

## Ejemplo de examen final - 3

1) Describir qué ocurrirá al ejecutarse el siguiente código en el navegador:

```
JavaScript
(async () => {
    const promesa = new Promise((resolve, reject) => {
        const n = Math.floor(Math.random()*25)+1;
        if (n%5 == 0)
            resolve(n);
        else
            reject("Error");
    });

    promesa
        .then((res) => console.info(res))
        .catch((error) => console.error(error));
})();
```

2) Suponer que una API devuelve una respuesta en el siguiente formato:

```
JavaScript
[ {
    nombre: "Pepe",
    apellido: "Luis",
    edad: 20
},
{
    nombre: "Luis",
    apellido: "Fonsi",
    edad: 40
},
{
    nombre: "Mariela",
    apellido: "Rodríguez",
    edad: 23
} ]
```

Al consumir dicha API utilizando el código a continuación:

```
JavaScript
fetch("https://url-api.com")
    .then((res) => res.json())
```

```
.then((datos) => <completar aquí>)  
.catch((error) => console.error(error));
```

El resultado obtenido y mostrado en la consola es:

```
JavaScript  
[ {  
  nacioEn: 2003  
},  
 {  
  nacioEn: 1983  
},  
 {  
  nacioEn: 2000  
}]
```

Escribir el código faltante en <completar aquí> para que se produzca la salida mencionada.

- 3) Al usar promesas, describir las ventajas, si existen, del uso de la sintaxis `async-await` respecto de la sintaxis clásica `then-catch`.
- 4) Dado el vector de números enteros  $v = [2, 4, 35, 83, 22, 78]$  escribir un reductor para obtener el número más grande.
- 5) Escribir una función booleana que tome como argumento un vector y un número entero. La función debe retornar verdadero si el vector contiene al número o falso en caso contrario