

EXAMEN. CREACIÓN DE UN BINGO.

Este bingo va a ir sacando bolas hasta que 5 de estas bolas coincidan con 5 números del cartón generado.

PRIMERA PARTE (2,5 puntos)

Incluir en el documento HTML la información necesaria de CSS y JS.

Los archivos Javascript componentes.js y http-provider.js, no es necesario modificarlos. Trabajar únicamente sobre index.js y funciones.js.

El archivo index.js incluye comentarios aconsejando la colocación de diferentes partes del código.

Crear en el documento funciones.js las funciones siguientes:

- Una función que genere todos los números del bingo (Desde el 1 al 90), que mezcle esas bolas y que las almacene en un arreglo. Este arreglo será devuelto por la función.
- Una función que genere el cartón de bingo (24 números aleatorios de los 90). Esta función será invocada por el botón **Pedir cartón bingo**. Cada vez que hagamos clic sobre el botón se generará un cartón nuevo. Esos 24 números se mostrarán ordenados. Las etiquetas HTML para los número tienen un id desde "square1" hasta "square24".



Create A Bingo Card

B	I	N	G	O
1	22	61	74	84
3	26	64	78	85
5	28	Free	79	86
11	30	65	80	88
14	36	66	83	90

Empezar a jugar Pedir carta de bingo

- Una función que vaya sacando una a una las bolas que se han generado previamente. Se aconseja usar la función setInterval incluida en el código. Si la bola coincide con el cartón, tendremos que añadirle la clase **bg-info** y **text-white**. Los números que vayamos sacando se irán mostrando en la etiqueta identificada como **jugando**. Cuando finalice la partida se mostrarán, en la etiqueta identificada como **resultado**, las 5 bolas premiadas y cuántas bolas hemos sacado del bingo para obtener el resultado.

Create A Bingo Card

B	I	N	G	O
4	28	46	65	76
6	29	51	69	79
8	34	Free	72	83
19	35	52	74	84
22	42	58	75	89

Empezar a jugar Pedir carta de bingo

22

Bola extraída y marcada como premiada en el cartón

Create A Bingo Card

B	I	N	G	O
2	26	49	69	79
3	32	50	74	81
12	40	Free	75	84
17	44	56	77	88
20	47	67	78	89

Empezar a jugar Pedir carta de bingo

12

Se han sacado 16 bolas, y las bolas premiadas son 88,74,56,78,12

Nº de bolas sacadas y bolas premiadas

NOTA: Hasta este punto es imprescindible tenerlo acabado y funcionando correctamente para poder valorar el resto del examen. Además, no podréis modificar bajo ningún concepto el index.html, salvo la inclusión de CSS y Javascript.

(FIN DE PRIMERA PARTE. SEGUNDA Y TERCERA PARTE EN LA SIGUIENTE PÁGINA)

SEGUNDA PARTE (2,5 puntos)

Utilizando el `http-provider`, `jugadores.json`, los componentes y toda la información que tenéis dentro, vamos a realizar lo siguiente:

- El botón de cerrar sesión estará oculto. Los botones de juego estarán desactivados hasta que el usuario no se haya identificado.
- En el momento que se haya identificado, se activarán los botones para que el usuario pueda jugar, se ocultará el botón de consultar y se mostrará el botón de cerrar sesión. En la etiqueta de identificador “saludo” se mostrará el nombre del jugador en una etiqueta H1.
- Trabajando con el objeto literal `jugador` disponible en `componentes.js`, almacenaremos toda la información del usuario en el `localStorage` para que en el caso de que el usuario cierre o actualice el navegador, pueda seguir jugando sin tener que volver a loguearse. **No se tiene en cuenta que todavía se haya iniciado la partida, solamente el acceso del usuario.**
- Cuando el usuario haga clic sobre el botón cerrar sesión se borrará el `localStorage`, se desactivarán los botones para que no se pueda jugar, y se mostrarán/activarán los cuadros de texto de usuario/contraseña y el botón de consultar. La etiqueta con identificador “saludo” se vaciará.
- Cada vez que el jugador haya terminado la partida, añadiremos al final del arreglo **resultado** en el objeto literal `jugador` cuántas bolas ha sacado hasta tener las 5 bolas de su cartón.

TERCERA PARTE (2,5 puntos)

Mediante el `localStorage` deberéis de hacer un control de errores para mantener la información de la partida guardada, es decir, que si el usuario cierra el navegador a media partida pueda reanudarla sin problema. (A media partida significa que mientras se están extrayendo bolas se cierra el navegador. Al acceder de nuevo a la página web, el juego debe continuar donde se quedó). Valorar con antelación la información necesaria para reanudar la partida y cuándo se debe guardar en el `localStorage`. Dicha información se debe almacenar en el objeto literal `enJuego` dentro de `componentes.js`, que es el objeto que se vuelca al `localStorage` y posteriormente se restaura.

El examen deberá de realizarse en orden, es decir, para que un apartado tenga valor, tendrá que estar acabado el apartado anterior. Un orden alternativo sería parte1, parte3 y parte2, aunque se recomienda seguir el orden establecido.

Las puntuaciones de cada parte se tendrá en cuenta la calidad de la solución aportada: código legible, uso de un lenguaje adecuado, comentarios, uso de funciones, uso de variables, etc.