

EXPLICACIÓN DEL CÓDIGO

LÓGICA DE PROGRAMACIÓN

JUEGO PIEDRA PAPEL O
TIJERA

Sebastián Rojas

Índice

- 1** DECLARACIÓN DE VARIABLES
- 2** SOLICITUD DE NOMBRES
- 3** INICIO DEL JUEGO CON CICLO WHILE
- 4** ELECCIÓN DEL JUGADOR Y VALIDAD
- 5** USO DE DICCIONARIO PARA COMPARAR
- 6** COMPARACIÓN DE ELECCIONES
- 7** PREGUNTAR SI DESEA SEGUIR
- 8** MOSTRAR RESUMEN DE RESULTADOS
- 9** DETERMINAR AL GANADOR GENERAL
- 10** MANEJO DE EXCEPCIONES

DECLARACIÓN DE VARIABLES

Estas variables se usan para almacenar los nombres de los jugadores, sus elecciones (piedra, papel o tijera), si desean seguir jugando y el conteo de resultados.

```
# Definimos variables para almacenar nombres y resultados
nombre1 = ""
nombre2 = ""
opcion1 = ""
opcion2 = ""
seguir = ""
gana1 = 0
gana2 = 0
empates = 0
```


SOLICITUD DE NOMBRES

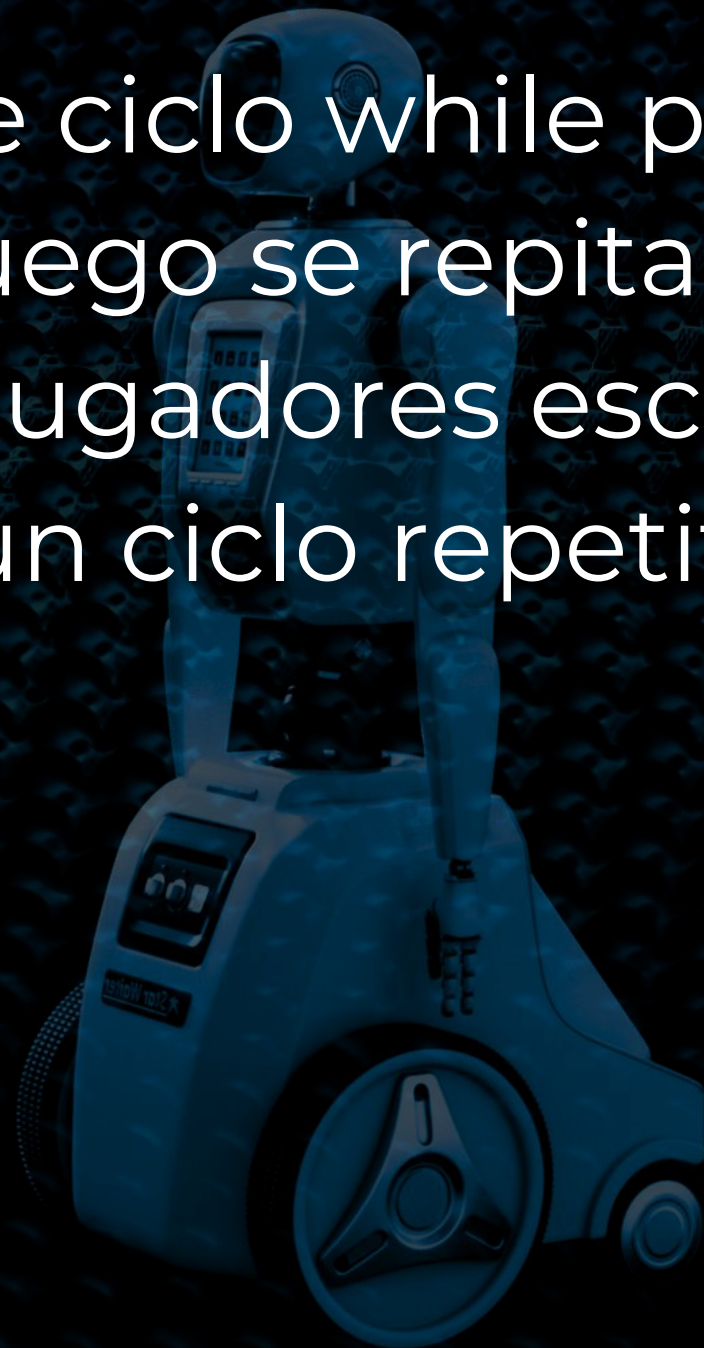
Aquí el programa pide al usuario que ingrese los nombres de los dos jugadores usando la función `input()`.

