

ACTIVIDAD DESARROLLO DESARROLLO WEB EN ENTORNO CLIENTE

**SARAI OLCINA GONZÁLEZ
ABRIL 2021**

INTRODUCCIÓN

Se ha creado una aplicación web basada en una máquina tragaperras.

El usuario tiene la opción de insertar monedas, leer las instrucciones, tirar o salir. Si selecciona la opción de instrucciones, aparece un cuadro de texto con las mismas. Si selecciona la opción de insertar monedas, aparecerá una alerta que le solicite ingresar el número de monedas.

Una vez insertadas las monedas, puede comenzar a jugar a través del botón Tirar. Cuando le da al botón Tirar, aparece un contador de monedas, así como un mensaje por cada tirada en la que ha ganado un premio donde se le especifica el número de monedas ganadas. Este mensaje aparecerá siempre en relación a las últimas 4 tiradas con premio.

Si selecciona la opción de Salir, le devolverá el número de monedas pendientes.

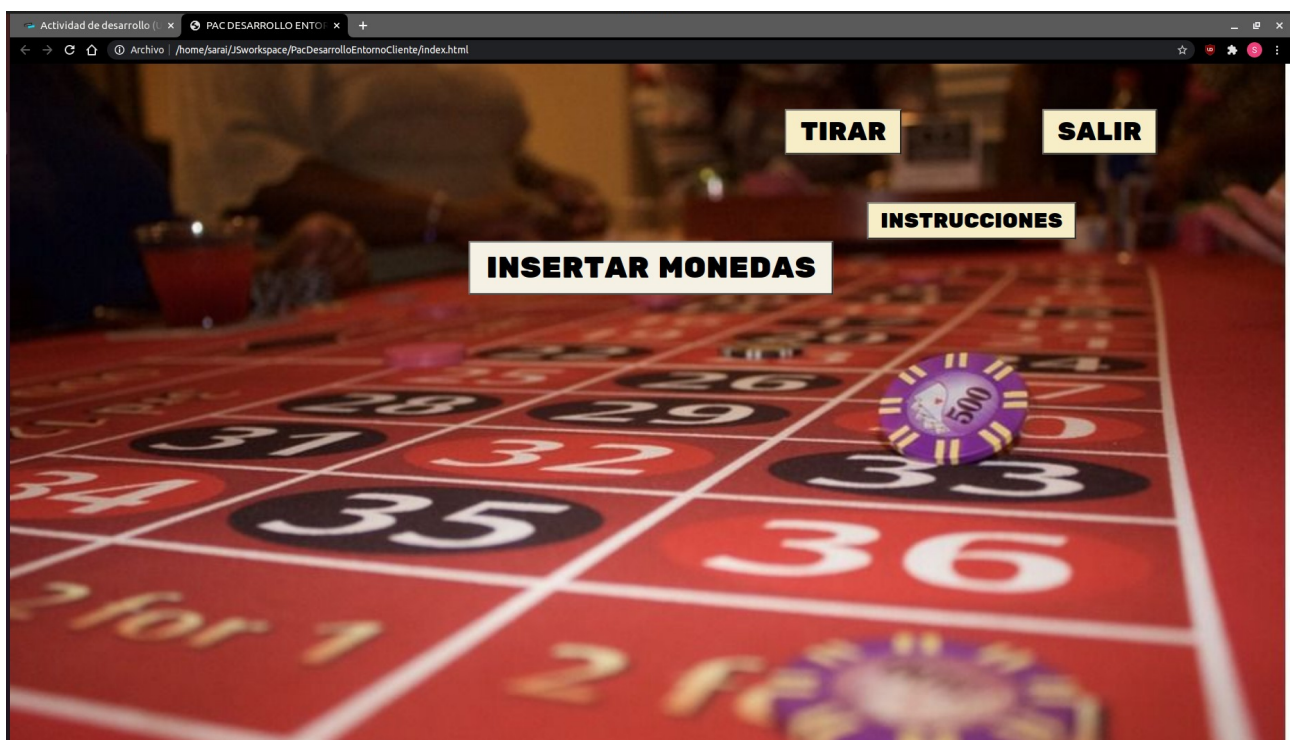


Figura 1: Inicio Aplicación

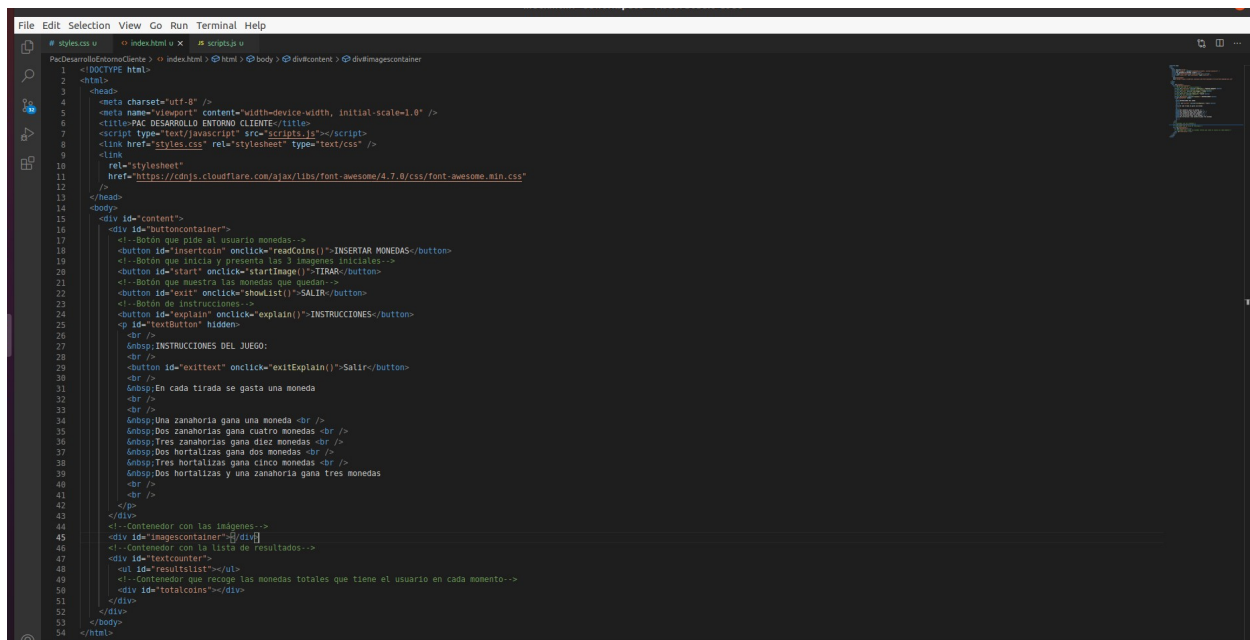
En caso de que el usuario decida seleccionar las opciones de Tirar o Salir antes de Insertar monedas, se devuelve un mensaje de alerta que le pide que inserte primero las monedas.

DESCRIPCIÓN DEL CÓDIGO

Para la elaboración de la práctica se han creado tres archivos: un documento html (index.html) que recoge todos los elementos que se van a utilizar en la aplicación; un archivo .css (styles.css) que le dará estilo a estos elementos; y un archivo .js (scripts.js), que dará funcionalidad a la misma.

Empezando con el documento html, en primer lugar empezamos importando los archivos y enlaces que vamos a utilizar desde el head.

Seguidamente, en el body, se muestran en primer lugar los botones de “Insertar Monedas”, “Tirar”, “Salir” e “Instrucciones”. El div que contiene las instrucciones, de primeras está oculto, solo cuando se seleccione el botón correspondiente aparecerá en pantalla (veremos cómo se hace en el archivo script.js).



