

22-12-2024

# EVALUACION FINAL

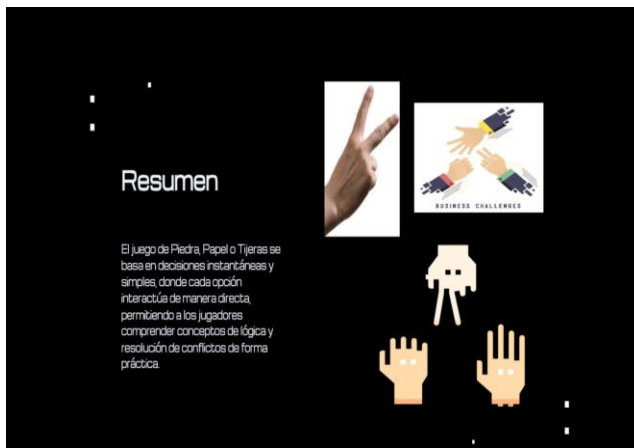
LOGICA DE PROGRAMACION 1-ECC-  
1B

## Contenido

PROYECTO INTEGRADOR .....	2
1. Desarrollo del Software .....	7
2. Enlace del video .....	7

**Facultad:** Facultad de Ciencias Técnicas  
**Escuela:** Facultad de Ciencias Técnicas  
**Carrera:** Ciberseguridad  
**Período académico:** 2024-2025  
**Docente:** MONICA PATRICIA SALAZAR TAPIA  
**Asignatura:** LOGICA DE PROGRAMACION 1-ECC-1B  
**Nombre del estudiante:** Shirley Escobar

## PROYECTO INTEGRADOR



Aquí podemos encontrar algunos de los objetivos

- Entretenimiento
- Aprendizaje de reglas y estrategias simples
- Desarrollo del pensamiento lógico

## OBJETIVO

Mantiene múltiples objetivos que van desde el entretenimiento hasta el desarrollo de habilidades sociales y cognitivas. Su versabilidad lo ha convertido en un juego popular en todo el mundo.

[+INFO](#)

Cita siempre al autor/a

**"recopilación de trabajos/uide/ShirleyEscobar"**

# ESTRUCTURA DEL JUEGO PIEDRA PAPEL O TIJERA

## CRONOLOGIA 1

### +UNIDAD 1

**Data**

**IDENTIFICACION DE LOS PROBLEMAS Y RESOLUCION**

La creación de un algoritmo para el juego de Piedra, Papel o Tijeras implica no solo definir las reglas y la lógica de decisión, sino también gestionar la entrada y validación de datos, asegurando que el sistema pueda manejar interacciones entre un jugador humano y un oponente virtual de manera eficiente y precisa.

### +UNIDAD 2

**Data**

**Manejo de Datos, algoritmos y diagramas.**

La creación de un algoritmo para el juego de Piedra, Papel o Tijeras implica no solo definir las reglas y la lógica de decisión, sino también gestionar la entrada y validación de datos, asegurando que el sistema pueda manejar interacciones entre un jugador humano y un oponente virtual de manera eficiente y precisa.

## CRONOLOGIA 2

### +UNIDAD 3

**Data**

**Logica de Programacion**

Piedra, Papel o Tijera es una excelente herramienta para:

- Comprender los fundamentos de la programación.
- Desarrollar habilidades de resolución de problemas.
- Crear programas sencillos y funcionales.

### +UNIDAD 4

**Data**

**Estructura de datos y Funciones.**

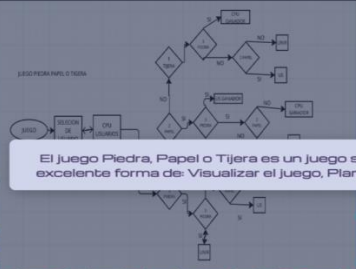

Es esencial en el juego de Piedra, Papel o Tijera porque permite organizar y gestionar las reglas del juego de manera eficiente, manejar múltiples rondas y resultados de manera estructurada, y facilitar la ampliación del juego con nuevas características.

**25% DE AVANCE**

## DIAGRAMAS

FUNCIONALIDAD Y ARQUITECTURA

**+ INFO**

El juego Piedra, Papel o Tijera es un juego sencillo, crear diagramas para puede ser una excelente forma de Visualizar el juego, Planificar el desarrollo y Documentar el proceso

**50% DE AVANCE**

**AMBIENTE**

## LOGICA DE PROGRAMACION

En este paso vamos a comenzar a preparar el ambiente para la etapa de codificación







**COMIENZO DEL JUEGO**

**SOLICITAR JUGADA**

**GENERAR JUGADA ALEATORIA**

**DETERMINAR EL GANADOR**

**PIEDRA GANA A TIJERA**



### Estructura de Datos y funciones

#### CODIFICACION DEL SOFTWARE

```

import random
def jugar_piedra_papel_tijera():
    ...
    Simula una partida de Piedra, Papel o Tijera.
    Returns:
    str: El resultado de la partida (gana jugador, gana
    computadora o empate).
    ...
    opciones = ["piedra", "papel", "tijera"]
    while True:
        # Jugada del usuario
        jugada_usuario = input("Elige tu jugada (piedra, papel o
        tijera): ").lower()
        while jugada_usuario not in opciones:
            print("Jugada inválida. Por favor, elige piedra, papel o
            tijera.")
        jugada_usuario = input("Elige tu jugada (piedra, papel o
        tijera): ").lower()
        # Jugada de la computadora
        jugada_computadora = random.choice(opciones)
        print(f"La computadora eligió: {jugada_computadora}")
  
