



## **PAC DESARROLLO: APLICACIÓN WEB**

Módulo 06: Desarrollo Web en Entorno Cliente

**Componentes descritos:**

Pruebas realizadas sobre la aplicación web del juego piedra-papel-tijera

Adrian Videira Rial



## Índice.

1. Introducción de usuario con datos no válidos. ....	3
1.1 Campo nombre vacío.....	3
1.2 Nombre muy corto o comienza por un número. ....	4
2. Introducción de cantidad de partidas con datos no válidos. ....	5
3. Acceso a la aplicación con datos válidos.....	7
4. Seleccionar una de las opciones y jugar al menos 5 partidas. ....	8
5. Pulsa el botón RESET y juega al menos 3 partidas. ....	9

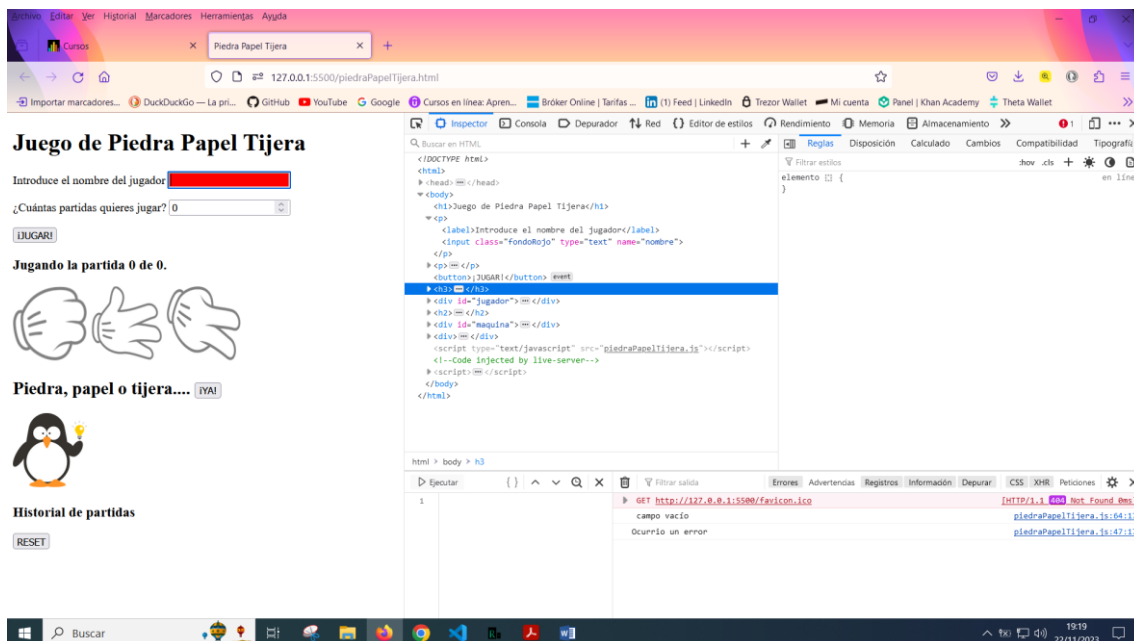
# 1. Introducción de usuario con datos no válidos.

