



PAC DESARROLLO UF 3

# Desarrollo web en entorno cliente

Martín Agüera Bitaubé

ÍNDICE

1. VARIABLES GLOBALES ..... 3

2. SECCIÓN DE EVENTOS..... 4

3. FUNCIÓN JUGAR ..... 4

4. FUNCIÓN SELECCION ..... 5

5. FUNCIÓN YA..... 6

6. FUNCIÓN GANADOR ..... 7

7. FUNCIÓN RESET..... 8

8. PRUEBAS ..... 9

9. BIBLIOGRAFÍA..... 11

## 1. VARIABLES GLOBALES

La actividad comienza con la declaración de una serie de variables y objetos globales que son necesarios para ir elaborando las diferentes funciones:

```
JS piedraPapelTijera.js X
C: > Users > aguera > Desktop > Actividades de desarrollo > entorno cliente > PAC > JS piedraPapelTijera.js > ...
1 // Este array no se puede modificar,
2 var posibilidades = ["piedra", "papel", "tijera"];
3 //
4
5 /* Variables y objetos globales */
6 var partidas = 0; //total de partidas
7 var partidaActual=1;
8 var nombre="";
9 var miMarcador=0;
10 var marcadorPc=0;
11 var seleccionado=posibilidades[0]; //seleccion del jugador por defecto
12 var seleccionadoPc=posibilidades[0]; //seleccion del pc por defecto
13 var ganadoresList=[];
14 class Ganador {
15     constructor(name,choice,choicePc){
16         this.name=name;
17         this.choice=choice;
18         this.choicePc=choicePc;
19     };
20 };
21
22 /*-----*/
23
```

La clase Ganador instancia un objeto con el nombre del ganador de esa partida, la elección del usuario (piedra, papel o tijera) y la elección de la máquina. Esta instancia se añade al array ganadoresList.

El siguiente bloque de código corresponde a la sección de eventos.

## 2. SECCIÓN DE EVENTOS

En la sección de eventos están declarados todos los eventos necesarios para desarrollar la aplicación. Cada botón de la aplicación tiene una escucha sobre su evento clic, y las tres imágenes correspondientes al jugador también tienen la escucha al evento clic, declaradas mediante un bucle for para ahorrar código y que quede más ordenado.

La función de la escucha del evento clic sobre las imágenes (función llamada seleccion), recibe como argumento el índice i de la imagen correspondiente.

```
/* Sección de eventos */
var botonJugar=document.getElementsByTagName("button")[0]; //captura del primer elemento boton
botonJugar.addEventListener("click",jugar,false); //creacion de la escucha del evento click sobre el botonJugar, para realizar la accion jugar.
var botonYa=document.getElementsByTagName("button")[1]; //captura del segundo elemento boton
botonYa.addEventListener("click",ya,false); //creacion de la escucha del evento click sobre el botonYa, para realizar la accion ya.
for (let i=0; i<3;i++){ //for para añadir addEventListener a las tres imagenes con su funcion seleccion, pasando el argumento i
    document.getElementsByTagName("img")[i].addEventListener("click",function(){seleccion(i);},false);
};
document.getElementsByTagName("button")[2].addEventListener("click",reset,false);

/*----- */
```

Si pulsamos el botón jugar, el botón ya, alguna de las tres imágenes o el botón reset, realizaremos las funciones jugar(), ya(), seleccion() o reset() respectivamente, funciones que detallaremos a continuación.

## 3. FUNCIÓN JUGAR

La función jugar tiene como finalidad recibir los parámetros de los inputs del html para preparar la partida.

