

# Primer Entrega: Simulador Interactivo de Filosofía

Estimado Profesor-tutor Facundo Casal,

Espero que te encuentres bien. Estoy contento de presentarte la primera entrega en el curso de JavaScript en Coderhouse. El proyecto consiste en un simulador interactivo de preguntas y respuestas de filosofía, diseñado para ayudar a los estudiantes que cursan esta materia a repasar conceptos y teorías filosóficas de una manera divertida y educativa.

En esta primera entrega, he aplicado los conceptos fundamentales de JavaScript, incluyendo condicionales, ciclos y funciones, de la siguiente manera:

## Uso de Condicionales en JavaScript:

- He implementado condicionales para validar las respuestas proporcionadas por los usuarios. Cuando un estudiante responde una pregunta, el sistema verifica si la respuesta es correcta o incorrecta utilizando declaraciones condicionales. Si la respuesta es correcta, se muestra un mensaje de felicitación y se procede a la siguiente pregunta en 2 segundos. Si la respuesta es incorrecta, muestra la respuesta que corresponde, pasando luego de 5 segundos a la próxima pregunta.
- También he utilizado condicionales para gestionar el flujo del juego, asegurando que las preguntas se presenten en el orden correcto y que el juego continúe hasta que se hayan respondido todas las preguntas.

---

### 1. Donde se aplica el condicional en: **verificarRespuesta(evento)**:

En la función **verificarRespuesta(evento)**: Se utiliza un condicional **if** para verificar si la respuesta seleccionada por el usuario es la respuesta correcta o no. Dependiendo de esto, se muestra un mensaje de "Respuesta correcta" o "Respuesta incorrecta" en la pantalla

```
function verificarRespuesta(evento) {
  const respuestaSeleccionada = evento.target.textContent;
  const preguntaActual = preguntas[indicePreguntaActual];

  if (respuestaSeleccionada === preguntaActual.respuestaCorrecta) {
    puntaje++;
    elementoRetroalimentacion.textContent = "¡Respuesta correcta!";
    // Avanzar automáticamente a la siguiente pregunta después de 2 segundos
    setTimeout(siguientePregunta, 2000);
  } else {
    elementoRetroalimentacion.textContent = `Respuesta incorrecta. La respuesta correcta es: ${preguntaActual.respuestaCorrecta}`;
    // Avanzar automáticamente a la siguiente pregunta después de 5 segundos
    setTimeout(siguientePregunta, 5000);
  }

  elementoPuntaje.textContent = `Puntuación: ${puntaje}`;
}
```

---

## 2. En la función **siguientePregunta()**:

En la función **siguientePregunta()**: Se utiliza un condicional **if** para verificar si aún hay más preguntas por responder. Si hay más preguntas, carga la siguiente pregunta; de lo contrario, muestra el mensaje "Juego Terminado" y activa el botón de reinicio.

```
function siguientePregunta() {  
    indicePreguntaActual++;  
  
    if (indicePreguntaActual < preguntas.length) {  
        cargarPregunta();  
        elementoRetroalimentacion.textContent = "";  
    } else {  
        // Juego terminado, muestra la puntuación final  
        elementoPregunta.textContent = "Juego Terminado";  
        elementoOpciones.innerHTML = "";  
        elementoRetroalimentacion.textContent = "";  
        elementoPuntaje.textContent = `Puntuación final: ${puntaje}`;  
        mostrarBotonReinicio(); // Muestra el botón de reinicio  
    }  
}
```

## Uso de Ciclos en JavaScript:

El ciclo principal en el código es manejado por la función **jugarJuego()**, que se ejecuta de forma repetitiva mientras **indicePreguntaActual** sea menor que la longitud de la matriz de preguntas (**preguntas.length**). Esta función verifica si todavía hay preguntas disponibles para mostrar y, si es así, carga una nueva pregunta con **cargarPregunta()** y espera la respuesta del usuario con **esperarRespuesta()**. Si todas las preguntas se han respondido, muestra la puntuación final y termina el juego.

---

## 1. Donde se aplica el Ciclo en la función **jugarJuego**:

```
function jugarJuego() {  
    if (indicePreguntaActual < preguntas.length) {  
        cargarPregunta();  
        esperarRespuesta();  
    } else {  
        // Juego terminado, muestra la puntuación final  
        elementoPregunta.textContent = "Juego Terminado";  
        elementoOpciones.innerHTML = "";  
        elementoRetroalimentacion.textContent = "";  
        elementoPuntaje.textContent = `Puntuación final: ${puntaje}`;  
        mostrarBotonReinicio(); // Muestra el botón de reinicio  
    }  
}
```

En resumen, los ciclos se utilizan en `jugarJuego` para recorrer las preguntas y esperar la respuesta del usuario antes de avanzar a la siguiente pregunta. Esto permite que el juego funcione de manera automática y que las preguntas se presenten secuencialmente.

## Uso de Funciones en JavaScript:

- Para mantener el código organizado y modular, he creado funciones que se encargan de verificar las respuestas del usuario y determinar su corrección. Estas funciones se llaman de manera consistente cada vez que un usuario responde una pregunta y se han diseñado de manera reutilizable en todo el juego.

---

### 1. `cargarPregunta()`:

```
function cargarPregunta() {  
    const preguntaActual = preguntas[indicePreguntaActual];  
    elementoPregunta.textContent = `Pregunta ${indicePreguntaActual + 1}: ${preguntaActual}`;  
}
```

---

### 2. `siguientePregunta()`:

```
function siguientePregunta() {  
    indicePreguntaActual++;  
}
```

---

### 3. verificarRespuesta(evento):

```
function verificarRespuesta(evento) {  
    const respuestaSeleccionada = evento.target.textContent;  
    const preguntaActual = preguntas[indicePreguntaActual];
```

---

### 4. mostrarBotonReinicio():

```
function mostrarBotonReinicio() {  
    elementoBotonReinicio.style.display = "block";  
}
```

---

### 5. reiniciarJuego():

```
function reiniciarJuego() {  
    indicePreguntaActual = 0;  
    puntaje = 0; // Reiniciamos el puntaje a 0  
    elementoPuntaje.textContent = `Puntuación: ${puntaje}`; // Actualizamos el e  
    cargarPregunta();  
    elementoRetroalimentacion.textContent = "";  
    elementoBotonReinicio.style.display = "none"; // Oculta el botón al reinicia  
}
```

En resumen, esta primera entrega de mi proyecto demuestra la aplicación efectiva de los conceptos de JavaScript, implementando condicionales, ciclos y funciones en la creación de un simulador interactivo de preguntas y respuestas de filosofía.

Estoy con muchas ganas de seguir incorporando conocimiento para mejorar las próximas entregas.

Todavía no tengo definido el proyecto final pero en breve me estaré comunicando.

Agradezco sinceramente su orientación y espero sus comentarios y sugerencias para poder seguir mejorando mi trabajo en las entregas futuras.

Atentamente,  
Mariano Pablo Vaccaro

GitHub: <https://github.com/Mariano581/simuladorFilo1.0>

Juego: <https://mariano581.github.io/simuladorFilo1.0/>

web: <https://simulador-filo1-0.vercel.app/>