

PRODUCTO INTEGRADOR DE APRENDIZAJE

Nombre de la Unidad de Aprendizaje: Actividad 2_ _____

Nombre de la Actividad: Repositorio de software compartido que contendrá los componentes desarrollados en esta fase.

Programa educativo: Licenciado en Tecnologías de Información

Semestre: 2

Grupo: Verano 7-11

Nombre del maestro: PEREZ SAUCEDA HIRAM ISAI _____

Nombre de los integrantes del equipo:

(En orden alfabético apellidos-nombres- # matrícula)

Torres Longoria Ximena Abigail #2011987

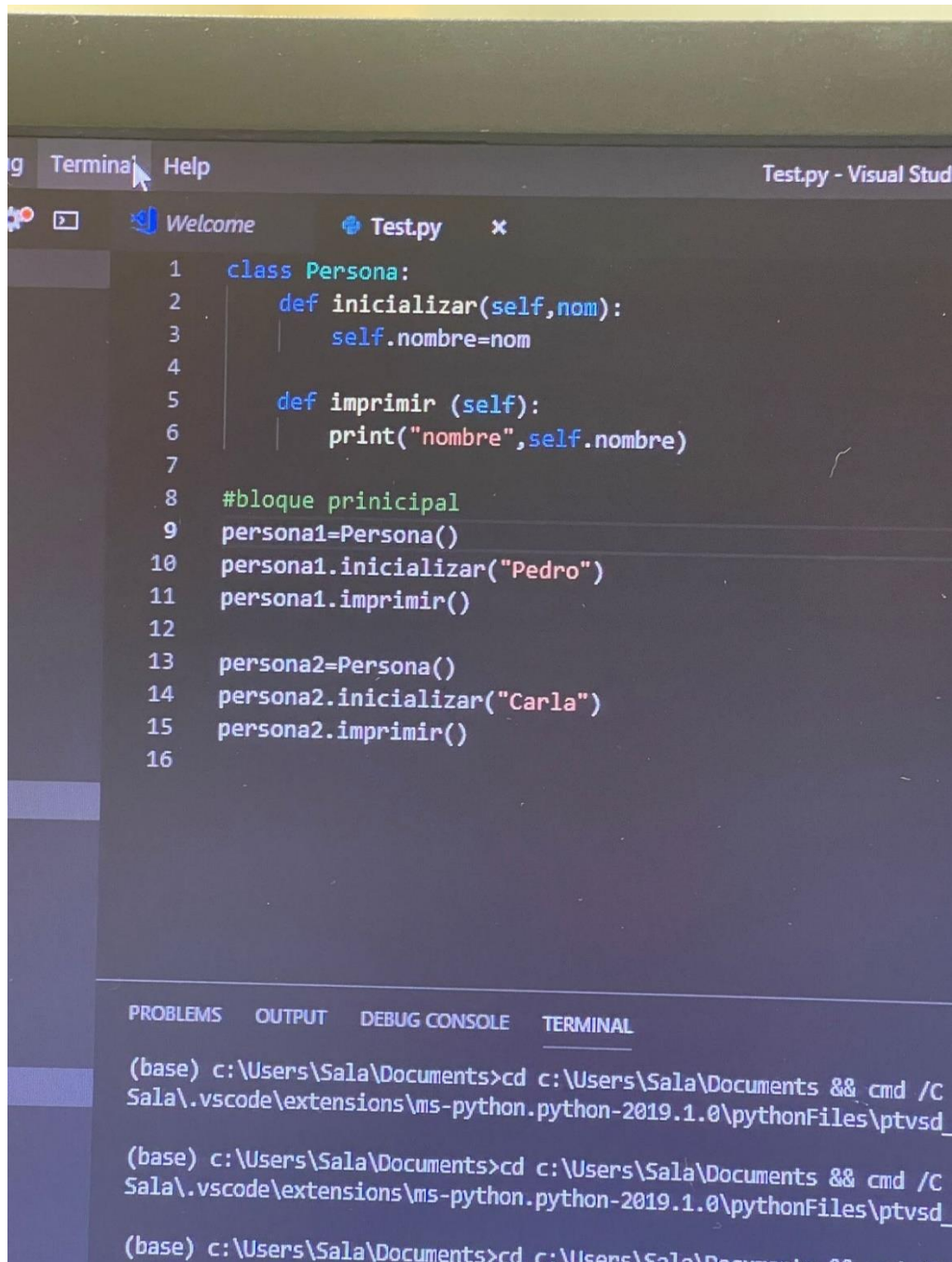
Contenido mínimo a evaluar	Cumplimiento
Índice	
Introducción .- incluye valores UANL aplicados	
Análisis y emisión de juicio	
Conclusiones individuales	
Conclusión del equipo	
Actividad en inglés	
Identificación de sub resultados de aprendizaje ANECA.	
Calificación PIA:	
Firma del maestro	

San Nicolás de los Garza, ciudad universitaria a (11/07/22)

DEF INICIALIZAR

def es una definición de función usada para crear objetos funciones definidas por el usuario.

