1. Debe construir un juego para adivinar un número aleatorio pensado para el juego en 5 intentos:
2. El juego funcionará íntegramente con controles de texto, o inputs de formulario, buttons para mostrar los números a adivinar, y los elementos html necesarios de tipo acierto y error desarrollados por el alumno, y con un framework de tipo cliente para el diseño visual y feedback.
3. La gestión de la funcionalidad será creada íntegramente por lenguaje de script ( javascript ) y totalmente funcional desarrollado por el alumno.
4. APARTADO A. El sitio debe cumplir las siguientes funcionalidades y servicios:
5. Debe pedir inicialmente el nombre del jugador y recogerlo para dar un feedback personalizado. -Sólo se aceptarán caracteres alfabéticos.
6. Debe pedir un número menor (entre 1 y 10 incluidos) para establecer el primer número del rango del aleatorio que pensará el juego - Sólo se aceptarán caracteres numéricos.
7. Debe pedir un número mayor (entre 30 y 40) para establecer el límite superior del aleatorio. – Sólo se aceptarán caracteres numéricos.
8. La aplicación pensará un número aleatorio entre el rango creado por el usuario y lo construirá en inputs de type button , o buttons .
9. Nuestro juego ofrecerá 5 intentos por adivinar el número aleatorio
10. En cada intento con los inputs de type button del usuario, la aplicación ofrecerá feedback orientando al jugador de su situación a cada tirón -Informará sobre si el input tecleado es menor o mayor que el pensado por el software- y de los intentos que le quedan.
11. Si acierta dentro de las 5 oportunidades, el juego le devolverá feedback positivo y le ofreceremos la posibilidad de volver a jugar. No será necesario volver a identificarse.
12. Si usted no acierta en las 5 oportunidades, el juego le dará feedback negativo pero le ofrecerá la oportunidad de volver a jugar. No será necesario volver a identificarse.
13. El juego debe ofrecer un botón para salir de la aplicación que borrará el nombre del jugador.
14. El juego deberá estar resuelto íntegramente en funciones, o expresiones de funciones tipo constante de tipo arrow functionss.
15. Todos los elementos del juego deben ser inputs, buttons , y/o divs para gestionar tanto las interacciones como los feedback de retorno.
16. No habrá programación estructurada -Sólo funcional, con arrow functions y expresiones de funciones-, ni alerts ,ni prompts o confirms por la gestión de los feedbacks.

C- APARTADO B. El sitio debe cumplir los siguientes requisitos de accesibilidad y usabilidad:

1. Se implementarán las técnicas de accesibilidad (Jerarquías, etiquetado semántico, alto contraste, y atributos ( aria role , o tabindex ) y usabilidad (tipografías, tamaños, colores, feedback) oportunas para mejorar la funcionalidad y la experiencia de usuario necesarias

Requisitos:

1. Html , css , y javascript en archivos independientes para poder garantizar la modularidad de la aplicación

*Entregar un archivo funcional. FORMATO entrega: nombre\_ApellidosM01941\_3.zip*