**Integrantes :** Harold Fernando Alvarez Montilla

Johan Amilkar Tocora Repizzo

Dylan Esteban Ricaurte Cuervo

Lilibeth Tatiana Gomez Mantilla

**TALLER 1 - ALGORITMOS**

**1. Cálculo de la media de N observaciones numéricas**

Algoritmo.

Inicio

Leer la cantidad de observaciones numéricas → cantidad

Asignar suma = 0

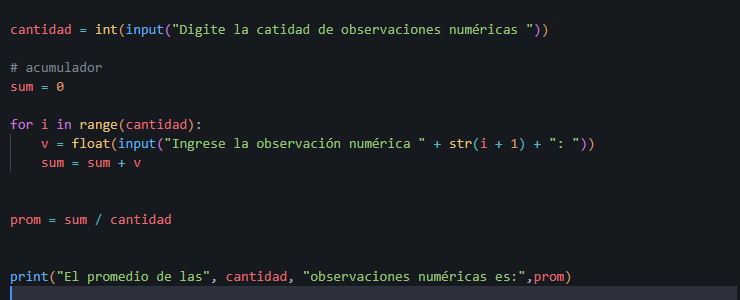
Para i desde 1 hasta cantidad hacer:  
 4.1. Leer la observación numérica = v  
 4.2. Sumar v a suma = suma = suma + v

Calcular el promedio = prom = suma / cantidad

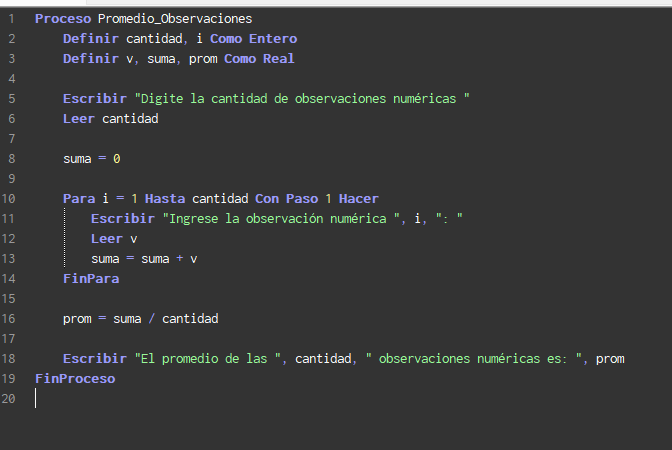
Mostrar: “El promedio de las”, cantidad, “observaciones numéricas es:”, prom

Fin

**PYTHON**

****

**PSEINT**

****

**2. Determinar si un número es primo o No**

TEXTO

Inicio

Leer un número → num

Si num ≤ 1 entonces

3.1. Mostrar: “El número no es primo”

Si no (es decir, si num > 1), entonces:

4.1. Asignar divisores = 0

4.2. Para i desde 1 hasta num hacer:

- Si num mod i = 0 entonces

divisores = divisores + 1

4.3. Si divisores = 2 entonces

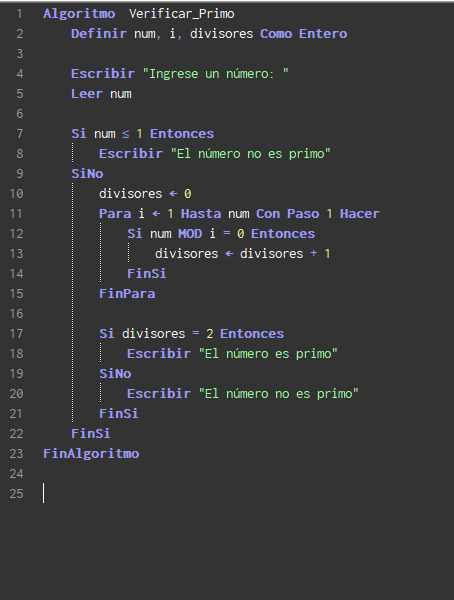
Mostrar: “El número es primo”