En este apartado, se han introducido restricciones para filtrar el tipo de nombres que se permiten introducir. Esto se hará mediante la función que he llamado **verificarNombre()**;

```
54 // Función para verificar Nombre introducido por el usuario.
55 function verificarNombre(){
56     // Obtenemos el texto introducido por el usuario en el primer input
57     let inputNombre = document.getElementsByTagName("input")[0].value;
58     let campoNombre = document.getElementsByTagName("input")[0];
59     //Verificamos si el campo está vacío.
60     if(inputNombre.length == 0 || inputNombre == null){
61         alert("Campo vacío");
62         // En caso de cumplir esta condición se pondrá de color rojo el campo.
63         campoNombre.classList.add("fondoRojo");
64         console.log(inputNombre + " campo vacío"); //test
65         return false;
66     }
67     //Verificamos si el campo tiene menos de 3 letras
68     else if(inputNombre.length < 4){
69         alert("El nombre es muy corto");
70         // En caso de cumplir esta condición se pondrá de color rojo el campo.
71         campoNombre.classList.add("fondoRojo");
72         console.log(inputNombre + " campo demasiado corto"); //test
73         return false;
74     }
75     //Verificamos si son números con una expresión regular.
76     else if(!/[a-z, ]/.test(inputNombre)){
77         alert("campo contiene caracteres no válidos");
78         // En caso de cumplir esta condición se pondrá de color rojo el campo.
79         campoNombre.classList.add("fondoRojo");
80         console.log(inputNombre + " campo con caracteres incorrectos"); //test
81         return false;
82     }
83     console.log(inputNombre + " nombre Ok!"); //test
84     campoNombre.classList.remove("fondoRojo"); // Eliminamos el fondo rojo ya que los datos ahora son correctos.
85     campoNombre.disabled = true; // Bloqueamos el campo una vez están introducidos los datos de forma correcta.
86     return true;
```

## 1.1 Campo nombre vacío.

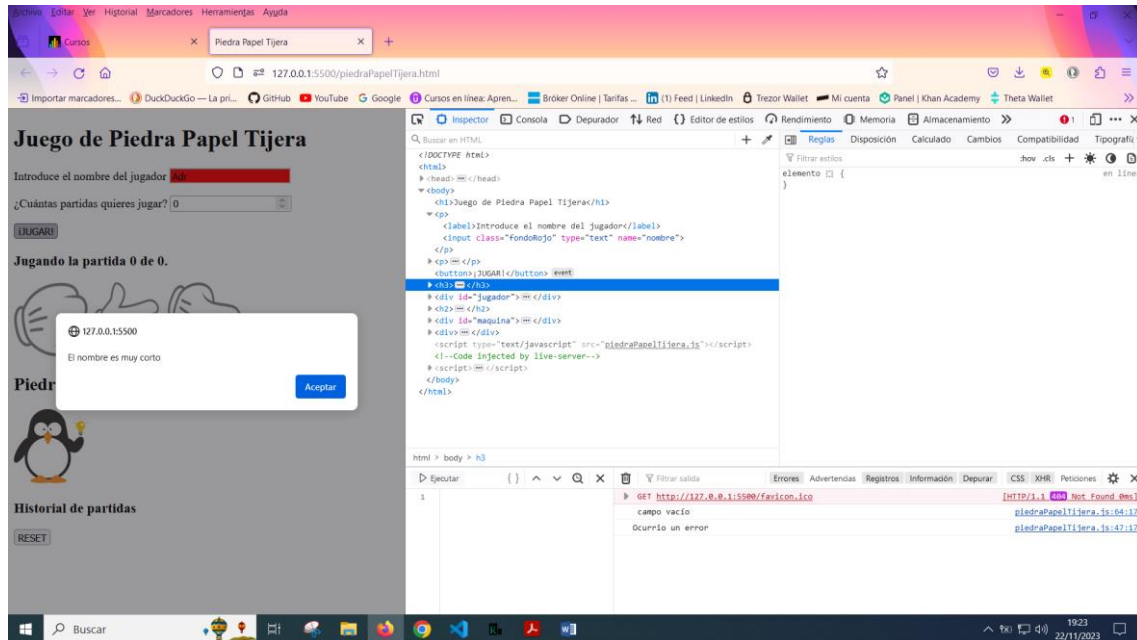
El input de "Introduce el nombre del jugador", nunca deberá estar en blanco, de lo contrario nos mostrará un error y se pondrá la casilla en rojo.



