

En este documento se pretende documentar las pruebas de la aplicación con diferentes casos, para así asegurar su funcionamiento, así como la lógica y codificación de la misma.

En primer lugar, ya dado el array de posibilidades con las que vamos a trabajar, inicializamos las variables con las que trabajaremos.

El nombre del jugador, el número de partidas seleccionadas, el número de partidas jugadas, y la opción o posibilidad seleccionada por el usuario. El resultado de las partidas será guardado en el historialPartidas.

```
// Array proporcionado que no se puede modificar
var posibilidades = ["piedra", "papel", "tijera"];

// Variables globales
let nombreJugador = '';
let partidasTotales = 0;
let partidasJugadas = 0;
let opcionSeleccionada = null;
let historialPartidas = [];
```

Luego, para la configuración de la aplicación, añadimos un evento del DOM que asegure que la página cargue, antes de que el usuario pueda interactuar con esta, ejecutamos la configuración que veremos más adelante. Luego, configuramos los eventos para los botones de jugar partida, realizar tirada y reiniciar juego.

```
// Función principal que se ejecuta al cargar la página
document.addEventListener('DOMContentLoaded', function() {
    // Configuración inicial de la aplicación
    configurarAplicacion();

    // Asignar eventos a los botones
    document.querySelector('button').addEventListener('click', comenzarPartida);
    document.querySelectorAll('button')[1].addEventListener('click', realizarTirada);
    document.querySelectorAll('button')[2].addEventListener('click', resetearJuego);
});
```

En la función configuración de la aplicación nos encargaremos de obtener las imágenes del jugador mediante un `querySelectorAll`, para así mostrar al jugador las opciones para jugar en cada tirada. Luego usaremos un bucle `for` para asignar a cada imagen el evento correspondiente con el array `posibilidades`. Y por último añadiremos un evento `click` a cada opción para ser seleccionadas por el usuario.

```

// Configuración inicial de la aplicación
function configurarAplicacion() {
    // Obtener todas las imágenes del jugador
    const imagenesJugador = document.querySelectorAll('#jugador img');

    // Asignar eventos y configurar imágenes del jugador
    for (let i = 0; i < imagenesJugador.length; i++) {
        if (i < posibilidades.length) {
            // Configurar la imagen correspondiente (sin modificar el array original)
            const opcion = posibilidades[i];
            imagenesJugador[i].src = `img/${opcion}Jugador.png`;

            // Asignar evento de clic
            imagenesJugador[i].addEventListener('click', function() {
                seleccionarOpcion(i);
            });
        } else {
            // Ocultar imágenes adicionales si no hay suficientes opciones
            imagenesJugador[i].style.display = 'none';
        }
    }
}

```

Luego para configurar las partidas, introduciremos los requisitos, primero obtenemos mediante un `querySelector` el nombre y las partidas que desea jugar el usuario. Luego de esto validamos los requisitos, revisando que el string introducido por el usuario sea superior a 3 caracteres y no comience por un número; después comprobamos que el usuario introduzca un número mayor a 0 partidas para jugar. Mientras estos requisitos no se cumplan, el juego no comenzará y el marcador estará en rojo como comprobaremos más adelante en el funcionamiento de la página.

Una vez hecho esto deshabilitamos los campos de entrada para evitar que el usuario sobreescriba.

```

// Función para comenzar una nueva partida
function comenzarPartida() {
    // Obtener valores de los campos
    const nombreInput = document.querySelector('input[name="nombre"]');
    const partidasInput = document.querySelector('input[name="partidas"]');

    // Validar nombre
    const nombre = nombreInput.value.trim();
    let nombreValido = nombre.length > 3 && isNaN(parseInt(nombre[0]));

    // Validar número de partidas
    const partidas = parseInt(partidasInput.value);
    let partidasValidas = partidas > 0;

    // Aplicar estilos de validación
    nombreInput.classList.toggle('fondoRojo', !nombreValido);
    partidasInput.classList.toggle('fondoRojo', !partidasValidas);

    // Si ambos campos son válidos, comenzar el juego
    if (nombreValido && partidasValidas) {
        nombreJugador = nombre;
        partidasTotales = partidas;
        partidasJugadas = 0;

        // Deshabilitar campos de entrada
        nombreInput.disabled = true;
        partidasInput.disabled = true;

        // Actualizar contadores
        document.getElementById('actual').textContent = '0';
        document.getElementById('total').textContent = partidasTotales;
    }
}