Programa que Imprima el nombre y la edad



```
1 class Persona:
2     def inicializar(self,nom):
3         self.nombre=nom
4
5     def imprimir (self):
6         print("nombre",self.nombre)
7
8 #bloque principal
9 persona1=Persona()
10 persona1.inicializar("Pedro")
11 persona1.imprimir()
12
13 persona2=Persona()
14 persona2.inicializar("Carla")
15 persona2.imprimir()
16
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

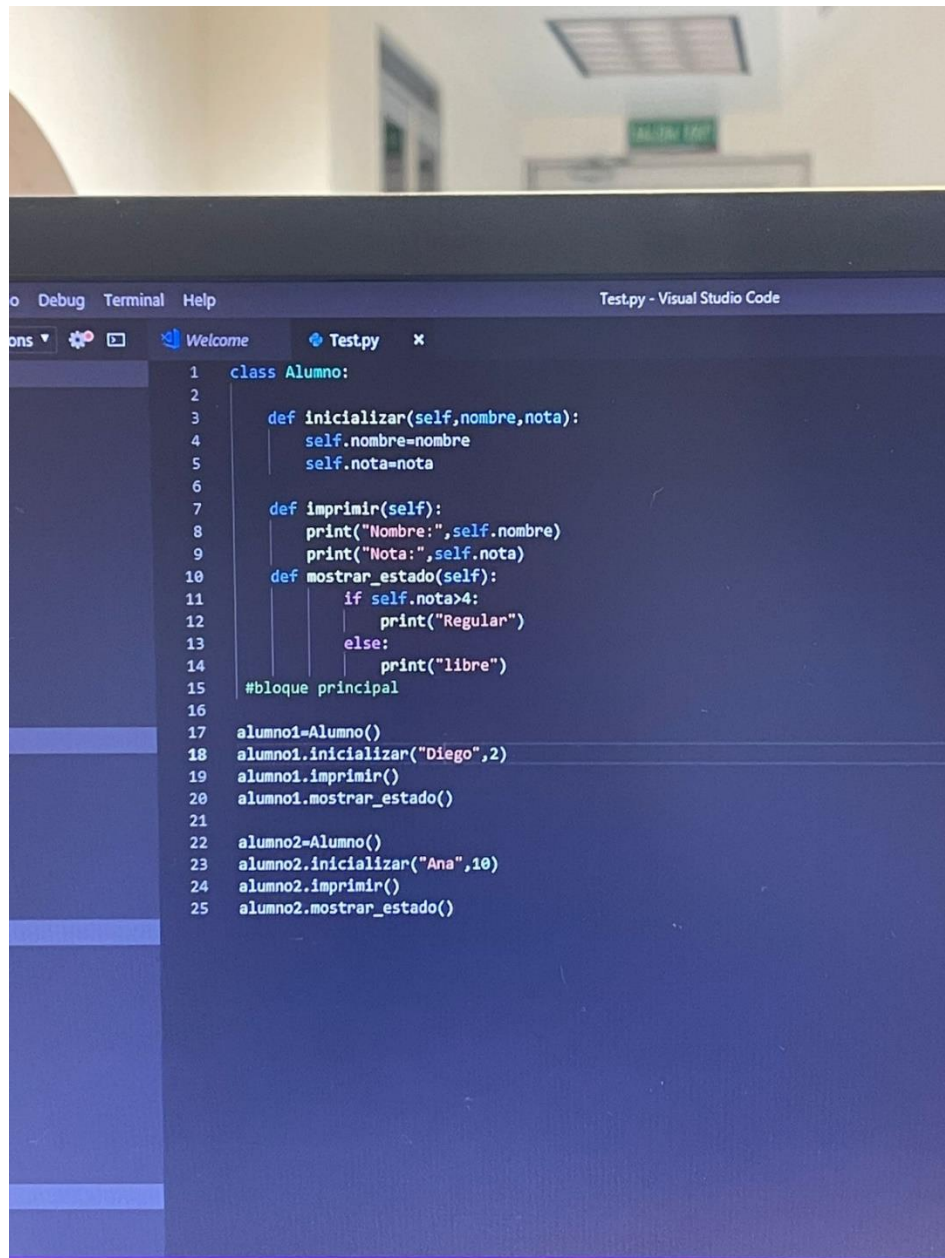
```
(base) c:\Users\Sala\Documents>cd c:\Users\Sala\Documents && cmd /C
Sala\.vscode\extensions\ms-python.python-2019.1.0\pythonFiles\ptvsd

(base) c:\Users\Sala\Documents>cd c:\Users\Sala\Documents && cmd /C
Sala\.vscode\extensions\ms-python.python-2019.1.0\pythonFiles\ptvsd

(base) c:\Users\Sala\Documents>cd c:\Users\Sala\Documents && cmd /C
```

DEF INICIALIZAR

Programa que imprima nombre y nota del alumno

A photograph of a computer screen displaying a Python script in the Visual Studio Code editor. The editor's interface includes a top menu bar with 'File', 'Edit', 'View', 'Test.py', and 'Help'. Below the menu is a toolbar with icons for Explorer, Search, Run and Debug, and a 'Test.py' button. The main editor area shows a Python file named 'Test.py' with the following code:

