

# **HITO INDIVIDUAL PROGRAMACIÓN**

**Felipe Muñoz Lopez**

## **Índice:**

***CUESTIÓN 1: Mostrar figuras por pantalla***

***CUESTIÓN 2: Juego de piedra papel o tijera***

***CUESTIÓN 3: Simular el funcionamiento de una cuenta bancaria***

# CUESTIÓN 1: Mostrar figuras por pantalla rectángulo o cuadrado

El código con el que he decidido hacerlo yo es el siguiente:

```
def mostrar_cuadrado(lado):
    for _ in range(lado):
        print('* ' * lado)

def mostrar_rectangulo(base, altura):
    for _ in range(altura):
        print('* ' * base)

def calcular_area_cuadrado(lado):
    return lado * lado

def calcular_perimetro_cuadrado(lado):
    return 4 * lado

def calcular_area_rectangulo(base, altura):
    return base * altura

def calcular_perimetro_rectangulo(base, altura):
    return 2 * (base + altura)

while True:
    print("Menú:")
    print("1 - Cuadrado")
    print("2 - Rectángulo")
    print("3 - Salir")

    opcion = input("Seleccione una opción: ")

    if opcion == '1':
        lado = float(input("Ingrese el lado del cuadrado: "))
        mostrar_cuadrado(int(lado))
        print(f"Área del cuadrado: {calcular_area_cuadrado(lado)}")
        print(f"Perímetro del cuadrado: {calcular_perimetro_cuadrado(lado)}")
    elif opcion == '2':
        base = float(input("Ingrese la base del rectángulo: "))
        altura = float(input("Ingrese la altura del rectángulo: "))
```

```

        mostrar_rectangulo(int(base), int(altura))
        print(f"Área del rectángulo: {calcular_area_rectangulo(base,
altura)}")

        print(f"Perímetro del rectángulo:
{calcular_perimetro_rectangulo(base, altura)}")
        elif opcion == '3':
            print("Saliendo...")
            break
        else:
            print("Opción no válida. Por favor, seleccione de nuevo.")

```

Y este es el funcionamiento que resultante:

```

PS C:\Progamacion> & C:/Users/CampusFP/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Progamacion/Mostrarfiguras.py
Menú:
1 - Cuadrado
2 - Rectángulo
3 - Salir
Seleccione una opción: 1
Ingrese el lado del cuadrado: 3
* * *
* * *
* * *
Área del cuadrado: 9.0
Perímetro del cuadrado: 12.0
Menú:
1 - Cuadrado
2 - Rectángulo
3 - Salir
Seleccione una opción: 2
Ingrese la base del rectángulo: 3
Ingrese la altura del rectángulo: 4
* * *
* * *
* * *
Área del rectángulo: 12.0
Perímetro del rectángulo: 14.0
Menú:
1 - Cuadrado
2 - Rectángulo
3 - Salir
Seleccione una opción: 3
Saliendo...
PS C:\Progamacion>

```

El pseudocódigo que he creado yo en base a este código es el siguiente:

Inicio

Función mostrar\_cuadrado(lado):

Para i desde 1 hasta lado hacer:

```
    Imprimir '*' * lado
  Fin Para
Fin Función
```

```
Función mostrar_rectangulo(base, altura):
  Para i desde 1 hasta altura hacer:
    Imprimir '*' * base
  Fin Para
Fin Función
```

```
Función calcular_area_cuadrado(lado):
  Retornar lado * lado
Fin Función
```

```
Función calcular_perimetro_cuadrado(lado):
  Retornar 4 * lado
Fin Función
```

```
Función calcular_area_rectangulo(base, altura):
  Retornar base * altura
Fin Función
```

```
Función calcular_perimetro_rectangulo(base, altura):
  Retornar 2 * (base + altura)
Fin Función
```

```
Mientras verdadero hacer:
  Imprimir "Menú:"
  Imprimir "1 - Cuadrado"
  Imprimir "2 - Rectángulo"
  Imprimir "3 - Salir"
```

```
Leer opcion
```

```
Si opcion es igual a '1' entonces:
  Leer lado
  Llamar mostrar_cuadrado(lado)
  Imprimir "Área del cuadrado: " + calcular_area_cuadrado(lado)
  Imprimir "Perímetro del cuadrado: " + calcular_perimetro_cuadrado(lado)
Sino Si opcion es igual a '2' entonces:
```

