

Programación II

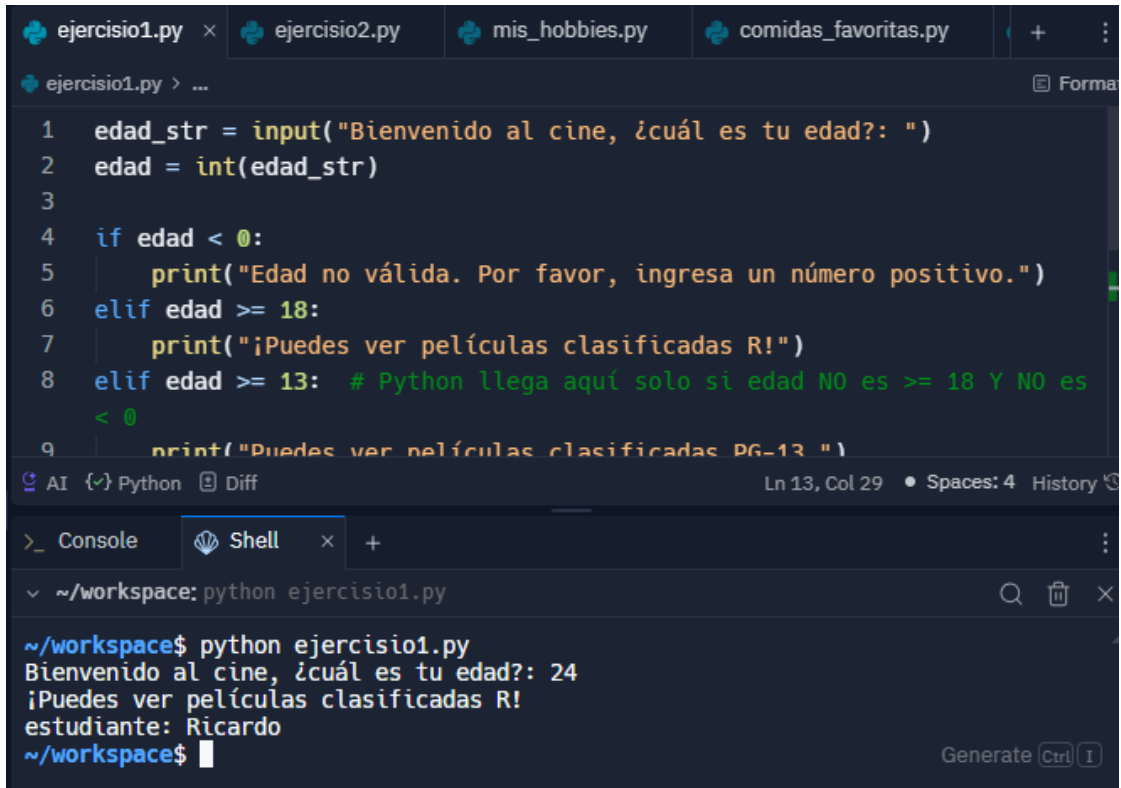
Ricardo Maldonado Suarez

CAPTURAS DE EJERCISIO



CamScanner

EDAD PARA VER PELICULAS



```
ejercicio1.py x ejercicio2.py mis_hobbies.py comidas_favoritas.py + ...
ejercicio1.py > ... Format

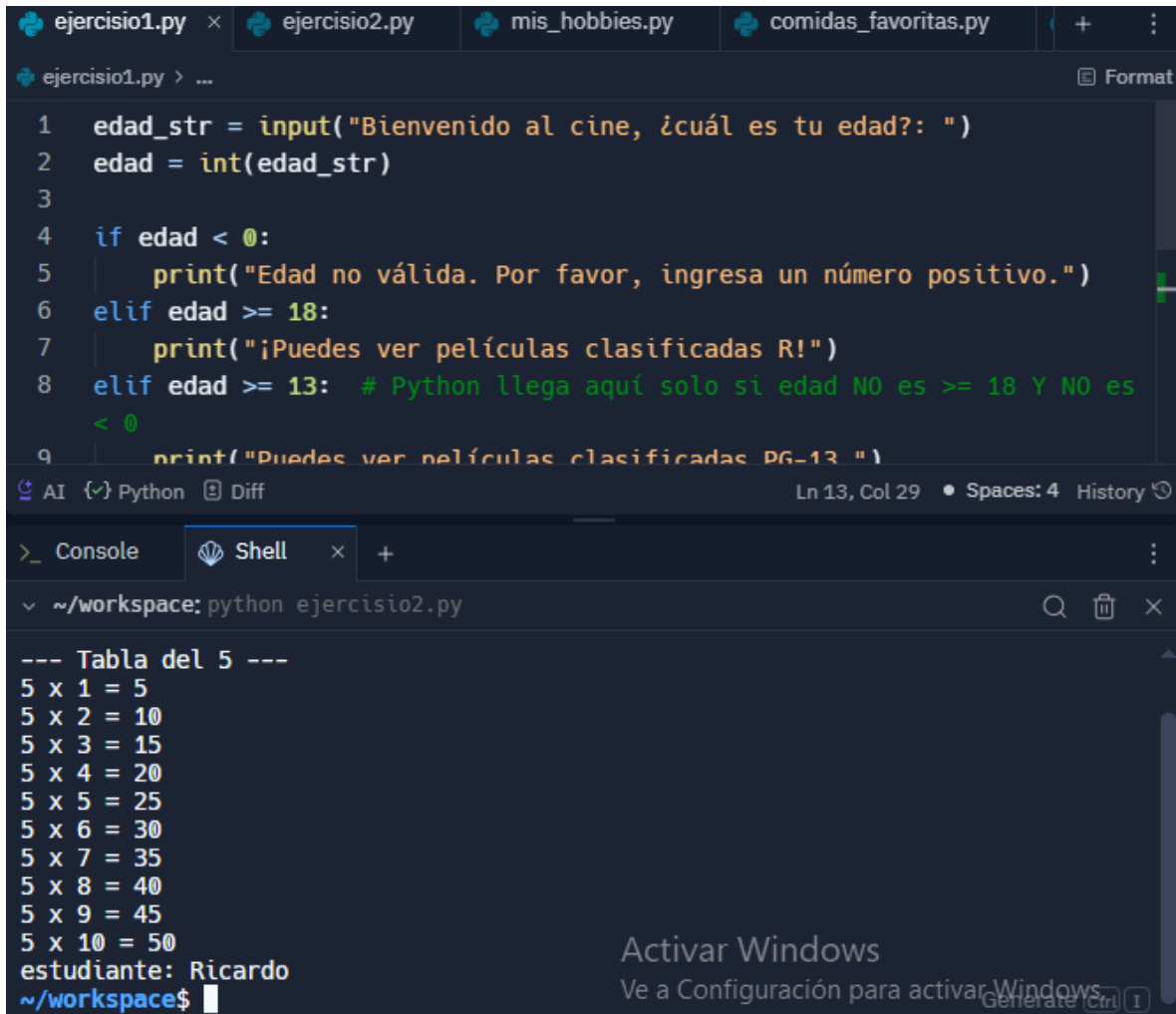
1 edad_str = input("Bienvenido al cine, ¿cuál es tu edad?: ")
2 edad = int(edad_str)
3
4 if edad < 0:
5     print("Edad no válida. Por favor, ingresa un número positivo.")
6 elif edad >= 18:
7     print("¡Puedes ver películas clasificadas R!")
8 elif edad >= 13: # Python llega aquí solo si edad NO es >= 18 Y NO es
9     < 0
10    print("Puedes ver películas clasificadas PG-13.")

Ln 13, Col 29 • Spaces: 4 History

>_ Console Shell x +
~/workspace: python ejercicio1.py

~/workspace$ python ejercicio1.py
Bienvenido al cine, ¿cuál es tu edad?: 24
¡Puedes ver películas clasificadas R!
estudiante: Ricardo
~/workspace$
```