Si no

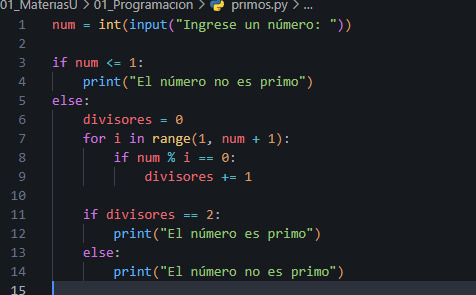
Mostrar: “El número no es primo”

Fin

**PSEINT**



**PYTHON**



**3. Cálculo del factorial de un número N**

**ALGORITMO**

Cálculo del Factorial

Datos de entrada: num (numero entero no negativo)

Datos de salida : Facto (factorial)

Procesos: facto = facto \* i

—--------------------------------------------------------------------------

1. Inicio

2. Leer el número num

3. Si num < 0 entonces

Mostrar: "No existe factorial para números negativos"

4. Si no (es decir, si num ≥ 0), entonces:

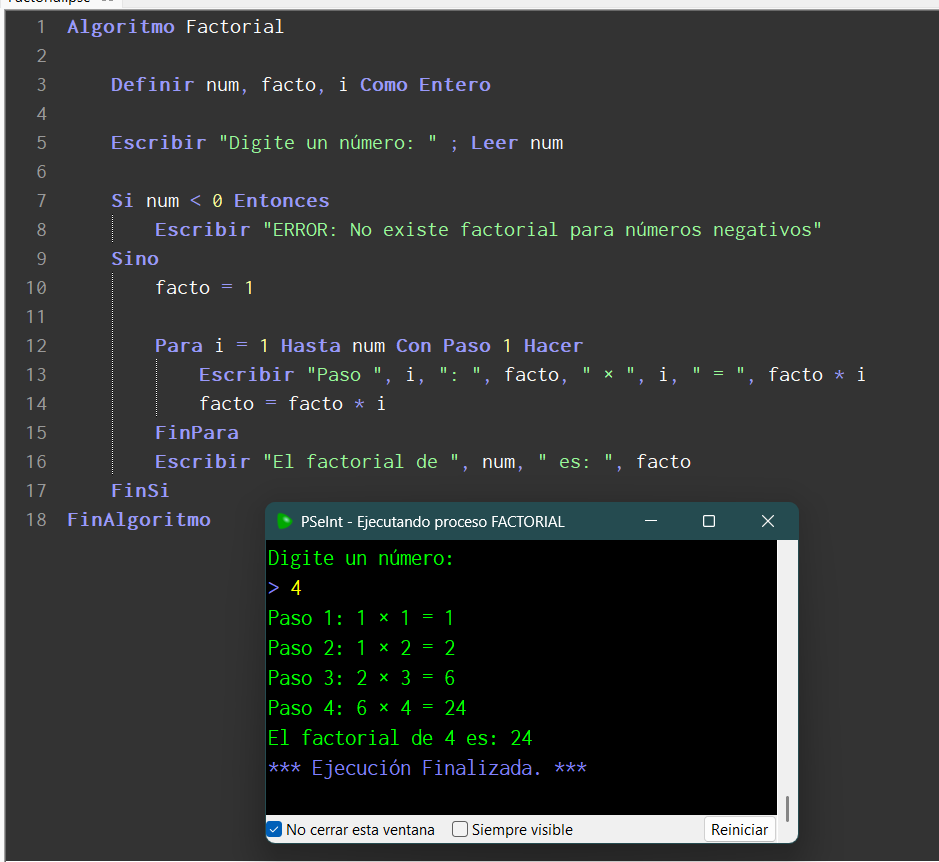
4.1. Asignar facto = 1

4.2. Para i desde 1 hasta num hacer:

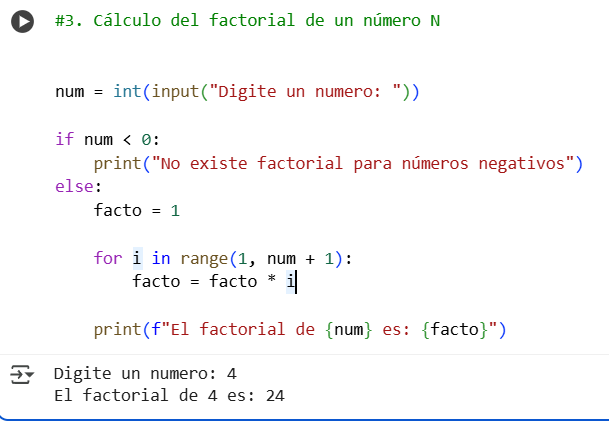
facto = facto \* i

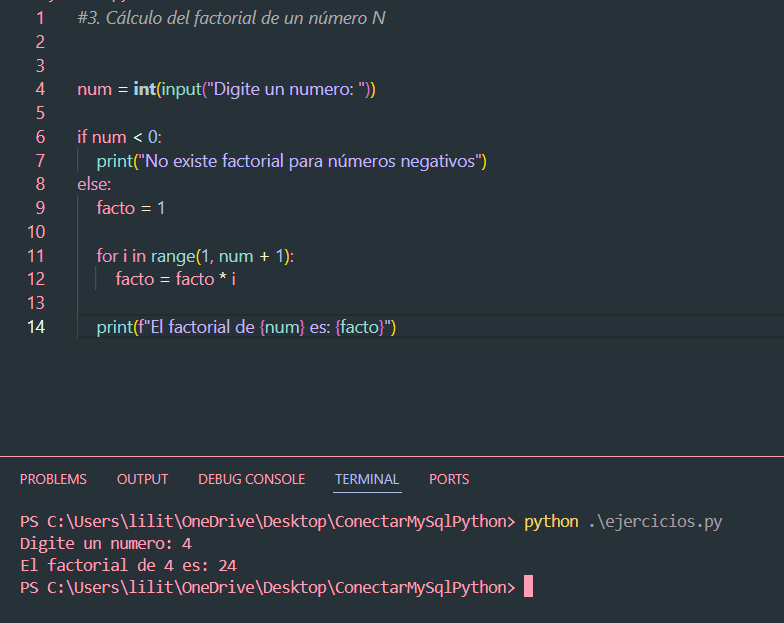
4.3. Mostrar: "El factorial de", num, "es:", facto

5. Fin

**PSEINT**

**PYTHON**





**4. Retiro de efectivo en un cajero automático**

**ALGORITMO**

**Retiro de cajero automático**

Datos de entrada: r (monto a retirar mayor a 0) s = saldo disponible

Datos de salida : s (saldo final actualizado)

Mensaje de confirmación o advertencia

“Fondos insuficientes”  
 “Retiro $x — Saldo restante $y”

