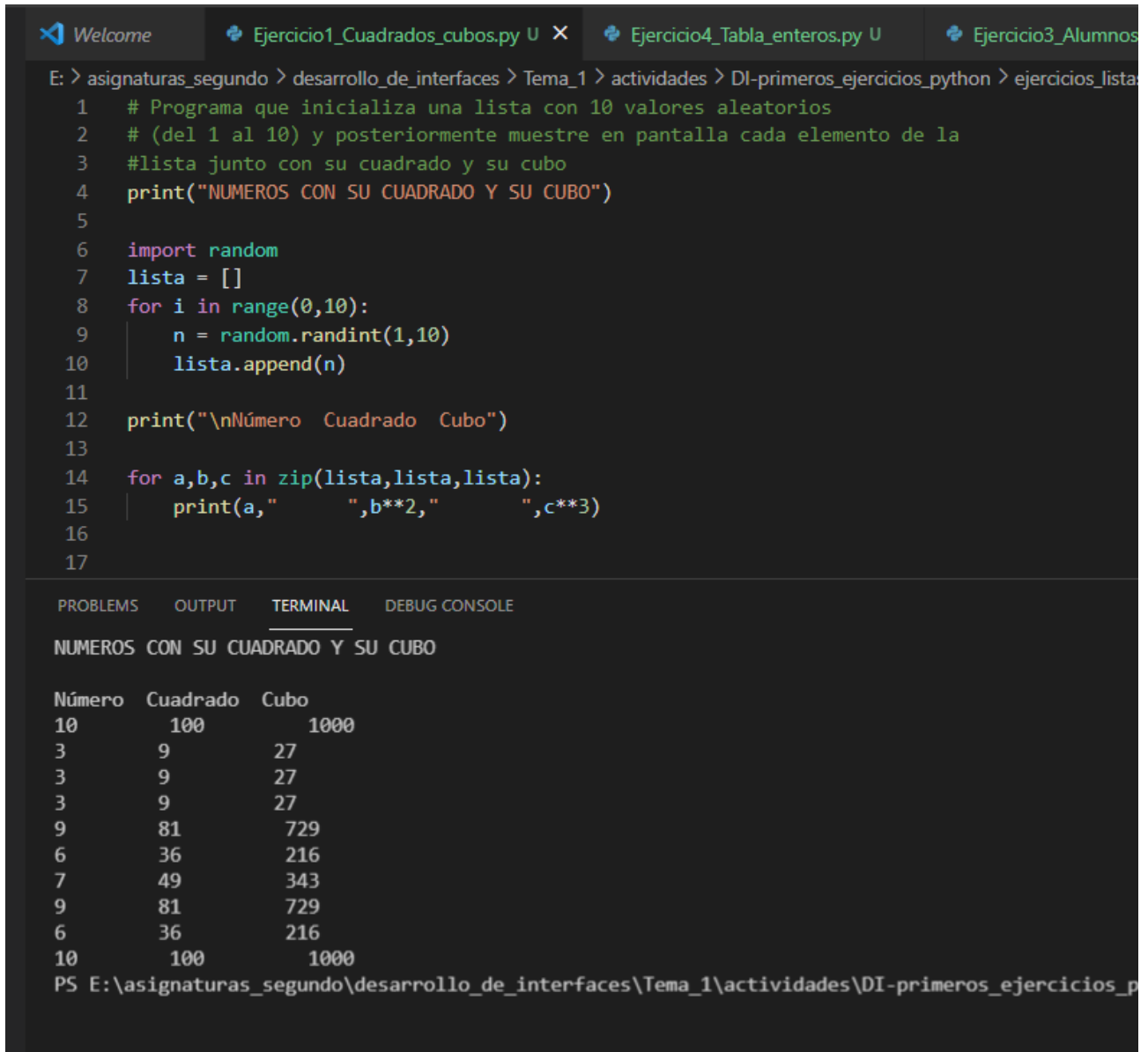


# Ejercicios listas

## Ejercicio 1: Cuadrados y cubos



The screenshot shows a code editor with four tabs: 'Welcome', 'Ejercicio1\_Cuadrados\_cubos.py U X', 'Ejercicio4\_Tabla\_enteros.py U', and 'Ejercicio3\_Alumnos'. The active tab is 'Ejercicio1\_Cuadrados\_cubos.py U X'. The code in the editor is as follows:

```
E: > asignaturas_segundo > desarrollo_de_interfaces > Tema_1 > actividades > DI-primeros_ejercicios_python > ejercicios_lista
1  # Programa que inicializa una lista con 10 valores aleatorios
2  # (del 1 al 10) y posteriormente muestre en pantalla cada elemento de la
3  # lista junto con su cuadrado y su cubo
4  print("NUMEROS CON SU CUADRADO Y SU CUBO")
5
6  import random
7  lista = []
8  for i in range(0,10):
9      n = random.randint(1,10)
10     lista.append(n)
11
12     print("\nNúmero  Cuadrado  Cubo")
13
14     for a,b,c in zip(lista,lista,lista):
15         print(a,"      ",b**2,"      ",c**3)
16
17
```

