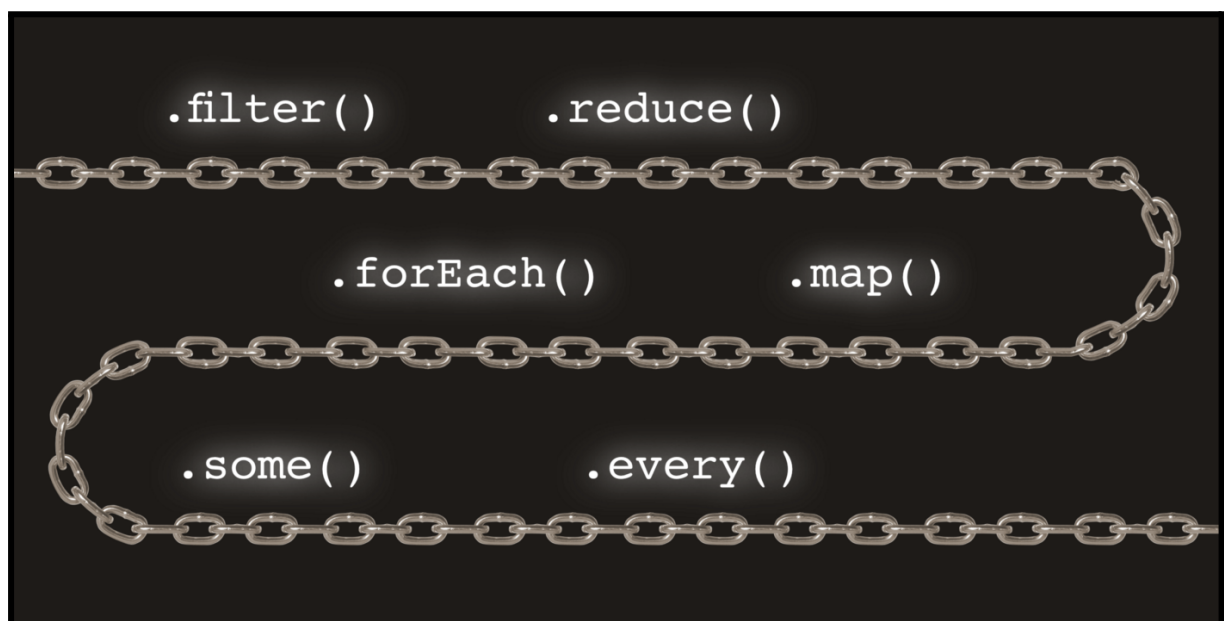


En esta tarea vas a poner en práctica nuevos métodos que ayudan a los desarrolladores a escribir código limpio y legible. Todos estos métodos son encadenables, lo que significa que se pueden usar en combinación entre sí y no mutan datos, lo cual es especialmente importante cuando se trabaja con frameworks como por ejemplo React. Con todos estos métodos de matriz, descubrirás que **no siempre es preciso utilizar bucles for o while para trabajar con arrays...**

Debes saber además que estos métodos reciben cómo parámetro una **función callback** o *función de devolución de llamada* que se ejecutará para cada elemento del array; no obstante, recuerda que en ECMAScript 6 es posible utilizar **Arrow Functions** en lugar de una callback.

Haz una lectura comprensiva del apartado correspondiente a los [métodos de iteración de arrays JavaScript](#) (recuerda que todos estos métodos operan en cada elemento del array) y realiza la tarea más abajo propuesta.



**Tarea16a** - El estudiante debe preparar ejemplos diferentes a los propuestos en el tutorial y *empleando arrow functions en lugar de funciones callback*, para al menos los siguientes métodos:

- **Array.map ()** - este método crea una nueva matriz y mapea los valores de la matriz origen manipulados por la función.
- **Array.filter ()** - este método crea un nuevo array con todos los elementos que cumplan la condición implementada por la función dada.
- **Array.reduce ()** - este método usa un acumulador para *reducir* todos los elementos de una matriz a un solo valor. Excelente para calcular totales.

**Tarea16b** Obtener a partir del array personas el array personasSimplificado

```
personas = [  
    { nombre: "Juan", edad: 22 },  
    { nombre: "Perico", edad: 50 },  
    { nombre: "Andrés", edad: 100 }];
```

```
personasSimplificado = [  
    { nombre: "Juan" },  
    { nombre: "Perico" },  
    { nombre: "Andrés" }];
```