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6     <title>PAC DESARROLLO ENTORNO CLIENTE</title>
7     <script type="text/javascript" src="scripts.js"></script>
8     <link href="styles.css" rel="stylesheet" type="text/css" />
9     <link
10       rel="stylesheet"
11       href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css"
12     />
13   </head>
14   <body>
15     <div id="content">
16       <div id="buttoncontainer">
17         <!-- Botón que pide al usuario monedas -->
18         <button id="insertcoin" onclick="readCoins()">INSERTAR MONEDAS</button>
19         <!-- Botón que inicia y presenta las 3 imágenes iniciales -->
20         <button id="start" onclick="startImage()">TIRAR</button>
21         <!-- Botón que muestra las monedas que quedan -->
22         <button id="exit" onclick="showList()">SALIR</button>
23         <!-- Botón de instrucciones -->
24         <button id="explain" onclick="explain()">INSTRUCCIONES</button>
25         <p id="textButton" hidden>
26           &nbsp;&nbsp;&nbsp;INSTRUCCIONES DEL JUEGO:
27         </p>
28         <button id="exittext" onclick="exitExplain()">Salir</button>
29         <br />
30         &nbsp;&nbsp;&nbsp;En cada tirada se gasta una moneda
31         <br />
32         &nbsp;&nbsp;&nbsp;Una zanahoria gana una moneda <br />
33         &nbsp;&nbsp;&nbsp;Dos zanahorias gana cuatro monedas <br />
34         &nbsp;&nbsp;&nbsp;Tres zanahorias gana diez monedas <br />