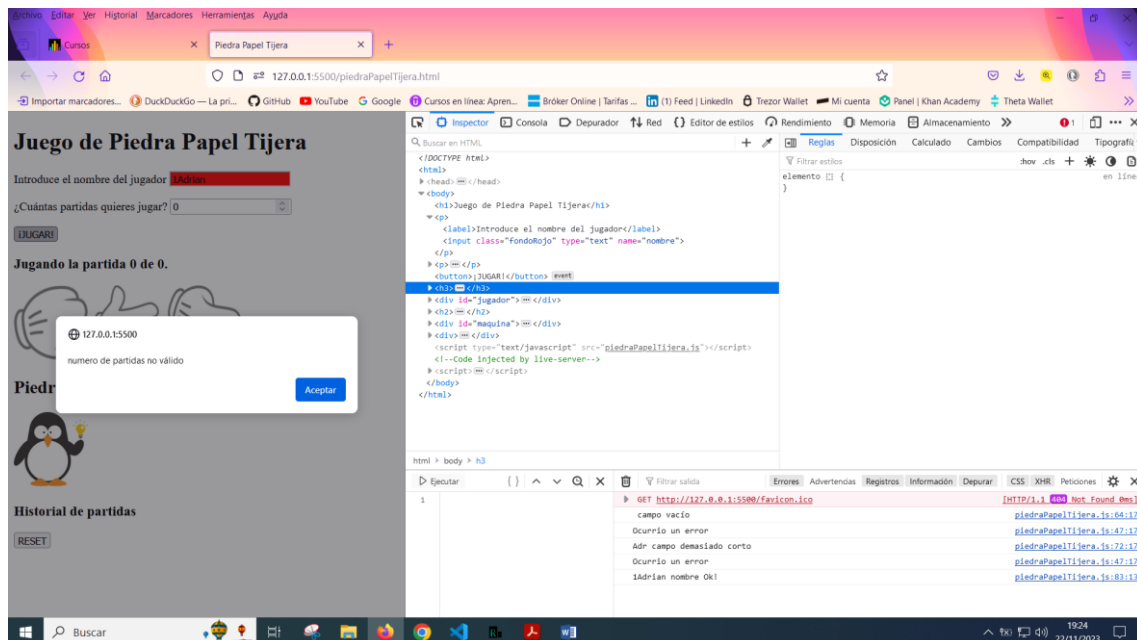
## 1.2 Nombre muy corto o comienza por un número.

El input de “Introduce el nombre del jugador”, nunca debe estar en blanco, de lo contrario nos mostrará un error y se pondrá la casilla en rojo.

- Nombre muy corto.



- Nombre que comienza por un valor numérico.



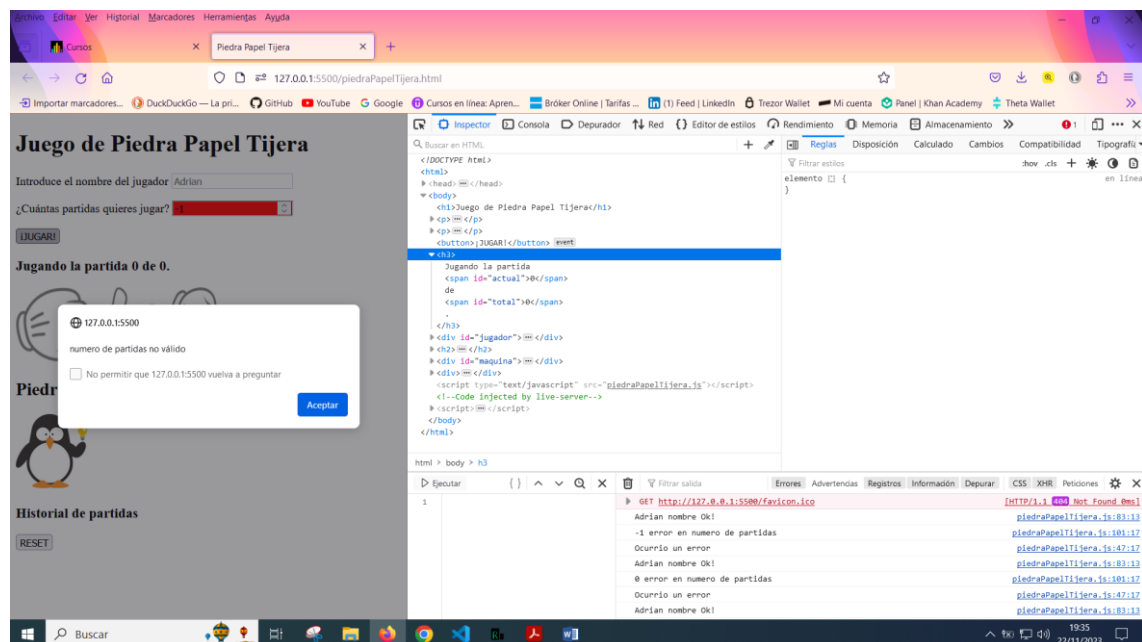
## 2. Introducción de cantidad de partidas con datos no válidos.

