

# Funciones, Objetos y JSON

## Práctica

### Preparación

Descargar la carpeta con los archivos de prueba y esqueleto de la solución desde el aula virtual. Se trabajará únicamente sobre estos archivos, salvo que se indique lo contrario.

### Desarrollar las siguientes funciones

#### **actualizarArchivosDeudas**

Recibe las rutas de cuatro archivos (de deudas viejo, de pagos, de deudas nuevo, y de log).

De los archivos de entrada sabemos lo siguiente:

- El archivo de deudas viejo no tiene un orden específico. Sus registros no presentan repetidos de acuerdo a su dni, ni a su nombre. Todos sus campos poseen datos válidos (no null, no vacíos). El campo 'debe' siempre es un número positivo.
- El archivo de pagos tampoco tiene un orden específico, y puede contener múltiples registros con el mismo dni y apellido. El campo 'fecha' no se repite entre ninguno de los registros. El campo 'pago' siempre es un número positivo.

Esta función no devuelve nada, y debe generar:

- Un archivo de deudas actualizado con todos los pagos realizados, a guardarse en la ruta especificada como 'deudas nuevo'. Este archivo debe tener las mismas características que el de entrada ('deudas viejo').
- Un archivo de log, en la ruta provista, en donde quede un registro de los eventos detallados en el último punto.

La función extrae el contenido de los archivos de la carpeta de entrada (in), los ordena, procesa las actualizaciones de los pagos sobre las deudas, y graba el resultado final en un nuevo archivo en la carpeta de salida (out).

#### **ordenarPorClave**

Recibe un array con objetos, y el nombre de un campo en formato string. Devuelve el array original, ordenado en forma ascendente por el campo provisto.

#### Opcional

En lugar de un único string, recibe un array de strings. Devuelve el array original, ordenado en forma ascendente por los campos provistos en el array de claves, siguiendo el mismo orden de importancia (primera clave, primer campo que determina el orden, etc).

## **actualizarDeudas**

Recibe un array con deudas, un array con pagos, y una función a la que podemos llamar, pasándole un mensaje, cada vez que precisemos loguear un evento. Devuelve un array con las deudas actualizadas según los pagos, siguiendo el siguiente criterio:

- Si aparece un registro de pago con un dni que no coincide con el de ninguna deuda, ese registro no se procesa, y se debe loguear como operación inválida.
- Si aparece un registro de pago que coincide con una deuda en su dni pero no en su apellido, el mismo no se procesa, y se deben loguear ambos, la deuda y el pago.
- Si un registro de deuda no posee pagos asociados, se agrega directamente al array actualizado, sin cambios.
- Si luego de aplicar todos los pagos correspondientes a una deuda, ésta queda aún queda en positivo, se agrega al array actualizado con el nuevo importe.
- Si luego de aplicar todos los pagos correspondientes a una deuda, ésta queda en cero o menos, la misma no debe agregarse al array de deudas actualizado.
- En particular, si la deuda queda en negativo (por debajo de cero), se deben loguear los datos del cliente y cuánto saldo a favor tiene.

## **Observación**

Para para facilitar la tarea, ya se cuenta con algunas funciones desarrolladas, y se proveen también las firmas de las funciones pedidas, junto con su comentario correspondiente (formato JSDOC). Además, se incluyen, como ejemplo, dos documentos de entrada con sus respectivos documentos de salida, para usar como lote de prueba / verificación.