TABLA DE MULTIPLICAR



The image shows a code editor with four tabs: `ejercicio1.py`, `ejercicio2.py`, `mis_hobbies.py`, and `comidas_favoritas.py`. The `ejercicio1.py` tab is active, displaying a Python script that checks a user's age and prints a message based on whether they are old enough to watch R-rated movies. The script is as follows:

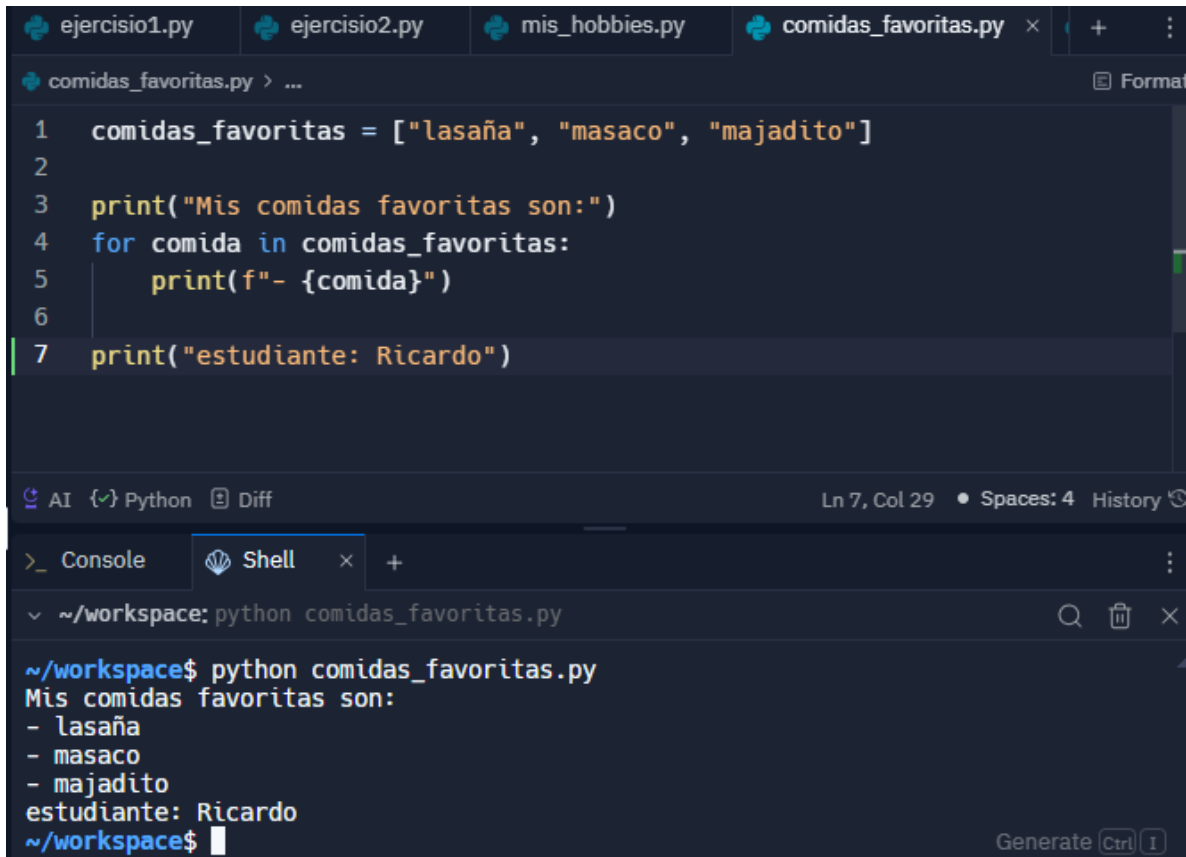
```
1 edad_str = input("Bienvenido al cine, ¿cuál es tu edad?: ")
2 edad = int(edad_str)
3
4 if edad < 0:
5     print("Edad no válida. Por favor, ingresa un número positivo.")
6 elif edad >= 18:
7     print("¡Puedes ver películas clasificadas R!")
8 elif edad >= 13: # Python llega aquí solo si edad NO es >= 18 Y NO es
9     print("Puedes ver películas clasificadas PG-13.")
```

Below the code editor is a terminal window titled "Shell". It shows the command `python ejercicio2.py` being executed. The output of the script is a multiplication table for the number 5, followed by the student's name and the current directory:

```
--- Tabla del 5 ---
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
estudiante: Ricardo
~/workspace$
```

An "Activar Windows" watermark is visible in the bottom right corner of the terminal window.

COMIDAS FAVORITAS



The image shows a code editor with a dark theme. At the top, there are four tabs: 'ejercicio1.py', 'ejercicio2.py', 'mis_hobbies.py', and 'comidas_favoritas.py'. The 'comidas_favoritas.py' tab is active. Below the tabs, the code editor shows the following Python code:

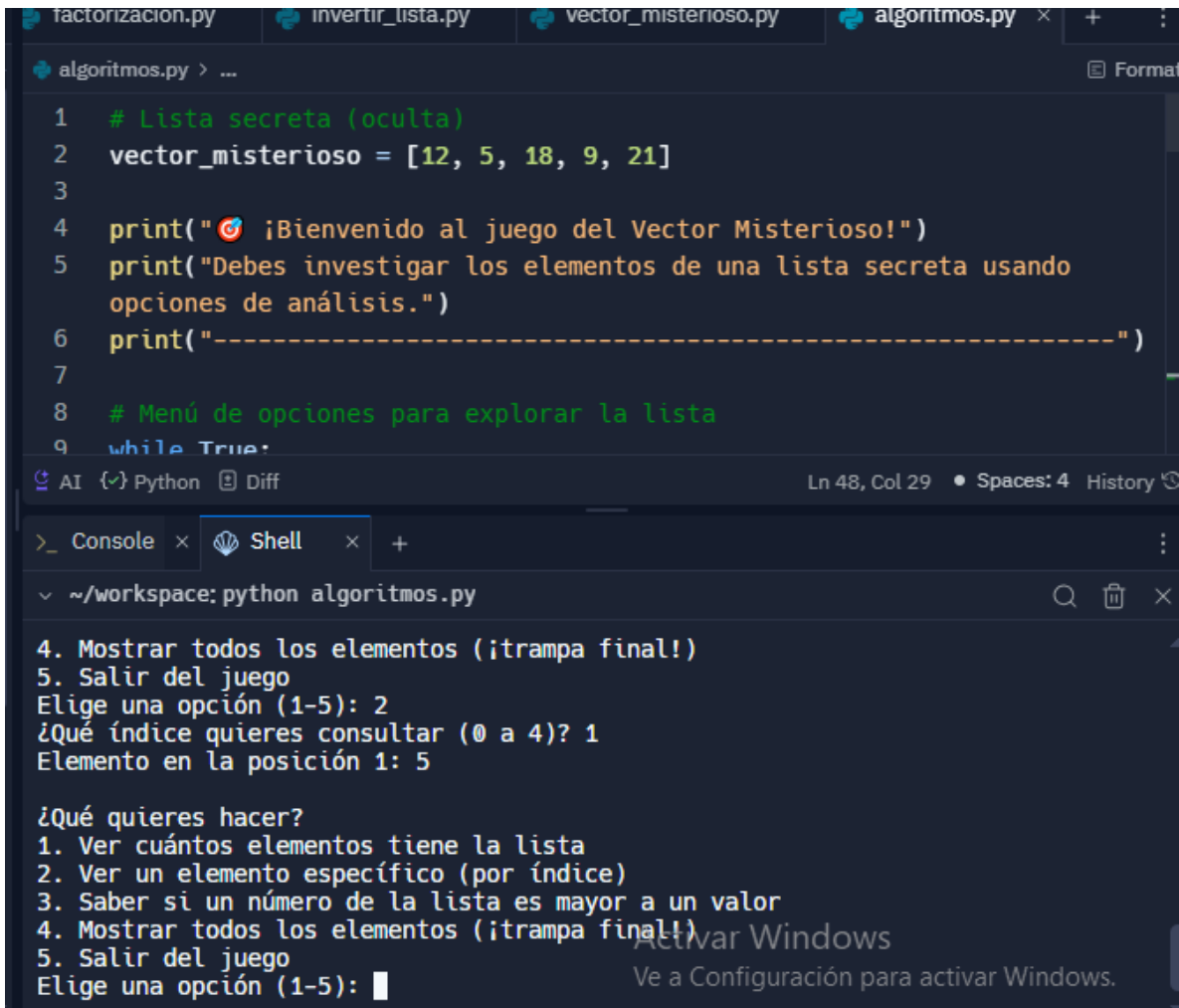
```
1  comidas_favoritas = ["lasaña", "masaco", "majadito"]
2
3  print("Mis comidas favoritas son:")
4  for comida in comidas_favoritas:
5      print(f"- {comida}")
6
7  print("estudiante: Ricardo")
```