En este apartado, se han introducido restricciones para que no se permitan introducir partida con signo negativo ( -1,-2,-3 etc...) ni con valor 0. EL número de partidas introducido debe ser siempre mayor a 0, de lo contrario nos mostrará un error y la casilla se pondrá con fondo rojo.

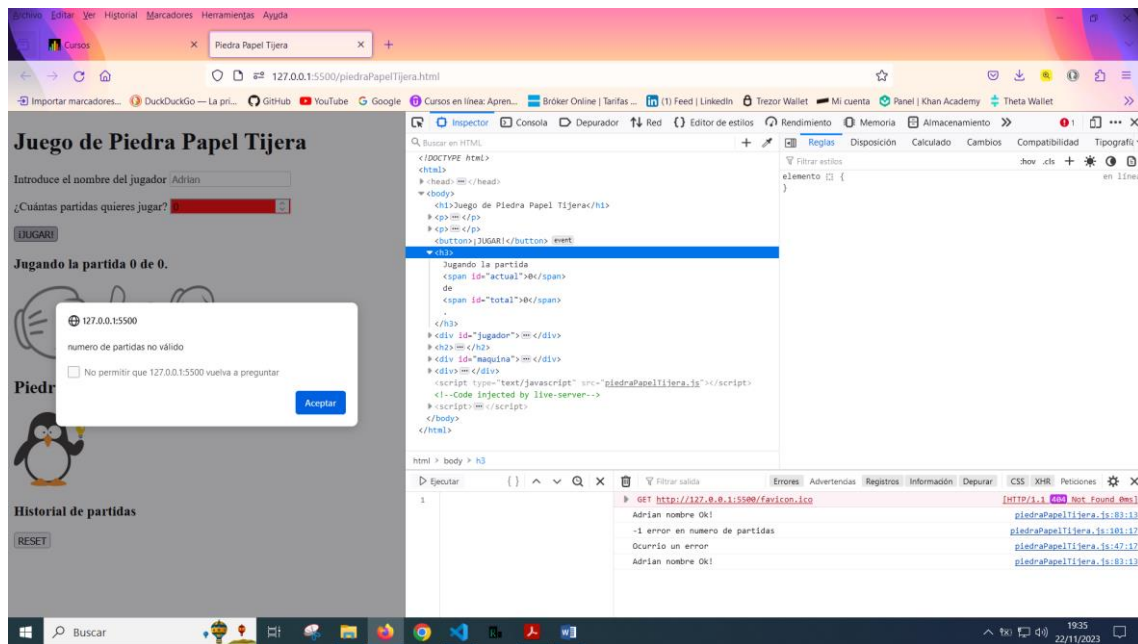
Esto se hace mediante la función **verificarPartidas()**.

```
90 // Función para verificar numero de partidas.
91 function verificarPartidas(){
92     //Obtenemos el numero de partidas introducidas por el usuario.
93     var inputPartidas = document.getElementsByTagName("input")[1].value; // las declaramos var para acceder a ellas fuera del bloque.
94     var campoPartida = document.getElementsByTagName("input")[1];
95     //console.log("inputPartidas " + inputPartidas); //test
96
97     if(inputPartidas <= 0 || inputPartidas == null){
98         alert("numero de partidas no válido");
99         // En caso de cumplir esta condición se pondrá de color rojo el campo.
100        campoPartida.classList.add("fondoRojo");
101        console.log(inputPartidas + " error en numero de partidas"); //test
102        document.getElementById("total").innerHTML = 0; // Ponemos el contador de partidas a 0;
103        return false;
104    }
105    console.log(inputPartidas + " numero de partidas Ok!"); //test
106    campoPartida.classList.remove("fondoRojo"); // Eliminamos el fondo rojo ya que los datos ahora son correctos.
107    campoPartida.disabled = true;
108    document.getElementById("total").innerHTML = inputPartidas; // Actualizamos numero total de partidas en el contador.
109    return true;
110 }
111
```