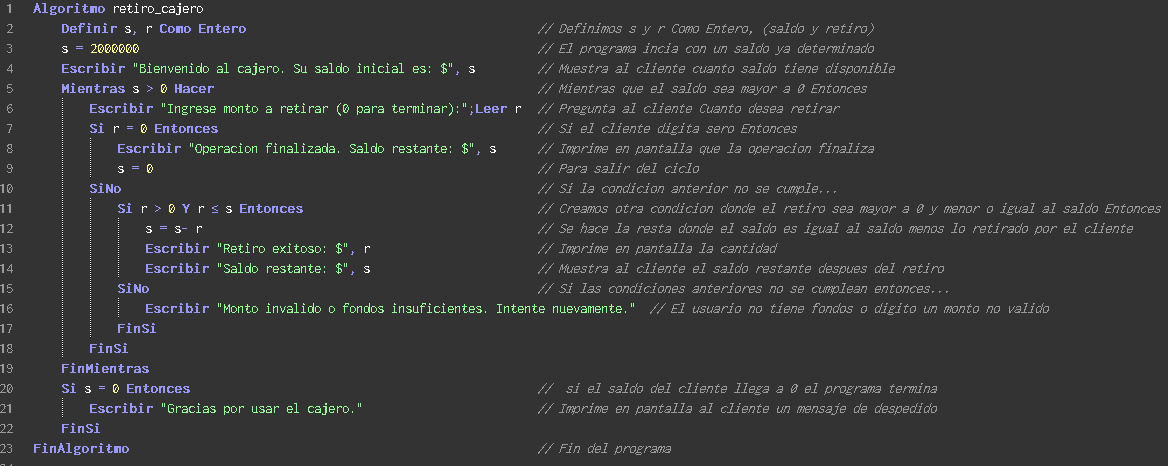
Procesos: s = s-r (saldo actualizado) r=0(sale del programa)

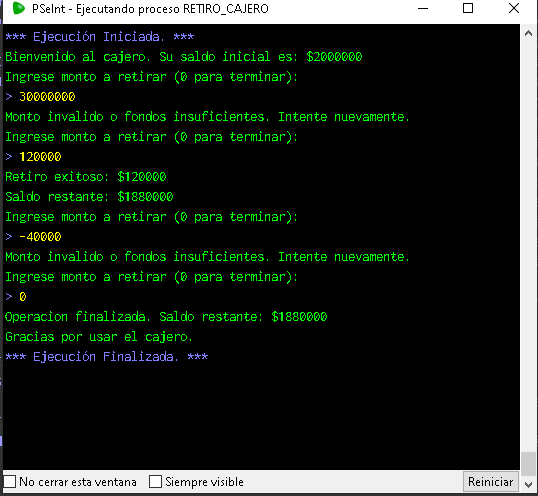
Si r es mayor a s = fondos insuficientes

Si r es menor a 0 = monto no valido

1. suficientes”  
     De lo contrario  
      7.2. Mostrar “Retiro $”, r  
      7.3. Calcular y mostrar “Saldo restante $”, s - r
2. Fin

**PSEINT**

****

****

**SCRIPT**

Algoritmo retiro\_cajero

Definir s, r Como Entero

s = 2000000

Escribir "Bienvenido al cajero. Su saldo inicial es: $", s

Mientras s > 0 Hacer

Escribir "Ingrese monto a retirar (0 para terminar):";Leer r

Si r = 0 Entonces

Escribir "Operacion finalizada. Saldo restante: $", s

s = 0

SiNo

Si r > 0 Y r <= s Entonces

s = s- r

Escribir "Retiro exitoso: $", r

Escribir "Saldo restante: $", s

SiNo

Escribir "Monto invalido o fondos insuficientes. Intente nuevamente."