Below the code editor, there is a 'Console' tab and a 'Shell' tab. The 'Shell' tab is active, showing the command prompt output of the script:

```
~/workspace$ python comidas_favoritas.py
Mis comidas favoritas son:
- lasaña
- masaco
- majadito
estudiante: Ricardo
~/workspace$
```

At the bottom right of the shell window, there is a 'Generate' button with a 'Ctrl' key icon.

ALGORITMOS



The screenshot shows a code editor with several tabs: `factorizacion.py`, `invertir_lista.py`, `vector_misterioso.py`, and `algoritmos.py`. The `algoritmos.py` tab is active, displaying the following Python code:

```
1 # Lista secreta (oculta)
2 vector_misterioso = [12, 5, 18, 9, 21]
3
4 print("🎮 ¡Bienvenido al juego del Vector Misterioso!")
5 print("Debes investigar los elementos de una lista secreta usando
   opciones de análisis.")
6 print("-----")
7
8 # Menú de opciones para explorar la lista
9 while True:
```

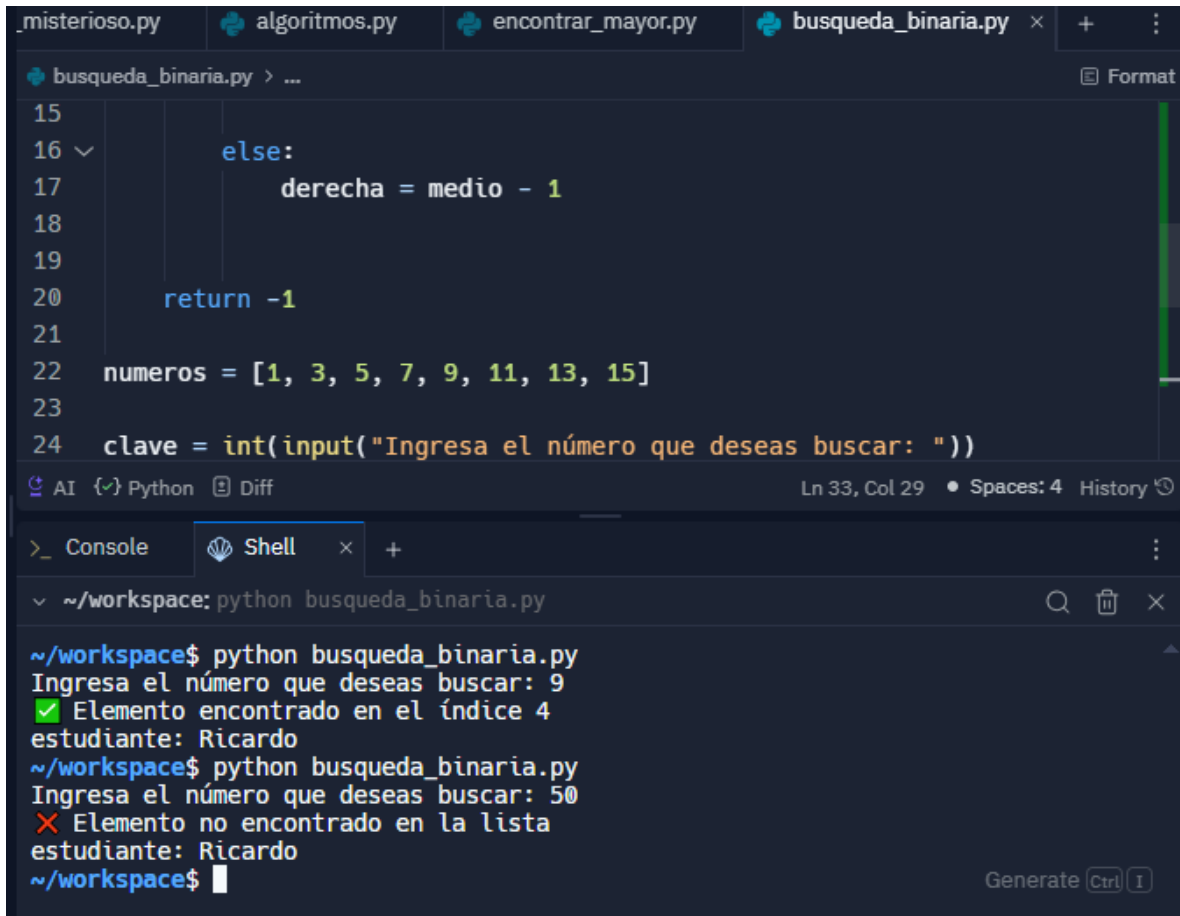
Below the code editor, there is a `Console` and a `Shell` tab. The `Shell` tab is active, showing the output of the script:

```
4. Mostrar todos los elementos (¡trampa final!)
5. Salir del juego
Elige una opción (1-5): 2
¿Qué índice quieres consultar (0 a 4)? 1
Elemento en la posición 1: 5

¿Qué quieres hacer?
1. Ver cuántos elementos tiene la lista
2. Ver un elemento específico (por índice)
3. Saber si un número de la lista es mayor a un valor
4. Mostrar todos los elementos (¡trampa final!)
5. Salir del juego
Elige una opción (1-5):
```

A watermark "Activar Windows" is visible in the background of the shell output.

