# Manejo de archivos

## Práctica

## Crear un proyecto que implemente las siguientes funciones

### leerArchivoComoString

Recibe la ruta del archivo que se quiere leer, y devuelve un único string con todo el contenido del mismo.

### escribirTextoEnArchivo

Recibe una ruta, un texto, y un flag, y graba ese texto en un archivo en la ruta dada. Si el directorio es válido pero el archivo no existe, decide que hacer según el flag:

- Con el flag en true, crea el archivo y lo escribe.
- Con el flag en false, lanza el error “el archivo no existe”.

### transformarStringEnArrayDeNumeros

Recibe un texto y una secuencia de caracteres que usará como separador. Devuelve un array con todos los números producto de partir el texto cada vez que aparezca la secuencia separadora. En el caso de que alguna de las partes no sea numérica, no se incluirá en el resultado, pero no debe lanzar ningún error.

Ejemplo

Input: texto = `‘123 | 456 | 789 | 1bc | 10’` , separador = `‘ | ’`
Output: `[123, 456, 789, 10]`

### transformarArrayDeNumerosAUnSoloString

Recibe un array con strings, y una secuencia de caracteres para usar como separador. Devuelve un único string que es la unión de todos los strings del array, intercalando la secuencia separadora entre cada uno.

Ejemplo

Input: array = `[123, 456, 789, 10]` , separador = `‘,’`
Output: `‘123,456,789,10’`

### combinarDosArrays

Recibe dos arrays, ambos con datos de tipo numérico, ambos *ordenados* en forma ascendente, y sin repetidos dentro de cada archivo (puede haber repetidos entre un archivo y otro). Devuelve un nuevo array, que contenga todos los datos de ambos arrays, también *ordenados* en forma ascendente, y también sin repetidos.

Ejemplo

Input: array1 = `[1, 5, 10]` , array2 = `[2, 3, 8, 11]`
Output: `[1, 2, 3, 5, 8, 10, 11]`

Observación

Si se te ocurrió una solución que requiere poder ordenar un array, pensá en otra forma de hacerlo.

### combinarNArrays

Igual que la función anterior, solo que ésta recibe un array de arrays de números ordenados en forma ascendente y sin repetidos, y devuelve un nuevo array, con la combinación de todos los números de todos los arrays recibidos, también *ordenados* en forma ascendente, y también sin repetidos.

Ejemplo

Input: arrays = `[[1, 10], [2, 3, 15, 16], [4], [6, 7, 13]]`
Output: `[1, 2, 3, 4, 6, 7, 10, 13, 15, 16]`

## Al finalizar

Escribir código que permita probar las funciones creadas, utilizando los datos de prueba provistos. Luego configurar un script que permita ejecutar nuestro código de prueba mediante la instrucción: *npm test*.