35         &nbsp;&nbsp;&nbsp;Cuatro zanahorias gana dos monedas <br />
36         &nbsp;&nbsp;&nbsp;Tres hortalizas gana cinco monedas <br />
37         &nbsp;&nbsp;&nbsp;Dos hortalizas y una zanahoria gana tres monedas
38         <br />
39       </div>
40     </div>
41     <!-- Contenedor con las imágenes -->
42     <div id="imagescontainer"></div>
43     <!-- Contenedor con la lista de resultados -->
44     <div id="resultlist"></div>
45     <div id="totalcoins"></div>
46   </body>
47 </html>
```

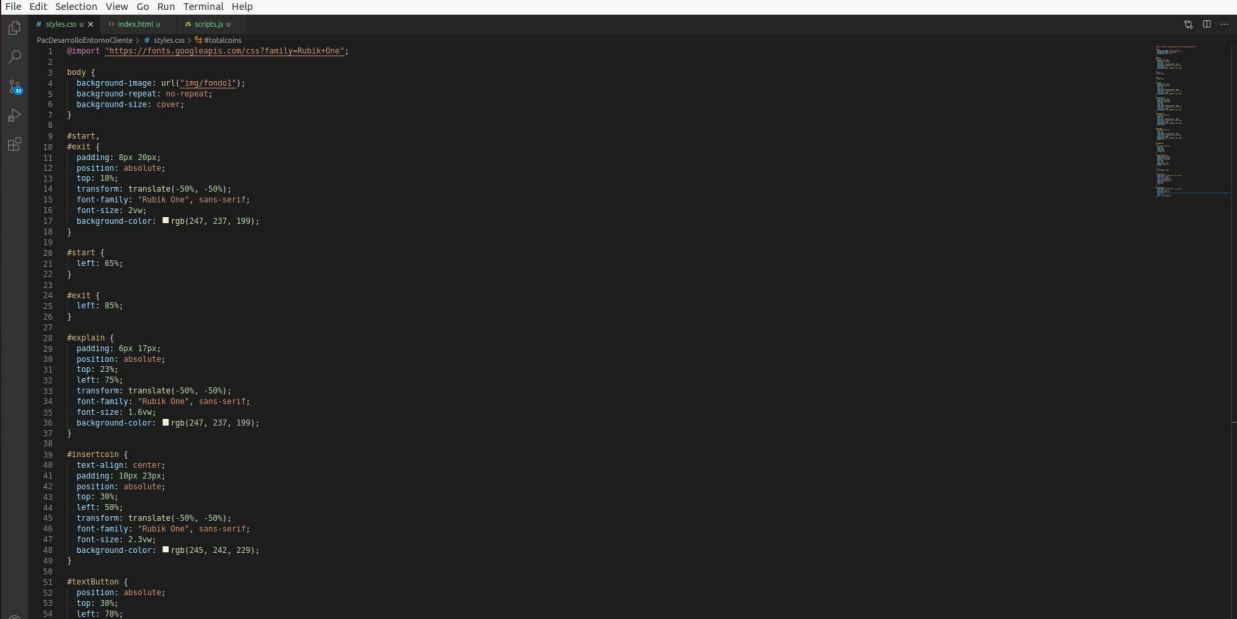
Figura 2: Código index.html

A continuación tenemos el contenedor de imágenes, que en un inicio está vacío pero que más tarde iremos rellenando con las imágenes aleatorias.

También tenemos el contenedor que muestra los resultados. Está formado de una una lista no ordenada y un elemento que muestra el total de monedas restantes en cada tirada. Igual que con el contenedor de imágenes, la lista la iremos rellenando de elementos conforme se vayan realizando tiradas, todo con funciones que se encuentran en el archivo script.js.

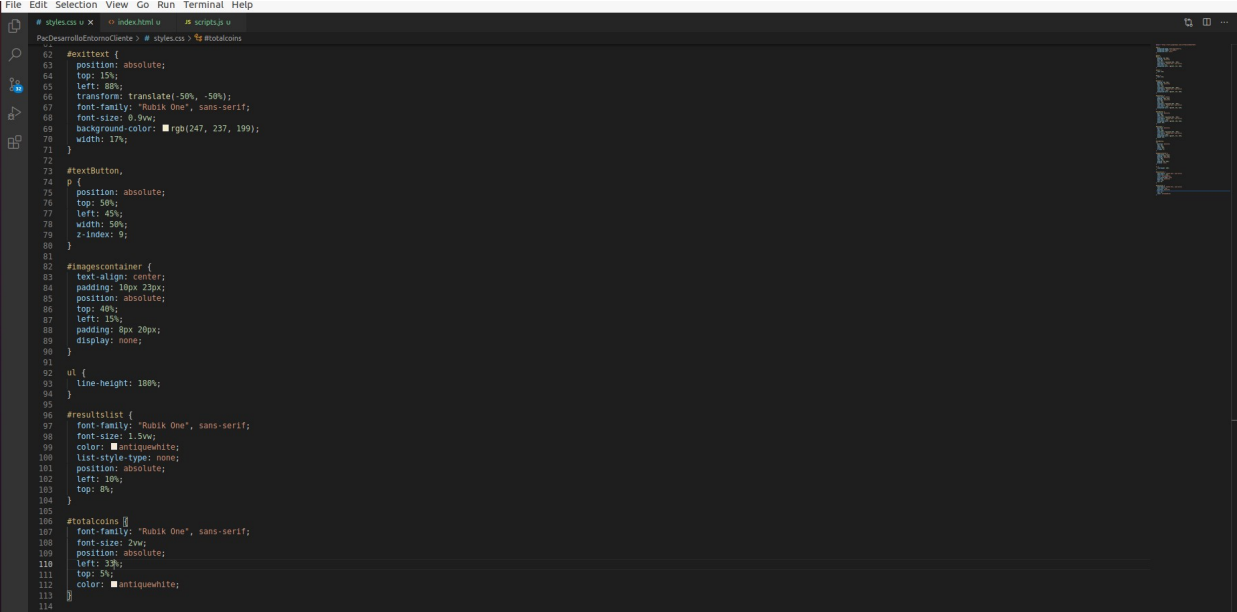
En cuanto a los estilos, en el archivo styles.css hemos elegido la misma fuente para todos los botones y textos, así como una imagen para el fondo de todo el documento.

Los elementos irán apareciendo según se vayan seleccionando los botones correspondientes, por eso, en el archivo html aparecían algunos como “hidden” o aparecen en el archivo .css con el atributo “display:none”.