BUSQUEDA BINARIA



The image shows a code editor with a dark theme. At the top, there are four tabs: `_misterioso.py`, `algoritmos.py`, `encontrar_mayor.py`, and `busqueda_binaria.py`. The `busqueda_binaria.py` tab is active, showing the following Python code:

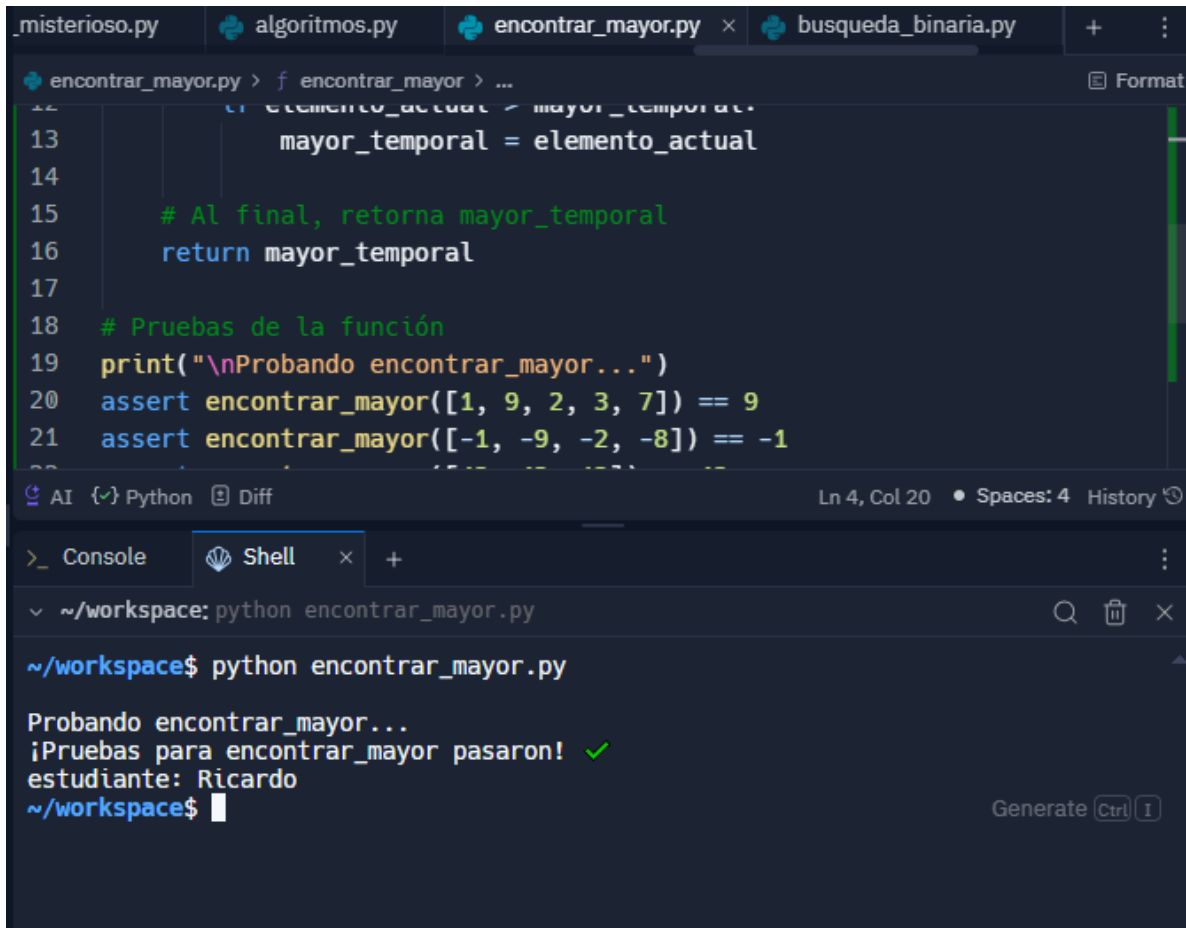
```
15
16     else:
17         derecha = medio - 1
18
19
20     return -1
21
22 numeros = [1, 3, 5, 7, 9, 11, 13, 15]
23
24 clave = int(input("Ingresa el número que deseas buscar: "))
```

Below the code editor is a terminal window with the title `~/workspace: python busqueda_binaria.py`. It shows two runs of the program:

```
~/workspace$ python busqueda_binaria.py
Ingresa el número que deseas buscar: 9
✅ Elemento encontrado en el índice 4
estudiante: Ricardo
~/workspace$ python busqueda_binaria.py
Ingresa el número que deseas buscar: 50
❌ Elemento no encontrado en la lista
estudiante: Ricardo
~/workspace$
```

The terminal output includes a green checkmark for a successful search and a red X for an unsuccessful search. The name "Ricardo" is entered after each prompt. At the bottom right of the terminal, there is a "Generate" button with a keyboard shortcut `Ctrl I`.

ENCONTRAR EL MAYOR



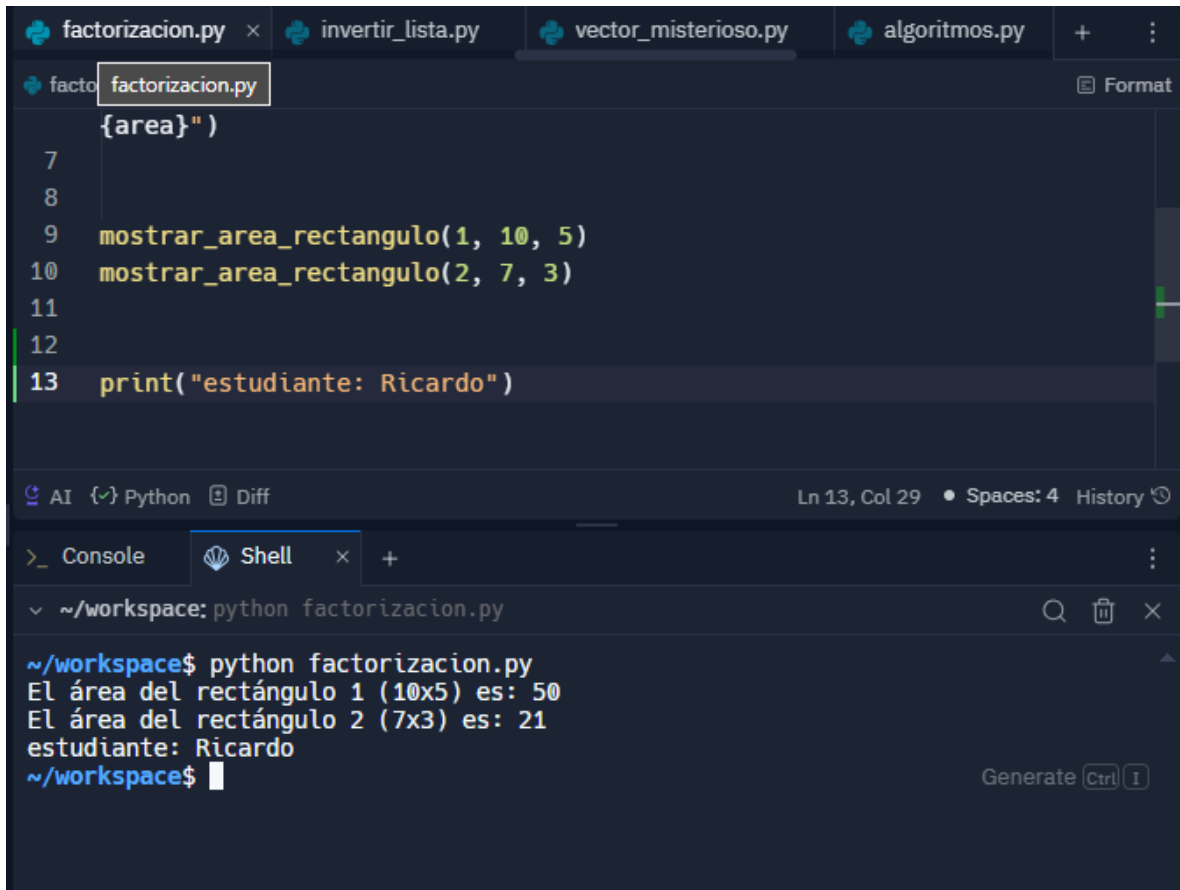
```
misterioso.py  algoritmos.py  encontrar_mayor.py  x  busqueda_binaria.py  +  ⋮
encontrar_mayor.py  >  f  encontrar_mayor  >  ...  Format
12  def encontrar_mayor(elemento_actual, mayor_temporal):
13      mayor_temporal = elemento_actual
14
15      # Al final, retorna mayor_temporal
16      return mayor_temporal
17
18  # Pruebas de la función
19  print("\nProbando encontrar_mayor...")
20  assert encontrar_mayor([1, 9, 2, 3, 7]) == 9
21  assert encontrar_mayor([-1, -9, -2, -8]) == -1
22
23  AI  {✓} Python  Diff  Ln 4, Col 20  • Spaces: 4  History ⌵

>_ Console  Shell  x  +  ⋮
~/workspace: python encontrar_mayor.py

~/workspace$ python encontrar_mayor.py

Probando encontrar_mayor...
¡Pruebas para encontrar_mayor pasaron! ✓
estudiante: Ricardo
~/workspace$  Generate Ctrl I
```