```
1 class Alumno:
2
3     def inicializar(self,nombre,nota):
4         self.nombre=nombre
5         self.nota=nota
6
7     def imprimir(self):
8         print("Nombre:",self.nombre)
9         print("Nota:",self.nota)
10
11     def mostrar_estado(self):
12         if self.nota>4:
13             print("Regular")
14         else:
15             print("libre")
16
17 #bloque principal
18
19 alumno1=Alumno()
20 alumno1.inicializar("Diego",2)
21 alumno1.imprimir()
22 alumno1.mostrar_estado()
23
24 alumno2=Alumno()
25 alumno2.inicializar("Ana",10)
26 alumno2.imprimir()
27 alumno2.mostrar_estado()
```

The code defines a class 'Alumno' with three methods: 'inicializar' to set attributes, 'imprimir' to print them, and 'mostrar_estado' to check the grade. It then creates two instances, 'alumno1' and 'alumno2', and calls their methods. The background of the photo shows a blurred hallway with a green exit sign.

DEF INICIALIZAR

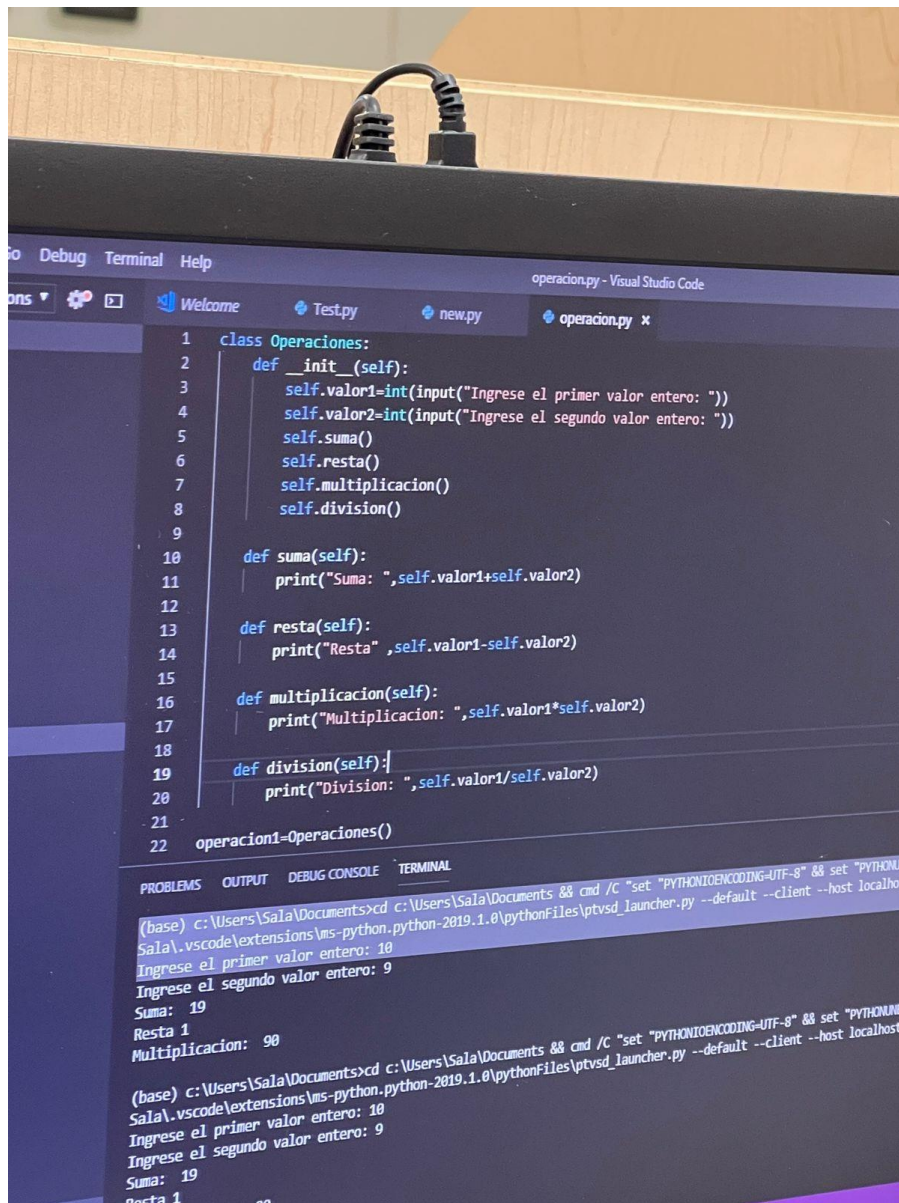
Programa en el cual ingresamos los lados de un triángulo y que identifique si es equilátero o no.

```
1 class Triangulo:
2
3     def inicializar(self):
4         self.lado1=int(input("Ingrese Primer lado:"))
5         self.lado2=int(input("Ingrese Segundo lado :"))
6         self.lado3=int(input("Ingrese Trecer lado:"))
7     def imprimir(self):
8         print("lado1",self.lado1)
9         print("lado2",self.lado2)
10        print("lado3",self.lado3)
11    def lado_mayor(self):
12        print("lado mayor")
13        if self.lado1>self.lado2 and self.lado1>self.lado3:
14            print(self.lado1)
15        else:
16            if self.lado2>self.lado3:
17                print(self.lado2)
18            else:
19                print(self.lado3)
20    def es_equilatero(self):
21        if self.lado1==self.lado2 and self.lado1==self.lado3:
22            print("El Triangulo es equilatero")
23        else:
24            print ("El triangulo no es equilatero")
25
26    #Bloque Principal
27
28    Triangulo1=Triangulo()
29    Triangulo1.inicializar()
30    Triangulo1.imprimir()
31    Triangulo1.lado_mayor()
32    Triangulo1.es_equilatero()
```


DEF __INIT__

init es una secuencia de comandos de shell ejecutable que termina todos los procesos activos en un sistema y, luego, sincroniza los discos antes de cambiar los niveles de ejecución.

Programa que llame directamente a las operaciones.



The image shows a computer monitor displaying the Visual Studio Code editor. The editor has a dark theme and shows a Python file named 'operacion.py'. The code defines a class 'Operaciones' with an '__init__' method and four other methods: 'suma', 'resta', 'multiplicacion', and 'division'. The '__init__' method prompts the user for two integers and calls the other four methods. The terminal window at the bottom shows the execution of the program, with the user entering '10' and '9', and the program outputting 'Suma: 19', 'Resta 1', and 'Multiplicacion: 90'.

```
1 class Operaciones:
2     def __init__(self):
3         self.valor1=int(input("Ingrese el primer valor entero: "))
4         self.valor2=int(input("Ingrese el segundo valor entero: "))
5         self.suma()
6         self.resta()
7         self.multiplicacion()
8         self.division()
9
10    def suma(self):
11        print("Suma: ",self.valor1+self.valor2)
12
13    def resta(self):
14        print("Resta" ,self.valor1-self.valor2)
15
16    def multiplicacion(self):
17        print("Multiplicacion: ",self.valor1*self.valor2)
18
19    def division(self):
20        print("Division: ",self.valor1/self.valor2)
21
22    operacion1=Operaciones()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
(base) c:\Users\Sala\Documents>cd c:\Users\Sala\Documents && cmd /C "set "PYTHONIOENCODING=UTF-8" && set "PYTHONUNBUFFERED=1" && python -u c:\Users\Sala\Documents\operacion.py --default --client --host localhost
Sala\vscode\extensions\ms-python.python-2019.1.0\pythonFiles\ptvsd_launcher.py
Ingrese el primer valor entero: 10
Ingrese el segundo valor entero: 9
Suma: 19
Resta 1
Multiplicacion: 90

(base) c:\Users\Sala\Documents>cd c:\Users\Sala\Documents && cmd /C "set "PYTHONIOENCODING=UTF-8" && set "PYTHONUNBUFFERED=1" && python -u c:\Users\Sala\Documents\operacion.py --default --client --host localhost
Sala\vscode\extensions\ms-python.python-2019.1.0\pythonFiles\ptvsd_launcher.py
Ingrese el primer valor entero: 10
Ingrese el segundo valor entero: 9
Suma: 19
Resta 1
Multiplicacion: 90
```

DEF __INIT__

Programa con menú que imprima una lista donde tengas que elegir la opción que desees. Y documentar la información que ingresamos.

