



**UNIVERSIDAD INTERNACIONAL DE
DESARROLLO**

LOGICA DE PROGRAMACIÓN

NOMBRE: ROJAS ESCOBAR JAIME SEBASTIAN

TAREA: TRABAJO AUTÓNOMO 2

FECHA DE ENTREGA: 15 DE JUNIO DEL 2025

PASO 1: Inicio del Desarrollo de Software. / Configuración del entorno.

1. Configuración de GitHub

Se ha configurado correctamente la cuenta de Github para gestionar el desarrollo del código. El repositorio del proyecto está disponible:

<https://github.com/SebasR2001/PUNTO-EXTRA.git>

2. DIAGRAMA DE FLUJO

Adjunto el diagrama de flujo general del software desarrollado, el cual visualiza las principales funcionalidades y estructuras del programa.

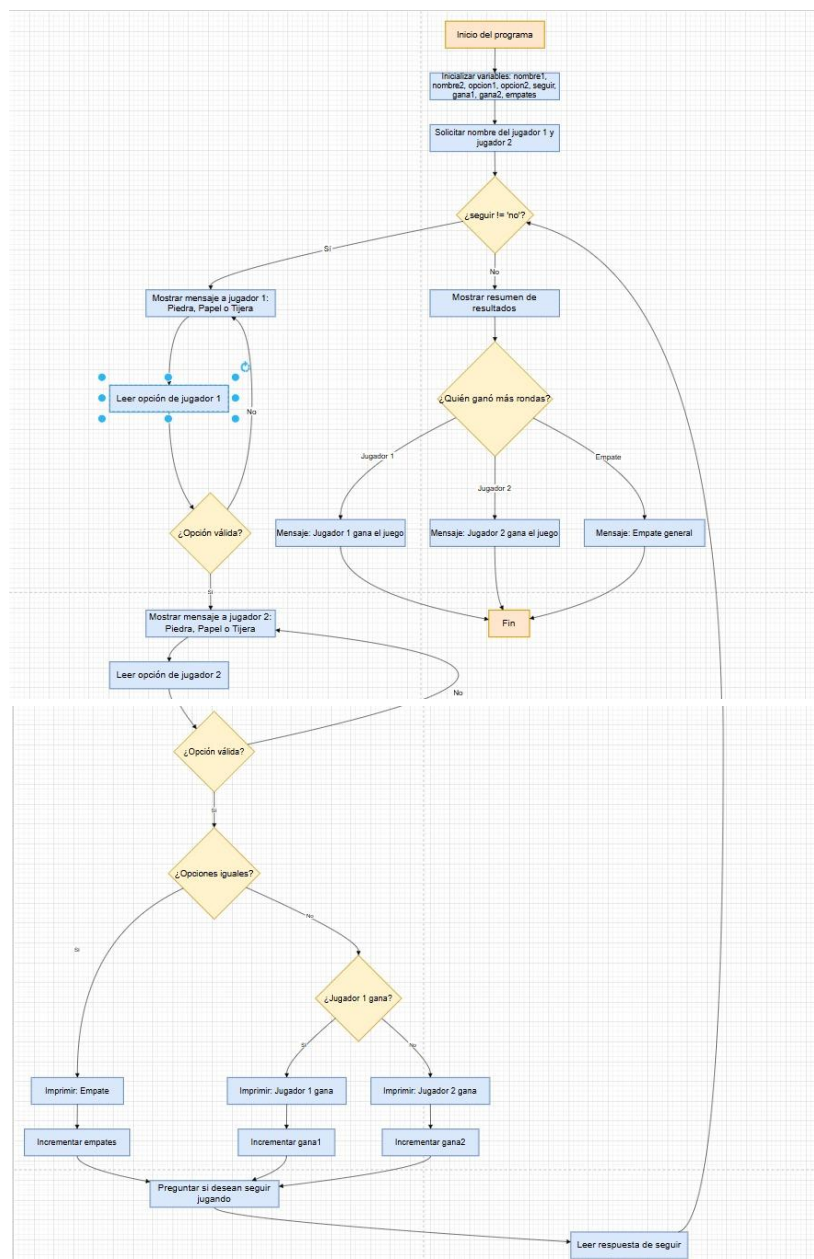


Figura 1. Diagrama de flujo

3. PREPARACION DEL AMBIENTE DE DESARROLLO

Para iniciar el desarrollo, se utilizó Visual Studio Code, donde se instaló la extensión del lenguaje de Python. Se configuró meticulosamente el compilador de la aplicación y se procedió con la clonación del repositorio de GitHub, asegurando así un entorno de desarrollo eficiente y bien preparado para la codificación.

PASO 2: Desarrollo del programa seleccionado

1. Se deben implementar funcionalidades estructuras lógicas.

En el juego de Piedra, Papel o Tijera, las estructuras lógicas en programación, como las instrucciones condicionales (if, elif, else) en Python, permiten controlar el flujo del programa según condiciones específicas. Estas estructuras son fundamentales para evaluar las elecciones de los jugadores y determinar quién gana cada ronda y el ganador general del juego.

```
# Determinar el resultado de la ronda
if opcion1 == opcion2: # Estructura condicional: if-elif-else
    print('Empate!')
    empates += 1
