

Ejemplo de examen final - 2

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()*10)+1;
        if (n%2 == 1)
            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(<completar aquí>)
  .catch((error) => console.error(error));
```

El resultado obtenido y mostrado en la consola es:

```
JavaScript
[
  {
    nombre_apellido: "Pepe Luis"
  },
  {
    nombre_apellido: "Luis Fonsi"
  },
  {
    nombre_apellido: "Mariela Rodríguez"
  }
]
```

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

- 3) Describir las ventajas del uso de promesas y la sintaxis async/await respecto del uso de callbacks
- 4) Dado el vector de números reales $v = [1.5, 2.2, 3.5, 8.3, 2.2, 7.8]$ escribir un reductor para calcular el valor promedio de sus valores
- 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