TEMAS 1-3 **EXAMEN DESARROLLO WEB EN ENTORNO CLIENTE - Recuperación**

Nombre:	**Fecha:**	**Nota**
Apellidos:	**Curso:**	

(10 puntos)

1) uber.js -Te acabas de registrar en Uber y te regalan 20€ para tu primer viaje. Quieres aprovechar ese dinero al máximo y vas a coger el mejor coche que te puedas permitir sin pasarte del dinero que te han regalado.

Hay 5 opciones, desde la más barata a la más cara: “UberX”, “UberXL”, “UberPlus”, “UberBlack”, “UberSUV”.

Sabiendo la longitud “length” en km y cuánto cuesta recorrer cada kilómetro para cada coche. Encuentra el mejor coche que te puedes permitir.

Ejemplo:

Para length=30 y coste (de cada coche) = [0.3, 0.5, 0.7, 1, 1.3], la respuesta esperada sería:

solution (length, money, cost) => “UberXL”

El coste del recorrido sería 15€, el cual te puedes permitir, pero “UberPlus” costaría 21€, lo cual es demasiado.

Crea para ello la función solution con los parámetros length (distancia), money (dinero), cost (array con el coste de cada coche por km) y devuelve el mejor coche que te puedes permitir.

(2 puntos)

2) Entregas para navidad.

Los elfos están preparando la víspera de Navidad y necesitan tu ayuda para calcular si van sobrados o no de tiempo ⏳.

Para ello te pasan un array con la duración de cada entrega. El formato de la duración es HH:mm:ss, las entregas empiezan a las 00:00:00 y el límite de tiempo es 07:00:00.

Tu función debe devolver el tiempo que les faltará o el tiempo que les sobrará para terminar las entregas. El formato de la duración devuelta debe ser HH:mm:ss.

Si terminan antes de las 07:00:00, el tiempo restante hasta las 07:00:00 debe ser mostrado con un signo negativo. Por ejemplo, si sobran 1 hora y 30 minutos, devuelve -01:30:00

calculateTime(['00:10:00', '01:00:00', '03:30:00']) // '-02:20:00'

calculateTime(['02:00:00', '05:00:00', '00:30:00']) // '00:30:00'

calculateTime(['00:45:00', '00:45:00', '00:00:30', '00:00:30']) // '-05:29:00'

(3 puntos)

3) Crea una clase para gestionar un sistema de alquiler de películas. Las películas están almacenadas en un array de objetos con la siguiente estructura:

- **Películas**:

```
{
 id: 101,
 titulo: 'Inception',
 director: 'Christopher Nolan',
 genero: ['Acción', 'Ciencia Ficción'],
 calificacion: 8.8,
 anio: 2010,
 copiasDisponibles: 3,
 alquiladoPor: [] // Array de IDs de clientes que han alquilado esta película
}
```

La clase se llamará **SistemaAlquilerPeliculas** y tendrá las siguientes propiedades:

- **peliculas**: Array que se inicializa en el constructor recibiendo por parámetro el array de películas existente.
- **clientes**: Array de objetos que representan a los clientes. Cada cliente tiene la siguiente estructura:

```
{
 id: 201,
 nombre: 'Ana Pérez',
 peliculasAlquiladas: [] // Array de IDs de películas alquiladas
}
```

- **peliculasAlquiladas**: Array vacío que posteriormente almacenará objetos de películas que están actualmente alquiladas.

Para esta clase, crea los siguientes métodos utilizando programación funcional:

a) obtenerPeliculasPorDirector(directorNombre):

Devuelve un array con los títulos de las películas dirigidas por el director especificado, ordenados alfabéticamente por el título de la película y teniendo en cuenta posibles mayúsculas o acentos.

b) obtenerPeliculasDisponiblesPorGenero(genero):

Devuelve un array con los títulos de las películas del género especificado que tienen al menos una copia disponible.

c) obtenerPeliculasMejorCalificadas(calificacionMinima):

Devuelve un array de objetos de películas que tienen una calificación igual o superior a **calificacionMinima**, ordenadas de mayor a menor calificación.

d) alquilarPelicula(clienteId, peliculaId):

Permite a un cliente alquilar una película si hay copias disponibles. Actualiza las propiedades **copiasDisponibles**, **alquiladoPor** de la película y **peliculasAlquiladas** del cliente. Y devuelve un array con la información de la película y del cliente, además del array de **peliculasAlquiladas**.

e) devolverPelicula(clienteId, peliculaId):

Permite a un cliente devolver una película alquilada. Actualiza las propiedades **copiasDisponibles**, **alquiladoPor** de la película y elimina el ID de la película de **peliculasAlquiladas** del cliente.

f) obtenerHistorialAlquilerCliente(clienteId):

Devuelve un array con los títulos de las películas que el cliente ha alquilado.

g) obtenerPeliculasMasPopulares():

Devuelve un array de títulos de películas ordenadas por el número de veces que han sido alquiladas, de mayor a menor popularidad.

h) eliminarPelicula(peliculaId):

Elimina una película del sistema si no está actualmente alquilada.

Notas adicionales:

- **Uso de Programación Funcional**: Se espera que todos los métodos implementados utilicen técnicas de programación funcional como **map**, **filter**, **reduce**, **sort**, etc.
- **Importante**: Si utilizas un **forEach** para resolver algún método, la pregunta te contará la mitad.
- **Actualización de Propiedades**: Cuando una película es alquilada o devuelta, actualiza correctamente las propiedades **copiasDisponibles** y los arrays **alquiladoPor** y **peliculasAlquiladas**.

(5 puntos)

Criterios de corrección:

Ejercicio 1:

1 punto – Calcula el precio que cuesta cada vehículo según los kms que van a recorrer.

1 punto – Devuelve el coche apropiado al presupuesto que tiene el usuario.

Ejercicio 2:

1 punto – La función calculateTime suma correctamente las duraciones en formato HH:mm:ss dadas en el array deliveries.

1 punto – La función compara correctamente el tiempo total acumulado con el límite de 07:00:00 para determinar si hay tiempo restante o tiempo excedido.

1 punto - La función muestra el tiempo restante con un signo negativo (-HH:mm:ss) si el total es menor a 07:00:00 y el tiempo excedido sin signo si es mayor o igual a 07:00:00.

Ejercicio 3:

0.5 – a) Muestra las películas filtradas por el director (0.25) y ordenadas por título (0.25).

0.5 – b) Devuelve las películas por género (0.25) y que tienen al menos una copia disponible (0.25).

0.5 – c) Devuelve un array de películas con una puntuación mayor a la que recibe por parámetro (0.25) y ordena este array de mayor a menor calificación (0.25).

1 – d) Permite realizar un alquiler de una película teniendo en cuenta si existen copias disponibles, y actualiza cada una de las propiedades que se piden.

0.75 – e) Devuelve una película actualizando cada uno de los campos correspondientes.

0.5 – f) Devuelve un array con el título de las películas que el cliente ha alquilado.

0.5 – g) Obtiene un array de títulos de películas más populares ordenadas por el número de veces que han sido alquiladas.

0.75 – h) Elimina una película del sistema (0.5) si no está actualmente alquilada (0.25).