Atividades Callback, Promises e Async Await

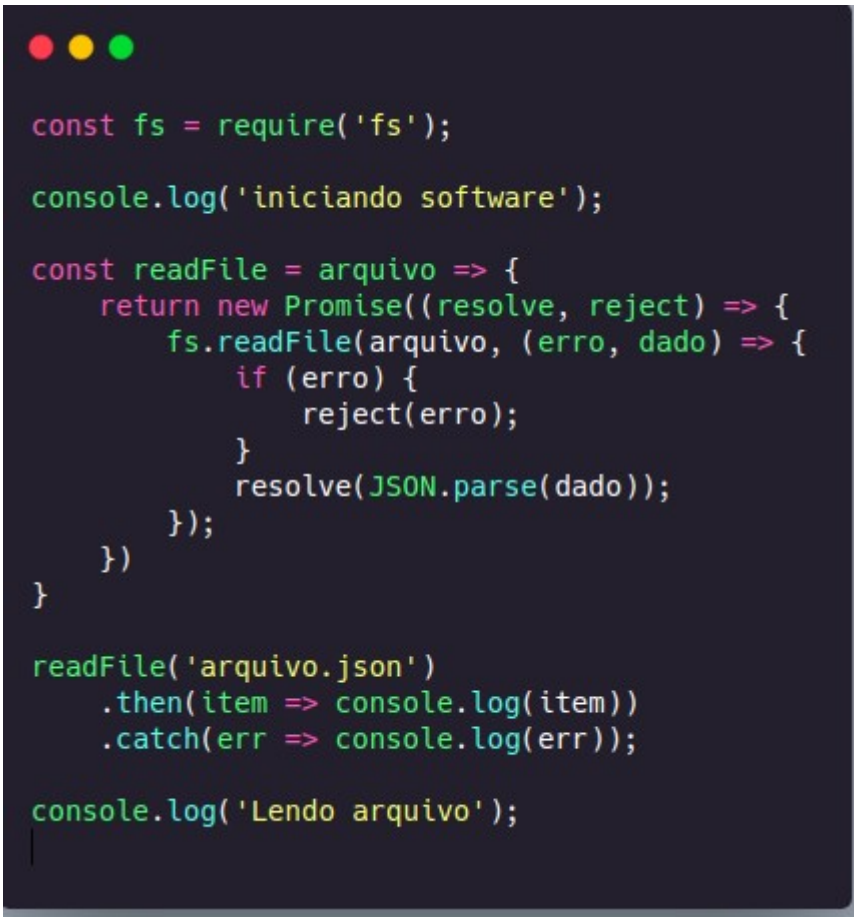
1 **(OBRIGATÓRIO)** - Crie um arquivo chamado atividade-1.js, dentro dele crie um comentário descrevendo qual será a saída no console do código abaixo sem utilizar o vscode.

```
const fs = require('fs');

console.log('iniciando software');

function callback(erro, dados) {
    if (erro) {
        console.log('aconteceu um erro');
        return;
    }
    console.log(JSON.parse(dados))
}

fs.readFile('arquivo.json', callback);

console.log('Lendo arquivo');
```

2 **(OBRIGATÓRIO)** - Crie um arquivo atividade-2.js, dentro dele crie um comentário descrevendo linha por linha o que o código abaixo esta fazendo (Observe que o método **readFile** é diferente do **readFileSync)**:

