

Prueba teórica:

1. Está recibiendo dos flujos de datos de dos sensores de temperatura (2 observables que devuelven números enteros), ¿qué operador de RxJS utilizaría para recibir estos dos datos en la misma suscripción?

Utilizaría el operador `combineLatest`

2. Si tiene dos llamadas al servidor y la segunda llamada depende de la primera, ¿cómo manejaría con RxJS esta secuencia de llamadas?

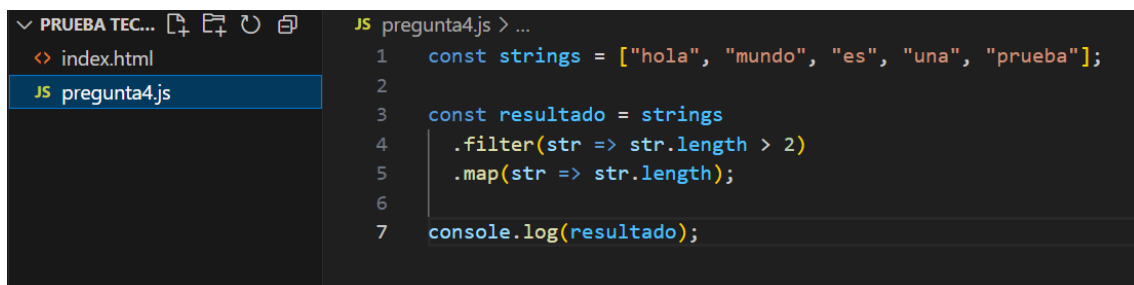
Utilizaría el operador `concatMap` para manejar la secuencia de llamadas.

3. Tengo en un servidor un archivo de texto que está en minúscula y ocupa 2GB en el disco duro, pero le solicitan que todo el archivo debe ser pasado a mayúsculas, ¿cómo lo haría?

Transformar con `.toUpperCase()` por bloques y guardar en nuevo archivo.

4. Tiene un arreglo de strings los cuales deben ser filtrados por su longitud mayor a dos y a la vez convertidos a un array de enteros con la longitud de cada string, ¿cómo lo haría? Ejemplo de entrada y salida: ["hola", "mundo", "es", "una", "prueba"] => [4, 5, 3, 6]

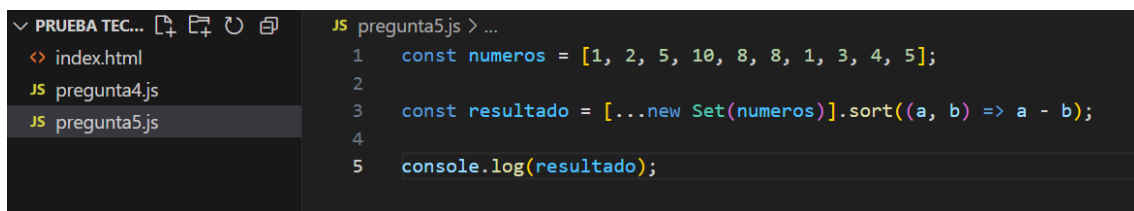
Utilizaría `filter()` y `map()` encadenados.



```
JS pregunta4.js > ...
1  const strings = ["hola", "mundo", "es", "una", "prueba"];
2
3  const resultado = strings
4    .filter(str => str.length > 2)
5    .map(str => str.length);
6
7  console.log(resultado);
```

5. Tiene un arreglo de números, los cuales pueden ser o no repetidos, ¿cómo eliminaría los repetidos? ¿Cómo los ordenaría en forma ascendente? Ejemplo de entrada y salida: [1, 2, 5, 10, 8, 8, 1, 3, 4, 5] => [1, 2, 3, 4, 5, 8, 10]

Utilizaría `Set` para eliminar duplicados y `sort()` para ordenar



```
JS pregunta5.js > ...
1  const numeros = [1, 2, 5, 10, 8, 8, 1, 3, 4, 5];
2
3  const resultado = [...new Set(numeros)].sort((a, b) => a - b);
4
5  console.log(resultado);
```