```
1 @import "https://fonts.googleapis.com/css?family=Rubik+One";
2
3 body {
4   background-image: url("img/fondol1");
5   background-repeat: no-repeat;
6   background-size: cover;
7 }
8
9 #start,
10 #exit {
11   padding: 8px 20px;
12   position: absolute;
13   top: 10%;
14   transform: translate(-50%, -50%);
15   font-family: "Rubik One", sans-serif;
16   font-size: 2vw;
17   background-color: #247, 237, 199;
18 }
19
20 #start {
21   left: 65%;
22 }
23
24 #exit {
25   left: 85%;
26 }
27
28 #explain {
29   padding: 8px 17px;
30   position: absolute;
31   top: 23%;
32   left: 73%;
33   transform: translate(-50%, -50%);
34   font-family: "Rubik One", sans-serif;
35   font-size: 1.6vw;
36   background-color: #247, 237, 199;
37 }
38
39 #insertcoin {
40   text-align: center;
41   padding: 10px 23px;
42   position: absolute;
43   top: 50%;
44   left: 50%;
45   transform: translate(-50%, -50%);
46   font-family: "Rubik One", sans-serif;
47   font-size: 2.3vw;
48   background-color: #245, 242, 229;
49 }
50
51 #textButton {
52   position: absolute;
53   top: 50%;
54   left: 78%;
```

Figura 3: Código styles.css I



```
55   top: 50%;
56   left: 78%;
57 }
58
59 #textButton,
60 p {
61   position: absolute;
62   top: 50%;
63   left: 45%;
64   width: 50%;
65   z-index: 9;
66 }
67
68 #imagescontainer {
69   text-align: center;
70   padding: 10px 23px;
71   position: absolute;
72   top: 40%;
73   left: 15%;
74   padding: 8px 20px;
75   display: none;
76 }
77
78 ul {
79   list-style-type: none;
80   line-height: 180%;
81 }
82
83 #resultlist {
84   font-family: "Rubik One", sans-serif;
85   font-size: 1.5vw;
86   color: #antiquewhite;
87   list-style-type: none;
88   position: absolute;
89   left: 10%;
90   top: 8%;
91 }
92
93 #totalcoins {
94   font-family: "Rubik One", sans-serif;
95   font-size: 2vw;
96   position: absolute;
97   left: 13%;
98   top: 5%;
99   color: #antiquewhite;
100 }
```

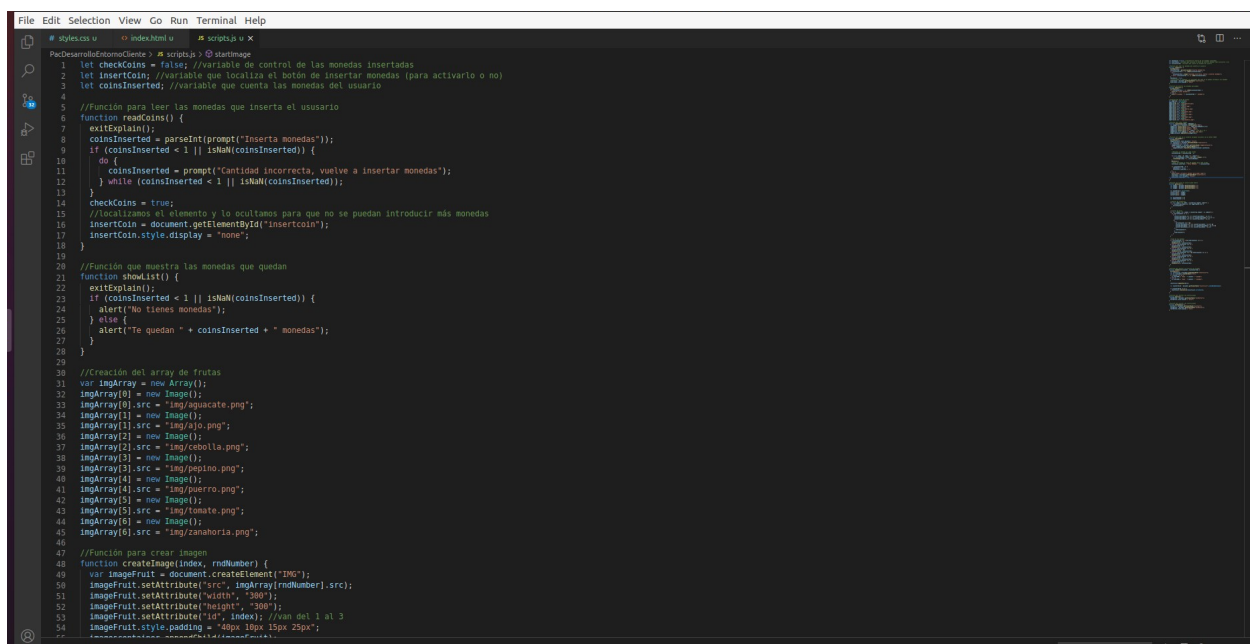
Figura 4: Código styles.css II

En el archivo html se ha decidido poner identificadores a los elementos, pues así es más sencillo poder darles un estilo único aunque se trate del mismo elemento (un div, un button, p...).

También se ha decidido poner los identificadores para luego poder seleccionar cada elemento concreto con una o varias funciones desde el archivo scripts.js.

En el archivo .js se van a recoger todas las funciones de la aplicación.

Ya sean funciones para hacer aparecer y desaparecer en pantalla los elementos del documento html como para generar aleatoriedad y resultados en las imágenes.