- Número de partidas negativo.

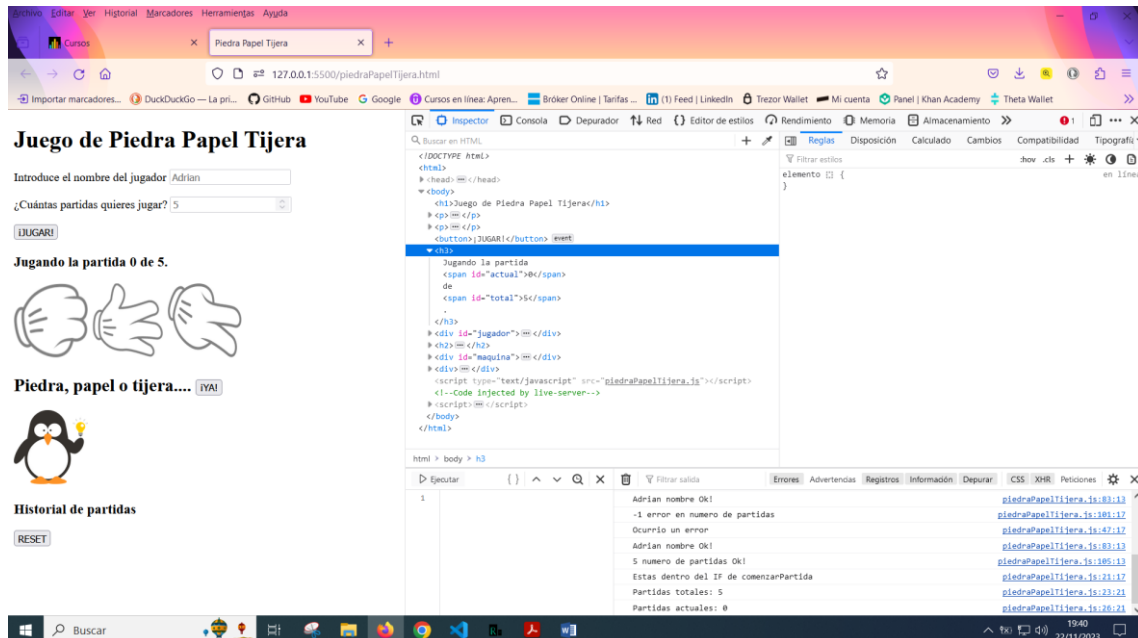


- Número de partidas igual a 0.



### 3. Acceso a la aplicación con datos válidos.

Una vez hayamos introducido el nombre del usuario y el número de partidas de forma correcta, no se mostrará ningún error, y nos dejará comenzar la partida, seleccionando la jugada que deseamos realizar. Hasta entonces esta parte del juego estará bloqueada mientras no introduzcamos unos datos válidos.