```
class Administrar:
    def __init__(self):
        self.nombres=[]
        self.notas=[]
    def menu(self):
        opcion=0
        while opcion!=4:
            print("1-cargar alumnos")
            print("2-Listar alumnos")
            print("3-listado de alumnos con notas mayor")
            print("4-finalizar programa")
            opcion=int(input("Ingrese su opcion:"))
            if opcion==1:
                self.cargar()
            elif opcion==2:
                self.listar()
            elif opcion==3:
                self.notas altas()

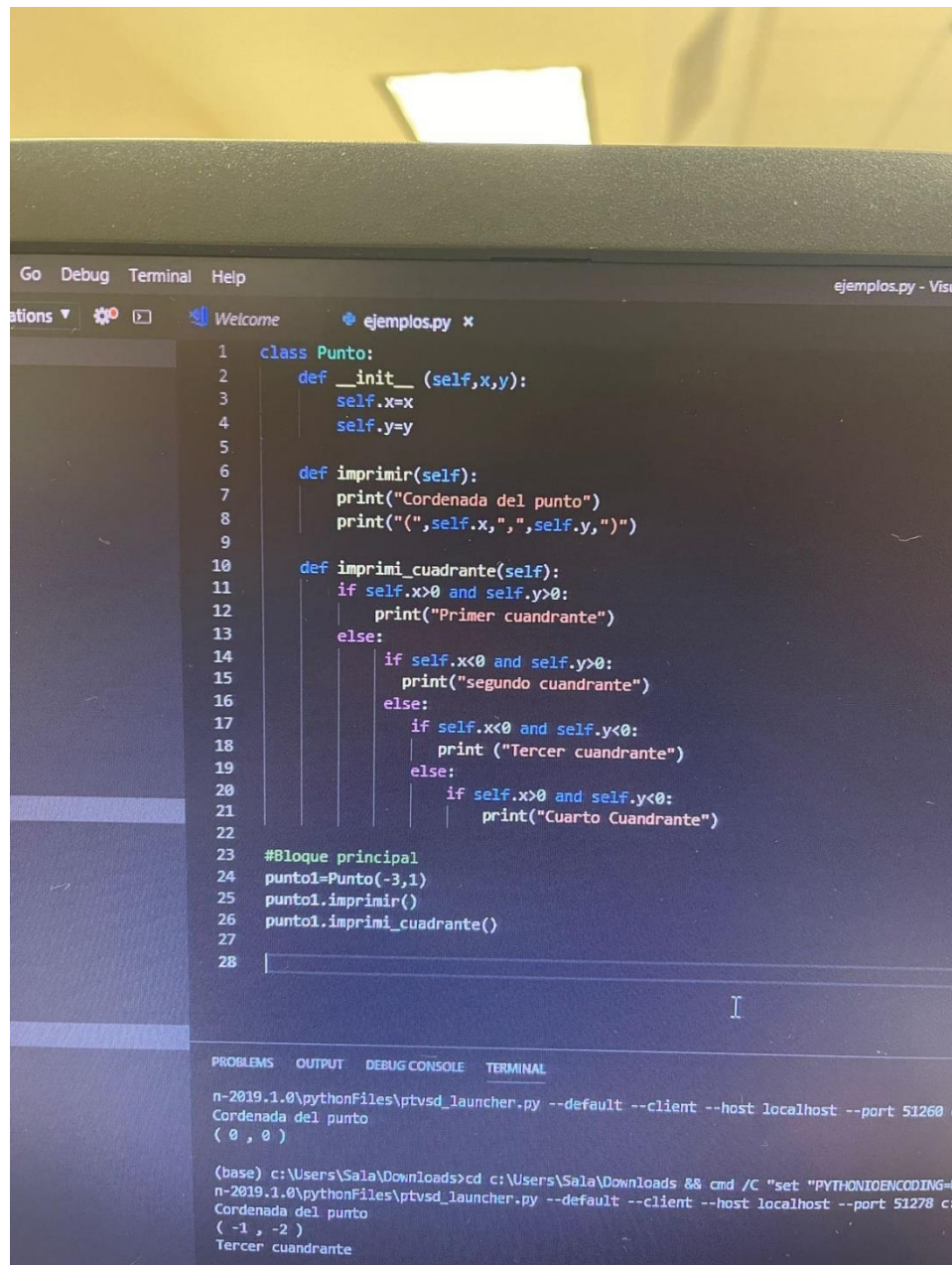
    def cargar(self):
        for x in range(5):
            nom=input("Ingrese nombre del alumno:")
            self.nombre.append(nom)
            no=int(input("nota del alumno:"))
            self.notas.append(no)

    def listar(self):
        print("listado completo de alumnos:")
        for x in range(5):
            print(self.nombres[x],self.notas[x])
            print("_____")

    def notas_altas(self):
        print("Alumnos con notas superiores o iguales a 7")
        for x in range (5):
            if self.notas[x]>=7:
                print(self.nombres[x],self.notas[x])
```


DEF __INIT__

Programa donde ingresas las coordenadas de un punto y te indica si es el primer, segundo, o tercer cuadrante.