elif (opcion1 == 'piedra' and opcion2 == 'tijera') or \
     (opcion1 == 'papel' and opcion2 == 'piedra') or \
     (opcion1 == 'tijera' and opcion2 == 'papel'):
    print(nombre1 + ' gana esta ronda!')
    gana1 += 1
else:
    print(nombre2 + ' gana esta ronda!')
    gana2 += 1
```

Figura 2. Estructura 1

```
# Determinar el ganador general
if gana1 > gana2: # Estructura condicional: if-elif-else
    print(nombre1 + ', ganaste, ¡felicidades!')
elif gana2 > gana1:
    print(nombre2 + ', ganaste, ¡felicidades!')
else:
    print('Hubo un empate general entre ' + nombre1 + ' y ' + nombre2)
```

Figura 3. Estructura 2

2. Agregar en la codificación manejo estructuras repetitivas.

La estructura repetitiva principal en el código proporcionado es un bucle while. Este bucle se utiliza para permitir que los jugadores jueguen múltiples rondas del juego de Piedra, Papel o Tijera hasta que decidan no seguir jugando. Aquí está el bucle while en el código:

```
# Iniciar el juego
while seguir != 'no': # Estructura repetitiva: bucle while
    # Jugador 1 elige opción
    print(nombre1 + ', elige: Piedra, Papel o Tijera')
    opcion1 = input().lower()
    while opcion1 != 'piedra' and opcion1 != 'papel' and opcion1 != 'tijera': # Estructura repetitiva: bucle while
        print('Opción inválida. Intenta de nuevo:')
        opcion1 = input().lower()

    # Jugador 2 elige opción
    print(nombre2 + ', elige: Piedra, Papel o Tijera')
    opcion2 = input().lower()
    while opcion2 != 'piedra' and opcion2 != 'papel' and opcion2 != 'tijera': # Estructura repetitiva: bucle while
        print('Opción inválida. Intenta de nuevo:')
        opcion2 = input().lower()
```

