



# **El impacto de las nuevas tecnologías en la sociedad: visualización del futuro.**

Visualizar el impacto de las nuevas tecnologías en la sociedad y reflexionen sobre su potencial futuro.

Nombre del estudiante: Pedro Javier Salguero Zúñiga.  
Carrera: Ingeniería en Ciberseguridad.  
Materia: Lógica de la programación.  
Docente: Paulina Vizcaino.  
Universidad: Universidad Internacional del Ecuador (UIDE).

## PROYECTO INTEGRADOR

**Instrucciones para el desarrollo del proyecto:** Definir alcances totales del proyecto de entrega desde semana 1 hasta la 8, tomando en cuenta los 8 temas.

### 1.- Software a desarrollar:

Piedra, papel o tijeras.

### 2.- Comprensión del problema:

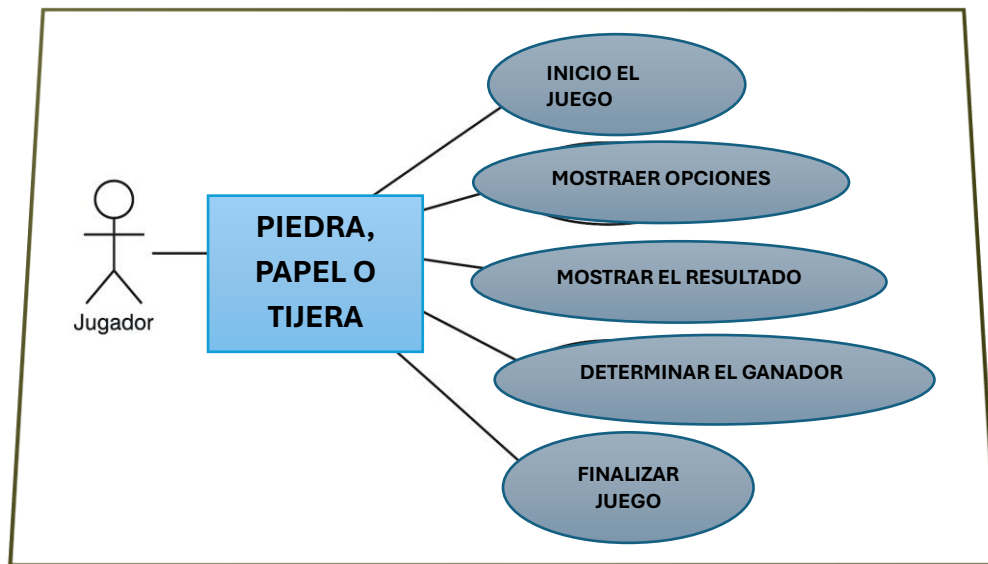
**Objetivo:** Encontrar el ganador del juego piedra, papel o tijera.

**Reglas:** Escoger una opción de las tres (piedra, papel, tijera) y determinar el ganador a la tercera ronda, para este juego hay dos jugadores (humano y/o computadora). A continuación, sus reglas del juego:

“Piedra” vence a “Tijeras”	→	Gana Piedra
“Tijeras” vence a “Papel”	→	Gana Tijeras
“Papel” vence a “Piedra”	→	Gana Papel
Si ambos eligen lo mismo	→	Empate

### 3.- Diseño de la funcionalidad:

**Diagrama de caso de uso:** Sirven para mostrar de una forma gráfica los distintos casos usos de nuestro software por parte del usuario.



**4.- Diagramas de flujo:** Aquí tuvimos que realizar 3 diagramas, uno para la opción de piedra, otro para la opción de papel y el ultimo tijeras.

