

## UD2E4. Ejercicios

### Ejercicio 1

Crea un closure llamado "jugarDados" que reciba como parámetro el número de lados de los dados ("numeroLados").

Este closure deberá definir internamente una función que devuelva un valor aleatorio entre 1 y el número de lados indicado.

Luego, el closure deberá devolver una función anónima que simule la tirada de dos dados, devolviendo un array con ambos resultados.

A continuación, escribe un script para probar el closure que:

1. Solicite al usuario el número de lados de los dados y el número de tiradas mediante prompts.
2. Utilice el closure para instanciar una función "tirarDados".
3. Realice la cantidad de tiradas indicada, comparando los resultados de la máquina y el jugador.
4. Muestra cada resultado individual y, al final, el número total de victorias de cada uno mediante alerts.

Una vez comprobado el código crea una página HTML que te permita jugar una partida de dados contra la máquina. Debe gestionar al menos el número de lados del dado, el número de tiradas de la partida, los resultados intermedios de las tiradas y el total de cada jugador. El jugador humano deberá pulsar un botón cada vez que quiera tirar.

Organiza el código adecuadamente.

### Ejercicio 2

A partir de un array de objetos JSON de tipo alumno (nombre, nota, modulo, convocatorias), vamos a crear funciones para trabajar con JSON y arrays.

Formato de los datos:

- Propiedad nota debe estar en el rango de 0-10.
- Propiedad convocatorias debe estar en el rango de 1-4.
- Los nombres de los módulos son DWEC y DWES.
- Incluye 5 o 6 datos de prueba

1. Lista de alumnos suspensos: Crea una función que devuelva un array de JSON con los alumnos que han suspendido (nota inferior a 5). El resultado debe incluir únicamente el nombre, la nota y el módulo del alumno, y debe estar ordenado alfabéticamente por el nombre del alumno.

2. Estadísticas por módulo: Crea una función que devuelva un array de JSON con el nombre del módulo, la nota media y el número de convocatorias medio por módulo. El

resultado debe estar ordenado por el número de convocatorias medio en orden descendente. Para calcular estas estadísticas:

- Primero, crea un objeto que agrupe los módulos y acumule las notas, convocatorias y **cantidad** de alumnos por módulo.
- Luego, recorre los datos agrupados para calcular la nota media y las convocatorias medias, truncando a dos decimales.

3. Función para devolver los datos en formato JSON: Implementa una función que devuelva la cadena JSON con todos los datos de los alumnos.

4. Función para cargar una nueva cadena JSON: Implementa una función que permita cargar una nueva cadena JSON. Utiliza un bloque ``try-catch`` para gestionar posibles errores durante el parseo del JSON. Si la cadena no es válida o no es un array, muestra un mensaje de error adecuado. Puedes comprobar si un objeto es un array con `Array.isArray(objeto)`.

Todo lo anterior debe estar contenido en una función autoinvocada y asociada a una constante denominada `$yedra`.

Crea una página HTML que permita probar la funcionalidad requerida.

Organiza el código adecuadamente.