```
1 let checkCoins = false; //variable de control de las monedas insertadas
2 let insertCoin; //variable que localiza el botón de insertar monedas (para activarlo o no)
3 let coinsInserted; //variable que cuenta las monedas del usuario
4
5 //Función para leer las monedas que inserta el usuario
6 function readCoins() {
7   exitExplain();
8   coinsInserted = parseInt(prompt("Inserta monedas"));
9   if (coinsInserted < 1 || isNaN(coinsInserted)) {
10     coinsInserted = prompt("Cantidad incorrecta, vuelve a insertar monedas");
11     while (coinsInserted < 1 || isNaN(coinsInserted));
12   }
13   checkCoins = true;
14   //localizamos el elemento y lo ocultamos para que no se puedan introducir más monedas
15   insertCoin = document.getElementById("insertCoin");
16   insertCoin.style.display = "none";
17 }
18
19 //Función que muestra las monedas que quedan
20 function showList() {
21   exitExplain();
22   if (coinsInserted < 1 || isNaN(coinsInserted)) {
23     alert("No tienes monedas");
24   } else {
25     alert("Te quedan " + coinsInserted + " monedas");
26   }
27 }
28
29 //Creación del array de frutas
30 var imgArray = new Array();
31 imgArray[0] = new Image();
32 imgArray[0].src = "img/squacate.png";
33 imgArray[1] = new Image();
34 imgArray[1].src = "img/sjo.png";
35 imgArray[2] = new Image();
36 imgArray[2].src = "img/cabolla.png";
37 imgArray[3] = new Image();
38 imgArray[3].src = "img/pesino.png";
39 imgArray[4] = new Image();
40 imgArray[4].src = "img/puerro.png";
41 imgArray[5] = new Image();
42 imgArray[5].src = "img/tomate.png";
43 imgArray[6] = new Image();
44 imgArray[6].src = "img/zanahoria.png";
45
46 //Función para crear imagen
47 function createImage(index, randomNumber) {
48   var imageFruit = document.createElement("img");
49   imageFruit.setAttribute("src", imgArray[randomNumber].src);
50   imageFruit.setAttribute("width", "300");
51   imageFruit.setAttribute("height", "300");
52   imageFruit.setAttribute("id", index); //van del 1 al 3
53   imageFruit.style.padding = "40px 10px 10px 20px";
54   imageFruit.innerHTML = "Imagen de la fruta";
55 }
```

Figura 5: Código script.js I

Comenzamos declarando algunas variables de control que nos van a servir para activar y desactivar elementos y para ver en qué momentos podemos retornar o no los resultados, cuadros de diálogo, botones...