Si estos valores cumplen los requisitos solicitados en la descripción de la actividad, se prepara el inicio de la partida.

```
function jugar(){ //funcion para obtener valor de los inputs y posicionar las imagenes
    nombre=document.getElementsByName("nombre")[0].value; //dato del input nombre
    partidas=parseInt(document.getElementsByName("partidas")[0].value); //dato del input partida parseado a Int;
    if (/^\d/.test(nombre)){ //utilizo una expresion regular que devuelve true si el nombre empieza por dígito
        alert("El nombre no puede empezar por un numero");
        document.getElementsByName("nombre")[0].className="fondoRojo";
    }
    else if (partidas<=0){
        alert("el número de partidas tiene que ser mayor a 0");
        document.getElementsByName("partidas")[0].className="fondoRojo";
    }
    else{
        document.getElementsByName("nombre")[0].className=""; //quitamos fondo rojo si lo tenía
        document.getElementsByName("partidas")[0].className=""; //quitamos fondo rojo si lo tenía
        document.getElementById("actual").innerHTML=partidaActual; // cambiamos contenido html actual de 0 a 1
        document.getElementById("total").innerHTML=partidas; // cambiamos contenido html total al total de partidas
        document.getElementsByTagName("img")[0].src="img/piedraJugador.png"; //cambiamos las imagenes por la de piedra, papel y tijera.
        document.getElementsByTagName("img")[1].src="img/papelJugador.png";
        document.getElementsByTagName("img")[2].src="img/tijeraJugador.png";
        for (let i=0; i<2;i++){ //desactivamos los 2 inputs
            document.getElementsByTagName("input")[i].disabled=true;
        }
    }
};
//-----
```

A continuación se pasaría a la función seleccion() si el usuario clicla alguna imagen, si no, quedaría por defecto seleccionada piedra y podría darle al botón ya.

## 4. FUNCIÓN SELECCION

La función selección se activa cuando el usuario clicla sobre alguna de las imágenes disponibles para elegir, recibiendo como argumento 0 si es piedra, 1 si es papel o 2 si es tijera.

En el caso de que el índice i coincida con el número de la imagen pasado en el argumento, se cambia la clase de esa imagen ha seleccionado y la del resto de imágenes a noSeleccionado.

```
function seleccion(numeroImagen){ /*funcion para obtener la imagen que selecciona el usuario y la variable seleccionado,
ya que coinciden en el mismo orden que en el array posibilidades;*/
    for(let i=0;i<3;i++){
        if (i==numeroImagen){
            document.getElementsByTagName("img")[i].className="seleccionado";
            seleccionado= posibilidades[i];
        }
        else{
            document.getElementsByTagName("img")[i].className="noSeleccionado";
        }
    }
};
//-----
```

## 5. FUNCIÓN YA

Al pulsar sobre el botón ya se realiza esta función.

En esta función se elige una imagen al azar para la máquina mediante el objeto Math y su método random, multiplicando este valor por 2 y redondeándolo, para obtener un valor entero entre 0 y 2.

Aprovechamos que los nombres de las imágenes comienzan por las mismas palabras que los elementos del array posibilidades para concatenar en la ruta hacia ellas.

En esta función he querido jugar con el método setTimeout para realizar las acciones de manera más lenta y así poder ver la imagen seleccionada de manera random antes de que finalice la partida.

Utilizamos la variable global seleccionado para realizar diferentes condiciones, empezando por comprobar si la elección del jugador coincide con la de la máquina, en cuyo caso alerta que la partida se ha empatado y no cuenta, en caso de no coincidir, se comparan las tres opciones disponibles con la elección de la máquina, y se accede a la función ganador, pasándole como argumento true, en el caso de ganar la partida, y false en el caso de que la gane la máquina.

```
function ya(){ //función para jugar contra el pc
  let random=Math.round(Math.random()*2); //random entre 0 y 2
  seleccionadoPc = posibilidades[random]; //seleccion del pc al azar
  document.getElementsByTagName("img")[3].src="img/"+ posibilidades[random]+"Ordenador.png"; // ponemos la imagen correspondiente al pc
  setTimeout(function(){//le ponemos tiempo a la realización de la función para que de tiempo a ver la imagen del Pc (0.7 seg)
    if (seleccionado!=seleccionadoPc){//si son diferentes se entra en la logica
      if(seleccionado=="piedra"){ //jugador escoge piedra
        if (seleccionadoPc=="papel"){
          ganador(false); //gana pc
        }
        else {
          ganador(true); //gana jugador
        }
      }
      else if (seleccionado=="papel"){ //jugador escoge papel
        if (seleccionadoPc=="piedra"){
          ganador(true);
        }
        else{
          ganador(false);
        }
      }
      else{ //jugador escoge tijeras
        if (seleccionadoPc=="piedra"){
          ganador(false);
        }
        else{
          ganador(true);
        }
      }
    }
    else{ //si son iguales, empate
      alert("Ha habido un empate, esta partida no cuenta y hay que repetirla!");
    }
  },700);
};
//-----
```