Como podemos observar, se carga automáticamente en la pantalla el número total de partidas que hemos seleccionado y se bloquean los inputs del nombre y el número de partidas.

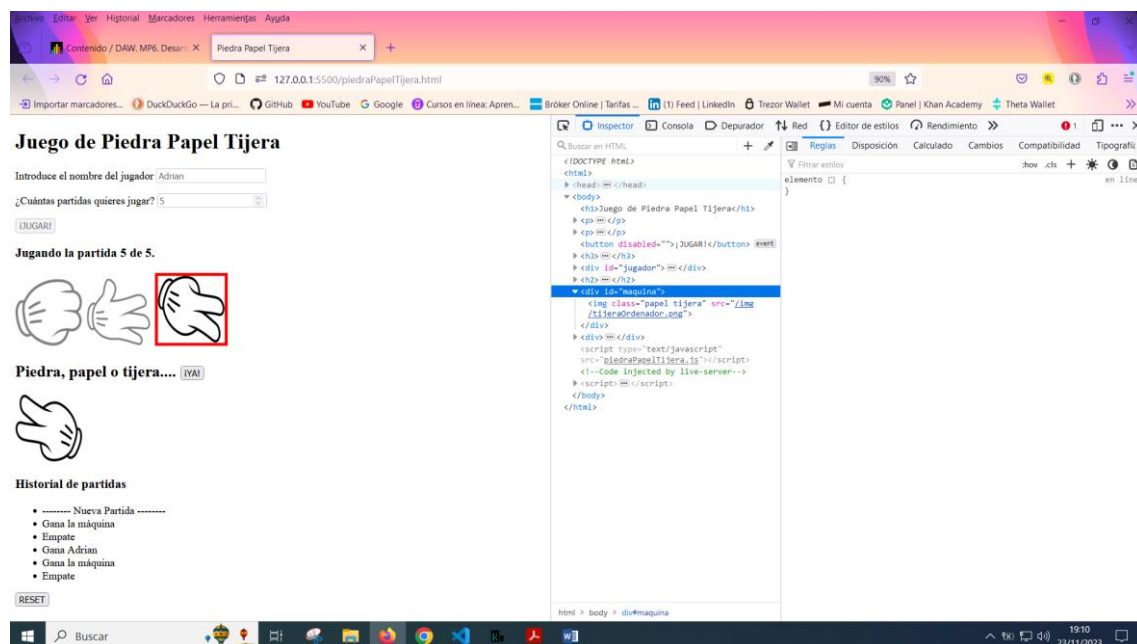


## 4. Seleccionar una de las opciones y jugar al menos 5 partidas.

Una vez introducido correctamente el nombre y las partidas a jugar, se nos permite pulsar el botón JUGAR y se entrará en la dinámica del juego, que es la siguiente:

- En primer lugar, el jugador selecciona una jugada de las tres disponibles (piedra, papel, tijera), mediante el método **seleccionJugadaUsuario()**, esto lo hará pulsando sobre una de las imágenes. Una vez pulsada la imagen esta se marcará con un borde de color rojo.
- En segundo lugar, pulsará el botón de YA, esto activa el método **seleccionMaquina()** que selecciona de forma aleatoria la jugada a la máquina.
- A continuación, se comparará la elección del usuario y la máquina, mediante el método **resultado()** y se mostrará en pantalla dicho resultado.
- Este bucle se repetirá tantas veces como partidas se hayan seleccionado y se añadirá el resultado obtenido uno detrás de otro en nuestra página web mediante una lista.

En la imagen del ejemplo, se muestra una partida de 5 juegos ya acabada con el historial de resultados obtenidos.