FinSi

FinSi

FinMientras

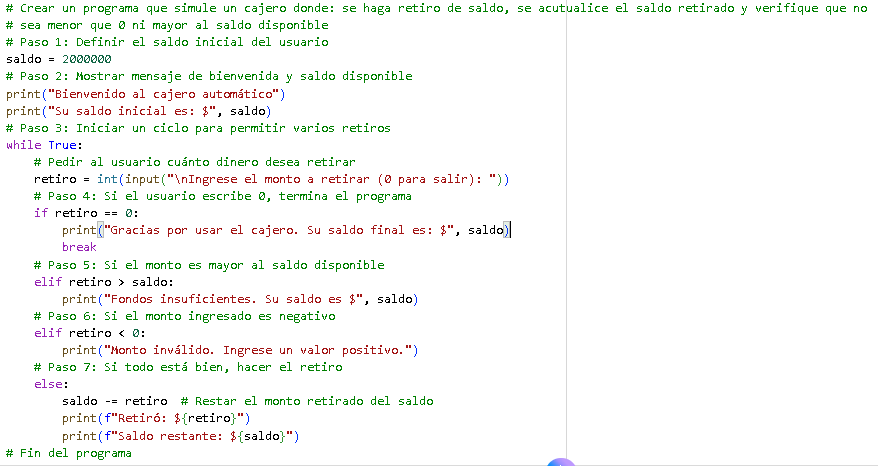
Si s = 0 Entonces

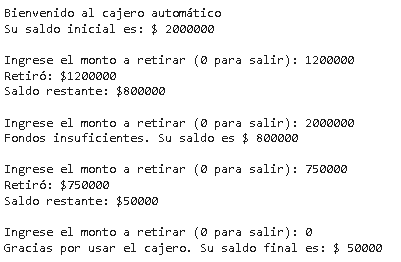
Escribir "Gracias por usar el cajero."

FinSi

FinAlgoritmo

**PYTHON**

****

****

**SCRIPT**

**# Crear un programa que simule un cajero donde: se haga retiro de saldo, se actualice el saldo retirado y verifique que no**

**# sea menor que 0 ni mayor al saldo disponible**

**# Paso 1: Definir el saldo inicial del usuario**

**saldo = 2000000**

**# Paso 2: Mostrar mensaje de bienvenida y saldo disponible**

**print("Bienvenido al cajero automático")**

**print("Su saldo inicial es: $", saldo)**

**# Paso 3: Iniciar un ciclo para permitir varios retiros**

**while True:**

**# Pedir al usuario cuánto dinero desea retirar**

**retiro = int(input("\nIngrese el monto a retirar (0 para salir): "))**

**# Paso 4: Si el usuario escribe 0, termina el programa**

**if retiro == 0:**

**print("Gracias por usar el cajero. Su saldo final es: $", saldo)**

**break**

**# Paso 5: Si el monto es mayor al saldo disponible**

**elif retiro > saldo:**

**print("Fondos insuficientes. Su saldo es $", saldo)**

**# Paso 6: Si el monto ingresado es negativo**

**elif retiro < 0:**

**print("Monto inválido. Ingrese un valor positivo.")**

**# Paso 7: Si todo está bien, hacer el retiro**

**else:**

**saldo -= retiro # Restar el monto retirado del saldo**

**print(f"Retiró: ${retiro}")**

**print(f"Saldo restante: ${saldo}")**

**# Fin del programa**

**5. Receta para preparar "spaghetti alla napoletana"**

**ALGORITMO**

Datos de entrada : Ingre (Ingredientes ) , Uten (utensilios)

Datos de salida : Spaghetti alla Napoletana listos para servir

Procesos : asignación, secuencia, iteración, entrada/salida

1. INICIO

2. PREPARAR INGREDIENTES:

- 400g spaghetti

- 500g tomates maduros

- 2 dientes de ajo

- 1 cebolla mediana

- Albahaca fresca

- Aceite de oliva

- Sal y pimienta

- Queso parmesano (opcional)