## 6. FUNCIÓN GANADOR

Esta función recibe como argumento los booleanos true o false, si se gana o se pierde la partida.

Se instancia un objeto de la clase ganador con los parámetros de nombre, elección del jugador y elección de la máquina, variando el nombre dependiendo de si gana el usuario o la máquina. Esta instancia se añade al array ganadoresList.

Se actualizan marcadores, se actualiza el historial en el html, se incrementa el valor de partidaActual y se comprueba si era la última partida, en cuyo caso se comprueban los marcadores, se alerta sobre quién ha ganado la partida y se ejecuta location.reload() para refrescar la página una vez pasados 2,5 segundos.

```
function ganador(winner){//funcion que crea la instancia ganador, la añade a ganadoreslist, actualiza el historial y actualiza marcadores.
  if (winner){
    const ganador=new Ganador(nombre,seleccionado,seleccionadoPc);
    ganadoresList.push(ganador);
    miMarcador++;
  }
  else{
    const ganador=new Ganador("Máquina",seleccionado,seleccionadoPc);
    ganadoresList.push(ganador);
    marcadorPc++;
  }
  let lista= document.getElementById("historial");
  let linea= document.createElement("li");
  linea.innerHTML=partidaActual + ". Gana: "+ganadoresList[ganadoresList.length-1].name+" (" +nombre+ " saca: "+seleccionado+ " | Máquina saca: "+seleccionadoPc+"");
  lista.appendChild(linea);
  partidaActual++;
  if (partidaActual>partidas){//comprobamos si era la ultima partida
    if (miMarcador>marcadorPc){
      setTimeout(function(){alert("Juego terminado.\nEnhorabuena, según la última serie de partidas has ganado!!!! "+miMarcador+" vs "+marcadorPc);},2000);
    }
    else if (miMarcador<marcadorPc){
      setTimeout(function(){alert("Juego terminado.\nLo siento, según la última serie de partidas has perdido!!! "+miMarcador+" vs "+marcadorPc);},2000);
    }
    else{
      setTimeout(function(){alert("Juego terminado.\n según la última serie de partidas Habéis empatado!!! "+miMarcador+" vs "+marcadorPc);},2000);
    }
    setTimeout(function(){location.reload();},2500);
  }
  else{
    document.getElementById("actual").innerHTML=partidaActual;
  }
}
//-----
```