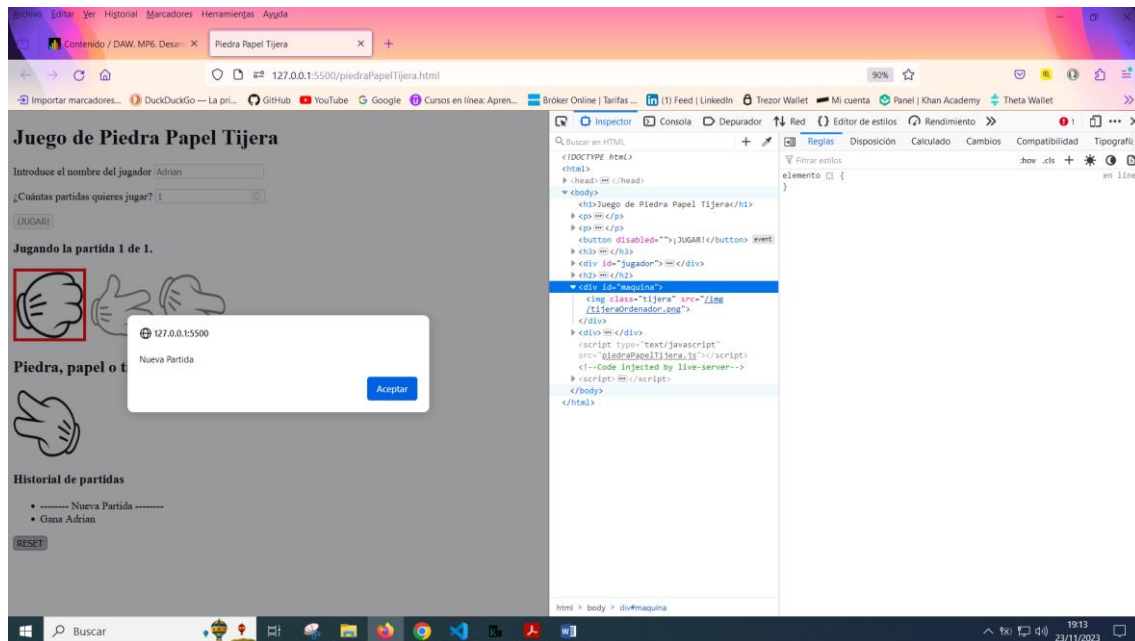


**5. Pulsa el botón RESET y juega al menos 3 partidas.**

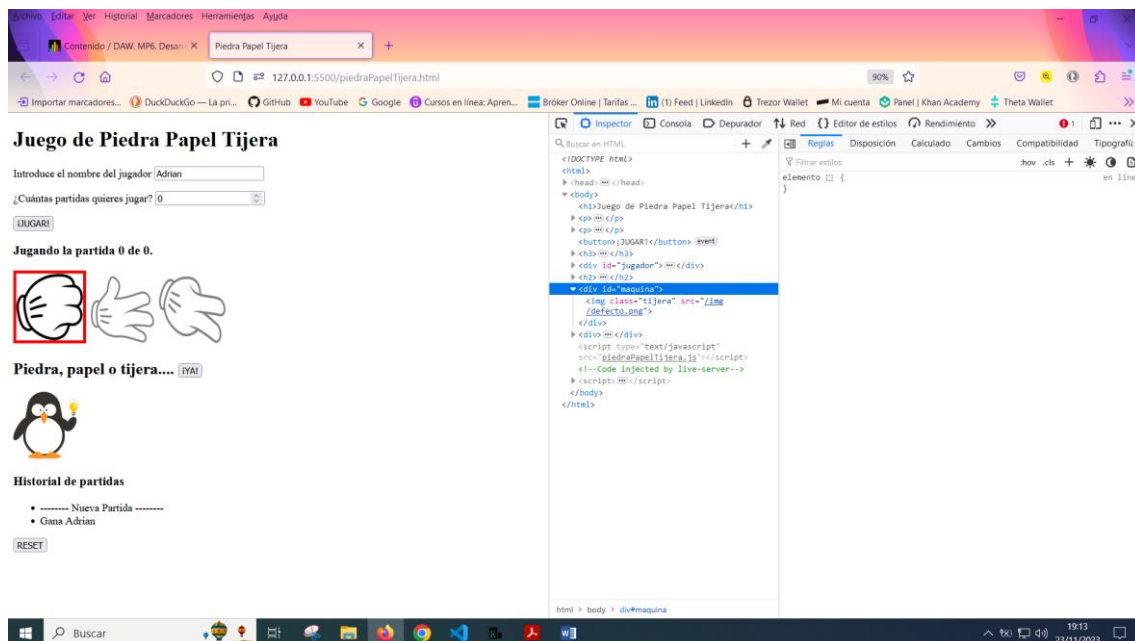
Una vez hemos finalizado todas las rondas de la partida, se activará el botón RESET, el cual, nos permitirá realizar otra partida.