Leer base  
Leer altura  
Llamar mostrar\_rectangulo(base, altura)  
Imprimir "Área del rectángulo: " + calcular\_area\_rectangulo(base, altura)  
Imprimir "Perímetro del rectángulo: " + calcular\_perimetro\_rectangulo(base, altura)  
Sino Si opcion es igual a '3' entonces:  
    Imprimir "Saliendo..."  
    Romper el bucle  
Sino:  
    Imprimir "Opción no válida. Por favor, seleccione de nuevo."  
Fin Si  
Fin Mientras  
  
Fin

## **CUESTIÓN 2: Juego de piedra papel o tijera**

El código que he usado es el siguiente:

```
import random

def juego_piedra_papel_tijera():
    punt_usuario = 0
    punt_maquina = 0

    while punt_usuario < 3 and punt_maquina < 3:
        # Solicitar la opción del usuario
        while True:
            try:
                usuario = int(input("Elige (1-Piedra, 2-Papel, 3-Tijera): "))
                if usuario in [1, 2, 3]:
                    break
            else:
```

```

        print("Opción no válida. Intenta de nuevo.")
    except ValueError:
        print("Por favor, ingresa un número válido.")

# Generar la opción de la máquina
maquina = random.randint(1, 3)

# Mostrar las elecciones
opciones = {1: "Piedra", 2: "Papel", 3: "Tijera"}
print(f"Tú elegiste: {opciones[usuario]}")
print(f"La máquina eligió: {opciones[maquina]}")

# Determinar el resultado
if usuario == maquina:
    print(";Es un empate!")
elif (usuario == 1 and maquina == 3) or (usuario == 2 and
maquina == 1) or (usuario == 3 and maquina == 2):
    print(";Has ganado esta ronda!")
    punt_usuario += 1
else:
    print("La máquina ha ganado esta ronda.")
    punt_maquina += 1

print(f"Puntuación - Tú: {punt_usuario}, Máquina:
{punt_maquina}")
print()

# Determinar el ganador final
if punt_usuario == 3:
    print(";Felicitades! Has ganado el juego.")
else:
    print("La máquina ha ganado el juego. Mejor suerte la próxima
vez.")

# Iniciar el juego
juego_piedra_papel_tijera()

```

y este es el resultado del juego creado:

```

PS C:\Progamacion> & C:/Users/CampusFP/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Progamacion/piedrapapelTijera.py
Elige (1-Piedra, 2-Papel, 3-Tijera): 2
Tú elegiste: Papel
La máquina eligió: Papel
¡Es un empate!
Puntuación - Tú: 0, Máquina: 0

Elige (1-Piedra, 2-Papel, 3-Tijera): 3
Tú elegiste: Tijera
La máquina eligió: Papel
¡Has ganado esta ronda!
Puntuación - Tú: 1, Máquina: 0

Elige (1-Piedra, 2-Papel, 3-Tijera): 1
Tú elegiste: Piedra
La máquina eligió: Papel
La máquina ha ganado esta ronda.
Puntuación - Tú: 1, Máquina: 1

Elige (1-Piedra, 2-Papel, 3-Tijera): 2
Tú elegiste: Papel
La máquina eligió: Tijera
La máquina ha ganado esta ronda.
Puntuación - Tú: 1, Máquina: 2

Elige (1-Piedra, 2-Papel, 3-Tijera): 2
Tú elegiste: Papel
La máquina eligió: Papel
¡Es un empate!
Puntuación - Tú: 1, Máquina: 2

Elige (1-Piedra, 2-Papel, 3-Tijera): 2
Tú elegiste: Papel
La máquina eligió: Tijera
La máquina ha ganado esta ronda.
Puntuación - Tú: 1, Máquina: 3

La máquina ha ganado el juego. Mejor suerte la próxima vez.
PS C:\Progamacion> 

```

Y el correspondiente pseudocódigo:

INICIO

```

punt_usuario = 0
punt_maquina = 0

```

MIENTRAS punt\_usuario < 3 Y punt\_maquina < 3 HACER

HACER

IMPRIMIR "Elige (1-Piedra, 2-Papel, 3-Tijera): "

LEER usuario

SI usuario NO ESTÁ EN [1, 2, 3] ENTONCES

IMPRIMIR "Opción no válida. Intenta de nuevo."