## 7. FUNCIÓN RESET

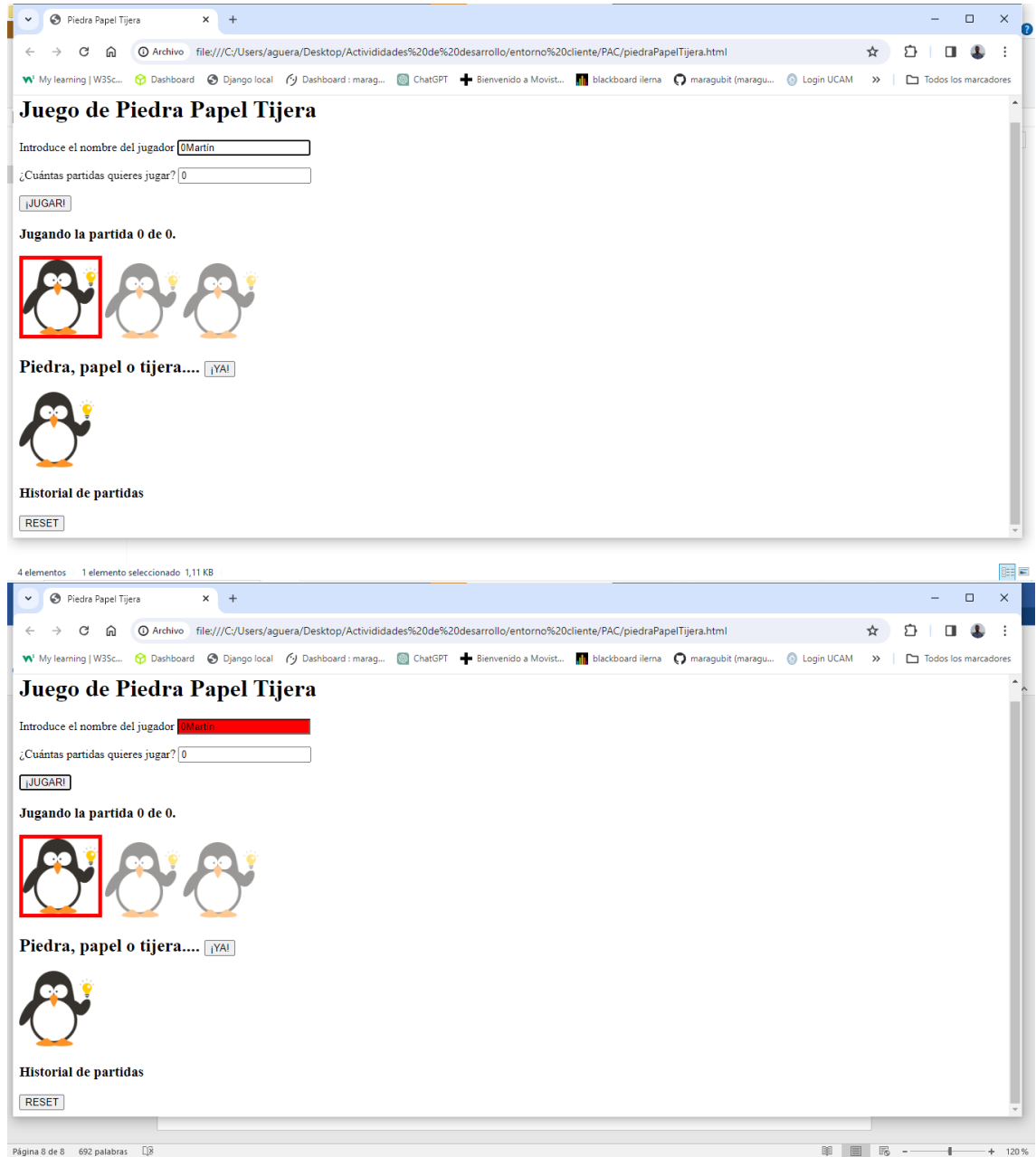
La función reset se ejecuta cuando se pulsa sobre el botón reset. Se trata de una función simple que resetea a 0 las partidas totales y los marcadores, las clases de las imágenes se ponen como estaban por defecto, y se vuelve a activar el input de cantidad de partidas.

```
function reset(){ //reinicia el número de partidas y marcadores
    alert("Nueva partida")
    partidas=0;
    partidaActual=1;
    miMarcador=0;
    marcadorPc=0;
    document.getElementById("actual").innerHTML=partidaActual;
    document.getElementById("total").innerHTML=partidas;
    document.getElementsByTagName("img")[3].src="img/defecto.png";
    document.getElementsByTagName("input")[1].disabled=false;
    document.getElementsByTagName("img")[0].className="seleccionado";
    document.getElementsByTagName("img")[1].className="noSeleccionado";
    document.getElementsByTagName("img")[2].className="noSeleccionado";
};
```

