EJERCICIOS CALLBACK, PROMESAS FETCH Y ASYNC_AWAIT

I. E. S. POLITÉCNICO H. LANZ
GRANADA

1. Dado el siguiente código, modifícalo para que devuelva una promesa.

```
function retardo(ms) {
    // tu código
}

delay(3000).then(() => alert('se ejecuta después de 3
segundos'));
```

2. Crear una promesa para que transcurridos 3 segundos cambie el color de fondo de una web.

3. Crear una promesa que en el "resolve" añada la fecha actual a una variable en el localStorage y en el "reject" muestre un mensaje de error. ¿Es una promesa segura, es decir que siempre se va a ejecutar?

4. Crear una función **obtenerNumero** que le pase 3 parámetros:
   NúmeroMáximo.
   NúmeroEsperado.
   Retardo.
   De tal forma que cuando llame a la función calcule un **número_aleatorio** entre **0** y el ***númeroMáximo***. Si el ***númeroEsperado*** es mayor que el NúmeroAleatorio obtenido mostrará un mensaje de error(reject) indicando dicha circunstancia; Si el número es menor, mostrará dicho número (resolve).
   Crear una promesa dentro de la función que ejemplifique lo que se pide. Además, dicha promesa se debe de ejecutar con un tiempo de espera que coincida con el parámetro ***retardo***.
   En caso de no pasar parámetros en la llamada a la función, establecer como parámetros por defecto 10, 5 , 3000.

5. Refactorizar el código del ejercicio anterior con Async/Await.

6. Dado el siguiente código:

```
const listaUsuarios = [
    {id: 1, nombre: "Luis", codPais: 3},
    {id: 2, nombre: "Alfred", codPais: 1},
    {id: 3, nombre: "Pascal", codPais: 2}
]
const paises = {1: "Francia", 2: "Bélgica", 3: "España"};
```

- Crea una promesa con un retardo de 3 segundos que me imprima por pantalla los usuarios.
- Crea una promesa con un retardo de 2 segundos que me imprima por pantalla los países de los usuarios.
- ¿Cómo obligarías a que se ejecute en primer lugar la obtención de los usuarios por pantalla y posteriormente se muestren los países?
- Modifica el código de consumición de la promesa para que, en la obtención de los usuarios, además de mostrarlos por pantalla me devuelva los países, de tal forma que pueda encadenar dos promesas seguidas.
- Refactoriza el código para usar Asinc/Await.

7. Dada la API https://jsonplaceholder.typicode.com/users, usando la API Rest se pide:
    - Crea un proyecto que obtenga (GET) todos los datos de todos los usuarios de forma colapsada, de tal forma que sólo aparezca en HTML el nombre de los usuarios y que cuando pulsemos en dicho nombre se desplieguen todas las características almacenadas de cada usuario.
    - Crea un formulario de tal forma que al rellenar los datos se haga un (POST) a la API `https://jsonplaceholder.typicode.com/users`.
    - Crea un formulario que me pida un email. Si dicho email se encuentra dentro de `https://jsonplaceholder.typicode.com/users` entonces lanzar un (DELETE) y elimina dicha información.