Diagrama #1 Tijeras

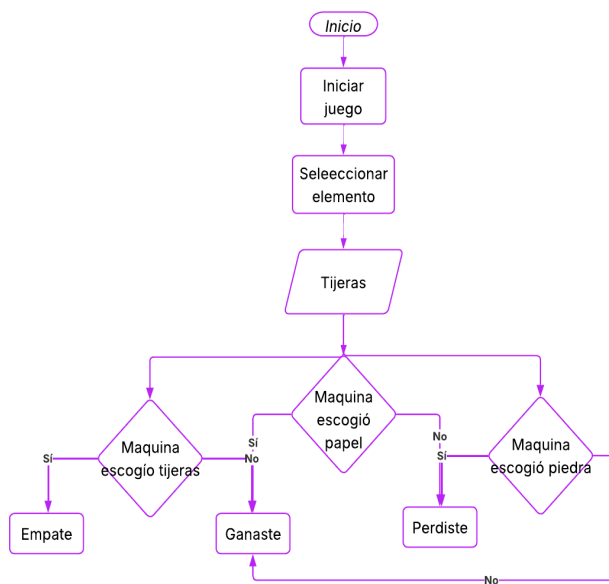


Diagrama #2 Piedra

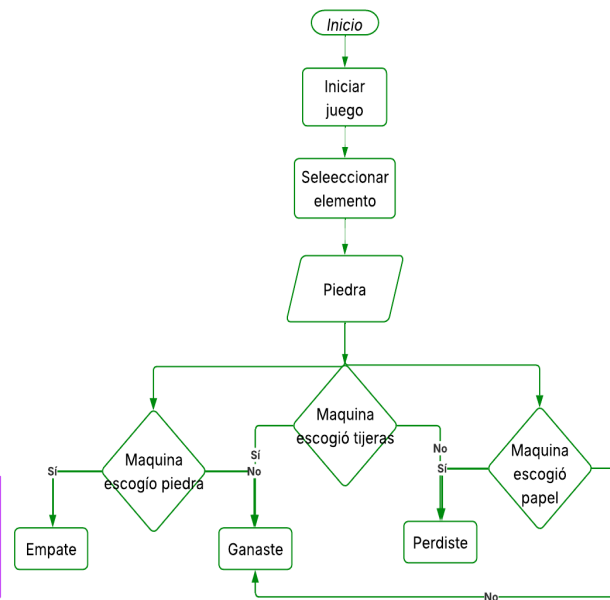
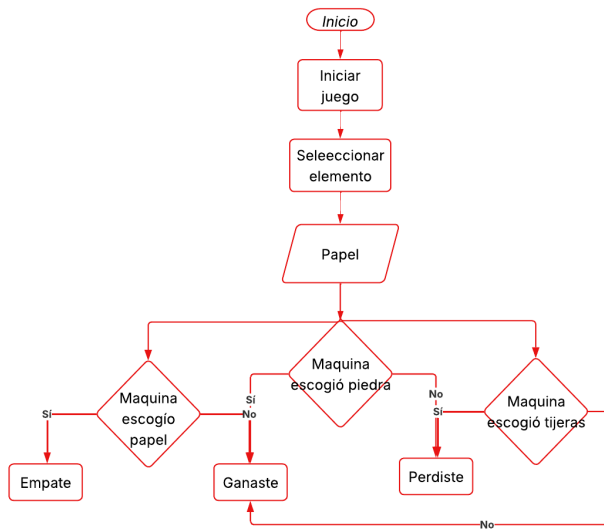


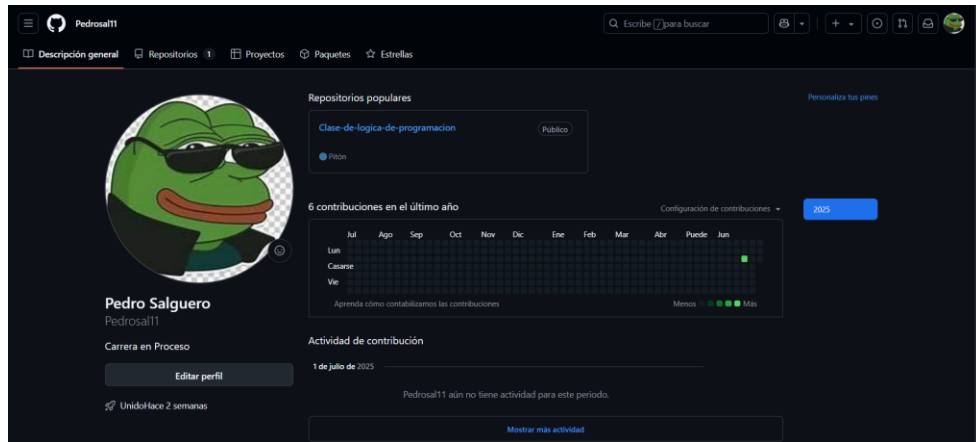
Diagrama #3 Papel



### 5.- Análisis a la fase de diseño de las funcionalidades:

- Iniciar juego.
- Mostrar las opciones (piedra, papel o tijeras).
- Ingresar la orden elegida.
- Selección de la computadora usando número random.
- Mostrar el resultado.
- Determinar el ganador.
- Desea continuar jugando.

**6.- Entorno de programación:** Personalizamos nuestro entorno de programación, nuestro perfil y repositorio de Github, en mi caso usé de fondo de pantalla el tema oscuro, además añadí algo de información personal y se hizo el cambio de la imagen de perfil.



**7.- Desarrollo del Software:** En la primera entrega del código que se subió al repositorio del Github tuve varios errores.

❖ **Errores sintácticos que se cometieron:**

- Palabras incorrectas como (importar, imprimir, aporte).
- Uso de “y” en lugar de “and”.
- Mal uso de mayúsculas en palabras clave como (Elif, si, demás).