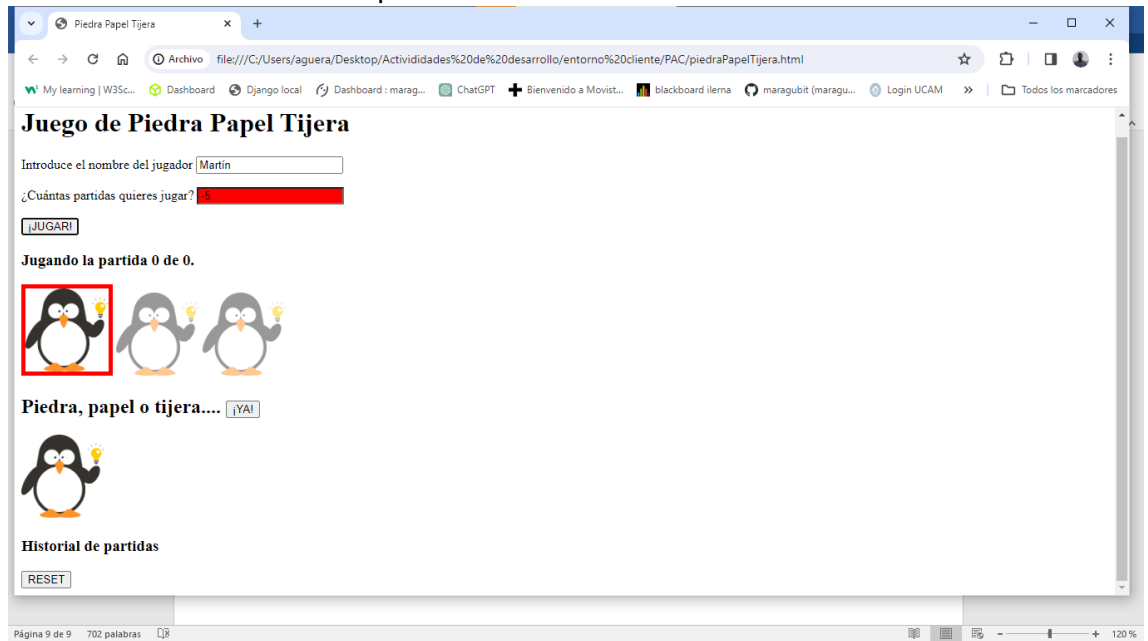


## 8. PRUEBAS

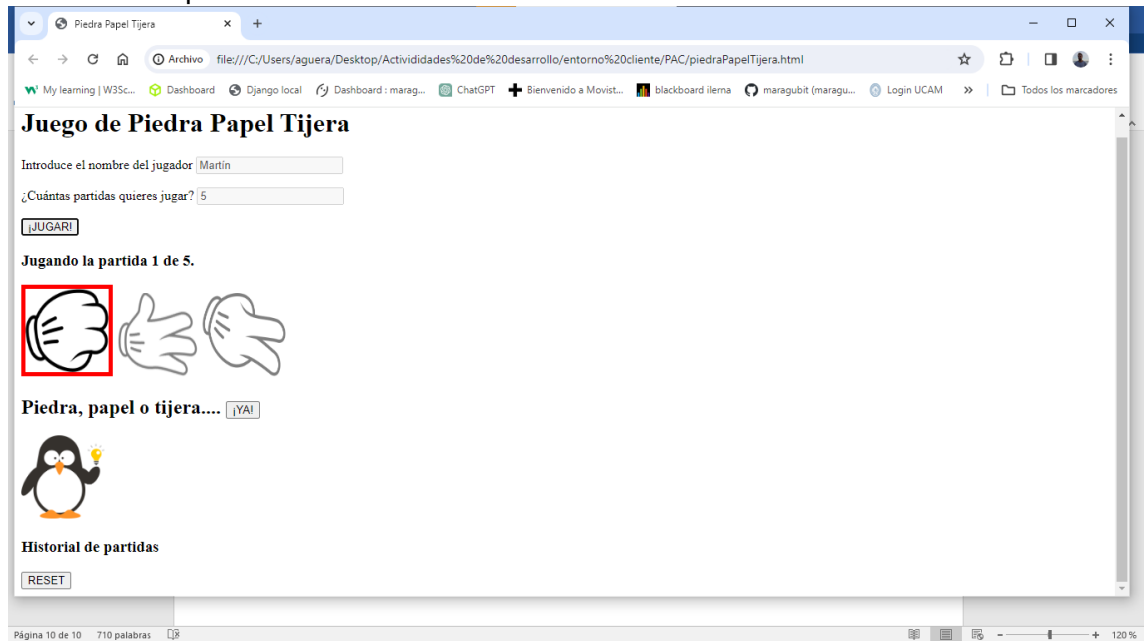
### 1. Introducción del usuario con datos no válidos