Below the code editor, the 'TERMINAL' tab is active, showing the output of the program:

```
NUMEROS CON SU CUADRADO Y SU CUBO

Número  Cuadrado  Cubo
10      100      1000
3       9       27
3       9       27
3       9       27
9       81      729
6       36      216
7       49      343
9       81      729
6       36      216
10     100     1000
```

The terminal output shows a table with three columns: 'Número', 'Cuadrado', and 'Cubo'. The numbers are generated randomly, and the output is formatted with spaces between the columns. The last line of the terminal output is 'PS E:\asignaturas\_segundo\desarrollo\_de\_interfaces\Tema\_1\actividades\DI-primeros\_ejercicios\_p'.

## Ejercicio 2: Notas alumnos

```
Welcome | Ejercicio1_Cuadrados_cubos.py U | Ejercicio2_Notas_alumnos.py U X | Ejercicio3_Alum...
E: > asignaturas_segundo > desarrollo_de_interfaces > Tema_1 > actividades > DI-primeros_ejercicios_python > ejercicios_l...
1  # Lee por teclado las 5 notas obtenidas por un alumno (comprendidas
2  # entre 0 y 10). A continuación, debe mostrar: todas las notas,
3  # la nota más alta que ha sacado y la menor
4  print("NOTAS DEL ALUMNO")
5
6  print("\nIntroduzca las notas del alumno una a una")
7
8  notas=[]
9
10 i=0
11 while(i<5):
12     nota = float(input("Nota: "))
13     if(nota<0 or nota>10):
14         print("Nota no válida. Las notas deben estar comprendidas entre 0 y 10.")
15     else:
16         notas.append(nota)
17         i=i+1
18
19
20 notamedia=sum(notas)/len(notas)
21 notaAlta=max(notas)
22 notaBaja=min(notas)
23
24 print("\nNotas: ",notas)
25 print("Nota media: ",notamedia)
26 print("Nota más alta: ",notaAlta)
27 print("Nota más baja: ",notaBaja)
```

PROBLEMS   OUTPUT   TERMINAL   DEBUG CONSOLE

I-primeros\_ejercicios\_python\ejercicios\_listas\Ejercicio2\_Notas\_alumnos.py'

NOTAS DEL ALUMNO

Introduzca las notas del alumno una a una  
Nota: 4  
Nota: 8.9  
Nota: 2.3  
Nota no válida. Las notas deben estar comprendidas entre 0 y 10.  
Nota: 2.3  
Nota: 7  
Nota: 10

Notas: [4.0, 8.9, 2.3, 7.0, 10.0]  
Nota media: 6.44  
Nota más alta: 10.0  
Nota más baja: 2.3  
PS E:\asignaturas\_segundo\desarrollo\_de\_interfaces\Tema\_1\actividades\DI-primeros\_ejercicio...

### Ejercicio 3: Alumnos y edades

```
Welcome | Ejercicio1_Cuadrados_cubos.py U | Ejercicio2_Notas_alumnos.py U | Ejercicio3_Alumnos_edades.py U X
E: > asignaturas_segundo > desarrollo_de_interfaces > Tema_1 > actividades > DI-primeros_ejercicios_python > ejercicios_listas > Ejercicio3_Alumnos

1  # Programa que registra nombre y edad de varios alumnos. El proceso
2  # de lectura de datos termina cuando se introduce como nombre un
3  # asterisco (*). Al finalizar se muestran:
4  # - Todos los alumnos mayores de edad
5  # - Los alumnos mayores (que tienen más edad)
6  print("ALUMNOS Y SUS EDADES")
7
8  nombres=[]
9  edades=[]
10
11 nombre=""
12 edad=0
13
14 while(nombre!="*"):
15     nombre=str(input("\nNombre: "))
16     if(nombre!="*"):
17         nombres.append(nombre)
18         edad=int(input("Edad: "))
19         edades.append(edad)
20
21 print("\nAlumnos mayores de edad:")
22 i=0
23 for num in edades:
24     if num>=18:
25         j=edades.index(num,i)
26         print(nombres[j],"",edades[j],"años.")
27         i=i+1
28
29 print("\nAlumnos más mayores:")
30 edadmax=max(edades)
31 i=0
32 for num in edades:
33     if num==edadmax:
34         j=edades.index(num,i)
35         print(nombres[j],"",edades[j],"años.")
36         i=i+1
37
```