```
1 class Punto:
2     def __init__(self,x,y):
3         self.x=x
4         self.y=y
5
6     def imprimir(self):
7         print("Cordenada del punto")
8         print("(" ,self.x," ,",self.y,")")
9
10    def imprimi_cuadrante(self):
11        if self.x>0 and self.y>0:
12            print("Primer cuadrante")
13        else:
14            if self.x<0 and self.y>0:
15                print("segundo cuadrante")
16            else:
17                if self.x<0 and self.y<0:
18                    print ("Tercer cuadrante")
19                else:
20                    if self.x>0 and self.y<0:
21                        print("Cuarto Cuadrante")
22
23    #Bloque principal
24    punto1=Punto(-3,1)
25    punto1.imprimir()
26    punto1.imprimi_cuadrante()
27
28
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

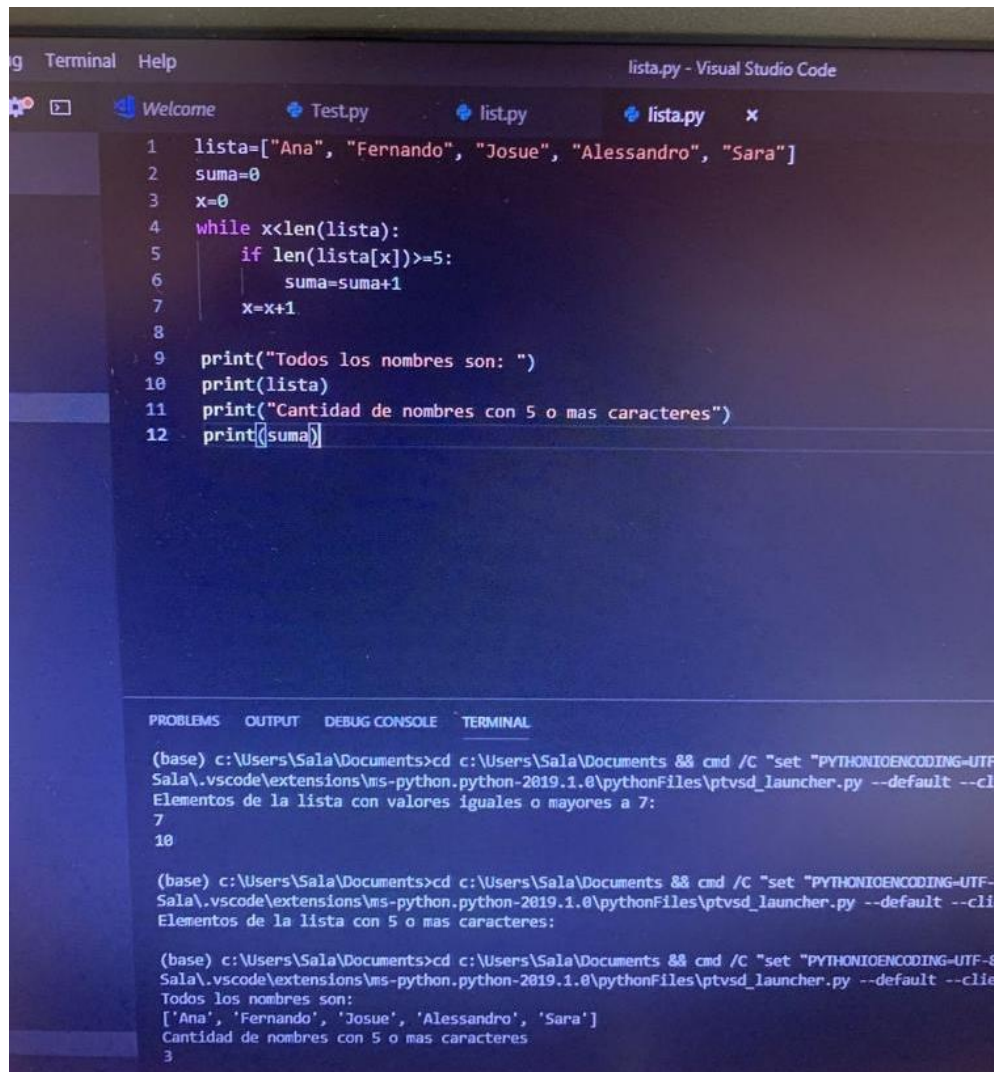
```
n-2019.1.0\pythonFiles\ptvsd_launcher.py --default --client --host localhost --port 51260 c
Cordenada del punto
( 0 , 0 )

(base) c:\Users\Sala\Downloads>cd c:\Users\Sala\Downloads && cmd /C "set "PYTHONIOENCODING=L
n-2019.1.0\pythonFiles\ptvsd_launcher.py --default --client --host localhost --port 51278 c:
Cordenada del punto
( -1 , -2 )
Tercer cuadrante
```

LISTAS

Son uno de los tipos de datos integrados en Python que se utilizan para almacenar colecciones de datos.