A continuación, la primera función que tenemos es la que lee las monedas que inserta el usuario (veremos que en muchas funciones se llama a otra función exitExplain(), esto es para cerrar el diálogo de las instrucciones en caso de que el usuario hubiera decidido leer las instrucciones antes de pulsar cualquier otro botón).

Una vez el usuario introduce las monedas correctas (valores >1 y numéricos), ocultamos esta opción para que no pueda volver a insertar monedas hasta que se agoten.

La función showList() devuelve con un alert el número de monedas que le quedan al usuario en cualquier momento de la partida.

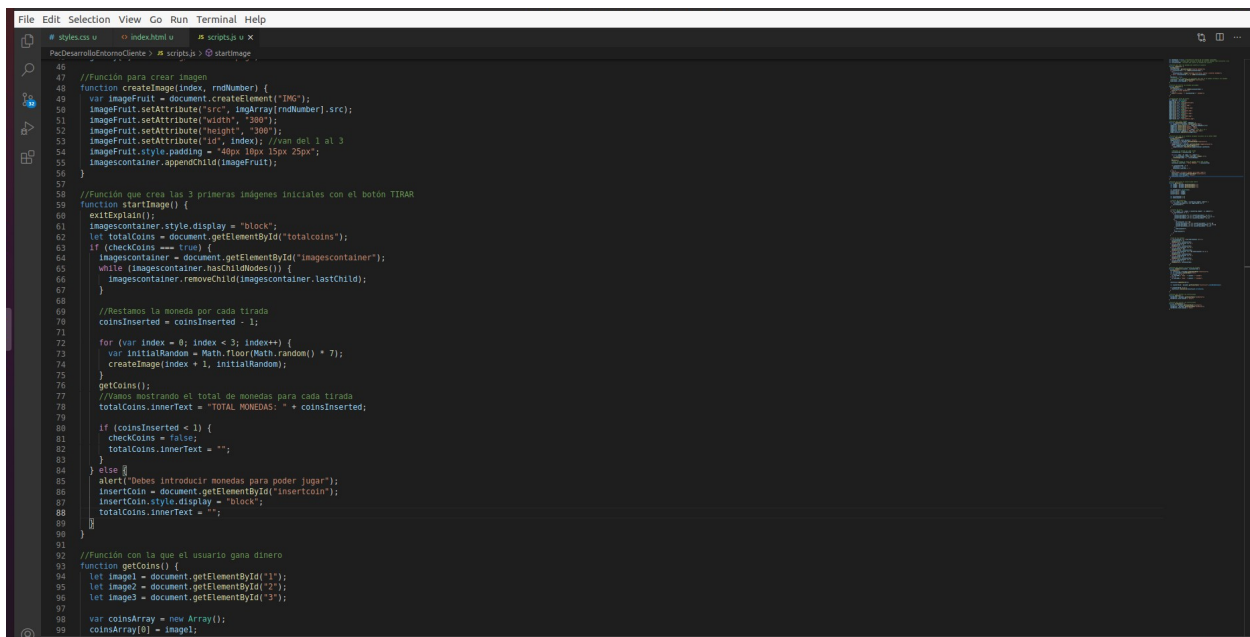


Figura 6: Código scripts.js II

A continuación creamos el array de imágenes y la función que crea la imagen. Usamos para ello la función `createElement` y le vamos asignando atributos al elemento creado (en este caso las características de la imagen). Por último, añadimos el elemento al contenedor padre.

La función que crea las imágenes en su conjunto, lo primero que hace es mostrar el elemento `div` padre (que de momento está vacío), seguidamente, a través de un bucle `for` de 3 iteraciones, llama a la función anterior para crear cada imagen. Todo esto una vez se ha seleccionado el botón "Tirar".

Esta función también se utiliza para descontar una moneda por cada tirada y mostrarlo por pantalla, así como para llamar a la función (`getCoins()`) que se va a encargar de almacenar los resultados de cada tirada y que generará en pantalla el listado de monedas ganadas en cada tirada.

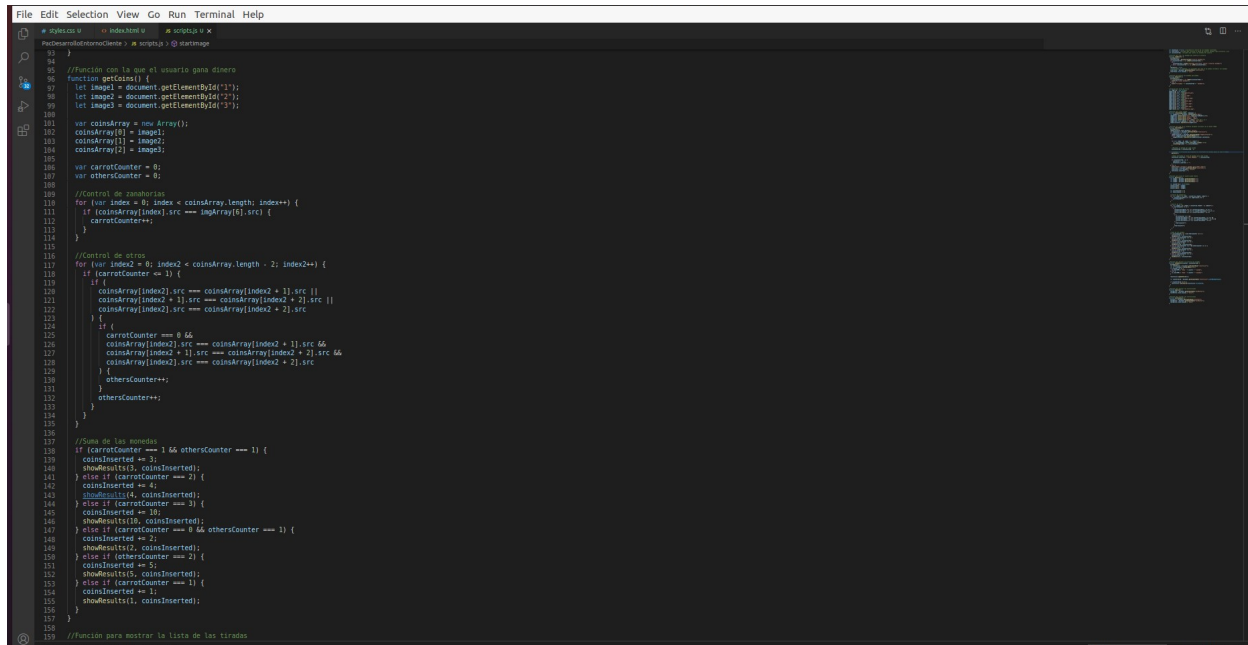
Esta función, `getCoins()`, como decíamos, va a ser la que nos va a controlar el número de monedas ganadas y perdidas en cada tirada y se encargará también de modificar la variable de las monedas iniciales que introdujo el usuario.

En primer lugar, se crea un nuevo array vacío y se rellena con las imágenes que se han creado para cada tirada (llamamos a cada imagen con `getElement`). De esta forma, tenemos un array "comodín" que irá cambiando en cada tirada (recordemos que llamábamos a esta función cada vez que el usuario selecciona el botón "Tirar" en la función `startImage()`).

A continuación, comparamos este nuevo array con el array de imágenes iniciales que creamos.

Si la imagen es la misma que la de la zanahoria (que ocupa la posición 6 en el array inicial), pondremos unas condiciones, y si no es igual, pondremos otras.

Para asignar las condiciones y por lo tanto sumar las monedas, hemos utilizado dos contadores diferentes y diversos condicionales.