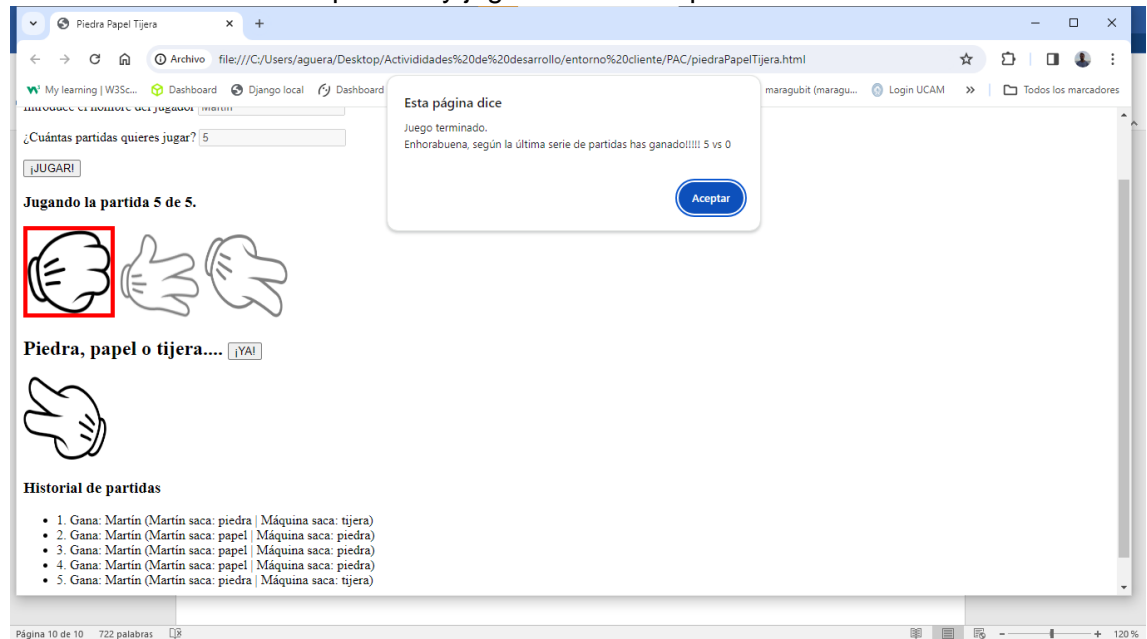
2. Introducción de cantidad de partidas con datos no válidos.



3. Acceso a la aplicación con datos válidos.



4. Seleccionar una de las opciones y jugar al menos 5 partidas.



## 9. BIBLIOGRAFÍA

*My learning / W3Schools.* (s. f.-c). <https://my-learning.w3schools.com/tutorial/js>

Ilerna Online. (2023). *Desarrollo web en entorno cliente*. Ilerna Online S.L.

Ilerna Online. (2023). *Lenguaje de marcas y sistemas de gestión de información*. Ilerna Online S.L.