

ADALAB



TEAM 49



- Equipo
- Metodología
- Descripción
- Logros
- Next Steps

EQUIPO TÉCNICO

SANDRA SIMÓN

AMANDA HERNÁNDEZ

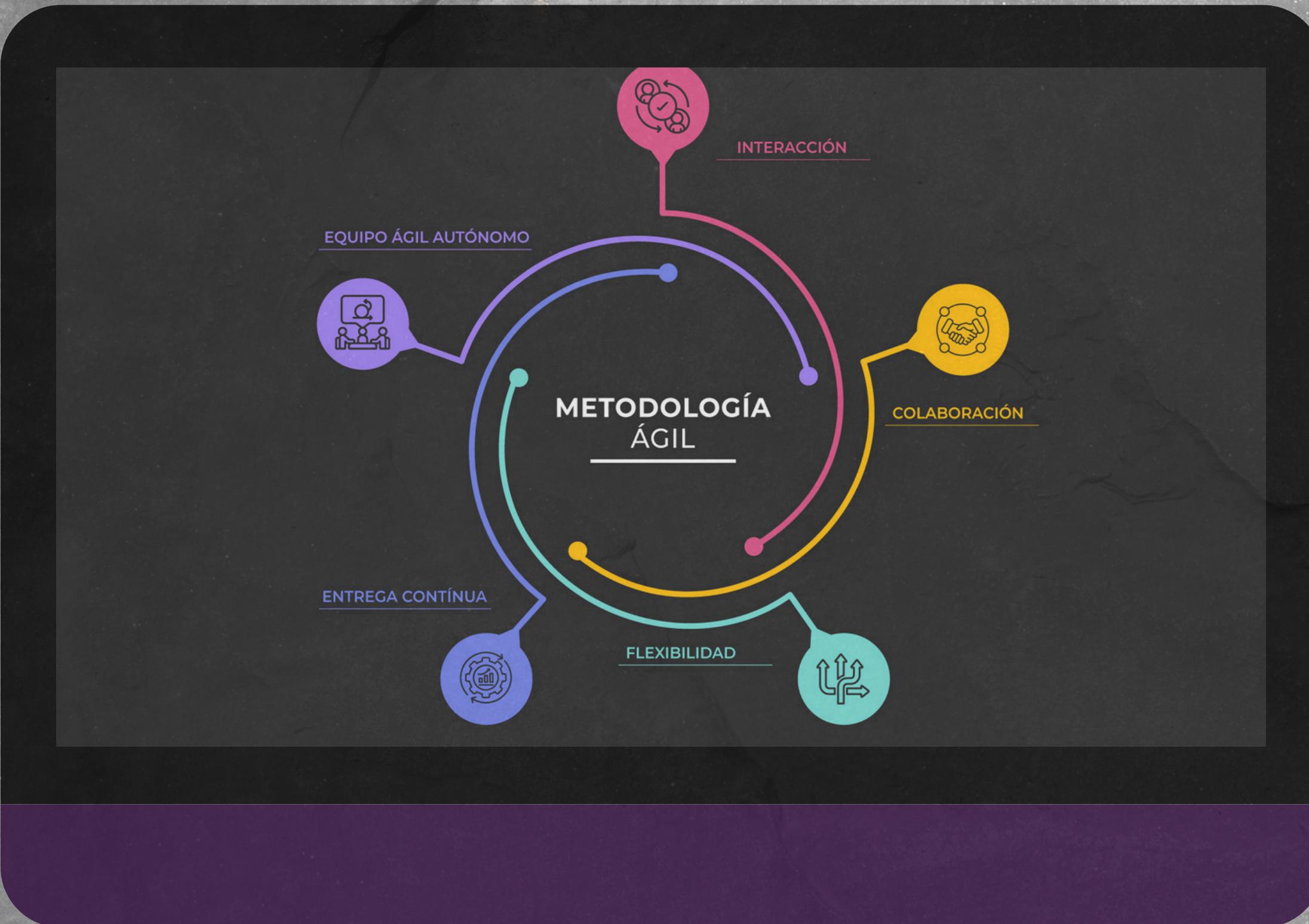
JAZMÍN SÁNCHEZ

CRISTINA MARTÍN

SCRUM MASTER

SARA ARRANZ





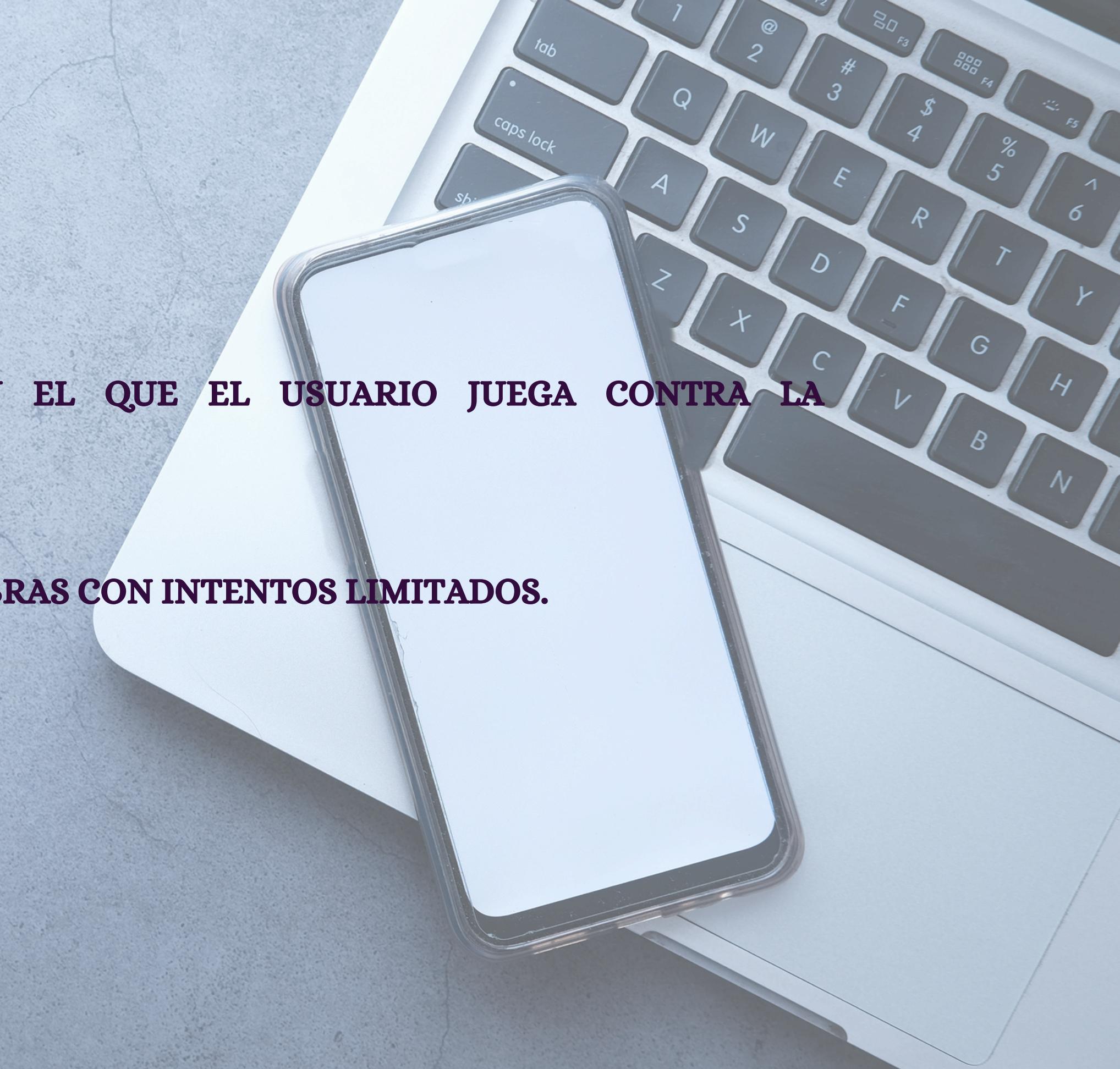
- CREACIÓN DE RED
- COMUNICACIÓN ASERTIVA
- MEJORA DE PROCESOS
- CONTROL DE TIEMPOS
- MANEJO DE DIFICULTADES
- DESARROLLO DE COMPETENCIAS

DESARROLLAMOS DOS JUEGOS TRADICIONALES:

- **PIEDRA-PAPEL-TIJERA:** JUEGO CLÁSICO EN EL QUE EL USUARIO JUEGA CONTRA LA COMPUTADORA.
- **AHORCADO:** JUEGO DE ADIVINANZA DE PALABRAS CON INTENTOS LIMITADOS.

LENGUAJE DE PROGRAMACIÓN PYTHON.