```
File Edit Selection View Go Run Terminal Help
//Función con la que el usuario gana dinero
function getCoin() {
  let img1 = document.getElementById("1");
  let img2 = document.getElementById("2");
  let img3 = document.getElementById("3");

  let coinsArray = new Array();
  coinsArray[0] = img1;
  coinsArray[1] = img2;
  coinsArray[2] = img3;

  let carrotCounter = 0;
  let othersCounter = 0;

  //Control de monedas
  for (let index = 0; index < coinsArray.length; index++) {
    if (coinsArray[index].src === imgArray[0].src) {
      carrotCounter++;
    }
  }

  //Control de otros
  for (let index2 = 0; index2 < coinsArray.length - 2; index2++) {
    if (carrotCounter === 1) {
      if (
        coinsArray[index2].src === coinsArray[index2 + 1].src ||
        coinsArray[index2 + 1].src === coinsArray[index2 + 2].src ||
        coinsArray[index2].src === coinsArray[index2 + 2].src
      ) {
        if (
          carrotCounter === 0 &&
          coinsArray[index2].src === coinsArray[index2 + 1].src &&
          coinsArray[index2 + 1].src === coinsArray[index2 + 2].src &&
          coinsArray[index2].src === coinsArray[index2 + 2].src
        ) {
          othersCounter++;
        }
        othersCounter++;
      }
    }
  }

  //Suma de las monedas
  if (carrotCounter === 1 && othersCounter === 1) {
    coinsInserted = 5;
    showResult(3, coinsInserted);
  } else if (carrotCounter === 2) {
    coinsInserted = 4;
    showResult(4, coinsInserted);
  } else if (carrotCounter === 3) {
    coinsInserted = 30;
    showResult(5, coinsInserted);
  } else if (carrotCounter === 0 && othersCounter === 1) {
    coinsInserted = 2;
    showResult(2, coinsInserted);
  } else if (othersCounter === 2) {
    coinsInserted = 5;
    showResult(5, coinsInserted);
  } else if (carrotCounter === 1) {
    coinsInserted = 1;
    showResult(1, coinsInserted);
  }
}