FIN SI

HASTA QUE usuario esté en [1, 2, 3]

maquina = GENERAR número aleatorio entre 1 y 3

IMPRIMIR "Tú elegiste: " + opciones[usuario]

```

IMPRIMIR "La máquina eligió: " + opciones[maquina]

SI usuario == maquina ENTONCES
    IMPRIMIR "¡Es un empate!"
SINO SI (usuario == 1 Y maquina == 3) O (usuario == 2 Y maquina == 1) O
(usuario == 3 Y maquina == 2) ENTONCES
    IMPRIMIR "¡Has ganado esta ronda!"
    punt_usuario = punt_usuario + 1
SINO
    IMPRIMIR "La máquina ha ganado esta ronda."
    punt_maquina = punt_maquina + 1
FIN SI

IMPRIMIR "Puntuación - Tú: " + punt_usuario + ", Máquina: " + punt_maquina
IMPRIMIR ""

FIN MIENTRAS

SI punt_usuario == 3 ENTONCES
    IMPRIMIR "¡Felicidades! Has ganado el juego."
SINO
    IMPRIMIR "La máquina ha ganado el juego. Mejor suerte la próxima vez."
FIN SI

FIN

```

### **CUESTIÓN 3: Simular el funcionamiento de una cuenta bancaria**

El código que he realizado yo es el siguiente:

```

def main():
    # Solicitar saldo inicial
    while True:
        try:
            saldo = float(input("Ingrese el saldo inicial de la cuenta
(debe ser mayor o igual a 0): "))
            if saldo >= 0:
                break
            else:

```



```

        print("El saldo debe ser mayor o igual a 0. Intente de
nuevo.")
    except ValueError:
        print("Entrada inválida. Por favor, ingrese un número
decimal.")

ingresos = 0
retiros = 0

while True:
    print("\nMenú:")
    print("1. Ingresar dinero")
    print("2. Retirar dinero")
    print("3. Mostrar saldo")
    print("4. Estadísticas")
    print("5. Salir")

    while True:
        try:
            opcion = int(input("Seleccione una opción: "))
            if opcion in [1, 2, 3, 4, 5]:
                break
            else:
                print("Opción no válida. Intente de nuevo.")
        except ValueError:
            print("Entrada inválida. Por favor, ingrese un número
del 1 al 5.")

    if opcion == 1:
        while True:
            try:
                cantidad = float(input("Ingrese la cantidad a
ingresar (no puede ser negativa): "))
                if cantidad >= 0:
                    saldo += cantidad
                    ingresos += 1
                    print(f"Se han ingresado {cantidad}. Nuevo
saldo: {saldo}")
                    break
                else:
                    print("La cantidad debe ser mayor o igual a 0.
Intente de nuevo.")
            except ValueError:

```

```

        print("Entrada inválida. Por favor, ingrese un
número decimal.")

    elif opcion == 2:
        while True:
            try:
                cantidad = float(input("Ingrese la cantidad a
retirar (no puede ser negativa): "))
                if cantidad < 0:
                    print("La cantidad debe ser mayor o igual a 0.
Intente de nuevo.")
                elif cantidad > saldo:
                    print("No puede retirar más de lo que tiene en
la cuenta. Intente de nuevo.")
                else:
                    saldo -= cantidad
                    retiros += 1
                    print(f"Se han retirado {cantidad}. Nuevo
saldo: {saldo}")
                    break
            except ValueError:
                print("Entrada inválida. Por favor, ingrese un
número decimal.")

    elif opcion == 3:
        print(f"Saldo actual: {saldo}")

    elif opcion == 4:
        print(f"Ingresos realizados: {ingresos}")
        print(f"Retiros realizados: {retiros}")

    elif opcion == 5:
        print("Saliendo del programa.")
        break

if __name__ == "__main__":
    main()