PROBLEMS    OUTPUT    TERMINAL    DEBUG CONSOLE

Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma <https://aka.ms/pscore6>

```
PS E:\asignaturas_segundo\desarrollo_de_interfaces\Tema_1\actividades\DI-primeros_ejercicios_
:\Users\Estefania\.vscode\extensions\ms-python.python-2021.9.1246542782\pythonFiles\lib\pytho
I-primeros_ejercicios_python\ejercicios_listas\Ejercicio3_Alumnos_edades.py'
ALUMNOS Y SUS EDADES
```

Nombre: Juan  
Edad: 34

Nombre: Pedro  
Edad: 23

Nombre: Lucía  
Edad: 17

Nombre: Paula  
Edad: 78

Nombre: Antonio  
Edad: 78

Nombre: Julia  
Edad: 12

Nombre: \*

Alumnos mayores de edad:  
Juan , 34 años.  
Pedro , 23 años.  
Paula , 78 años.  
Antonio , 78 años.

Alumnos más mayores:  
Paula , 78 años.  
Antonio , 78 años.

```
PS E:\asignaturas_segundo\desarrollo_de_interfaces\Tema_1\actividades\DI-primeros_ejercicios_
```

## Ejercicio 4: Tabla 5x5

```

Welcome  Ejercicio1_Cuadrados_cubos.py U  Ejercicio2_Notas_alumnos.py U  Ejercicio3_Alumnos_edades.py U  Ejercicio4_Tabla_enteros.py U
E: > asignaturas_segundo > desarrollo_de_interfaces > Tema_1 > actividades > DI-primeros_ejercicios_python > ejercicios_listas > Ejercicio4_Tabla_enteros.py > ...
1  # Algoritmo que:
2  # - Crea una tabla (lista con dos dimensiones) de 5x5 enteros
3  # - Carga la tabla con valores numéricos enteros
4  # - Suma los elementos de cada fila y todos los elementos de cada
5  #   columna visualizando los resultados por pantalla
6  print("TABLA DE ENTEROS 5X5")
7  # No se puede crear una tabla de 5x5 vacía. Una tabla vacía por defición tendrá
8  # dimensiones 0x0. Si tiene una dimensión determinada distinta de cero,
9  # tiene que tener elementos dentro. Por ello, el algortimo creará la tabla vacía
10 # y luego añadirá los valores enteros hasta alcanzar las dimensiones 5X5.
11 # (Podría interpretarse que el primer paso pide una tabla 5x5 llena de ceros,
12 # pero no se trataría de una tabla vacía).
13
14 print("\nCreación de tabla vacía:")
15 tabla=[]
16 print(tabla)
17
18 print("\nCreación de tabla 5x5 con valores enteros:")
19
20 import random
21
22 for j in range(0,5):
23     fila = []
24     for i in range(0,5):
25         n = random.randint(0,50)
26         fila.append(n)
27     tabla.append(fila)
28
29 print(tabla)
30
31 print("\nSuma elementos de cada fila")
32 j=0
33 for fila in tabla:
34     print("Fila",j+1,":",sum(fila))
35     j=j+1
36
37 print("\nSuma elementos de cada columna")
38 j=0
39 while(j<len(tabla)):
40     i=0
41     sumaColumna=0
42     while (i<len(tabla)):
43         sumaColumna=sumaColumna+tabla[i][j]
44         i=i+1
45     print("Columna",j+1,":",sumaColumna)
46     j=j+1
47
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS E:\asignaturas_segundo\desarrollo_de_interfaces\Tema_1\actividades\DI-primeros_ejercicios_python\ejercicios_listas> & '
:\Users\Estefania\.vscode\extensions\ms-python.python-2021.9.1246542782\pythonFiles\lib\python\debugpy\launcher' '57299' '-
I-primeros_ejercicios_python\ejercicios_listas\Ejercicio4_Tabla_enteros.py'
TABLA DE ENTEROS 5X5

Creación de tabla vacía:
[]

Creación de tabla 5x5 con valores enteros:
[[22, 42, 48, 44, 25], [1, 7, 25, 38, 37], [49, 36, 10, 36, 49], [20, 38, 30, 49, 23], [16, 24, 50, 39, 35]]

Suma elementos de cada fila
Fila 1 : 181
Fila 2 : 108
Fila 3 : 180
Fila 4 : 160
Fila 5 : 164

Suma elementos de cada columna
Columna 1 : 108
Columna 2 : 147
Columna 3 : 163
Columna 4 : 206
Columna 5 : 169
PS E:\asignaturas_segundo\desarrollo_de_interfaces\Tema_1\actividades\DI-primeros_ejercicios_python\ejercicios_listas>

```

