

## 1. O que é Node.js e suas principais características

**Node.js** é um ambiente de execução JavaScript **do lado do servidor**, construído sobre o motor V8 do Google Chrome. Ele permite rodar código JavaScript fora do navegador.

**Principais características:**

- **Assíncrono e orientado a eventos** (non-blocking I/O)
- **Single-threaded**, porém com alto desempenho graças ao Event Loop
- Suporte ao **JavaScript moderno**
- **Multiplataforma**
- Utiliza o **motor V8** para execução rápida de código
- Possui um grande ecossistema de pacotes via **npm**

## 2. Diferença entre modelo Single-threaded e Multi-threaded

No modelo **Single-threaded**, apenas **uma única thread** de execução é utilizada. Isso significa que as tarefas são executadas de forma sequencial, uma após a outra. No entanto, em ambientes como o Node.js, é possível lidar com tarefas assíncronas (como leitura de arquivos, requisições de rede etc.) por meio de um mecanismo chamado **Event Loop**, que permite tratar eventos e callbacks de forma eficiente, sem bloquear a thread principal.

Já no modelo **Multi-threaded**, o sistema utiliza **várias threads** para executar múltiplas tarefas **simultaneamente**. Cada thread pode lidar com uma tarefa diferente ao mesmo tempo, o que pode aumentar o desempenho em aplicações que exigem muita computação paralela. Linguagens como Java e C++ permitem esse tipo de programação por meio de bibliotecas de threads, como as **Java Threads** ou **pthreads** em C/C++.

**Resumindo:**

- O modelo **Single-threaded** (como o Node.js) é baseado em uma única thread que lida com tarefas de forma assíncrona e eficiente.
- O modelo **Multi-threaded** distribui as tarefas entre múltiplas threads, podendo realizar várias operações em paralelo.
- **3. O que é o Event Loop no Node.js**
- O **Event Loop** é o mecanismo central que permite que Node.js trate operações assíncronas, como leitura de arquivos e requisições HTTP, **sem bloquear a thread principal**. Ele escuta eventos e executa callbacks registrados quando esses eventos ocorrem.

#### 4. Três aplicações onde o Node.js é amplamente utilizado

- **APIs RESTful** para back-end de aplicações web e mobile
- **Aplicações em tempo real** como chats e jogos multiplayer
- **Ferramentas de linha de comando (CLI)** e automação de tarefas

#### 5. Como o npm facilita o desenvolvimento de aplicações

O npm (Node Package Manager):

- Gerencia dependências de forma simples (npm install)
- Possui um **repositório enorme** de pacotes reutilizáveis
- Facilita **atualizações** de pacotes
- Permite **scripts personalizados** para automação (testes, build, etc.)

#### 6. Passos para criar um projeto Node.js

1. Criar pasta do projeto e navegar até ela:

```
mkdir meu-projeto && cd meu-projeto
```

2. Iniciar o projeto com npm:

```
npm init -y
```

3. Criar o arquivo principal (ex: index.js)
4. Escrever o código no index.js
5. Instalar dependências com npm install
6. Executar o projeto com:

```
bash
```

```
CopiarEditar
```

```
node index.js
```

#### 7. Dois exemplos de pacotes populares no npm

- **Express:** Framework web minimalista e rápido para criação de APIs e servidores HTTP.

- **Nodemon:** Ferramenta que reinicia automaticamente o servidor ao detectar mudanças no código.

## 8. O que é o arquivo package.json e sua importância

É o **manifesto** do projeto Node.js. Ele contém:

- Informações do projeto (nome, versão, autor, etc.)
- Scripts de execução
- Lista de **dependências**
- Permite **reprodutibilidade** e automação da instalação do projeto

## 9. Conceito de callback no Node.js

Um **callback** é uma função passada como argumento para outra função, a ser executada após uma operação assíncrona ser concluída. Exemplo:

```
fs.readFile('arquivo.txt', (err, data) => {  
  
  if (err) throw err;  
  
  console.log(data.toString());  
  
});
```

## 10. Qual é o papel do motor V8 no Node.js

O **V8** é o motor JavaScript do Google, usado no Chrome. No Node.js, ele:

- Compila e executa código JavaScript diretamente em **código de máquina**
- Oferece alta performance na execução
- Permite que o JavaScript rode **fora do navegador**