

PIEDRA PAPEL O TIJERA EN PYTHON

**EVALUACIÓN EN CONTACTO CON EL DOCENTE
LÓGICA DE PROGRAMACIÓN 2-SIN-1B
UIDE**

ANA CAROLINA VINTIMILLA

CONTENIDO

1. NOMBRE DEL PROYECTO
2. DESCRIPCIÓN GENERAL
DEL PROBLEMA
3. PROPÓSITO DEL
PROYECTO
4. INSTRUCCIONES PARA EL
DESARROLLO DEL
PROYECTO
5. INTEGRACIÓN DE LAS 4
UNIDADES
6. PROBLEMAS Y
SOLUCIONES
7. CONCLUSIONES

1. NOMBRE DEL PROYECTO

Juego Interactivo “Piedra, Papel o Tijera” para la aplicación de fundamentos de programación en Python.

Este proyecto integra los contenidos de las cuatro unidades mediante el desarrollo de un programa que simula el clásico juego de Piedra, Papel o Tijera entre el usuario y la computadora.

2. DESCRIPCIÓN GENERAL DEL PROBLEMA

El proyecto busca desarrollar una aplicación en Python que permita al usuario jugar múltiples rondas de Piedra, Papel o Tijera contra la computadora, en donde se cumpla:

- La validación correcta de entradas del usuario
- El uso de estructuras de control
- La implementación de programación funcional
- La simulación de decisiones aleatorias

El programa atiende el objetivo del juego en donde se deben cumplir las reglas del juego:

- Piedra gana a Tijera
- Tijera gana a Papel
- Papel gana a Piedra

3. PROPÓSITO DEL PROYECTO

Desarrollar un programa interactivo en Python que integre variables, estructuras de control, ciclos y funciones para simular el juego Piedra, Papel o Tijera.

- Aplicar variables y tipos de datos.
- Implementar condicionales para determinar el ganador.
- Utilizar ciclos para permitir múltiples rondas.
- Desarrollar funciones para modularizar el código.
- Incorporar la librería random para generar decisiones aleatorias de la computadora.
- Validar correctamente las entradas del usuario.

4. INSTRUCCIONES PARA EL DESARROLLO DEL PROYECTO

1. Solicitar al usuario el número de rondas a jugar.
2. Validar que la jugada ingresada sea correcta (piedra, papel o tijera).
3. Generar una jugada aleatoria para la computadora utilizando la librería random.
4. Comparar ambas jugadas mediante una función que determine el ganador.
5. Llevar un contador de puntuaciones.
6. Mostrar el resultado final al terminar todas las rondas.
7. Organizar el código utilizando programación funcional.
8. Subir el proyecto completo a un repositorio de GitHub con README explicativo.

5. INTEGRACIÓN DE LAS 4 UNIDADES

- Unidad 1 – Variables y Tipos de Datos: Uso de strings, enteros y listas.
- Unidad 2 – Condicionales: Uso de if, elif y else para determinar el ganador.
- Unidad 3 – Ciclos: Uso de for y while para repetir rondas y validar entradas.
- Unidad 4 – Funciones: Creación de funciones para validar jugadas y comparar resultados.

6 . P R O B L E M A S Y S O L U C I O N E S

1. Bucle infinito en estructuras while

- Problema:

El programa quedaba ejecutándose indefinidamente.

- Solución:

Se agregó una condición de salida adecuada y se utilizó correctamente el contador de rondas.

2. Error por función no definida

- Problema:

Se intentó llamar una función antes de declararla.

- Solución:

Se reorganizó el código colocando las funciones antes del bloque principal del programa.

6 . P R O B L E M A S Y S O L U C I O N E S

3. Validación incorrecta de entradas

- Problema:

El usuario podía ingresar valores inválidos y el programa fallaba o no contaba correctamente la jugada.

- Solución:

Se implementó una función con un ciclo while que valida la entrada hasta que sea correcta.

4. Repetición innecesaria de código

- Problema:

El programa tenía partes repetidas (como pedir las rondas dos veces).

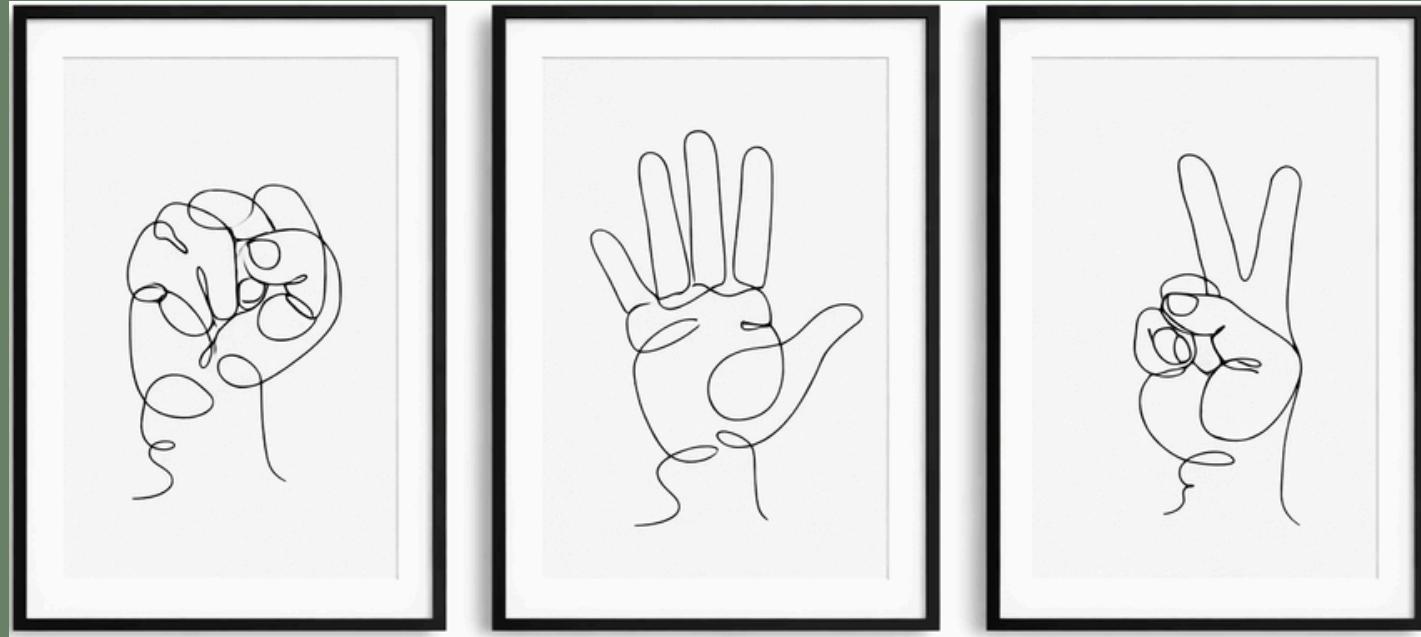
- Solución:

Se reorganizó el flujo del programa eliminando redundancias y mejorando la estructura lógica.

1. CONCLUSIONES

- El desarrollo del programa permitió aplicar de forma práctica los conocimientos adquiridos durante las cuatro unidades.
- Se fortaleció la comprensión de estructuras condicionales y ciclos.
- La implementación de funciones mejoró la organización y modularidad del código.
- Se comprendió la importancia de validar datos ingresados por el usuario.
- Se aprendió a integrar lógica de programación con generación aleatoria mediante la librería random

M U C H A S G R A C I A S



A N A C A R O L I N A V I N T I M I L L A