```

Para la selección del jugador verificamos que el índice u opción del jugador sea válida dentro del array y añadimos los eventos de seleccionar y deseleccionar a cada opción.

```

// Función para seleccionar una opción del jugador
function seleccionarOpcion(indice) {
    // Verificar que el índice sea válido
    if (indice >= 0 && indice < posibilidades.length) {
        opcionSeleccionada = indice;
        const imagenesJugador = document.querySelectorAll('#jugador img');

        // Aplicar estilos de selección
        for (let i = 0; i < imagenesJugador.length; i++) {
            if (i === indice) {
                imagenesJugador[i].classList.add('seleccionado');
                imagenesJugador[i].classList.remove('noSeleccionado');
            } else {
                imagenesJugador[i].classList.add('noSeleccionado');
                imagenesJugador[i].classList.remove('seleccionado');
            }
        }
    }
}

```

Mediante la función determinar resultado manipulamos aquellos casos en los que el jugador gana mediante un if, los empates mediante un segundo condicional, y aquellos donde el jugador pierde.

```

function determinarResultado(jugador, maquina) {
    // jugador y maquina son índices: 0 = piedra, 1 = papel, 2 = tijera

    // Empate
    if (jugador === maquina) {
        return 'empate';
    }

    // Casos donde gana el jugador
    if (
        (jugador === 0 && maquina === 2) || // piedra gana a tijera
        (jugador === 1 && maquina === 0) || // papel gana a piedra
        (jugador === 2 && maquina === 1)    // tijera gana a papel
    ) {
        return 'jugador';
    }

    // Si no es empate ni gana el jugador, gana la máquina
    return 'maquina';
}

```

Luego, para la función realizar tirada, comprobamos que el número de partidas jugadas no iguale ni supere a las partidas seleccionadas por el usuario; que el usuario haya seleccionado una opción y que el nombre del jugador sea válido.

```
// Función para realizar una tirada
function realizarTirada() {
  // Verificar que se pueda jugar
  if (partidasJugadas >= partidasTotales || opcionSeleccionada === null || !nombreJugador) {
    return;
  }

  // Generar opción aleatoria para la máquina (usando el array original sin modificarlo)
  const opcionMaquina = Math.floor(Math.random() * posibilidades.length);
  const opcionMaquinaNombre = posibilidades[opcionMaquina];
  const imagenMaquina = document.querySelector('#maquina img');
  imagenMaquina.src = `img/${opcionMaquinaNombre}Ordenador.png`;

  // Incrementar contador de partidas jugadas
  partidasJugadas++;
  document.getElementById('actual').textContent = partidasJugadas;

  // Determinar el resultado usando el array original
  const resultado = determinarResultado(opcionSeleccionada, opcionMaquina);

  // Registrar en el historial
  agregarAlHistorial(resultado);
}
```

Hecho esto, generamos una opción seleccionada por la máquina mediante un math random. Ya que esto devuelve un valor entre 0 y 1, multiplicamos por el número de elementos en el array posibilidades; así el valor obtenido es entre 0 y 3, apuntando a un índice del array. Luego aumentamos el número de la variable partidas jugadas y determinamos el resultado, el cual es almacenado en el historial y mostrado por pantalla al usuario.

```

// Función para agregar un resultado al historial
function agregarAlHistorial(resultado) {
    let mensaje = '';

    switch (resultado) {
        case 'jugador':
            mensaje = `Gana ${nombreJugador}`;
            break;
        case 'maquina':
            mensaje = 'Gana la máquina';
            break;
        case 'empate':
            mensaje = 'Empate';
            break;
        default:
            mensaje = 'Nueva partida';
    }

    // Agregar al array de historial
    historialPartidas.push(mensaje);

    // Actualizar la lista en el DOM
    const listaHistorial = document.getElementById('historial');
    listaHistorial.innerHTML = '';

    for (const item of historialPartidas) {
        const li = document.createElement('li');
        li.textContent = item;
        listaHistorial.appendChild(li);
    }
}

```

Esto último lo haremos manejando los distintos casos en los que el jugador gana, empata o pierde, mediante un switch.

```

// Función para resetear el juego
function resetearJuego() {
    // Agregar mensaje de nueva partida al historial
    agregarAlHistorial('nueva');

    // Habilitar campos de entrada (excepto el nombre)
    document.querySelector('input[name="partidas"]').disabled = false;
    document.querySelector('input[name="partidas"]').value = '0';

    // Reiniciar contadores
    partidasJugadas = 0;
    document.getElementById('actual').textContent = '0';
    document.getElementById('total').textContent = '0';

    // Reiniciar imagen de la máquina
    document.querySelector('#maquina img').src = 'img/defecto.png';

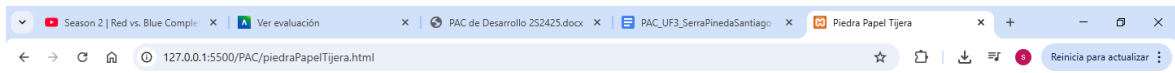
    // Reiniciar selección
    seleccionarOpcion(0);
}

```

Finalmente para resetear el juego, mostramos por pantalla el mensaje “nueva partida” en el historial, reiniciamos el contador y la selección del jugador.