Figura 4. Estructura repetitiva - while

3. Agrega comentarios en las funcionalidades más complejas para su adecuado entendimiento.

Este proyecto implementa un juego de Piedra, Papel o Tijera en Python, utilizando comentarios para clarificar las funcionalidades más complejas del código.

```
# Definir variables para almacenar nombres, opciones y resultados del juego
nombre1 = "" # Sirve para Nombre del jugador 1
nombre2 = "" # Sirve para Nombre del jugador 2
opcion1 = "" # Opción elegida por el jugador 1 (Piedra, Papel o Tijera)
opcion2 = "" # Opción elegida por el jugador 2 (Piedra, Papel o Tijera)
seguir = "" # Variable para controlar si los jugadores desean seguir jugando
gana1 = 0 # Contador de victorias del jugador 1
gana2 = 0 # Contador de victorias del jugador 2
empates = 0 # Contador de empates

# Solicita el nombres de los jugadores
nombre1 = input("Ingrese el nombre del jugador 1: ")
nombre2 = input("Ingrese el nombre del jugador 2: ")

# Iniciar el juego y repetir mientras los jugadores desean seguir jugando
while seguir != 'no':
    # Jugador 1 elige una opción (Piedra, Papel o Tijera)
    print(nombre1 + ', elige: Piedra, Papel o Tijera')
    opcion1 = input().lower()

    # Validar la opción elegida por el jugador 1
    while opcion1 != 'piedra' and opcion1 != 'papel' and opcion1 != 'tijera':
        print('Opción inválida. Intenta de nuevo:')
        opcion1 = input().lower() # Permite al jugador 1 ingresar su elección
        # (piedra, papel o tijera) en minúsculas para comparaciones precisas.

    # Jugador 2 elige una opción (Piedra, Papel o Tijera)
    print(nombre2 + ', elige: Piedra, Papel o Tijera')
    opcion2 = input().lower()

    # Validar la opción elegida por el jugador 2
    while opcion2 != 'piedra' and opcion2 != 'papel' and opcion2 != 'tijera':
        print('Opción inválida. Intenta de nuevo:')
        opcion2 = input().lower()
```

Figura 5. Código con comentarios

```
# Determinar el resultado de la ronda
if opcion1 == opcion2:
    print('Empate!')
    empates += 1 # Incrementar el contador de empates si las opciones son iguales
elif (opcion1 == 'piedra' and opcion2 == 'tijera') or \
     (opcion1 == 'papel' and opcion2 == 'piedra') or \
     (opcion1 == 'tijera' and opcion2 == 'papel'):
    print(nombre1 + ' gana esta ronda!')
    gana1 += 1 # Incrementar el contador de victorias del jugador 1 si gana
else:
    print(nombre2 + ' gana esta ronda!')
    gana2 += 1 # Incrementar el contador de victorias del jugador 2 si gana

# Preguntar a los jugadores si desean seguir jugando
seguir = input('¿Quieren seguir jugando? (si/no)').lower()

# Mostrar resumen de resultados al finalizar el juego
print('Resumen de resultados:')
print('-----')
print(nombre1 + ' ganó ' + str(gana1) + ' veces.')
print(nombre2 + ' ganó ' + str(gana2) + ' veces.')
print('Empates: ' + str(empates))
print('-----')

# Determinar y mostrar al ganador general del juego
if gana1 > gana2:
    print(nombre1 + ', ganaste, ¡felicidades!')
elif gana2 > gana1:
    print(nombre2 + ', ganaste, ¡felicidades!')
else:
    print('Hubo un empate general entre ' + nombre1 + ' y ' + nombre2)
```

Figura 6. Código con comentarios

ENTREGABLES:**VIDEO GitHub:**

<https://drive.google.com/file/d/1bWhg1wMnPcGqcq1V6eAsg33J2YnAVWUr/view?usp=sharing>

VIDEO EXPLICATIVO DEL AVANCE:

<https://drive.google.com/file/d/1rZFp2Szm1bRI9XLI9D3IH3-RIgsorwgx/view?usp=sharing>

VIDEO DEMOSTRACIÓN DE FUNCIONALIDADES:

<https://drive.google.com/file/d/1D19nx6cqxn36K4LurwmxfYhLdaKcyZ/view?usp=sharing>

REFERENCIA BIBLIOGRAFICA

Shaffer, C. A. (2016). Python Programming: An Introduction to Computer Science (3rd ed.). Franklin, Beedle & Associates Inc.

McKinney, W. (2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython (2nd ed.). O'Reilly Media.

Sweigart, A. (2015). Automate the Boring Stuff with Python: Practical Programming for Total Beginners. No Starch Press.