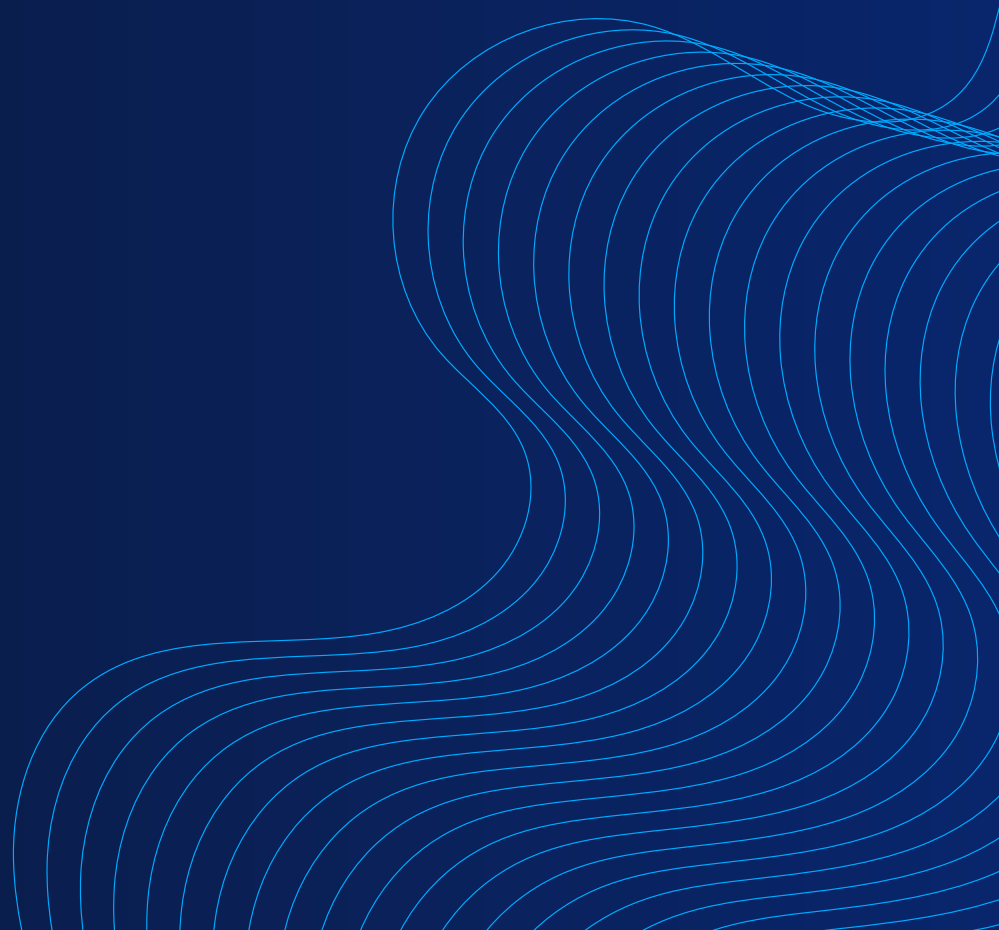




DNC⁷

Material de Apoio

Node.js



➤ Node

- **Definição:** Node geralmente se refere à plataforma como um todo, incluindo o motor JavaScript V8, o núcleo do Node.js e os diversos módulos que permitem a execução de JavaScript fora do navegador.
- **Plataforma:** É a plataforma que permite a execução de código JavaScript no servidor, além de proporcionar a base sobre a qual o Node.js opera.

➤ Node.js

- **Definição:** Node.js é um runtime de código aberto que permite a execução de código JavaScript no lado do servidor.
- **Implementação:** É construído sobre o motor JavaScript V8 do Google, o mesmo motor usado pelo navegador Chrome.
- **Componentes:** Inclui várias bibliotecas e módulos para facilitar tarefas de servidor, como manipulação de arquivos, operações de rede, streaming de dados e comunicação entre processos.

Principais Características do Node.js

➤ Assíncrono e Baseado em Eventos:

- O Node.js adota um modelo de I/O assíncrono e não bloqueante, o que significa que atividades como leitura de arquivos, consultas a bancos de dados e requisições HTTP não interrompem outras tarefas em execução.
- Com o loop de eventos, o Node.js consegue gerenciar operações de I/O de forma eficaz, sendo ideal para aplicações com múltiplas conexões simultâneas.

➤ V8 JavaScript Engine:

- Utilizando o mecanismo V8 do Google Chrome, o Node.js compila o JavaScript para código de máquina, garantindo alto desempenho.
- Isso possibilita que o Node.js execute operações de maneira extremamente rápida se comparado a outras linguagens de script.

Principais Características do Node.js

➤ Módulos:

- O Node.js inclui um sistema de módulos integrado (CommonJS), que facilita a modularização do código, promovendo reutilização e manutenção simplificadas.
- Os módulos podem ser importados e exportados entre diferentes arquivos de forma fácil usando **'require'** e **'module.exports'**.

➤ NPM (Node Package Manager):

- O NPM é o gerenciador de pacotes padrão do Node.js, permitindo que os desenvolvedores instalem, compartilhem e gerenciem as dependências de seus projetos.
- Com uma extensa coleção de mais de um milhão de pacotes disponíveis, o NPM oferece diversas bibliotecas e ferramentas para acelerar o processo de desenvolvimento.

Como Funciona?

Node.js opera em um único thread, mas usa um modelo de I/O assíncrono e não bloqueante para suportar operações de alto desempenho.

Isso é alcançado através do uso do loop de eventos, que gerencia todas as operações assíncronas, delegando tarefas para o sistema operacional e chamando callbacks quando as operações são concluídas.

Aplicações do Node.js

Node.js é amplamente utilizado para construir aplicações de tempo real e escaláveis, tais como:

- **APIs RESTful:**
 - Devido à sua capacidade de lidar com muitas conexões simultâneas, Node.js é uma escolha popular para criar APIs RESTful eficientes e escaláveis.
- **Aplicações de Tempo Real:**
 - Aplicações como chats, jogos multiplayer e plataformas de colaboração em tempo real se beneficiam do modelo de I/O assíncrono do Node.js.
- **Aplicações de Streaming de Dados:**
 - Node.js é ideal para aplicações que requerem o processamento de grandes volumes de dados em tempo real, como plataformas de streaming de vídeo e áudio.

Vantagens e Desvantagens do Node.js

➤ Vantagens

- Alta performance devido ao motor V8.
- Manejo eficiente de operações de I/O.
- Grande ecossistema de módulos e pacotes.
- Ideal para aplicações de tempo real e escaláveis.

➤ Desvantagens

- Pode ser difícil de depurar devido ao código assíncrono.
- Não é adequado para operações intensivas de CPU.
- A natureza assíncrona pode levar a código complexo e difícil de manter.



E aí, curtiu?

Esperamos que esse resumo tenha enriquecido sua perspectiva estratégica para enfrentar os desafios.

Salve esse PDF para consultar sempre que precisar.