MANUAL TECNICO SIMPLEQL

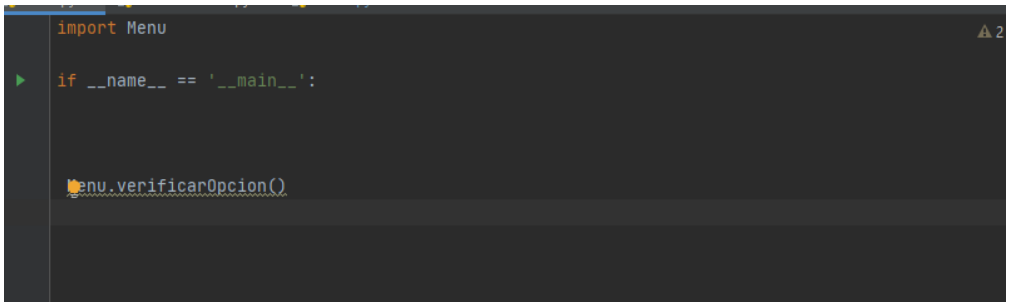
LFPSS2020 SECCION A

Selvin Orlando Hernández

201700603

CLASES EMPLEADAS EN LA PRACTICA

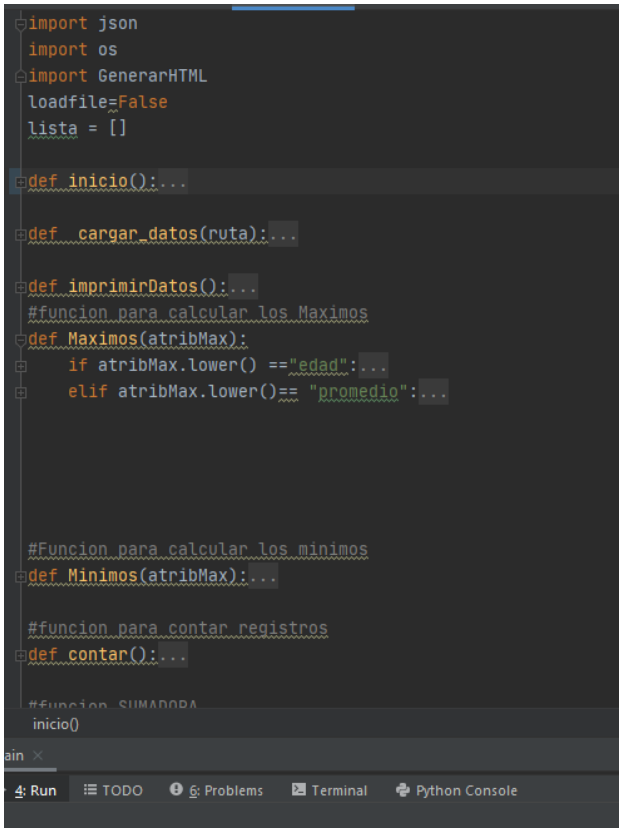
CLASE main



En ella se encentra la invocación del método **verificarOpcion** que se encuentra en la clase menú y que como su nombre lo indica analiza lo que el usuario quiere realizar