❖ **Errores semánticos que se cometieron:**

- No validar la entrada del jugador
- Comparar valores sin estandarizarlos ("Tijeras" vs "tijeras")

```

1
2     #PIEDRA, PAPEL O TIJERAS
3
4     importar aleatorio
5     opciones=("papel","piedra","tijeras")
6
7     jugador = aporte("Elige papel, piedra o tijeras:")
8     máquina = aleatorio.elección(opciones)
9
10    imprimir("máquina de selección",máquina)
11
12    si máquina == jugador:
13        imprimir("empatar")
14    Elif(jugador== "piedra" y máquina== "tijeras")o\
15        (jugador== "papel" y máquina== "piedra")o\
16        (jugador)==("tijeras" y máquina== "papel"):
17        imprimir("Ganaste")
18    demás:
19        imprimir("Lo siento, perdiste")

```

**8.- Desarrollo del Software:** Con la ayuda de las clases brindadas se incluyeron varias funcionalidades como las tuplas, variable, for, un `while` que dio por concluido con nuestro software a desarrollar. “piedra, papel o tijeras” y así quedó.

- ✚ Se importa el módulo **random**, que permite a la computadora elegir una opción aleatoria entre las opciones (piedra, papel o tijeras).
- ✚ Se crea una **tupla** llamada opciones con las 3 posibles jugadas.
- ✚ Las **tuplas** son inmutables y se usan aquí porque los valores no cambiarán durante el juego.
- ✚ Se muestran mensajes de bienvenida usando “**print**” al usuario y se explica la regla principal: gana quien alcance primero 3 rondas.
- ✚ Se implementa un **bucle “while”** que seguirá repitiéndose mientras ni el jugador ni la máquina hayan llegado a 3 victorias.
- ✚ Se inserta un `(string).lower()` esto hace que convierta lo que escribe el jugador a minúsculas, para evitar errores por diferencias de mayúsculas/minúsculas.

```

1  import random
2
3  # Opciones disponibles:
4  opciones = ("piedra", "papel", "tijeras")
5
6  # Puntos de jugador y máquina:
7  puntos_jugador = 0
8  puntos_maquina = 0
9
10 print("¡Bienvenido al juego Piedra, Papel o Tijeras!")
11 print("Primero en ganar 3 rondas es el campeón.")
12
13 while puntos_jugador < 3 and puntos_maquina < 3:
14     jugador = input("Elige piedra, papel o tijeras: ").lower()
15
16     # Validación de entrada:
17     if jugador not in opciones:
18         print("Opción inválida. Intenta de nuevo.")
19         continue
20
21     maquina = random.choice(opciones)
22     print(f"La máquina eligió: {maquina}")

```

- ✚ Se introduce un **if** que evalúa si una **condición es verdadera** y si lo es, ejecuta el bloque de código que está intentado justo debajo.
- ✚ Si el jugador escribe algo incorrecto (como "piedraa" o "fuego"), se muestra un error y se salta el resto del bucle usando **continue**.

## Lógica del juego:

- ✚ Si ambos eligieron la misma jugada, es un empate.
- ✚ Estas son todas las combinaciones en que el jugador **gana**:

(Piedra vence a Tijeras); (Papel vence a Piedra) o (Tijeras vencen a Papel)

Si se cumple alguna de estas, se muestra un mensaje de victoria y se suma un punto al jugador.

```

24 # Lógica del juego:
25 if jugador == maquina:
26     print("Empate.")
27 elif (jugador == "piedra" and maquina == "tijeras") or \
28      (jugador == "papel" and maquina == "piedra") or \
29      (jugador == "tijeras" and maquina == "papel"):
30     print("¡Ganaste esta ronda!")
31     puntos_jugador += 1
32 else:
33     print("Perdiste esta ronda.")
34     puntos_maquina += 1
35
36 # Mostrar marcador:
37 print(f"Puntos -> Tú: {puntos_jugador} | Máquina: {puntos_maquina}")
38 print("-" * 40)
39
40 # Resultado final:
41 if puntos_jugador == 3:
42     print("¡Felicidades! Ganaste el juego. ")
43 else:
44     print(" La máquina ganó el juego. ¡Suerte la próxima vez!")
45

```

- ✚ Se muestra el mensaje final de victoria o derrota usando **if** si es ganador y un **else** si es perdedor.
- ✚ Y así es como nos quedaría la terminal dándole play, en este ejemplo el usuario ganó el juego, ahí se muestra el puntaje del usuario y maquina.

```

Elige piedra, papel o tijeras: piedra
La máquina eligió: papel
Perdiste esta ronda.
Puntos -> Tú: 2 | Máquina: 2
-----
Elige piedra, papel o tijeras: tijeras
La máquina eligió: papel
¡Ganaste esta ronda!
Puntos -> Tú: 3 | Máquina: 2
-----
¡Felicidades! Ganaste el juego.
PS C:\Users\GOVIS>

```