Factorizacion



The screenshot shows a code editor with four tabs: `factorizacion.py`, `invertir_lista.py`, `vector_misterioso.py`, and `algoritmos.py`. The `factorizacion.py` tab is active, displaying the following Python code:

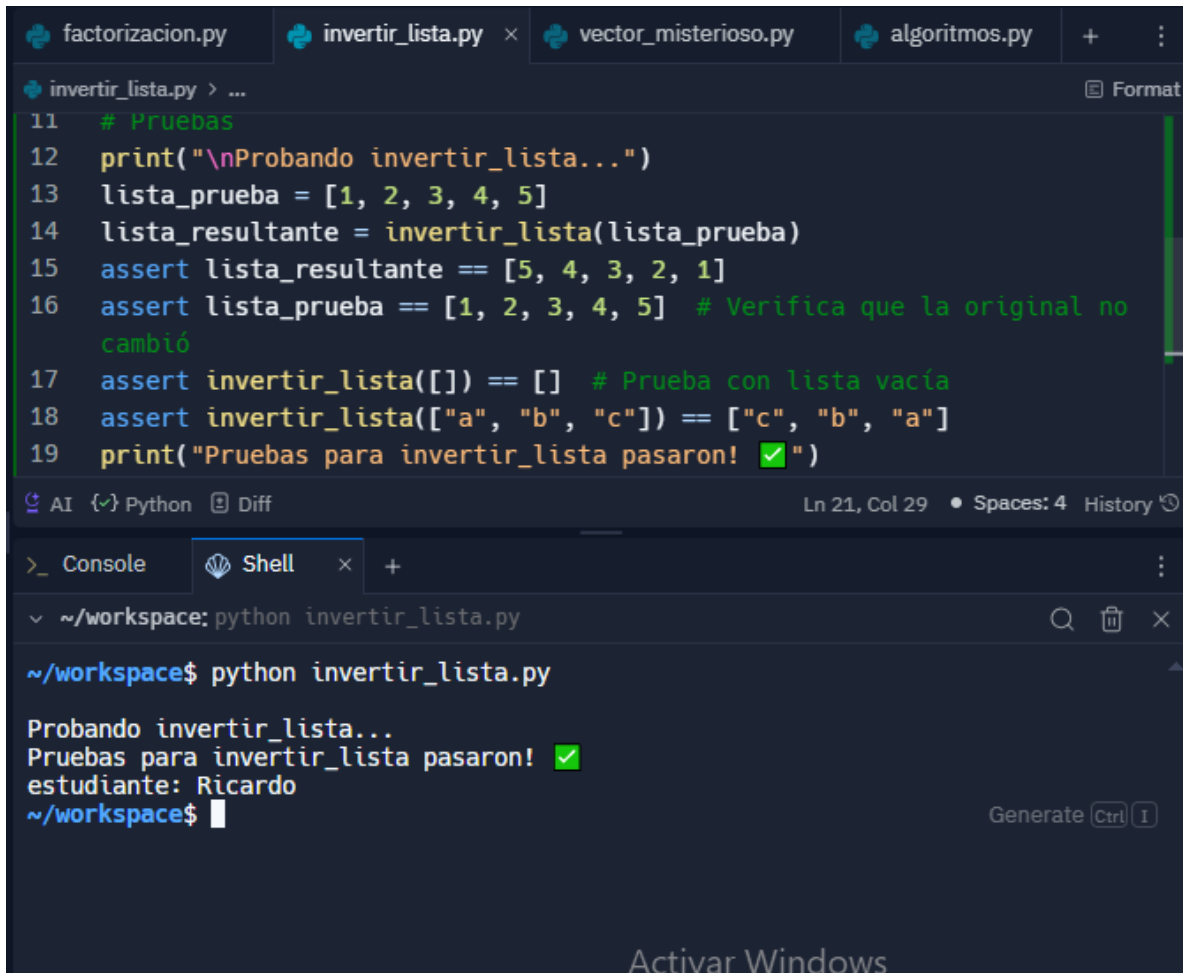
```
{area}")  
7  
8  
9 mostrar_area_rectangulo(1, 10, 5)  
10 mostrar_area_rectangulo(2, 7, 3)  
11  
12  
13 print("estudiante: Ricardo")
```

Below the code editor, the `Shell` tab is active, showing the execution of the script:

```
~/workspace: python factorizacion.py  
El área del rectángulo 1 (10x5) es: 50  
El área del rectángulo 2 (7x3) es: 21  
estudiante: Ricardo  
~/workspace$
```

The status bar at the bottom of the editor indicates the current position is `Ln 13, Col 29`, with `Spaces: 4` and a `History` button.

INVERTIR LISTA



The image shows a code editor with four tabs: `factorizacion.py`, `invertir_lista.py` (active), `vector_misterioso.py`, and `algoritmos.py`. The active tab displays Python code for reversing a list. Below the code editor is a terminal window showing the execution of the script.

```
11 # Pruebas
12 print("\nProbando invertir_lista...")
13 lista_prueba = [1, 2, 3, 4, 5]
14 lista_resultante = invertir_lista(lista_prueba)
15 assert lista_resultante == [5, 4, 3, 2, 1]
16 assert lista_prueba == [1, 2, 3, 4, 5] # Verifica que la original no
    cambió
17 assert invertir_lista([]) == [] # Prueba con lista vacía
18 assert invertir_lista(["a", "b", "c"]) == ["c", "b", "a"]
19 print("Pruebas para invertir_lista pasaron! ✅")
```