3. PREPARAR UTENSILIOS:

- Olla grande

- Sartén grande

- Cuchillo

- Tabla de cortar

- Colador

4. PREPARAR SALSA:

4.1. Picar finamente la cebolla y el ajo

4.2. Cortar los tomates en cubos pequeños

4.3. Calentar 3 cucharadas de aceite en la sartén

4.4. Sofreír la cebolla y ajo hasta que estén dorados

4.5. Agregar los tomates picados

4.6. Cocinar a fuego lento por 20 minutos

4.7. Agregar sal, pimienta y albahaca picada

5. COCINAR PASTA:

5.1. Llenar la olla con agua y agregar sal

5.2. Hervir el agua

5.3. Agregar los spaghetti

5.4. Cocinar por 8-10 minutos (seguir instrucciones del paquete)

5.5. Escurrir la pasta

6. MEZCLAR:

6.1. Agregar los spaghetti a la salsa

6.2. Mezclar bien durante 2 minutos a fuego bajo

7. SERVIR:

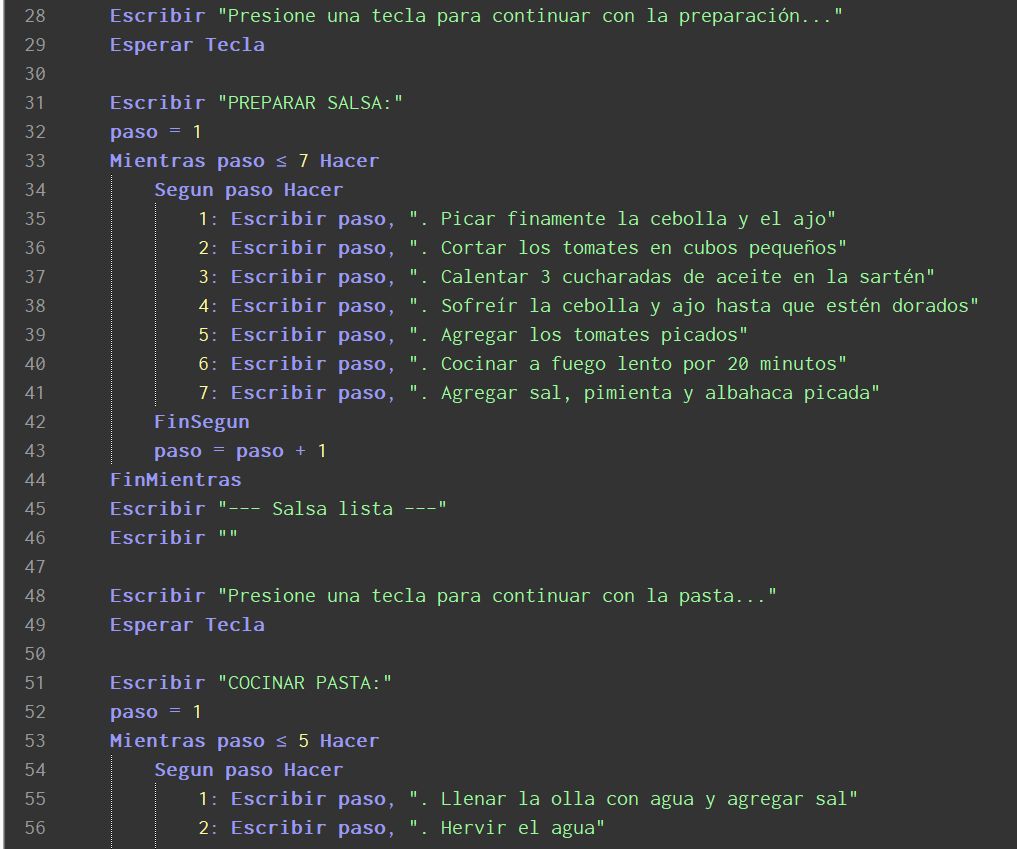
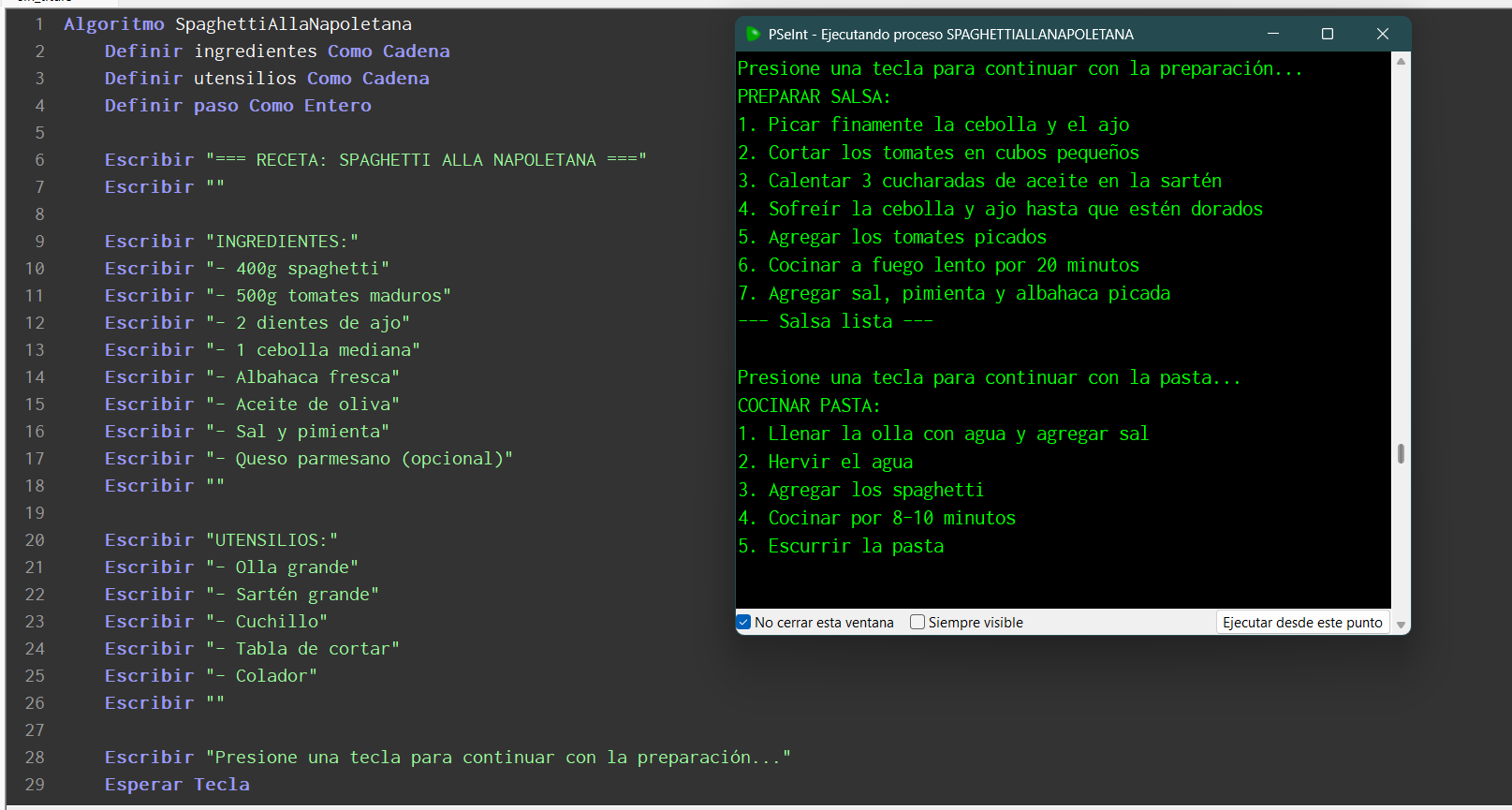
7.1. Colocar en platos