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


INICIO DEL JUEGO CON CICLO WHILE

Este ciclo while permite que el juego se repita hasta que los jugadores escriban "no". Es un ciclo repetitivo.

```
while seguir != 'no':
```





ELECCIÓN DEL JUGADOR Y VALIDACIÓN

Se pide la opción del jugador 1 y se valida que sea válida. Si no lo es, se repite hasta que lo sea (estructura doble, while dentro de otro while).

```
opcion1 = input(f"{nombre1}, elige: Piedra, Papel o Tijera: ").lower()
while opcion1 not in ['piedra', 'papel', 'tijera']:
    print("Opción inválida. Intenta de nuevo:")
    opcion1 = input().lower()
```

Elección del jugador 2 y validación:
Mismo proceso que con el jugador 1.
Se valida la opción con una estructura similar.

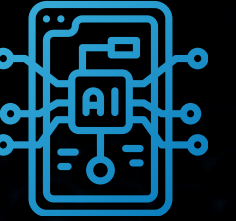
USO DE UN DICCIONARIO PARA COMPARAR



```
# Diccionario para saber qué opción gana a cuál
ganador = {
    'piedra': 'tijera',
    'tijera': 'papel',
    'papel': 'piedra'
}
```

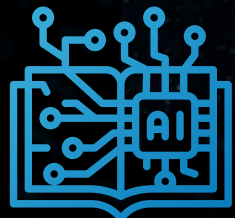
Este diccionario nos dice qué elemento gana a cuál. Por ejemplo, piedra gana a tijera.

COMPARACIÓN DE ELECCIONES(CONDICIONALES)

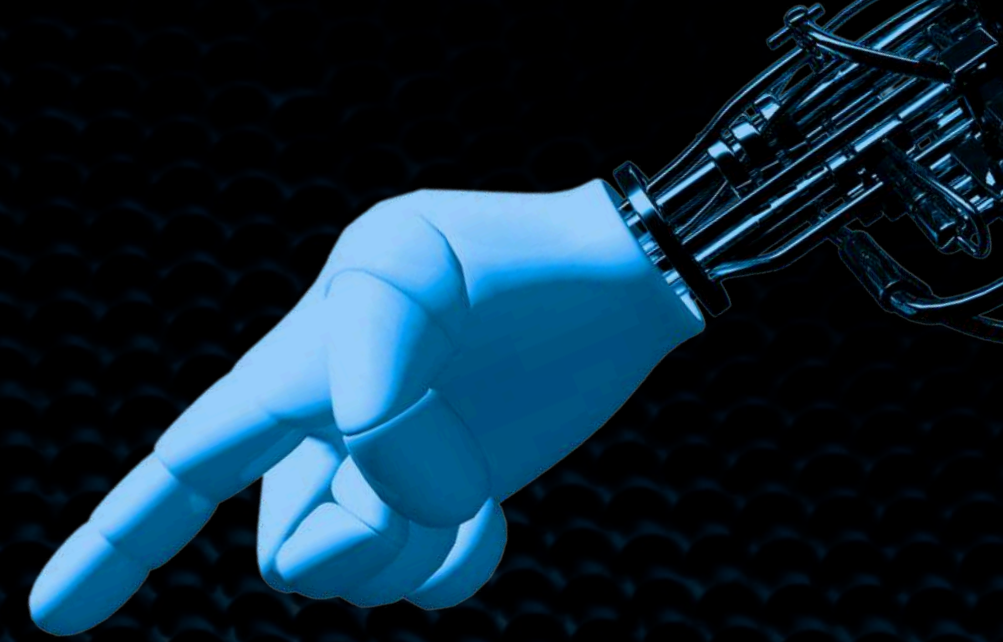


Se determina el ganador de la ronda usando if, elif, y else. Se compara usando el diccionario ganador.

```
# COMPARAR OPCIONES Y ASIGNAR RESULTADO
if opcion1 == opcion2:
    print("¡Empate!")
    empates += 1
elif ganador[opcion1] == opcion2:
    print(nombre1 + " gana esta ronda!")
    gana1 += 1
else:
    print(nombre2 + " gana esta ronda!")
    gana2 += 1
```



PREGUNTAR SI DESEAN SEGUIR



Después de cada ronda, se pregunta si desean seguir. Si escriben "no", el ciclo termina.

```
# Preguntar si desean seguir jugando  
seguir = input("¿Quieren seguir jugando? (si/no): ").lower()
```