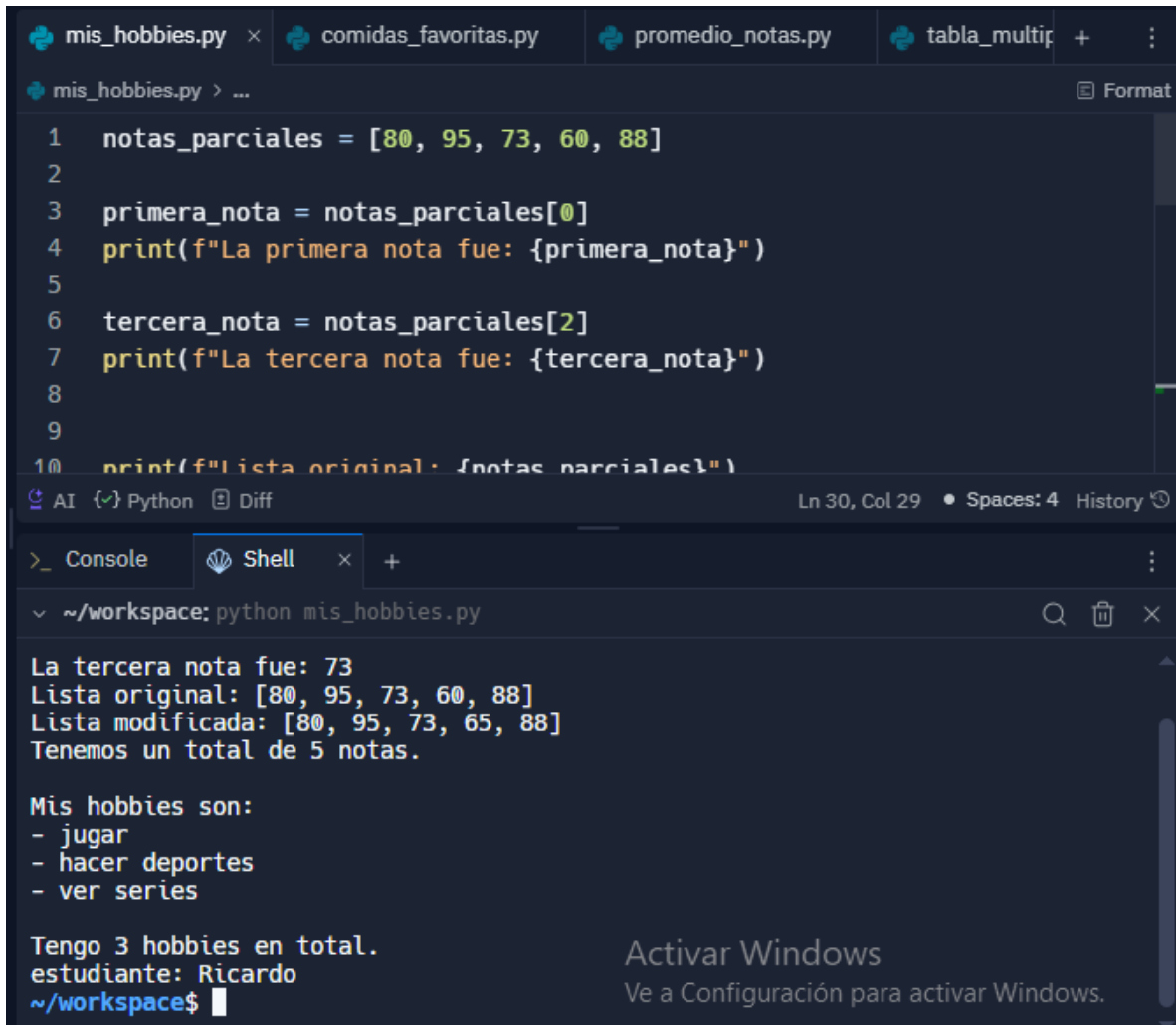
Terminal output:

```
~/workspace$ python invertir_lista.py

Probando invertir_lista...
Pruebas para invertir_lista pasaron! ✅
estudiante: Ricardo
~/workspace$
```

At the bottom of the terminal window, there is a "Generate" button with a "Ctrl I" shortcut and a watermark that says "Activar Windows".

PROMEDIOS Y HOBBIES



The image shows a VS Code editor window with four tabs: `mis_hobbies.py`, `comidas_favoritas.py`, `promedio_notas.py`, and `tabla_multi...`. The `mis_hobbies.py` tab is active, displaying the following Python code:

```
1 notas_parciales = [80, 95, 73, 60, 88]
2
3 primera_notas = notas_parciales[0]
4 print(f"La primera nota fue: {primera_notas}")
5
6 tercera_notas = notas_parciales[2]
7 print(f"La tercera nota fue: {tercera_notas}")
8
9
10 print(f"Lista original: {notas_parciales}")
```

Below the code editor is a terminal window titled "Shell" with the command `python mis_hobbies.py` executed. The output of the script is as follows:

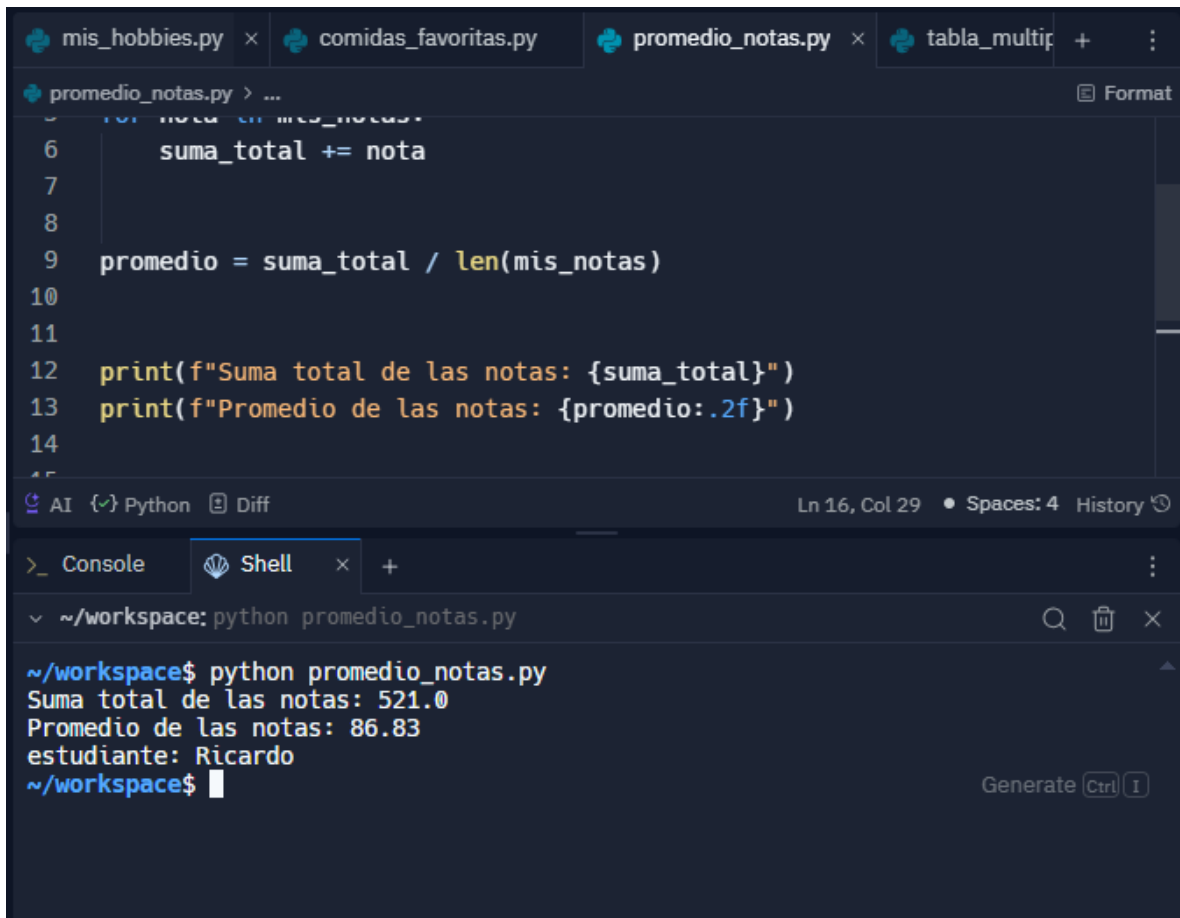
```
La tercera nota fue: 73
Lista original: [80, 95, 73, 60, 88]
Lista modificada: [80, 95, 73, 65, 88]
Tenemos un total de 5 notas.

Mis hobbies son:
- jugar
- hacer deportes
- ver series

Tengo 3 hobbies en total.
estudiante: Ricardo
~/workspace$
```

An "Activar Windows" watermark is visible in the bottom right corner of the terminal area.

