

# Para todos los ejercicios

- Solamente se puede utilizar map, find, filter y reduce.
- No se puede utilizar ningún método salvo los mencionados.
- No se puede utilizar ninguna estructura de control (if, for, while, etc).
- Solamente tienes que escribir una línea de código.
- Los datos de los arrays pueden variar.

## EJERCICIO 1

Se pide:

Nombre de las frutas tropicales con un stock mayor de 21. Una línea de código.

### CÓDIGO STARTER:

```
<html>
<head>
<title>retos JavaScript</title>
</head>
<body>
<h1>retos JavaScript</h1>
<script>
let frutas = ["aguacate", "pera", "manzana", "naranja", "tomate", "granada",
"mango", "litchi", "nispero"];
let tropicales = ["aguacate", "mango", "litchi", "nispero"];
let tropicStock =
[ {fruta:"manzana", stock:55}, {fruta:"aguacate", stock:50},
{fruta:"mango", stock:10},
{fruta:"litchi", stock:20}, {fruta:"nispero", stock:25} ];
//Nombre de las frutas tropicales con un stock mayor de 21
</script>
</body>
</html>
```

## EJERCICIO 2

Se pide que hagas un programa que muestre la media de TODAS las notas de los 3 alumnos que tenemos en nuestro JSON.

CÓDIGO STARTER:

```
<!DOCTYPE html>
<html>
<body>
<p>
```

Se pide que hagas un programa que muestre la media de TODAS las notas de DWEC de los alumnos que tenemos en nuestro JSON. Hoy hay 3 alumnos pero podrían ser 3000.

```
</p>
<script>
var valumnos = {
  "alumnos": [{
    "nombre": "Fulanito",
    "notas": {
      "DWEC": 8,
      "DWESE": 0.71,
      "ED": 7.04,
      "SSOO": 3.37,
      "BBDD": 0.22,
      "PROG": 1.09,
      "EINEM": 6.17
    }
  },
  {
    "nombre": "Menganito",
    "notas": {
      "DWEC": 7,
      "DWESE": 0.71,
      "ED": 7.04,
      "SSOO": 3.37,
      "BBDD": 0.22,
      "PROG": 1.09,
      "EINEM": 6.17
    }
  },
  {
    "nombre": "Cetanito",
```

```
"notas": {  
  "DWECC": 6,  
  "DWESE": 0.71,  
  "ED": 7.04,  
  "SSOO": 3.37,  
  "BBDD": 0.22,  
  "PROG": 1.09,  
  "EINEM": 6.17  
}  
}  
];  
};
```

```
// ESCRIBE TU CÓDIGO AQUÍ
```

```
</script>  
</body>  
</html>
```

Se valorará de tu programa:

- El programa realiza la funcionalidad requerida.
- Que no haya errores de consola.
- La eficiencia.
- La modularidad (alta cohesión y bajo acoplamiento).
- Que el código sea limpio, flexible, reutilizable y mantenible.
- Sigue los principios SOLID.
- Otros criterios.