```

Su funcionamiento:

```

PS C:\Progamacion> & C:/Users/CampusFP/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Progamacion/funcionamiento de una cuenta bancaria.py"
Ingrese el saldo inicial de la cuenta (debe ser mayor o igual a 0): 20

Menú:
1. Ingresar dinero
2. Retirar dinero
3. Mostrar saldo
4. Estadísticas
5. Salir
Seleccione una opción: 3
Saldo actual: 20.0

Menú:
1. Ingresar dinero
2. Retirar dinero
3. Mostrar saldo
4. Estadísticas
5. Salir
Seleccione una opción: 1
Ingrese la cantidad a ingresar (no puede ser negativa): 37'80
Entrada inválida. Por favor, ingrese un número decimal.
Ingrese la cantidad a ingresar (no puede ser negativa): 37
Se han ingresado 37.0. Nuevo saldo: 57.0

Menú:
1. Ingresar dinero
2. Retirar dinero
3. Mostrar saldo
4. Estadísticas
5. Salir
Seleccione una opción: 4
Ingresos realizados: 1
Retiros realizados: 0

Menú:
1. Ingresar dinero
2. Retirar dinero
3. Mostrar saldo
4. Estadísticas
5. Salir
Seleccione una opción: 5
5. Salir
Seleccione una opción: 5
Saliendo del programa.
PS C:\Progamacion>

```

y su pseudocódigo:

INICIO

Solicitar saldo inicial

HACER

PEDIR "Ingrese el saldo inicial de la cuenta (debe ser mayor o igual a 0): " A  
saldo

SI saldo < 0 ENTONCES

IMPRIMIR "El saldo debe ser mayor o igual a 0. Intente de nuevo."

FIN\_SI

MIENTRAS saldo < 0

// Inicializar contadores

ingresos <- 0

retiros <- 0

Bucle principal del programa

HACER

IMPRIMIR "Menú:"

IMPRIMIR "1. Ingresar dinero"

IMPRIMIR "2. Retirar dinero"

IMPRIMIR "3. Mostrar saldo"

IMPRIMIR "4. Estadísticas"

IMPRIMIR "5. Salir"

Validar opción

HACER

PEDIR "Seleccione una opción: " A opción

MIENTRAS opción NO EN [1, 2, 3, 4, 5]

SI opción = 1 ENTONCES

// Opción 1: Ingresar dinero

HACER

PEDIR "Ingrese la cantidad a ingresar (no puede ser negativa): " A cantidad

SI cantidad < 0 ENTONCES

IMPRIMIR "La cantidad debe ser mayor o igual a 0. Intente de nuevo."

SINO

saldo <- saldo + cantidad

ingresos <- ingresos + 1

IMPRIMIR "Se han ingresado ", cantidad, ". Nuevo saldo: ", saldo

FIN\_SI

MIENTRAS cantidad < 0

SINO SI opción = 2 ENTONCES

Opción 2: Retirar dinero

HACER

PEDIR "Ingrese la cantidad a retirar (no puede ser negativa): " A cantidad

SI cantidad < 0 ENTONCES

IMPRIMIR "La cantidad debe ser mayor o igual a 0. Intente de nuevo."

SINO SI cantidad > saldo ENTONCES

IMPRIMIR "No puede retirar más de lo que tiene en la cuenta. Intente de nuevo."

SINO

saldo <- saldo - cantidad

retiros <- retiros + 1

IMPRIMIR "Se han retirado ", cantidad, ". Nuevo saldo: ", saldo

```
    FIN_SI
    MIENTRAS cantidad < 0 O cantidad > saldo

    SINO SI opcion = 3 ENTONCES
        Opción 3: Mostrar saldo
        IMPRIMIR "Saldo actual: ", saldo

    SINO SI opcion = 4 ENTONCES
        Opción 4: Estadísticas
        IMPRIMIR "Ingresos realizados: ", ingresos
        IMPRIMIR "Retiros realizados: ", retiros

    SINO SI opcion = 5 ENTONCES
        Opción 5: Salir
        IMPRIMIR "Saliendo del programa."
        SALIR
    FIN_SI
    MIENTRAS VERDADERO

FIN
```