

## Lista de Exercícios 02 - NODE

1. Em linguagens síncronas é comum a função *sleep(tempo)*. Assim, pode-se por exemplo escrever algo no console, pausar o script por x tempo e escrever algo a mais como:

```
console.log('Olá!')
sleep(2000)
console.log('Depois de 2secs')
```

simule o *sleep* de maneira correta em Javascript (sem travar a thread principal) utilizando promises e uma função *async/await*.

2. A função *readdir* disponível no módulo *fs* do Node, permite ler um diretório, e retornar seu conteúdo como um vetor. Crie uma versão em Promise da função.

**Dica:** a assinatura da função é: *fs.readdir(caminho, (err, files) => {})*

3. A lista de arquivos retornado no exercício anterior, também retorna diretórios. Crie uma função que retorne todos os sub-diretórios em um vetor (deve-se ignorar os arquivos retornados pelo *readdir*).

Algumas funções do *fs* e de *Promises* que te ajudarão:

- *fs.stat(caminhoArquivoDiretorio, (err, stats) => {})*
- *stats* que é retornado no callback possui um *isDirectory()* e um *isFile()*
- *Promise.all(vetorDePromises)*: aguarda um vetor de promises ser executado, e retorna uma promise que retorna para um vetor com o resultado.
- Exemplo:

```
Promise.all([promise1, promise2]).then( results => {
    results[0] // resultado de promise1
    results[1] // resultado de promise2
})
```