[Nombre del autor]

[Nombre de la empresa]  [Dirección de la compañía]

PRÁCTICAS DE BUCLES, IF Y MATCH CASE

Contenido

[1 · CAJERO AUTOMÁTICO SENCILLO 2](#_Toc210652505)

[2 · CONTROL DE TEMPERATURA SEMANAL 4](#_Toc210652506)

[3 · LISTA DE LA COMPRA 6](#_Toc210652507)

[4 · CALCULADORA CON MENÚ 8](#_Toc210652508)

[5 · PIRÁMIDE DE ASTERISCOS 10](#_Toc210652509)

# 1 · CAJERO AUTOMÁTICO SENCILLO

**Tipo de estructuras:** while, match case

**Descripción del ejercicio:** Crea un programa que simule un pequeño cajero automático.  
 El usuario comienza con un saldo de **100 €**.  
 El programa debe mostrar el siguiente menú:

1. Consultar saldo

2. Ingresar dinero

3. Retirar dinero

4. Salir

**Instrucciones:**

1. Usa un **bucle while** que se repita mientras la opción elegida no sea “4”.
2. Dentro del bucle, utiliza **match case** para ejecutar cada acción.
3. En la opción 2, permite ingresar dinero y suma la cantidad al saldo.
4. En la opción 3, permite retirar dinero solo si el saldo es suficiente.  
     
   

saldo = 100

menu = "1. Consultar saldo\n2. Ingresar dinero\n3. Retirar dinero\n4. Salir"

print(menu)

while True:

    opcion = int(input("Introduce la opcion que desees: "))

    ingreso = 0

    retirada = 0

    dinerototal = 0

    if opcion == 4:

        print("Hasta pronto")

        break

    match opcion:

        case 1:

            print(f"Tu saldo es: {saldo}")

        case 2:

            ingreso = float(input("Introduce la cantidad de dinero que quieras ingresar: "))

            if ingreso > 0:

                saldo += ingreso

                print(f"Has ingresado {ingreso}€. Tu saldo actual es: {saldo}€")

            else:

                print("Cantidad no valida")

        case 3:

            retirada = float(input("Introduce la cantidad de dinero que quieres retirar: "))

            if 0 < retirada <= saldo :

                saldo -= retirada

                print(f"Has retirado {retirada}€. Tu saldo actual es: {saldo}")

            else:

                print("Cantidad no valida")

# 2 · CONTROL DE TEMPERATURA SEMANAL

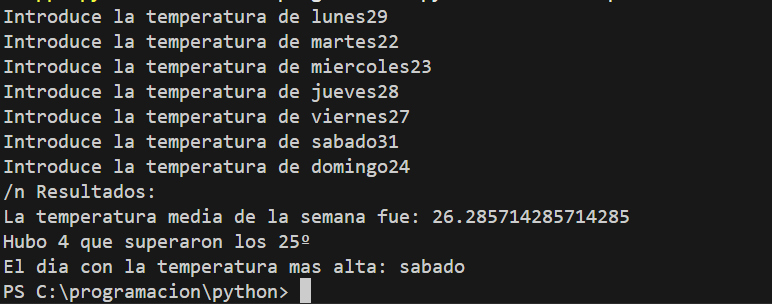
**Tipo de estructuras:** for, if

**Descripción del ejercicio:** Crea un programa que pida al usuario la temperatura media de cada día de la semana.  
 Al final, el programa mostrará:

* La temperatura media total.
* Cuántos días superaron los 25 °C.
* El día con la temperatura más alta.

**Instrucciones:**

1. Crea una lista con los nombres de los días de la semana.
2. Usa un **bucle for** para pedir una temperatura por cada día.
3. Suma todas las temperaturas para calcular la media.
4. Usa **if** para contar los días con más de 25 °C.
5. Al final, muestra todos los resultados.



semana = ['lunes', 'martes', 'miercoles', 'jueves', 'viernes', 'sabado', 'domingo']

temperaturas = []

suma = 0

dias\_25 = 0

dia\_max = ""

temperatura\_max = 0

media = 0

for dia in semana:

    temp = float(input(f"Introduce la temperatura de {dia}"))

    temperaturas.append(temp)

    suma += temp

    if temp >25:

        dias\_25 += 1

    if temp > temperatura\_max:

        temperatura\_max = temp

        dia\_max = dia

media = suma / len(semana)

print("/n Resultados:")

print(f"La temperatura media de la semana fue: {media}")

print(f"Hubo {dias\_25} que superaron los 25º")

print(f"El dia con la temperatura mas alta: {dia\_max}")

# 3 · LISTA DE LA COMPRA

**Tipo de estructuras:** while, if

**Descripción del ejercicio:** Crea un programa que permita al usuario crear su lista de la compra.  
 El usuario escribirá los productos uno a uno.  
 El programa debe evitar añadir productos repetidos.  
 Cuando el usuario escriba “fin”, se mostrará la lista completa.

**Instrucciones:**

1. Crea una lista vacía para almacenar los productos.
2. Pide al usuario un producto.
3. Mientras el producto no sea “fin”:  
   * Si el producto **no está** en la lista, añádelo.
   * Si el producto **ya está**, muestra un aviso.
   * Pide otro producto.
4. Al terminar, muestra la lista completa en pantalla.

# 4 · CALCULADORA CON MENÚ

**Tipo de estructuras:** while, match case

**Descripción del ejercicio:** Crea una calculadora que permita realizar operaciones básicas entre dos números.  
 El menú debe incluir:

1. Sumar

2. Restar

3. Multiplicar

4. Dividir

5. Salir

**Instrucciones:**

1. Muestra el menú al usuario y pide una opción.
2. Mientras la opción no sea “5”, pide dos números.
3. Usa **match case** para realizar la operación seleccionada.
4. Muestra el resultado y vuelve a mostrar el menú.
5. Cuando el usuario elija “5”, finaliza con un mensaje de despedida.

# 5 · PIRÁMIDE DE ASTERISCOS

**Tipo de estructuras:** for

**Descripción del ejercicio:** Crea un programa que dibuje en pantalla una pirámide de asteriscos (\*) con la altura que indique el usuario.

**Ejemplo si el usuario introduce 5:**

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

**Instrucciones:**

1. Pide al usuario un número entero que represente la altura.
2. Usa un **bucle for** para controlar las filas.
3. En cada fila, muestra el número de asteriscos correspondiente.
4. El número de asteriscos debe aumentar en cada línea.