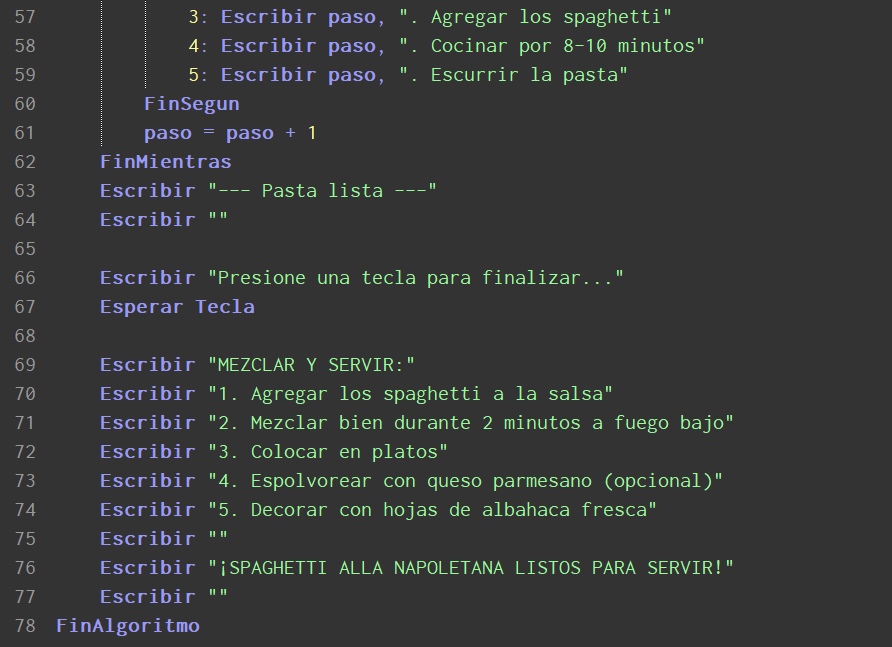
7.2. Espolvorear con queso parmesano (opcional)