MOSTRAR RESUMEN DE RESULTADOS

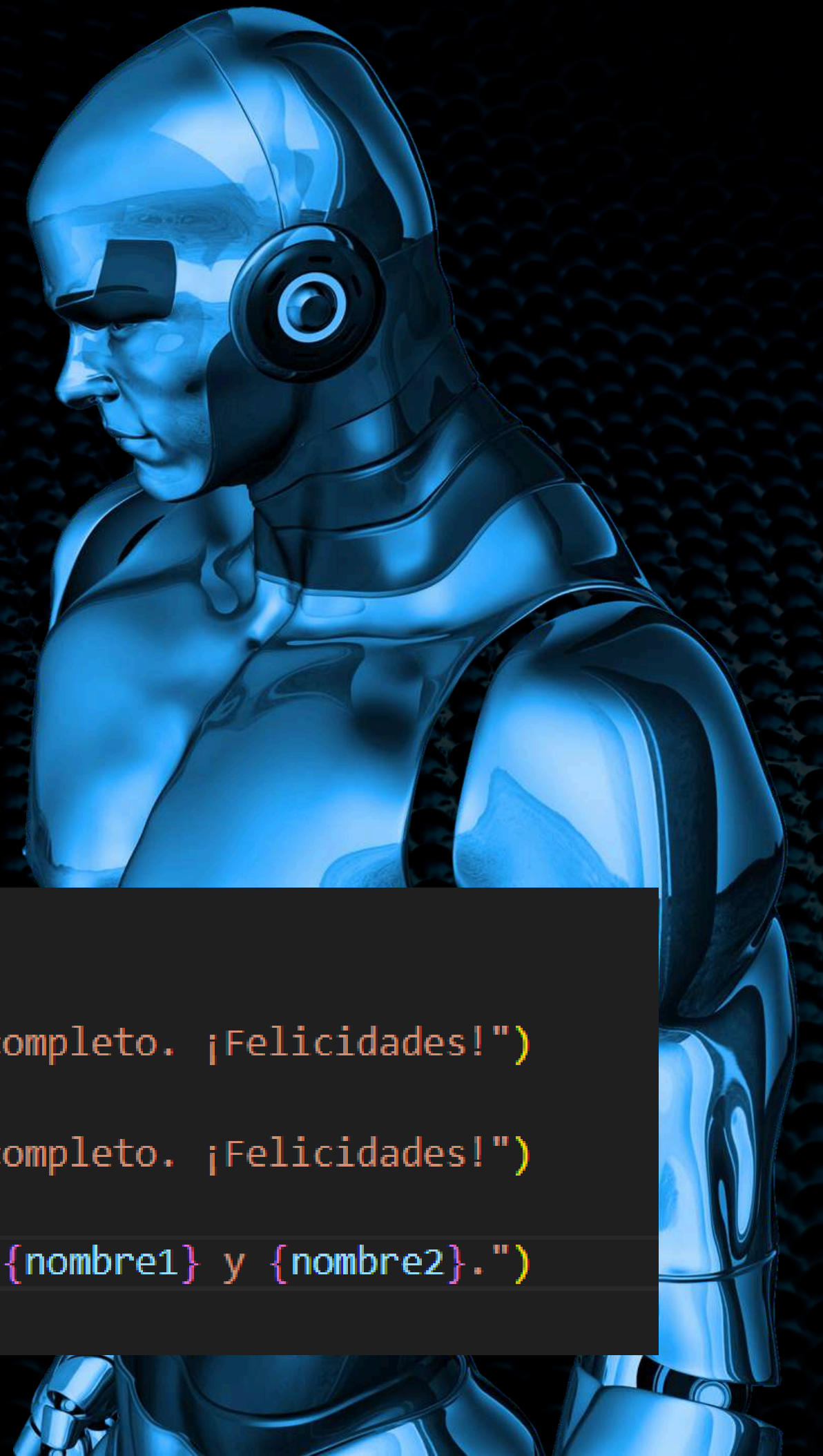
Al finalizar el juego, se muestra un resumen de cuántas veces ganó cada jugador y cuántos empates hubo.

```
# MOSTRAR RESUMEN FINAL
print("\nResumen de resultados:")
print('-----')
print(f"{nombre1} ganó {gana1} veces.")
print(f"{nombre2} ganó {gana2} veces.")
print(f"Empates: {empates}")
print('-----')
```


DETERMINAR AL GANADOR GENERAL

Se usa otro bloque condicional para declarar al ganador general del juego.

```
# DETERMINAR AL GANADOR GENERAL
if gana1 > gana2:
    print(f"{nombre1}, ganaste el juego completo. ¡Felicidades!")
elif gana2 > gana1:
    print(f"{nombre2}, ganaste el juego completo. ¡Felicidades!")
else:
    print(f"Hubo un empate general entre {nombre1} y {nombre2}.")
```



MANEJO DE EXCEPCIONES

Podemos incluir esto para evitar errores si algo sale mal al ingresar datos. Esto es una excepción con try-except.

```
try:
    opcion1 = input(f"{nombre1}, elige: Piedra, Papel o Tijera: ").lower()
    while opcion1 not in ['piedra', 'papel', 'tijera']:
        print("Opción inválida. Intenta de nuevo:")
        opcion1 = input().lower()
except:
    print("Ocurrió un error al ingresar la opción. Intenta de nuevo.")
    continue # vuelve a empezar el bucle
```


THANK YOU