- **MANEJO DE SINTAXIS BÁSICA**
- **USO DE ESTRUCTURAS DE CONTROL (BUCLLES)**
- **DISEÑO DE FUNCIONES**



LOGROS

```
while True:
    eleccion = input('Elige piedra, papel o tijera: ').lower()
    if eleccion in opciones:
        return eleccion
    print('Opción no válida. Inténtalo de nuevo.')

usuario_puntos = 0
puntos_objetivo = 3
pc_puntos = 0

print('¡¡¡Bienvenido al juego de Piedra, Papel y Tijera!!!')
print(f'El primero en llegar a {puntos_objetivo} puntos gana.\n')

while usuario_puntos < puntos_objetivo and pc_puntos < puntos_objetivo:
    usuario = obtener_eleccion()

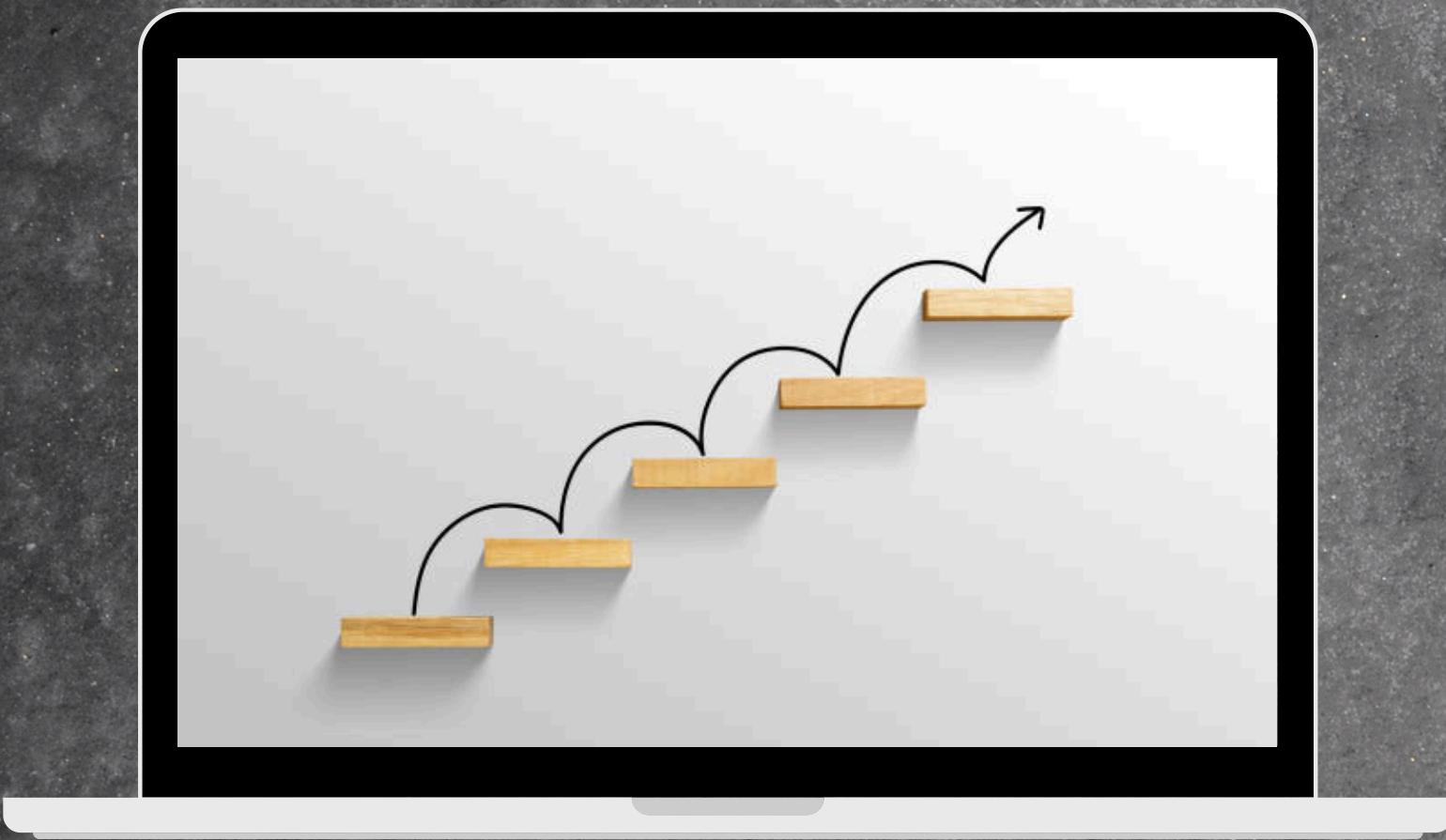
    pc = random.choice(['piedra', 'papel', 'tijera'])

    print(f'Tú eliges: {usuario}')
    print(f'Tu contrincante eligió: {pc}')

    if usuario == pc:
        print('Has empatado 😑')
    elif (usuario == 'piedra' and pc == 'tijera') or\
        (usuario == 'tijera' and pc == 'papel') or\
        (usuario == 'papel' and pc == 'piedra'):
        print('¡Has ganado esta ronda! 🎉')
        usuario_puntos += 1
    else:
        print('Has perdido esta ronda. 😞')
```

- Código efectivo
- Programación defensiva
- Limpieza visual
- Usabilidad

NEXT STEPS



- Ampliación del código que facilite la opción de volver a jugar: "¿jugar otra vez? S/N"
- Mayor versatilidad en el diseño
- Mayor flujo de control de las excepciones posibles

GRACIAS