7.3. Decorar con hojas de albahaca fresca

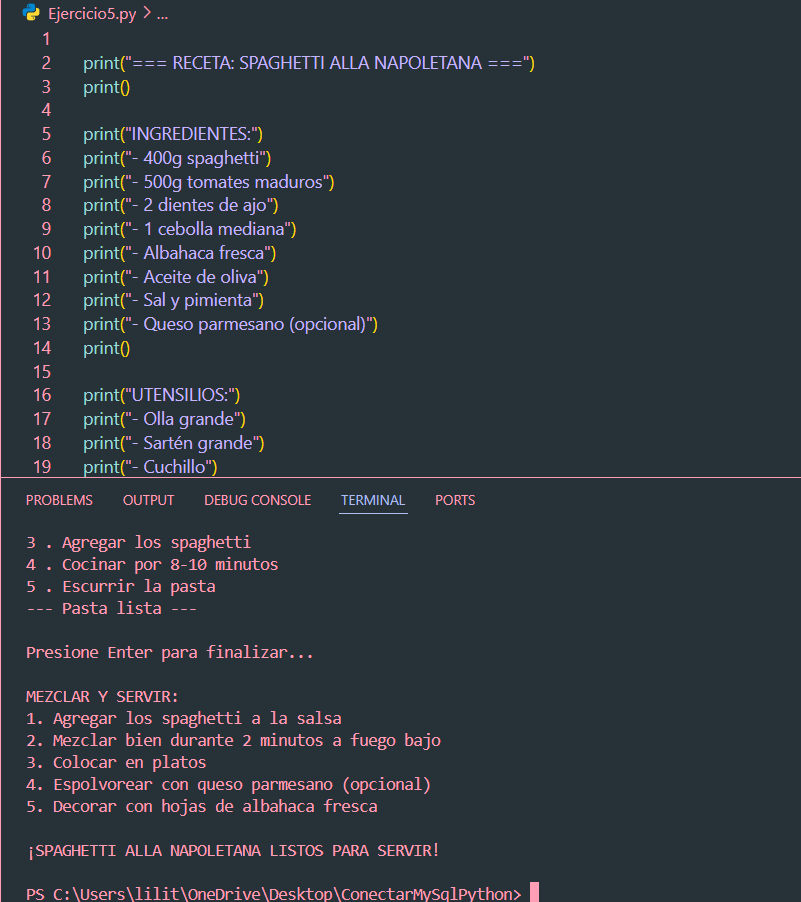
8. FIN

**PSEINT**

****

****

**PYTHON**

****