//Función para mostrar la lista de las tiradas
```

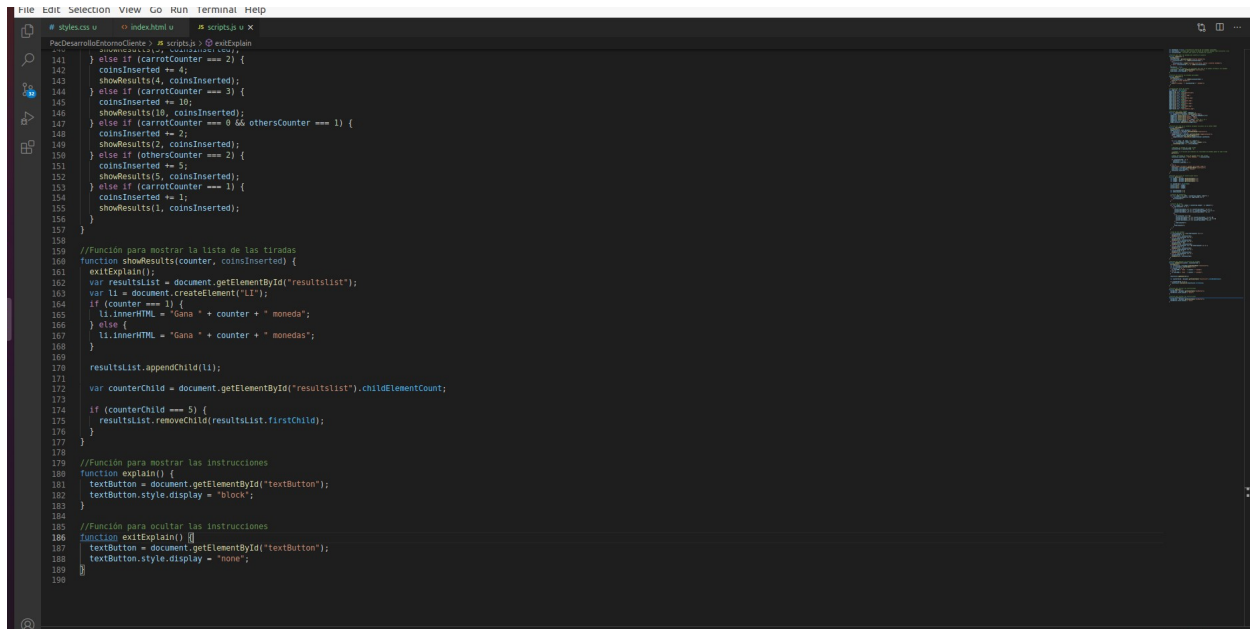
Figura 7: Código script.js III

También tenemos la función para crear los elementos de la lista que en cada tirada aparecerá por pantalla (solo si el usuario ha ganado monedas, si no, no aparece nada). Nuevamente, creamos el elemento “li” y lo añadimos con appendChild al contenedor padre.

En este caso queremos que solo se muestren por pantalla 4 líneas, es decir, 4 elementos li, por lo que a través de la función childElementCount y removeFirstChild, vamos añadiendo cada tirada en la que el usuario gana algún premio y por lo tanto, eliminando la primera.

Es decir, se añade el último elemento con appendChild pero se elimina el primero con removeFirstChild.

Por último, simplemente nos quedan las funciones que muestran y ocultan el contenedor de instrucciones.



```
110 } else if (carrotCounter === 2) {
111   coinsInserted = 4;
112   showResults(4, coinsInserted);
113 } else if (carrotCounter === 3) {
114   coinsInserted = 10;
115   showResults(10, coinsInserted);
116 } else if (carrotCounter === 0 && othersCounter === 1) {
117   coinsInserted = 2;
118   showResults(2, coinsInserted);
119 } else if (othersCounter === 2) {
120   coinsInserted = 5;
121   showResults(5, coinsInserted);
122 } else if (carrotCounter === 1) {
123   coinsInserted = 1;
124   showResults(1, coinsInserted);
125 }
126 }
127
128 //Función para mostrar la lista de las tiradas
129 function showResults(counter, coinsInserted) {
130   exitExplain();
131   var resultList = document.getElementById("resultslist");
132   var li = document.createElement("li");
133   if (counter === 1) {
134     li.innerHTML = "Gana " + counter + " monedas";
135   } else {
136     li.innerHTML = "Gana " + counter + " monedas";
137   }
138   resultList.appendChild(li);
139   var counterChild = document.getElementById("resultslist").childElementCount;
140   if (counterChild === 5) {
141     resultList.removeChild(resultList.firstChild);
142   }
143 }
144
145 //Función para mostrar las instrucciones
146 function explain() {
147   textButton = document.getElementById("textButton");
148   textButton.style.display = "block";
149 }
150
151 //Función para ocultar las instrucciones
152 function exitExplain() {
153   textButton = document.getElementById("textButton");
154   textButton.style.display = "none";
155 }
```

Figura 8: Código script.js IV

BIBLIOGRAFÍA

- Material didáctico asignatura
- <https://www.javatpoint.com/>
- <https://www.mclibre.org/>
- <https://www.w3schools.com/>
- <https://es.wikipedia.org/>