En esta lista documentamos los nombres y le pedimos que nos indique los nombres con mas de 5 caracteres.



```
1 lista=["Ana", "Fernando", "Josue", "Alessandro", "Sara"]
2 suma=0
3 x=0
4 while x<len(lista):
5     if len(lista[x])>=5:
6         suma=suma+1
7     x=x+1
8
9 print("Todos los nombres son: ")
10 print(lista)
11 print("Cantidad de nombres con 5 o mas caracteres")
12 print(suma)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

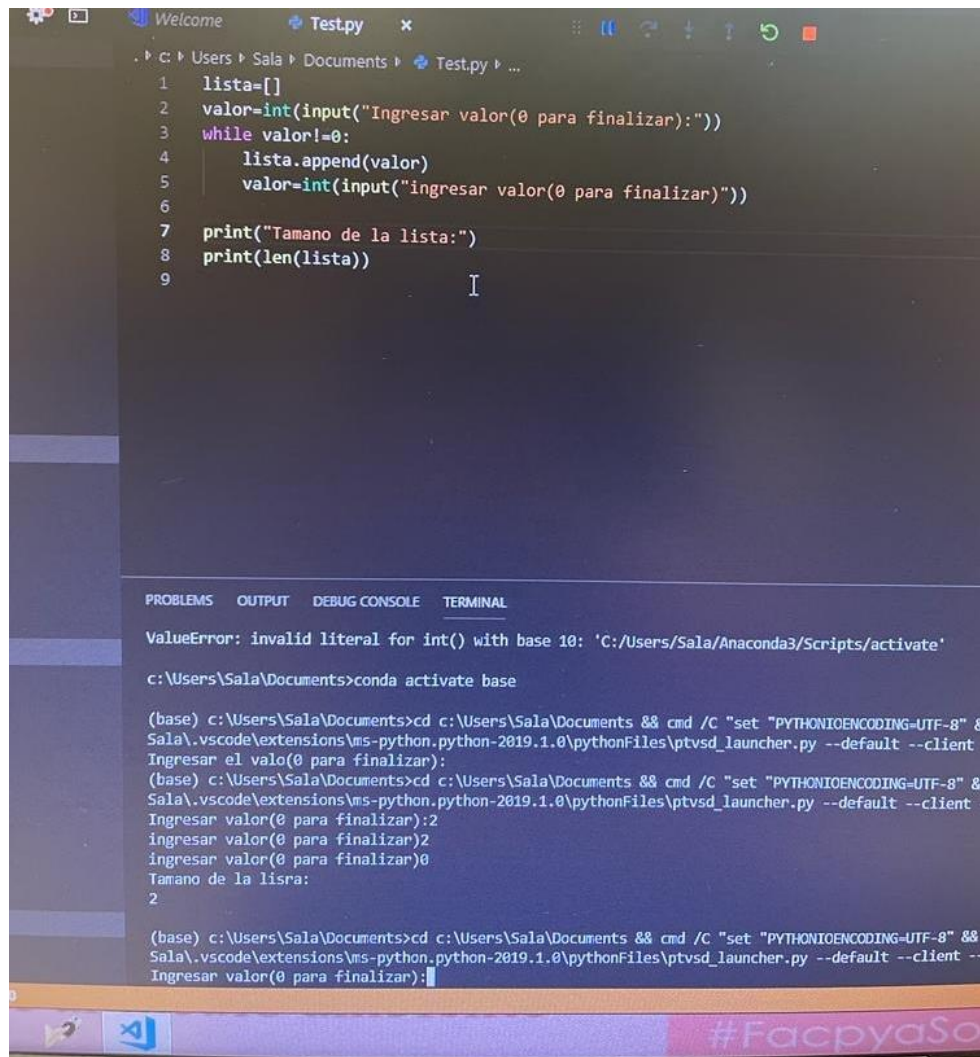
```
(base) c:\Users\Sala\Documents>cd c:\Users\Sala\Documents && cmd /C "set "PYTHONIOENCODING=UTF-8" && python lista.py"
Elementos de la lista con valores iguales o mayores a 7:
7
10

(base) c:\Users\Sala\Documents>cd c:\Users\Sala\Documents && cmd /C "set "PYTHONIOENCODING=UTF-8" && python lista.py"
Elementos de la lista con 5 o mas caracteres:

(base) c:\Users\Sala\Documents>cd c:\Users\Sala\Documents && cmd /C "set "PYTHONIOENCODING=UTF-8" && python lista.py"
Todos los nombres son:
['Ana', 'Fernando', 'Josue', 'Alessandro', 'Sara']
Cantidad de nombres con 5 o mas caracteres:
3
```


LISTAS

En estas lista pedimos que ingrese dos valores para ir agregando a la lista y ingresar 0 para finalizar , al finalizar nos imprime el tamaño de la lista.



```
1 lista=[]
2 valor=int(input("Ingresar valor(0 para finalizar):"))
3 while valor!=0:
4     lista.append(valor)
5     valor=int(input("ingresar valor(0 para finalizar)"))
6
7 print("Tamano de la lista:")
8 print(len(lista))
9
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

ValueError: invalid literal for int() with base 10: 'C:/Users/Sala/Anaconda3/Scripts/activate'

c:\Users\Sala\Documents>conda activate base

(base) c:\Users\Sala\Documents>cd c:\Users\Sala\Documents && cmd /C "set "PYTHONIOENCODING=UTF-8" && Sala\.vscode\extensions\ms-python.python-2019.1.0\pythonFiles\ptvsd_launcher.py --default --client

Ingresar el valo(0 para finalizar):

(base) c:\Users\Sala\Documents>cd c:\Users\Sala\Documents && cmd /C "set "PYTHONIOENCODING=UTF-8" && Sala\.vscode\extensions\ms-python.python-2019.1.0\pythonFiles\ptvsd_launcher.py --default --client

Ingresar valor(0 para finalizar):2

ingresar valor(0 para finalizar)2

ingresar valor(0 para finalizar)0

Tamano de la lisra:

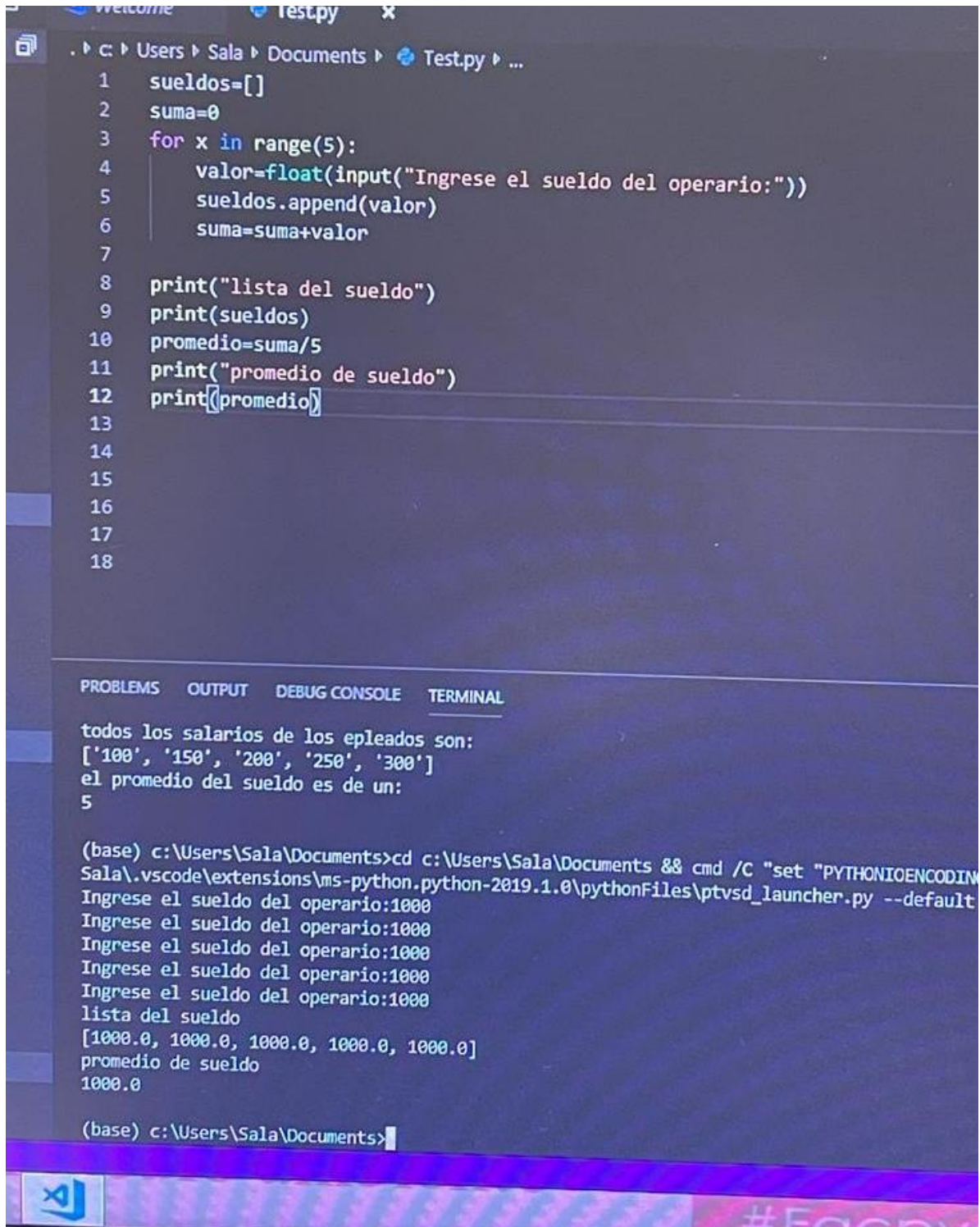
2

(base) c:\Users\Sala\Documents>cd c:\Users\Sala\Documents && cmd /C "set "PYTHONIOENCODING=UTF-8" && Sala\.vscode\extensions\ms-python.python-2019.1.0\pythonFiles\ptvsd_launcher.py --default --client --

Ingresar valor(0 para finalizar):

LISTAS

En esta lista pedimos al usuario que ingrese el sueldo del operario y que en una lista imprima el sueldo y el promedio del operario.



The image shows a screenshot of a Visual Studio Code editor window. The editor is open to a file named 'Test.py' located at 'c:\Users\Sala\Documents'. The code in the file is as follows:

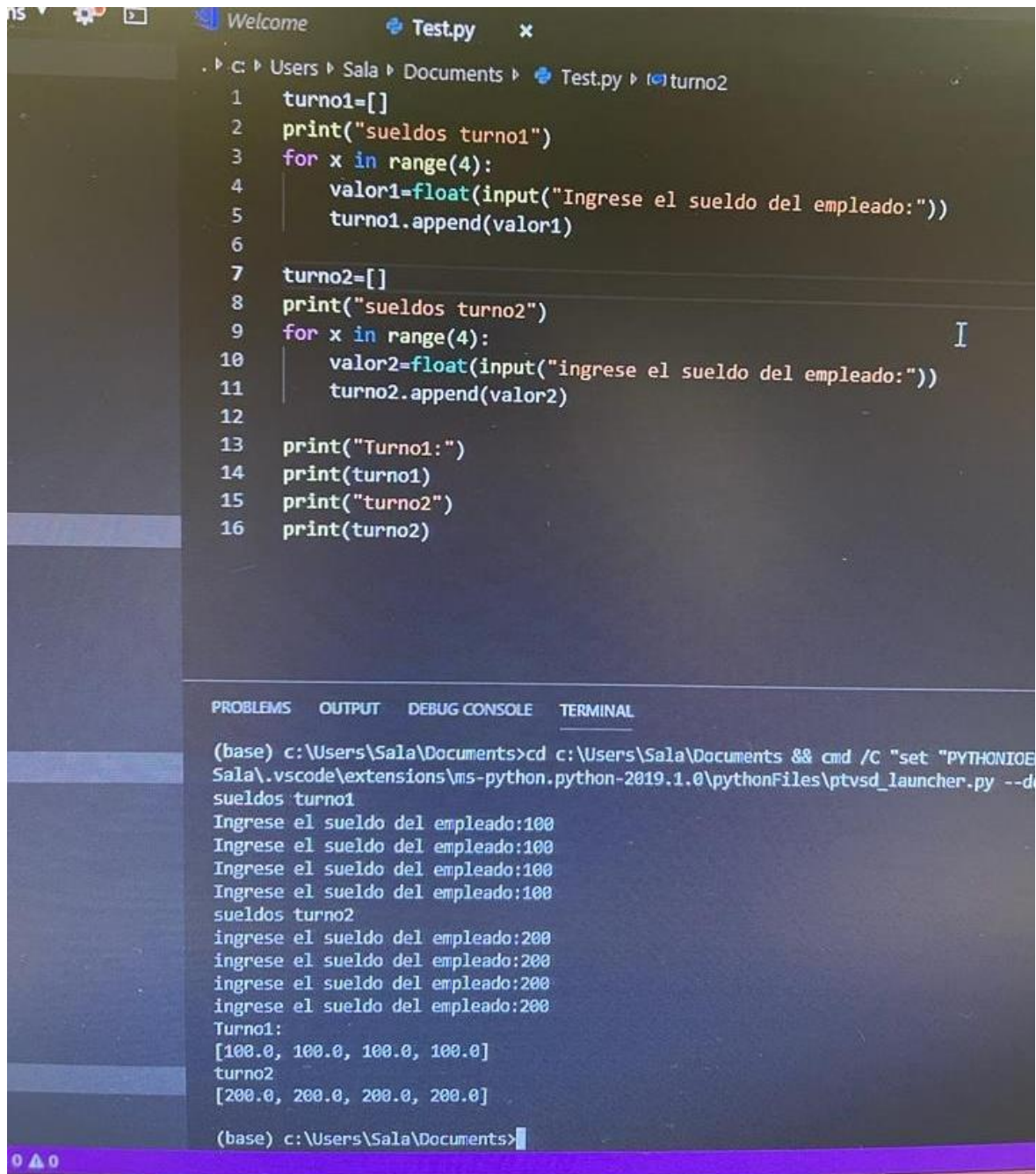
```
1 sueldos=[]
2 suma=0
3 for x in range(5):
4     valor=float(input("Ingrese el sueldo del operario:"))
5     sueldos.append(valor)
6     suma=suma+valor
7
8 print("lista del sueldo")
9 print(sueldos)
10 promedio=suma/5
11 print("promedio de sueldo")
12 print(promedio)
```