Clase Menu

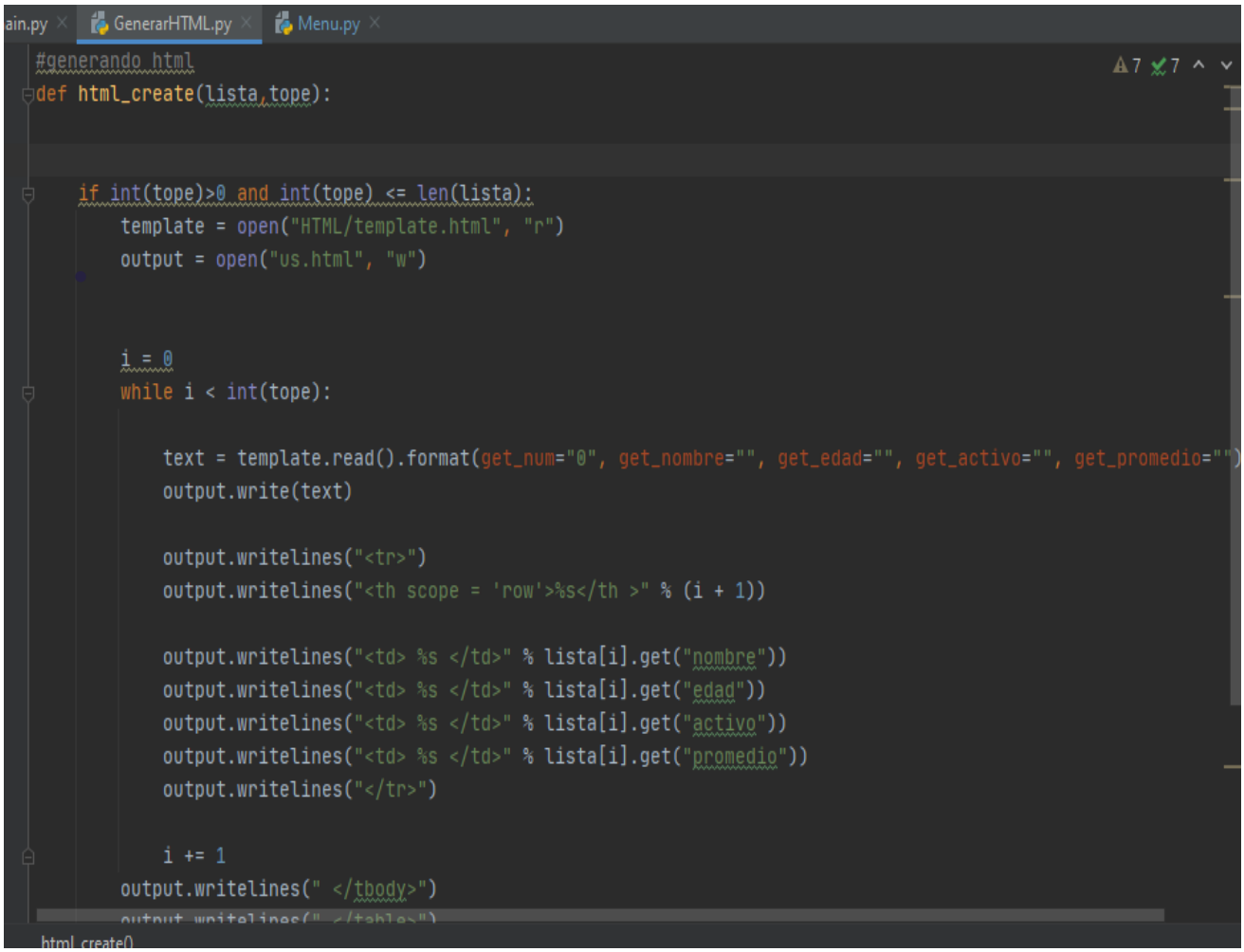
En esta clase se pondrá encontrar los métodos que verifican y realizan las tareas elegidas por el usuario como cargar, seleccionar, sumar, etc.



CLASES EMPLEADAS EN LA PRACTICA

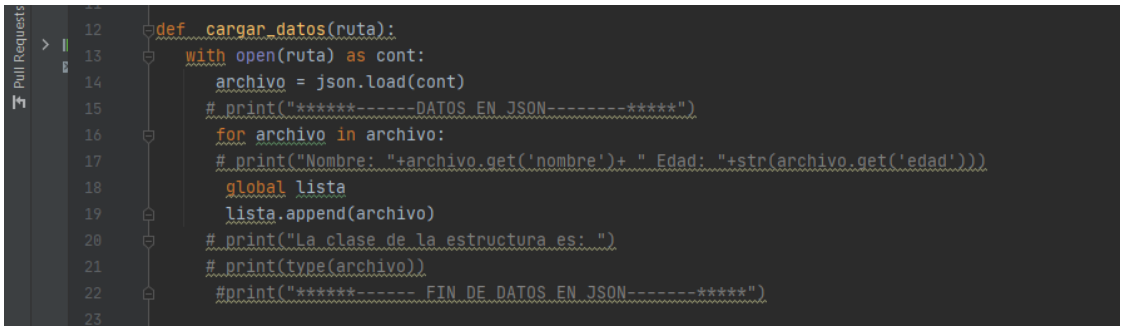
CLASE GenerarHTML

- Esta es la encargada de crear el reporte en HTML y almacenarla en la ubicación de la clase principal



PRINCIPALES METODOS DE LA CLASE MENU

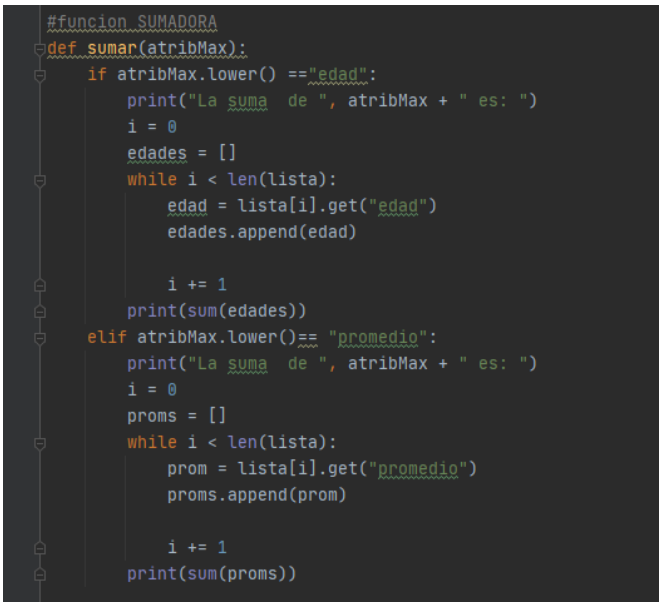
Cargar_datos Mediante la librería json podemos importa archivos en un diccionario y mantenerlo en memoria mientras se ejecuta el script



```
def  cargar_datos(ruta):
    with open(ruta) as cont:
        archivo = json.load(cont)
        # print("******------DATOS EN JSON--------*****")
        for archivo in archivo:
        # print("Nombre: "+archivo.get('nombre')+ " Edad: "+str(archivo.get('edad')))
         global lista
         lista.append(archivo)
        # print("La clase de la estructura es: ")
        # print(type(archivo))
         #print("******------ FIN DE DATOS EN JSON-------*****")
```