Una vez pulsemos este botón, nos saltará un mensaje de aviso con el siguiente mensaje: “Nueva Partida”. También pondrá los marcadores a 0 y restaurará la imagen por defecto de la máquina.

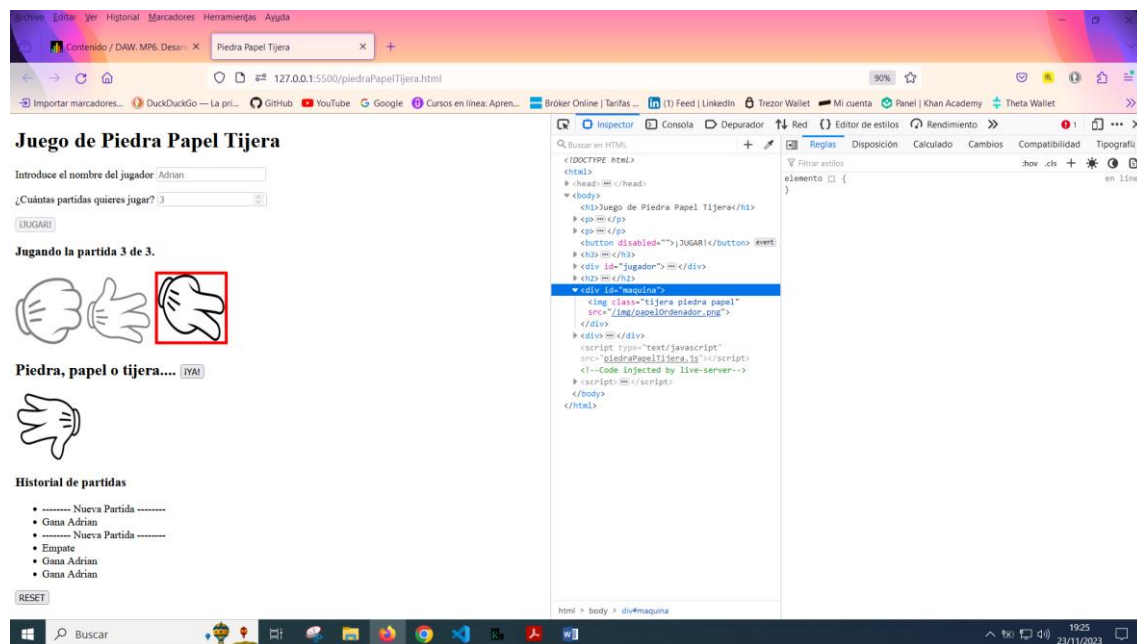
El historial de la partida anterior se mantendrá.



Vemos que después de pulsar el botón y salir el mensaje de aviso, el marcador se reinicia a 0 y el historial se mantiene.



Los resultados de la nueva partida se irán añadiendo en el historial de nuestra aplicación.



En caso de pulsar el botón RESET antes de finalizar la partida, nos avisara de que no es posible, y de que debemos finalizar todas las tiradas antes de poder usarlo.

