Em Node.js, um **callback** é uma função passada como argumento para outra função, que é executada após a conclusão de uma operação assíncrona ou de longa duração. Como o Node.js é baseado em um modelo de **I/O não bloqueante** e **assíncrono**, os callbacks são fundamentais para lidar com operações como leitura de arquivos, requisições de rede ou consultas de bases de dados sem bloquear a execução do programa.

**Características Principais:**

1. **Assincronia**: Permite que operações demoradas sejam executadas em segundo plano, enquanto o código continua a processar outras tarefas.
2. **Padrão Error-First**: Convenção em Node.js onde o primeiro parâmetro do callback é reservado para um objeto de erro (error), e os demais parâmetros são para os resultados em caso de sucesso.
3. **Reutilização**: Permite definir comportamentos personalizados para serem executados após a conclusão de uma operação.

**Qual dos dois exemplos fornecidos em baixo faz o uso de callbacks?**

Uma imagem com texto, captura de ecrã, Tipo de letra, file

Os conteúdos gerados por IA poderão estar incorretos.

Estes dois exemplos explicam o conceito de pedidos **bloqueantes** e **não-bloqueantes**. O primeiro exemplo mostra que o programa **bloqueia** até ler o ficheiro, o segundo exemplo mostra que o programa **não espera pela leitura** do ficheiro e imprime o "Programa Finalizado" ao mesmo tempo que o programa continua a ler o ficheiro.

Em nodejs todo o trabalho está numa única thread, o que resulta num consumo inferior de cursos.