



# Proyecto: Piedra, papel o tijera

## Curso de Fundamentos

### Introducción

Para este proyecto, crearás el juego [Piedra, papel o tijera](#) . Este juego se jugará completamente en la consola.

Algunas de las soluciones para estudiantes que se muestran a continuación contienen botones, texto y otros elementos. Estos elementos forman parte de lo que se denomina una interfaz gráfica de usuario (GUI). Crearás la GUI en una lección posterior. Mientras tanto, recuerda enviar tu código a GitHub.

### Enfoque de resolución de problemas

Dado que este es el primer proyecto de JavaScript que se crea desde cero, es importante recordar las sabias palabras de la [lección de resolución de problemas](#) . Para cada paso de este proyecto, asegúrese de hacer lo siguiente

1. Planifique o pseudocodifique su solución.
2. Escribe el código.
3. Pruebe su código para asegurarse de que funciona.

### Asignación

¡Recuerde realizar los compromisos con anticipación y con frecuencia! Para refrescar su memoria, consulte la [lección sobre los mensajes de compromiso](#).

### Paso 1: Configurar la estructura del proyecto

1. Crea un nuevo repositorio Git para tu proyecto.
2. Crea un documento HTML en blanco con una etiqueta de script.
3. Compruebe si JavaScript está vinculado correctamente:
  - Escribir `console.log("Hello World")` en JavaScript.
  - Comprueba si "Hola mundo" está registrado en la consola del navegador una vez que abras tu página web.

Se recomienda crear un vínculo a un archivo JavaScript externo dentro de esta etiqueta de script. El uso de un archivo JavaScript externo mantiene el archivo HTML limpio y organizado.

No es necesario escribir código adicional en el archivo HTML. Este juego se juega completamente a través de la consola.

### Paso 2: Escribe la lógica para obtener la elección de la computadora

Jugarás contra la computadora. Escribirás una función que devuelva aleatoriamente "piedra", "papel" o "tijeras".

1. Crea una nueva función llamada `getComputerChoice`.
2. Escriba el código para que `getComputerChoice` se obtenga aleatoriamente `return` uno de los siguientes valores de cadena: "piedra", "papel" o "tijeras".
  - **Sugerencia** : El método [Math.random](#) devuelve un número aleatorio mayor o igual a 0 y menor que 1. Piense en cómo puede usar esto para devolver condicionalmente una de las opciones múltiples.
  - **Nota** : Algunos estudiantes pueden pensar que las matrices son necesarias para este problema, pero no lo son. Si no estás familiarizado con las matrices, no te preocupes: las abordaremos más adelante en el programa.

3. Pruebe que su función devuelva lo esperado utilizando `console.log` las [herramientas para desarrolladores del navegador](#) antes de avanzar al siguiente paso.

Paso 3: Escribe la lógica para obtener la elección humana.

Tu juego será jugado por un jugador humano. Escribirás una función que tome la elección del usuario y la devuelva.

1. Crea una nueva función llamada `getHumanChoice`.
2. Escriba el código para que `getHumanChoice` devuelva una de las opciones válidas dependiendo de lo que ingrese el usuario.
  - **Sugerencia** : utilice el método [de solicitud](#) para obtener la entrada del usuario.
3. Pruebe lo que su función devuelve utilizando `console.log`.

Paso 4: Declarar las variables de puntuación de los jugadores

Tu juego llevará un registro de la puntuación de los jugadores. Escribirás variables para llevar un registro de la puntuación de los jugadores.

1. Crea dos nuevas variables llamadas `humanScore` y `computerScore` en el ámbito global.
2. Inicialice esas variables con el valor de `0`.

Paso 5: Escribe la lógica para jugar una sola ronda

Tu juego se jugará ronda por ronda. Escribirás una función que tome las elecciones de los jugadores humanos y de la computadora como argumentos, juegue una sola ronda, incremente el puntaje del ganador de la ronda y registre un anuncio del ganador.

1. Crea una nueva función llamada `playRound`.
2. Defina dos parámetros para `playRound`: `humanChoice` y `computerChoice`. Utilice estos dos parámetros para tomar las opciones de la persona y la computadora como argumentos.
3. Haga que el parámetro de su función `humanChoice` no distinga entre mayúsculas y minúsculas para que los jugadores puedan ingresar "rock", "ROCK", "Rock" u otras variaciones.

4. Escribe el código de tu `playRound` función en `console.log` un valor de cadena que represente al ganador de la ronda, como por ejemplo:  
"¡Pierdes! El papel le gana a la piedra".
5. Incrementa la variable `humanScore` o `computerScore` en función del ganador de la ronda.

Código de ejemplo:

```
1 function playRound(humanChoice, computerChoice) {  
2   // your code here!  
3 }  
4  
5 const humanSelection = getHumanChoice();  
6 const computerSelection = getComputerChoice();  
7  
8 playRound(humanSelection, computerSelection);
```

Paso 6: Escribe la lógica para jugar todo el juego.

Tu juego se jugará en 5 rondas. Escribirás una función llamada `playGame` que llame `playRound` para jugar 5 rondas, lleve un registro de los puntajes y declare un ganador al final.

1. Crea una nueva función llamada `playGame`.
2. Mueva sus `playRound` variables de función y puntuación para que se declaren dentro de la nueva `playGame` función
3. Juega 5 rondas llamando `playRound` 5 veces.
  - **Sugerencia** : cuando asigna una llamada de función a una variable, el valor de retorno de esa función se asigna a la variable. Acceder a la variable posteriormente solo proporcionará el valor asignado; no recuperará la función. Debe recuperar las funciones de elección para obtener nuevas opciones para cada ronda.
  - Si es necesario, vuelva a trabajar en sus funciones anteriores o cree más funciones auxiliares. En concreto, es posible que desee cambiar los valores de retorno por otros más útiles.

- Si ya conoces los bucles, puedes usarlos. Si no, ¡no te preocupes! Los bucles se tratarán en la próxima lección.

Al realizar proyectos interactivos, como este, es posible que sienta la tentación de agregar más funciones, mejorar la interactividad, la experiencia del usuario, el diseño y el estilo de su sitio web, etc.

Le recomendamos no hacer eso y guardar este esfuerzo para sus proyectos de cartera.

Para obtener más información sobre la mentalidad de aprendizaje y las piezas del portafolio, lea [la Parte 5](#) y [la Parte 7](#) de [Cómo convertirse en una historia de éxito](#) [TOP](#)

 [Mejorar en GitHub](#)

 [Informar un problema](#)

[Ver registro de cambios de la lección](#)

## Tu solución

[Ver soluciones comunitarias](#)



Envíe su solución



[Ver curso](#)

[Marcar como completado](#)

[Próxima lección](#)

## ¡Apóyanos!

El Proyecto Odin está financiado por la comunidad. ¡Únase a nosotros para ayudar a estudiantes de todo el mundo apoyando el Proyecto Odin!

[Más información](#)[Dona ahora](#)

THE ODIN PROJECT

Educación en codificación de alta calidad mantenida por una comunidad de código abierto.



### Sobre nosotros

[Acerca de](#)[Equipo](#)[Blog](#)[Casos de éxito](#)

### Apoyo

[Preguntas frecuentes](#)[Contribuir](#)[Contáctenos](#)

### Guías

[Guías de la comunidad](#)[Guías de instalación](#)

### Legal

[Términos](#)[Privacidad](#)

© 2025 El Proyecto Odin. Todos los derechos reservados.