

## UD02-T02: Arrays, funciones y objetos en JS

Realizar una Web que presente en una única página todas los apartados que se piden en el ejercicio, de forma que puedan validarse de forma independiente y sin necesidad de realizar recargas de la página para su ejecución. Esta página contendrá un menú desde el que accederemos (mediante desplazamiento vertical) a cada una de las opciones y en estas habrá un enlace que nos permita volver al principio (menú).

Se debe separar el código HTML del código JS y la forma de realizar la ejecución de los apartados será a través de funciones que serán llamadas desde los eventos clic de los diferentes apartados. Por lo tanto, será necesario incluir botones que respondan a acciones determinadas.

Validar correctamente la entrada de datos para evitar errores de formato/tipo o de límites/restricciones.

1. Gestione las notas de una clase de 5 alumnos de los cuales sabemos el nombre y la nota (se puede asignar directamente en el programa). Los datos deben ser almacenados en un array de objetos que deben tener las propiedades y métodos adecuados. El menú de opciones será el que se muestra a continuación:
  - Buscar un alumno.
  - Modificar su nota.
  - Realizar la media de todas las notas.
  - Realizar la media de las notas menores de 5.
  - Mostrar el alumno(s) que mejor(es) nota(s) ha sacado.
  - Mostrar el alumno que peores notas ha sacado.
2. Un garaje de estacionamiento cobra una tarifa mínima de 2.00€ por aparcar durante un máximo de tres horas. Además, el garaje cobra 0.50€ por cada hora. También cobrará 0.50€ por cada grupo de tres horas a partir de la sexta hora. El cargo máximo por día es 10.00€. De tal forma que la tarifa quedaría:
  - Primeras 3 horas: 2.00 €
  - A partir de la tercera hora, cada hora sumará 0.50 €
  - A partir de la sexta hora, cada 3 horas sumará 0.50 € (p.e. estar entre 7 y 9 horas supondrá añadir 0.5€ a la tarifa)
  - Un día completo costará 10.00 €
  - Si sobrepasa un día, se sumará la tarifa máxima de un día a las horas que esté del día siguiente.

Ejemplos:

- Si "Juan" dice que ha permanecido 2 horas, se le cobrarán 2 €
- Si "Ana" dice que ha estado 24 horas, se le cobrarán 10 €
- Si "Roberto" dice que ha dejado el coche 6 horas, se le cobrarán 3,5€ (2€ de las primeras 3 horas y 1,5€ de las 3 horas siguientes)
- Si "Lola" dice que tiene el coche 8 horas se le cobrarán 4€

Escribir una función (o funciones) en JS que calcule y muestre lo que ha de abonar varios clientes. Por cada cliente, se ha de introducir las horas que lleva estacionado el coche de dicho cliente. El programa debe mostrar el cobro para el cliente actual en la Web y debe calcular y mostrar el total acumulado de los ingresos en la consola de la página Web. Debes crear una función llamada *calculaCargos* para determinar el cobro por cada cliente. Cada vez que se termina de mostrar los datos de cobro de un cliente se preguntará (mediante *un cuadro de diálogo*) si se desea realizar otro cobro. El programa finalizará cuando el usuario le de al botón de cancelar.

3. Partiendo de la definición de objetos siguiente:

```
function Libro(titulo, propietario, paginas) {  
    this.titulo = titulo;  
    this.propietario = propietario;  
    this.paginas= paginas;  
}  
  
Libro.prototype.mostrarConsola = function() {  
    console.log("Has leído las " + this.paginas + " páginas de " + this.titulo);  
};
```

- Crea un objeto nuevo llamado Libro. Su constructor debería también incluir el parámetro autor.
- Añade otro método a Libro llamado leiste() que utilice console.log para mostrar un mensaje del tipo "Has leído 'La sombra de los vientos'", donde el título del libro será sustituido por el correcto en cada caso.
- Utiliza un array y un bucle for para instanciar 5 objetos Libro y cumplimentar su información con valores aleatorios. Esta sería la información inicial antes de que el usuario pueda realizar alguna modificación sobre ellos.
- Mostrar las opciones necesarias para poder realizar el mantenimiento de este objeto. Es decir, incluir opciones de modificación, borrado e insertado de libros, manteniendo la estructura de un array de objetos. Para ello tendrás que modificar el objeto para incorporar métodos que modifiquen estas propiedades.