

Vamos a desarrollar el juego del “Ahorcado” para poder ejecutarlo en un navegador web. Está será la pantalla inicial del juego:



Al acceder a la página, se cargará una palabra al azar de un array de palabras, y se mostrará un guion por cada letra de la palabra. Para jugar, podemos hacer “click” sobre una de las letras del abecedario o pulsando la letra en el teclado. Si la letra se encuentra entre las letras que componen la palabra a adivinar, se sustituirán los guiones por la letra elegida, en todos los lugares en los que se encuentre en la palabra a adivinar:



Si la letra no existe, se añadirá al listado de “Letras erróneas” y se cambiará la imagen para mostrar el error:



El juego terminará cuando todas las letras sean descubiertas, o cuando el usuario cometa tantos errores que la imagen se complete. En ambos casos, se mostrará un mensaje de alerta para informar al jugador.

Deberás completar las funciones que están vacías en el documento HTML para poder jugar.

Ejercicio 1: **cargarLetras()**

Añadirá un elemento `` por cada letra del array global **alfabeto** (por ejemplo: `A`). Estos elementos colgarán del elemento `<ul id="letras">` y cuando se haga *click* sobre cada uno de ellos, se ejecutará la función **letraRaton**.

(1 punto)

Ejercicio 2: **palabraAzar()**

Cogerá una palabra al azar de entre las palabras del array global **palabras**, y devolverá un array que tendrá en cada posición una letra de esa palabra. Ejemplo:

VAINILLA se debe devolver como `['V', 'A', 'I', 'N', 'I', 'L', 'L', 'A']`

Puedes usar la función **numeroAzar** como ayuda. Debes pasarle la cantidad de números distintos que necesitas.

(1 punto)

Ejercicio 3: **valorInicialJuego()**

Escribe en todas las posiciones del array **palabraJuego** los caracteres “- ” (guion y espacio en blanco).

(1 punto)

Ejercicio 4: **mostrarPalabraJuego()**

Escribe en el elemento *palabraAciertos* los caracteres del array **palabraJuego**.

(1 punto)

Ejercicio 5: **letraRaton()**

Es la función que maneja los eventos de ratón sobre los `` de las letras. Recogerá del elemento pulsado la letra (si pulsamos sobre `B`, recogerá la B), y llamará a **comprobarLetra** pasándole como parámetro la letra recogida.

(0,5 puntos)

Ejercicio 6: **actualizarAciertos(letra)**

Comprueba si la letra que recibe por parámetro está en la variable array **palabraAdivinar**. Actualizará el array **palabraJuego**, poniendo el valor **letra** seguido de espacio en blanco en los lugares donde la **palabraAdivinar** tiene la letra que recibimos. Ejemplo:

letra: A

palabraAdivinar: ['V', 'A', 'I', 'N', 'I', 'L', 'L', 'A']

palabraJuego: ['- ', 'A ', '- ', '- ', '- ', '- ', '- ', 'A ']

Devolverá **true** si la letra está en la palabra a adivinar, y **false** si no es así

(1 punto)

Ejercicio 7: **actualizarPalabraJuego()**

Recoge el elemento HTML *palabraAciertos* y vacía su contenido. Después llama a la función **mostrarPalabraJuego()**.

(0,5 puntos)

Ejercicio 8: **actualizarLetrasError(letra)**

Recoge el elemento HTML *letrasErroneas* y le añade la letra pasada por parámetro. También actualiza la imagen del juego, cambiando el atributo *src* del elemento html *hangman*. Su valor deberá ser el siguiente:

Si en la variable errores tenemos un **2**, la imagen que tenemos que usar será:

<https://examenjs.000webhostapp.com/imagenes/hangman2.png>

(1 punto)

Ejercicio 9: **comprobarPartidaTerminada()**

Comprueba si hemos acertado todas las letras, mirando si quedan letras por descubrir en la variable **palabraJuego**. Si hemos acertado todas las letras, mostrará una alerta para decirle al jugador que ha ganado!

(1 punto)

Ejercicio 10: **letraTeclado()** y **letraValida(letra)**

Vamos a ampliar el programa para poder jugar desde el teclado. Añade el código necesario para poder escuchar los eventos del teclado desde la ventana. Además, hay que completar estas 2 funciones:

1. **letraTeclado()**: será la manejadora del evento. Recogerá la letra pulsada, la pasará a mayúsculas y si la letra es válida (función **letraValida** más abajo), llamará a **comprobarLetra** pasándole como parámetro la letra pulsada. Si la letra no es válida, mostrará una alerta informando al usuario.
2. **letraValida(letra)**: devolverá *true* si la letra que recibe por parámetro se encuentra en el array **alfabeto**, y *false* si no es así.

(1 punto)

Ejercicio 11: **palabraAzarAJAX**

Vamos a cambiar el funcionamiento del programa para que la palabra a adivinar sea recogida de un servidor en lugar de usar un array global, mediante una llamada AJAX. Tendremos que llamar a la URL <https://examenjs.000webhostapp.com/palabras.php>, a la que le pasaremos el parámetro *numLetras*, que contendrá el número de letras que queremos que tenga la palabra a adivinar. El servidor nos devolverá un archivo XML con este formato (para *numLetras*=5):

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<palabras>
  <palabra>ARBOL</palabra>
  <palabra>CALLE</palabra>
  <palabra>CALOR</palabra>
  <palabra>CLASE</palabra>
  <palabra>SILLA</palabra>
</palabras>
```

Debemos elegir una palabra al azar de entre las que tenemos en el XML y devolverla como un array, como en el ejercicio 2.

(1 punto)