PROMEDIO DE NOTAS



The image shows a code editor with four tabs: `mis_hobbies.py`, `comidas_favoritas.py`, `promedio_notas.py`, and `tabla_multiplicar.py`. The `promedio_notas.py` tab is active, displaying the following Python code:

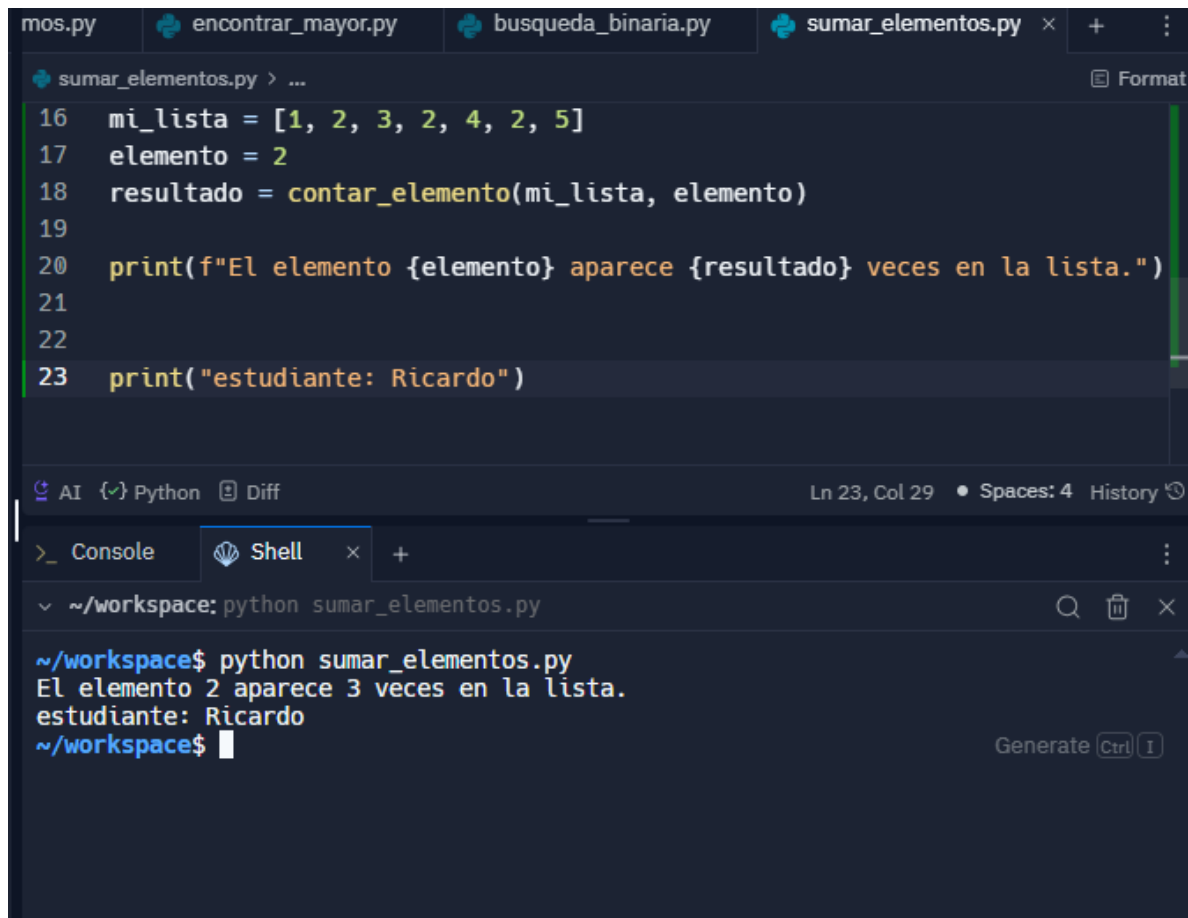
```
1 for nota in mis_notas:
2     suma_total += nota
3
4
5
6     suma_total += nota
7
8
9 promedio = suma_total / len(mis_notas)
10
11
12 print(f"Suma total de las notas: {suma_total}")
13 print(f"Promedio de las notas: {promedio:.2f}")
14
```

Below the code editor is a terminal window with the following output:

```
~/workspace$ python promedio_notas.py
Suma total de las notas: 521.0
Promedio de las notas: 86.83
estudiante: Ricardo
~/workspace$
```

The terminal output shows the sum of grades as 521.0 and the average as 86.83. The student's name, Ricardo, is also displayed. The terminal prompt is `~/workspace$`.

SUMA DE ELEMENTOS



The screenshot shows a code editor with four tabs: `mos.py`, `encontrar_mayor.py`, `busqueda_binaria.py`, and `sumar_elementos.py`. The `sumar_elementos.py` tab is active, displaying the following Python code:

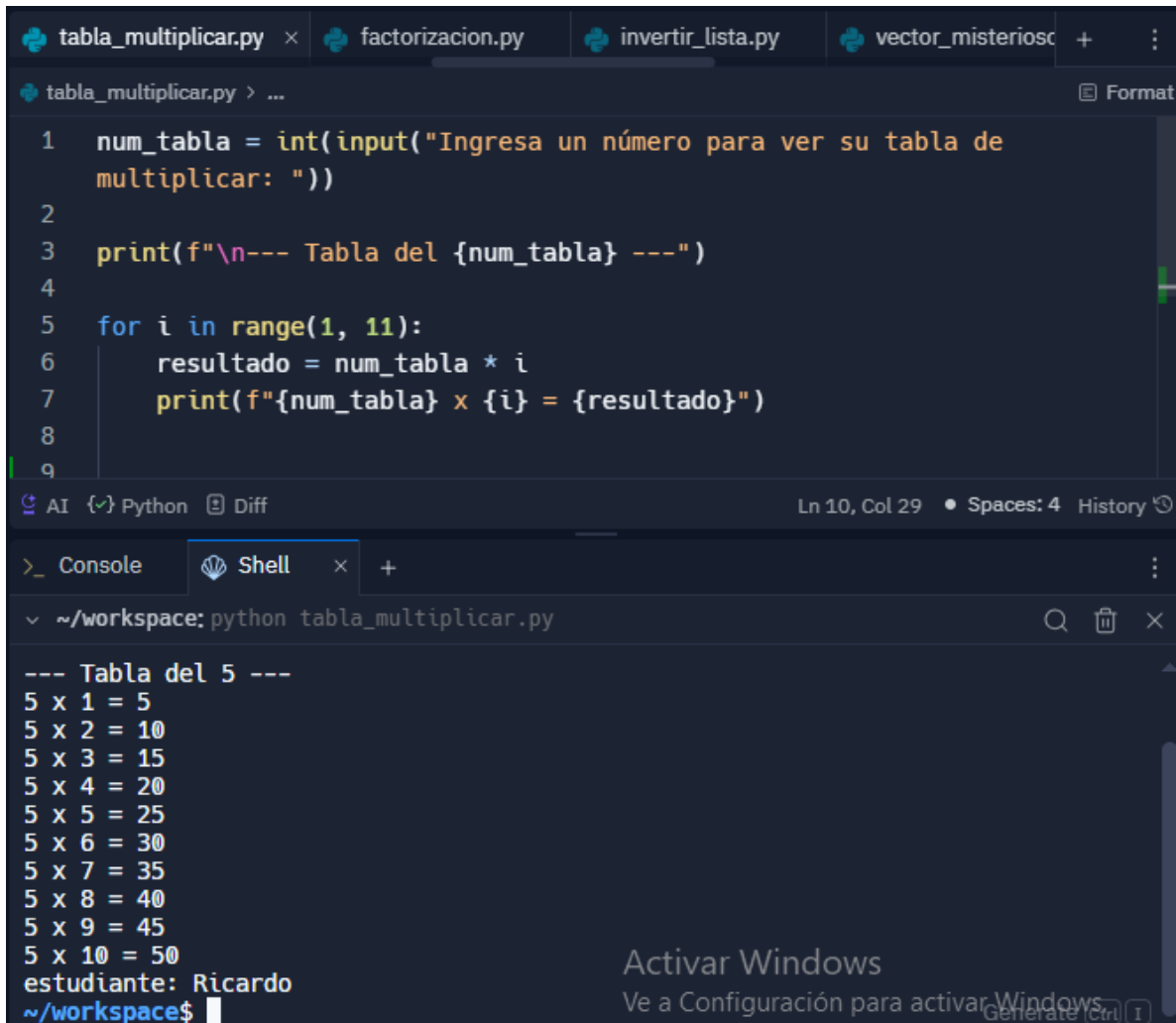
```
16 mi_lista = [1, 2, 3, 2, 4, 2, 5]
17 elemento = 2
18 resultado = contar_elemento(mi_lista, elemento)
19
20 print(f"El elemento {elemento} aparece {resultado} veces en la lista.")
21
22
23 print("estudiante: Ricardo")
```