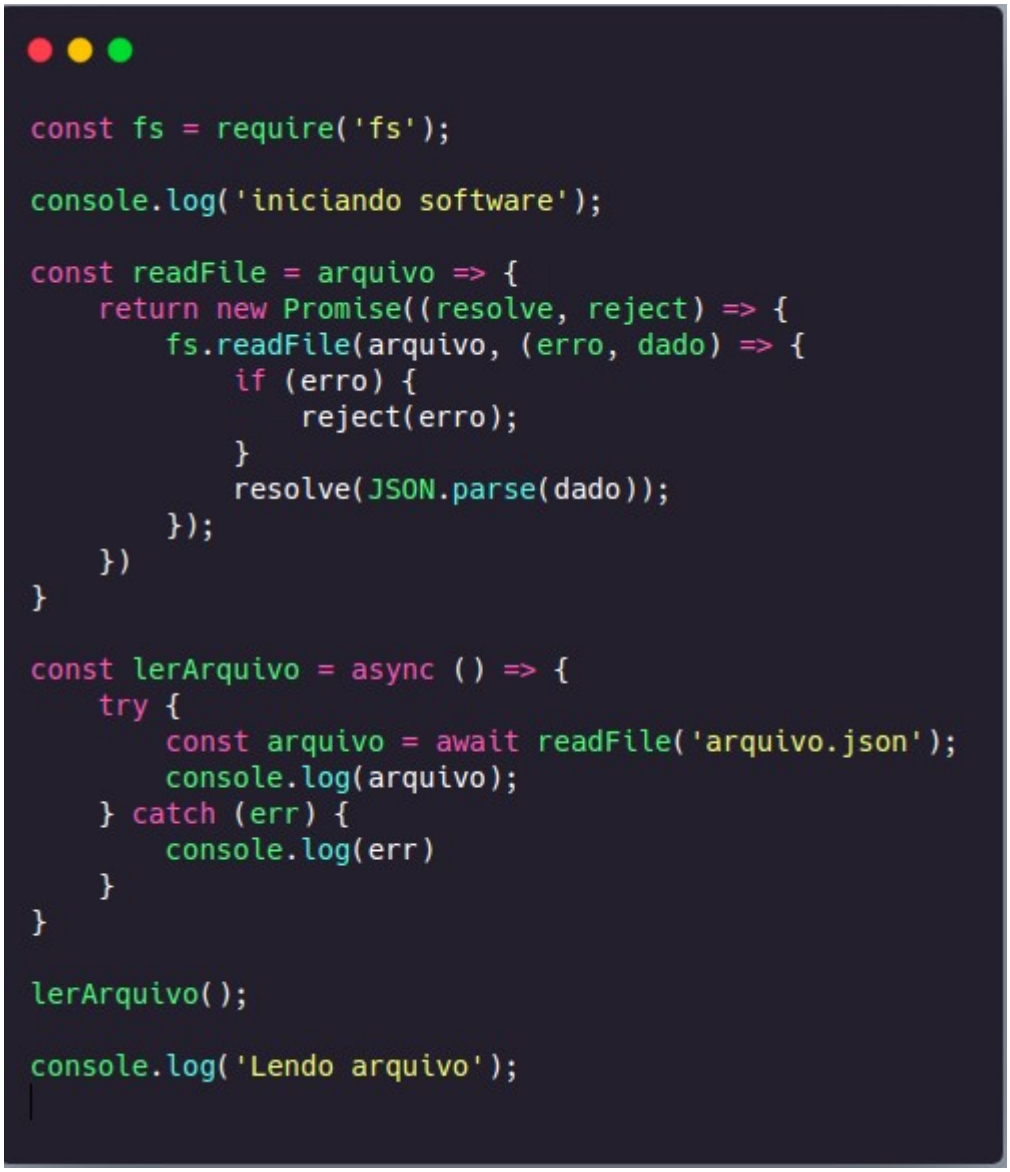
```
const fs = require('fs');

console.log('iniciando software');

function callback(erro, dados) {
    if (erro) {
        console.log('aconteceu um erro');
        return;
    }
    console.log(JSON.parse(dados))
}

fs.readFile('arquivo.json', callback);

console.log('Lendo arquivo');
```

3 **(OBRIGATÓRIO)** - Crie um arquivo atividade-3.js, dentro dele crie um comentário descrevendo linha por linha o que o código abaixo esta fazendo.



```
const fs = require('fs');

console.log('iniciando software');

const readFile = arquivo => {
    return new Promise((resolve, reject) => {
        fs.readFile(arquivo, (erro, dado) => {
            if (erro) {
                reject(erro);
            }
            resolve(JSON.parse(dado));
        });
    })
}

readFile('arquivo.json')
    .then(item => console.log(item))
    .catch(err => console.log(err));

console.log('Lendo arquivo');
```

4 **(OBRIGATÓRIO)** - Crie um arquivo atividade-4.js, dentro dele crie um comentário descrevendo linha por linha o que o código abaixo esta fazendo.



```
const fs = require('fs');

console.log('iniciando software');

const readFile = arquivo => {
    return new Promise((resolve, reject) => {
        fs.readFile(arquivo, (erro, dado) => {
            if (erro) {
                reject(erro);
            }
            resolve(JSON.parse(dado));
        });
    })
}

const lerArquivo = async () => {
    try {
        const arquivo = await readFile('arquivo.json');
        console.log(arquivo);
    } catch (err) {
        console.log(err)
    }
}

lerArquivo();

console.log('Lendo arquivo');
```

5 **(OBRIGATÓRIO) –** Faça um clone do repositóro → https://github.com/arbyte-br/callback_nodejs
e refatore o código utilizando apenas Promises (Não utilizar async/ await) e métodos que julgar eficiente.

6 **(OBRIGATÓRIO) –** Faça um clone do repositóro → https://github.com/arbyte-br/callback_nodejs
e refatore o código utilizando Promises e async/ await.

7 - Crie um programa que contenha duas funções, uma delas deve simular o download de uma imagem e, a outra deve simular o download dessa imagem.
O programa deve funcionar de forma Síncrona.

8 - Crie um programa que contenha duas funções, uma delas deve simular o download de uma imagem e, a outra deve simular o download dessa imagem.
O programa deve funcionar de forma Assíncrona.

9 - Crie um programa que simula o download de múltiplas imagens de forma assíncrona. O programa deve conter ‘.then’ e ‘.catch’ para tratar o sucesso e a falha da promisse.

10 - Crie um programa que simula a compra de um produto no supermercado, o programa deve simular o processamento de uma compra e imprimir as seguintes mensagens no console:
1 - Compra em processamento
2 - Compra aprovada OU Negada (para erros)

O programa deve tratar tanto os erros quanto os sucessos da promisse.