Below the editor, the 'TERMINAL' tab is active, showing the output of the script. The output indicates that the salaries of the employees are: ['100', '150', '200', '250', '300'] and the average salary is 5. Below this, the command prompt shows the execution of the script, where the user enters '1000' five times, and the output shows the list of salaries [1000.0, 1000.0, 1000.0, 1000.0, 1000.0] and the average salary 1000.0.

```
(base) c:\Users\Sala\Documents>cd c:\Users\Sala\Documents && cmd /C "set "PYTHONIOENCODING"
Sala\.vscode\extensions\ms-python.python-2019.1.0\pythonFiles\ptvsd_launcher.py --default
Ingrese el sueldo del operario:1000
Ingrese el sueldo del operario:1000
Ingrese el sueldo del operario:1000
Ingrese el sueldo del operario:1000
Ingrese el sueldo del operario:1000
lista del sueldo
[1000.0, 1000.0, 1000.0, 1000.0, 1000.0]
promedio de sueldo
1000.0
(base) c:\Users\Sala\Documents>
```

LISTAS

En esta lista, imprime dos listas en la cual una imprime el sueldo de los empleados del primer turno , y en la segunda lista imprime el sueldo de los empleados del segundo turno.



The image shows a VS Code editor window with a file named 'Test.py' open. The file path is 'C:\Users\Sala\Documents\Test.py'. The code in the file is as follows:

```
1  turno1=[]
2  print("sueldos turno1")
3  for x in range(4):
4      valor1=float(input("Ingrese el sueldo del empleado:"))
5      turno1.append(valor1)
6
7  turno2=[]
8  print("sueldos turno2")
9  for x in range(4):
10     valor2=float(input("ingrese el sueldo del empleado:"))
11     turno2.append(valor2)
12
13  print("Turno1:")
14  print(turno1)
15  print("turno2")
16  print(turno2)
```

Below the editor, the 'TERMINAL' tab is active, showing the execution of the script. The output is as follows:

```
(base) c:\Users\Sala\Documents>cd c:\Users\Sala\Documents && cmd /C "set "PYTHONIOE
Sala\.vscode\extensions\ms-python.python-2019.1.0\pythonFiles\ptvsd_launcher.py --d
sueldos turno1
Ingrese el sueldo del empleado:100
Ingrese el sueldo del empleado:100
Ingrese el sueldo del empleado:100
Ingrese el sueldo del empleado:100
sueldos turno2
ingrese el sueldo del empleado:200
ingrese el sueldo del empleado:200
ingrese el sueldo del empleado:200
ingrese el sueldo del empleado:200
Turno1:
[100.0, 100.0, 100.0, 100.0]
turno2
[200.0, 200.0, 200.0, 200.0]
(base) c:\Users\Sala\Documents>
```


LISTAS

En esta listas te pide el sueldo de 5 dias y te imprime una lista sin ordenar, y después te imprime la lista con el ultimo elemento ordenado

```
1  sueldos=[]
2  for x in range(5):
3      valor=int(input("Ingrese el sueldo:"))
4      sueldos.append(valor)
5
6  print("Lista sin ordenar")
7  print(sueldos)
8
9  for x in range(4):
10     if sueldos[x]>sueldos[x+1]:
11         aux=sueldos[x]
12         sueldos[x]=sueldos[x+1]
13         sueldos[x+1]=aux
14
15  print("lista con el ultimo elemento ordenado")
16  print(sueldos)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Lista sin ordenar
[900, 100, 500, 1100, 30]

LISTAS

En esta lista ingresamos 5 valores y nos imprime las listas acomodadas de menor a mayor , y después de mayor a menor.

```
1  lista=[]
2  for x in range(5):
3      valor=int(input("ingrese el valor:"))
4      lista.append(valor)
5
6  #Ordenamos de menor a mayor
7  for k in range(4):
8      for x in range(4-k):
9          if lista[x]>lista[x+1]:
10             aux=lista[x]
11             lista[x]=lista[x+1]
12             lista[x+1]=aux
13
14  print("lista ordenada de menor a mayor")
15  print(lista)
16
17  #ordenamos de mayor a menor
18  for k in range(4):
19      for x in range (4-k):
20          if lista [x]>lista[x+1]:
21             aux=lista[x]
22             lista[x+1]=aux
23
24  print("Lista ordenanda de meyor a menor")
25  print(lista)
```