Below the code editor, there is a `Shell` tab showing the execution of the script:

```
~/workspace$ python sumar_elementos.py
El elemento 2 aparece 3 veces en la lista.
estudiante: Ricardo
~/workspace$
```

The output of the script is: "El elemento 2 aparece 3 veces en la lista." followed by "estudiante: Ricardo".

MULTIPLICAR



The image shows a code editor with four tabs: `tabla_multiplicar.py`, `factorizacion.py`, `invertir_lista.py`, and `vector_misterioso.py`. The `tabla_multiplicar.py` tab is active, displaying the following Python code:

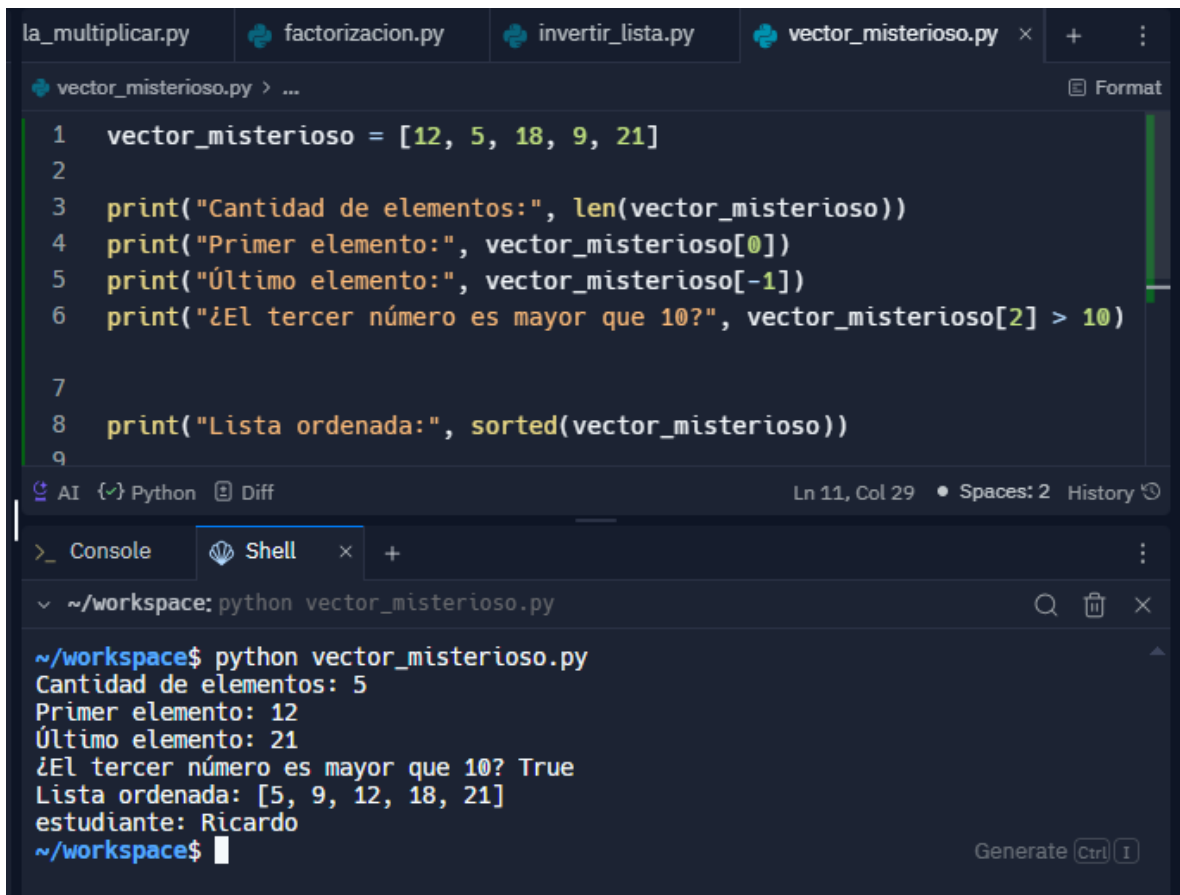
```
1 num_tabla = int(input("Ingresa un número para ver su tabla de multiplicar: "))
2
3 print(f"\n--- Tabla del {num_tabla} ---")
4
5 for i in range(1, 11):
6     resultado = num_tabla * i
7     print(f"{num_tabla} x {i} = {resultado}")
8
9
```

Below the code editor is a terminal window titled "Shell". It shows the command `python tabla_multiplicar.py` being executed. The output of the program is:

```
--- Tabla del 5 ---
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
estudiante: Ricardo
~/workspace$
```

An "Activar Windows" watermark is visible in the bottom right corner of the terminal area.

VECTOR MISTERIOSO



The image shows a code editor with four tabs: `la_multiplicar.py`, `factorizacion.py`, `invertir_lista.py`, and `vector_misterioso.py`. The `vector_misterioso.py` tab is active, displaying the following Python code:

```
1 vector_misterioso = [12, 5, 18, 9, 21]
2
3 print("Cantidad de elementos:", len(vector_misterioso))
4 print("Primer elemento:", vector_misterioso[0])
5 print("Último elemento:", vector_misterioso[-1])
6 print("¿El tercer número es mayor que 10?", vector_misterioso[2] > 10)
7
8 print("Lista ordenada:", sorted(vector_misterioso))
9
```

Below the code editor is a terminal window with the following output:

```
~/workspace$ python vector_misterioso.py
Cantidad de elementos: 5
Primer elemento: 12
Último elemento: 21
¿El tercer número es mayor que 10? True
Lista ordenada: [5, 9, 12, 18, 21]
estudiante: Ricardo
~/workspace$
```

The terminal window also shows a "Generate" button with a "Ctrl" icon.