Ahora en la prueba de la página:

En primer lugar comprobamos que , según código, la aplicación descarte casos en los que el usuario quiera utilizar un nombre de usuario no válido, bien sea menor a cuatro caracteres o el cual empiece por un número en vez de una letra.



Juego de Piedra Papel Tijera

Introduce el nombre del jugador

¿Cuántas partidas quieres jugar?

Jugando la partida 0 de 0.



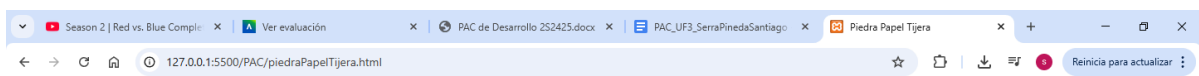
Piedra, papel o tijera....



Historial de partidas



Ejemplo de ambos errores



Juego de Piedra Papel Tijera

Introduce el nombre del jugador

¿Cuántas partidas quieres jugar?

Jugando la partida 0 de 0.



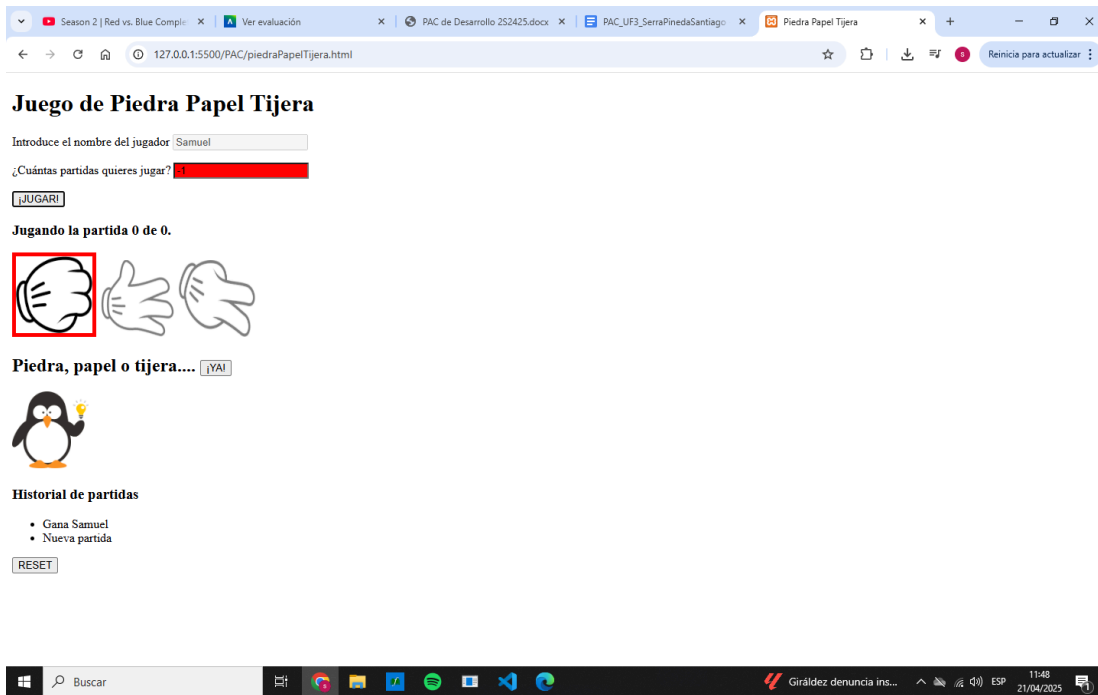
Piedra, papel o tijera....



Historial de partidas



Ejemplo de número de partidas igual a 0.



Ejemplo de número negativo de partidas para jugar.

Juego de Piedra Papel Tijera

Introduce el nombre del jugador

¿Cuántas partidas quieres jugar?

Jugando la partida 3 de 3.



Piedra, papel o tijera....



Historial de partidas

- Gana la máquina
- Gana la máquina
- Gana la máquina
- Gana Santiago
- Gana la máquina
- Nueva partida
- Empate
- Gana la máquina
- Gana la máquina

Y ejemplo jugando, reiniciando y volviendo a jugar.