Luego de que se haya cargado en memoria se puede hacer uso de los diferentes métodos como lo son seleccionar, sumar, contar,etc.

Sumar: este método suma los valores de edad o promedio que se encuentren cargados en memoria



```
#funcion SUMADORA
def sumar(atribMax):
    if atribMax.lower() =="edad":
        print("La suma  de ", atribMax + " es: ")
        i = 0
        edades = []
        while i < len(lista):
            edad = lista[i].get("edad")
            edades.append(edad)

            i += 1
        print(sum(edades))
    elif atribMax.lower()== "promedio":
        print("La suma  de ", atribMax + " es: ")
        i = 0
        proms = []
        while i < len(lista):
            prom = lista[i].get("promedio")
            proms.append(prom)

            i += 1
        print(sum(proms))
```

Contar: este método devuelve la cantidad de registros cargados en memoria, el método únicamente utiliza la función len que retorna dicho valor

verficarOpcion:

Este es el encargado de realizar las distintas operaciones que permite el programa y básicamente es una serie de splits que comparan que comando y que valores se está ingresando

```
# INICIA BLOQUE DE OPCIONES
def verificarOpcion():



    while True:
        inicio()
        cadena = input("$ ")
        global loadfile
        verificar = cadena.split(sep=' ')

        #MENU cargar archivos
        try:
            if verificar[0].lower() == "cargar":
             # print("ESta en la funcion cargar")
             # print(verificar)
              i = 1
              while i < len(verificar):
               cargar_datos(verificar[i].replace(",", "") + ".json")
              # print(verificar[i].replace(",", ""))
               i += 1
               print("Archivo(s) cargado(s)")
            elif loadfile==False:
                loadfile = not loadfile
        except:
verificarOpcion()
```

```
            elif loadfile==False:
                loadfile = not loadfile
    except:
        print("ERROR: Archivo Inexistente, compruebe la escritura del nombre")

   #MENU SELECT
    try:

        if verificar[0].lower()=="seleccionar":
            if loadfile==True:
#               print("estas en la func. seleccionar")
                verificar.pop(0)
                eleccion= " ".join(verificar)
                imprimirDatos2(eleccion)
            else:
                print("No se ha cargado ningun archivo")

    except:
        print("ERROR: comando invalido o dato inexisten, vuelva a intentarlo")


   #Menu maximos

    try:

        if verificar[0].lower()=="maximo":
            if loadfile==True:
```

```
        if verificar[0].lower()=="maximo":
            if loadfile==True:
                print("")
                verificar.pop(0)
                eleccion= " ".join(verificar)
                Maximos(eleccion)
            else:
                print("No se ha cargado ningun archivo")

    except:
        print("ERROR: comando invalido o dato inexisten, vuelva a intentarlo")

       #menu minimos
    try:

        if verificar[0].lower()=="minimo":
            if loadfile==True:
                print("")
                verificar.pop(0)
                eleccion= " ".join(verificar)
                Maximos(eleccion)
            else:
                print("No se ha cargado ningun archivo")

    except:
        print("ERROR: comando invalido o dato inexisten, vuelva a intentarlo")
```

```
    #menu SUMAR
        try:

            if verificar[0].lower()=="suma":
                if loadfile==True:
                    print("")
                    verificar.pop(0)
                    eleccion= " ".join(verificar)
                    sumar(eleccion)
                else:
                    print("No se ha cargado ningun archivo")

        except:
            print("ERROR: comando invalido o dato inexisten, vuelva a intentarlo")

    #menu CONTAR
        try:

            if verificar[0].lower()=="cuenta":
                if loadfile==True:
                    print("")
                    contar()

                else:
                    print("No se ha cargado ningun archivo")

        except:
            print("ERROR: comando invalido o dato inexisten, vuelva a intentarlo")


    #menu REPORTAR
        try:

            if verificar[0].lower()=="reportar":
                if loadfile==True:
                    print("reportando")
                    verificar.pop(0)
                    eleccion = " ".join(verificar)
                    print(eleccion)
                    GenerarHTML.html_create(lista,eleccion)
                else:
                    print("No se ha cargado ningun archivo")

        except:
            print("ERROR: comando invalido o dato inexisten, vuelva a intentarlo")
```