```

75%

porcentaje de avance del software

+ INFO

### Estructura de Datos y funciones

#### CODIFICACION DEL SOFTWARE

```

import random
def jugar_piedra_papel_tijera():
    ...
    Simula una p
    Returns:
    str: El resul
    computadora
    ...
    opciones = [
    while True:
        # Jugada de
        jugada_usua
        tijera): ").lowe
        while jugada
        print("Jugad
        tijera.")
        jugada_usua
        tijera): ").lowe
        # Jugada de
        jugada_com
        print(f"La co
  
```

75%

porcentaje de avance del software

X

```

# Determinar ganador
if jugada_usuario == jugada_computadora:
    print("Empate!")
elif (jugada_usuario == "piedra" and jugada_computadora ==
    "tijera") or \
    (jugada_usuario == "papel" and jugada_computadora ==
    "piedra") or \
    (jugada_usuario == "tijera" and jugada_computadora ==
    "papel"):
    print("Ganaste!")
  
```

**100%**  
porcentaje de avance del software

**software completado**

```

JUEGO PIEDRA PAPEL O TIJERA
else:
    print("¡Perdiste!")
    # ¿Jugar otra vez?
    jugar_de_nuevo = input("¿Quieres jugar otra
    vez? (si/no): ").lower()
    if jugar_de_nuevo != "si":
        break
    # Iniciar el juego
    jugar_piedra_papel_tijera()
  
```

Tabla de partidas

	PIEDRA	PAPEL	TIJERA	EMPATE	PIERDE	GANA
US	1	1	1	0	2	1
CPU	1	0	2	0	2	1
US	0	2	1	1	1	1
CPU	2	0	1	0	3	0



# Comprobacion del juego

## EJECUCION DEL JUEGO

```
def jugar_piedra_papel tijera():
    print("La computadora eligió: (jugada_computadora)")

    # Interacción jugador
    if jugada_usuario == jugada_computadora:
        print("Empate!")
    elif (jugada_usuario == "piedra" and jugada_computadora == "tijera") or \
        (jugada_usuario == "tijera" and jugada_computadora == "piedra") or \
        (jugada_usuario == "papel" and jugada_computadora == "papel") or \
        (jugada_usuario == "papel" and jugada_computadora == "tijera"):
        print("Ganaste!")
    else:
        print("Perdiste!")

    # Jugador gana o no
    jugador_gana = input("¿Quieres jugar otra vez? (si/no): ").lower()
    if jugador_gana == "si":
        jugar_piedra_papel_tijera()
    else:
        print("¡Gracias!")
```

GANA EL US

```
def jugar_piedra_papel_tijera():
    print("La computadora eligió: (jugada_computadora)")

    # Interacción jugador
    if jugada_usuario == jugada_computadora:
        print("Empate!")
    elif (jugada_usuario == "piedra" and jugada_computadora == "tijera") or \
        (jugada_usuario == "tijera" and jugada_computadora == "piedra") or \
        (jugada_usuario == "papel" and jugada_computadora == "papel") or \
        (jugada_usuario == "papel" and jugada_computadora == "tijera"):
        print("Ganaste!")
    else:
        print("Perdiste!")

    # Jugador gana o no
    jugador_gana = input("¿Quieres jugar otra vez? (si/no): ").lower()
    if jugador_gana == "si":
        jugar_piedra_papel_tijera()
    else:
        print("¡Gracias!")
```

GANA CPU

```
def jugar_piedra_papel_tijera():
    print("La computadora eligió: (jugada_computadora)")

    # Interacción jugador
    if jugada_usuario == jugada_computadora:
        print("Empate!")
    elif (jugada_usuario == "piedra" and jugada_computadora == "tijera") or \
        (jugada_usuario == "tijera" and jugada_computadora == "piedra") or \
        (jugada_usuario == "papel" and jugada_computadora == "papel") or \
        (jugada_usuario == "papel" and jugada_computadora == "tijera"):
        print("Ganaste!")
    else:
        print("Perdiste!")

    # Jugador gana o no
    jugador_gana = input("¿Quieres jugar otra vez? (si/no): ").lower()
    if jugador_gana == "si":
        jugar_piedra_papel_tijera()
    else:
        print("¡Gracias!")
```

EMPATE



## 1. Desarrollo del Software

```
import random
def jugar_piedra_papel_tijera():
    """
    Simula una partida de Piedra, Papel o Tijera.
    Returns:
    str: El resultado de la partida (gana jugador, gana computadora o empate).
    """
    opciones = ["piedra", "papel", "tijera"]
    while True:
        # Jugada del usuario
        jugada_usuario = input("Elige tu jugada (piedra, papel o tijera): ").lower()
        while jugada_usuario not in opciones:
            print("Jugada inválida. Por favor, elige piedra, papel o tijera.")
            jugada_usuario = input("Elige tu jugada (piedra, papel o tijera): ").lower()
        # Jugada de la computadora
        jugada_computadora = random.choice(opciones)
        print(f"La computadora eligió: {jugada_computadora}")
        # Determinar ganador
        if jugada_usuario == jugada_computadora:
            print("¡Empate!")
        elif (jugada_usuario == "piedra" and jugada_computadora == "tijera") or \
            (jugada_usuario == "papel" and jugada_computadora == "piedra") or \
            (jugada_usuario == "tijera" and jugada_computadora == "papel"):
            print("¡Ganaste!")
        else:
            print("¡Perdiste!")
        # ¿Jugar otra vez?
        jugar_de_nuevo = input("¿Quieres jugar otra vez? (si/no): ").lower()
        if jugar_de_nuevo != "si":
            break
    # Iniciar el juego
    jugar_piedra_papel_tijera()
```

## 2. Enlace del video

<https://view.genially.com/6768e5d853faf8c77b96e1b1/presentation-evaluacion-final>