y PIA_MODULO

Declaración de variables:

LISTA [()]: una lista que guarda los valores del api en tuplas

OPCION: número entero que selecciones de un menú de opciones

PASE: es el indicador que valida que primero deben de pasar por la opción 1 del menú

TOKEN: cadena de carácter con el token del api elegido

URL: cadena de carácter con la url del api elegido

DATOS: variable que convierte la petición a json

VALORES: lista de valores de las diferentes poblaciones que da el api

DATO: variable entera que representa el contador del ciclo de datos

MEDIA: variable que saca los valores de la lista VALORES en media

MEDIANA: variable que saca los valores de la lista VALORES convertidos a mediana

MODA: variable que saca los valores de la lista VALORES en moda

VARIANZA: variable que saca los valores de la lista VALORES convertidos a varianza

DESVIACION: variable que saca los valores de la lista VALORES convertidos a desviación estándar

NOMBRE [()]: lista que contiene la descripción del api en tuplas

EXCEL_GUARDAR: variable que manda a llamar a un archivo Excel

SHEET: variable que activa el guardado en el archivo de Excel

INDEX: variable entera que representa la columna en la repetición

POS: variable entera que representa la fila en la repetición

RESPONSE: hace un pedido a la url del api

INFO: hace el pedido a formato json del api

I: valor entero que representa el contador en el ciclo de info

PLT: variable que convierte valores para una grafica

- 1.-Inicio
- 2.- importar request, json, openpyxl, matplotlib, statistics
- 3.-hacer LISTA=[("Indicador", "Periodo", "Total de personas")]
- 4.-hacer TOKEN <- cadena de carácter con token del api
- 5.-hacer URL <- cadena de carácter con la url del api
- 6-hacer PASE=0
- 7.- leer OPCION
- 8.-mientras (OPCION !=4) repetir:
- 9.- si OPCION==1, entonces:
- 10.- Hacer PASE=1
- 11.- Hacer PETICION <- hace una petición a la url del api
- 12.- Hacer DATOS=PETICION <-hace la petición a json
- 13.- hacer VALORES=[]
- 14.- repetir DATO hasta DATOS["Series"]:
- 15.- hacer LISTA <-guardar datos en la lista
- 16.- Mostar DATO["INDICADOR"]
- 17.- Mostrar DATO["OBSERVATIONS"][0]["TIME_PERIOD"]
- 18.- Mostrar DATO["OBSERVATIONS"][0]["OBS_VALUE"]
- 19.- hacer VALORES<- guardar DATO["OBSERVATIONS"][0]["OBS_VALUE"]
- 20.- fin del ciclo paso 14
- 21.- hacer MEDIA <-saca la media de VALORES con statistic
- 22.- hacer MEDIANA <-saca la mediana de VALORES con statistic
- 23.- definir:
- 24.- hacer MODA <- saca la moda de VALORES con statistic
- 25.- excepción hay error en statistic
- 26.- hacer MODA="No hay una sola moda"
- 27.- hacer VARIANZA <- saca varianza de VALORES con statistic
- 28.- hacer DESVIACION <- saca la desviación estándar de VALORES con statistic
- 29.- mostrar MEDIA, MEDIANA, MODA, VARIANZA, DESVIACION
- 30.- fin de la condición paso9

```

31.- si OPCION==2, entonces:
32.-     si PASE==0, entonces:
33.-         mostrar "Primero extraiga los datos para guardar en Excel"
34.-     si no:
35.-         hacer NOMBRES= []<-contiene cadenas de carácter con descripciones
36.-         hacer EXCEL_GUARDAR <- abre un libro de Excel
37.-         SHEET=EXCEL_GUARDAR <-activa el guardado en Excel
38.-         repetir INDEX, POS en LISTA:
39.-             hacer SHEET ["A {INDEX}"]=POS [0]
40.-             hacer SHEET ["C {INDEX}"]=POS [1]
41.-             hacer SHEET ["D {INDEX}"]=POS [2]
42.-         Fin del ciclo paso 38
43.-         repetir INDEX, POS en NOMBRES:
44.-             hacer SHEET ["B {INDEX}"]=POS [0]
45.-         Fin del ciclo paso 43
46.-         hacer EXCEL_GUARDAR <- guarda los datos en "datos.xlsx"
47.-         mostrar "Se guardaron los datos con éxito"
48.-     fin de la condición paso 32
49.- fin de la condición paso 31
50.- si OPCION==3, entonces:
51.-     si PASE==0, entonces:
52.-         Mostar "Primeo extraiga los datos para ver la gráfica"
53.-     si no:
54.-         hacer RESPONSE <-manda una petición a la URL del api
55.-         hacer INFO <- devuelve la petición a json
56.-         hacer NOMBRES= []<-contiene cadenas de carácter con descripciones
57.-         hacer VALORES=[]
58.-         repetir I en INFO ("Series", []):
59.-             VALUE=I consigue
60.-             en ("OBSERVATIONS",{[]})[0] consigue en ("OBS_VALUE", None)

```

61.- si VALUES no es None, entonces:

62.- hacer VALORES <- guardar VALUES en VALORES

63.- fin de la condición paso 61

64.- fin del ciclo paso 58

65.- hacer PLT <-guarda los NOMBRE y VALORES en la grafica

66.- hacer PLT <-marca las ubicaciones en x de la grafica

67.- hacer PLT <- ajusta el parámetro de las figuras

68.- hacer PLT <- ejecuta la gráfica hecha

69.- fin de la condición paso 51

70.- fin de la condición paso 50

71.- leer OPCION

72.-fin del ciclo paso 8

73.-Fin