Desenvolvendo aplicações com Node.JS

{ Aprenda a desenvolver aplicações back-end }



Anderson Oliveira
Rodrigo Bressan

CLUBE DE AUTORES

Todos os direitos reservados aos autores. LEI 9.610/98 e atualizações. Nenhuma parte desta publicação poderá ser reproduzida ou transmitida, sejam quais forem os meios empregados.

Copyright© 2024 – Clube de Autores.

Autores: Anderson Luiz de Oliveira, Rodrigo Bressan.

Revisão Ortográfica: Fernanda Bortoleto Michetti Muniz de Faria.

Capa: Daina Pinheiro.

Diagramação: Daina Pinheiro.

Ilustrações: Freepik.Com.

Supervisão: Márcia Cardoso Dias.

1ª Edição - 09/2024 - Atibaia / SP – Versão Digital.

Bibliografia. ISBN 978-65-01-14972-1
1. Desenvolvimento de sistemas 2. Informática 3. Programação (Computadores)

Índices para catálogo sistemático:

CDD: 005.107
1. Programação : Informática : Estudo e ensino

Eliete Marques da Silva - Bibliotecária - CRB-8/9380

Sumário

INTRODUÇÃO 5

TERMINOLOGIA BÁSICA 6

1. Desenvolvimento Back-End com Node.Js 7

1.1. O que é Node.js e por que é usado para o desenvolvimento back-end. 7

1.2. Configuração do ambiente de desenvolvimento Node.js. 7

1.3. Primeira aplicação back-end simples com Node.js. 10

1.4. Configuração do ambiente Node.js e criação de um servidor simples. 14

1.5. Resumo 16

1.6. Atividade de Fixação 16

2. Conceitos Fundamentais de Node.js 18

2.1. Módulos em Node.js e como usá-los. 18

2.1.1. Como Criar um módulo. 19

2.1.2. Atividade de Fixação 22

2.2. Programação assíncrona com call-backs. 22

2.3. Promises e async/await para lidar com operações assíncronas. 22

2.4. Uso do sistema de eventos para criar aplicativos reativos. 23

2.5. Utilização de módulos e programação assíncrona em Node.js. 24

2.6. Resumo. 27

2.7. Atividade de Fixação 27

3. Criação de APIs RESTful com Express.js 29

3.1. Introdução ao Express.js e seu papel na criação de APIs. 29

3.2. Roteamento de URLs e manipulação de requisições HTTP. 32

3.3. Middlewares para processar requisições e respostas 34

3.4. Tratamento de erros e status de resposta. 37

3.5. Criação de uma API simples com Express.js. 39

3.6. Resumo. 43

3.7. Atividade de Fixação 44

4. Banco de Dados e CRUD com Node.js 45

4.1 Conexão com banco de dados usando Sequelize 45

4.2. Criação de modelos e esquemas de dados. 48

4.3. Operações CRUD: criação, leitura, atualização e exclusão de registros. 52
4.4. Uso de consultas para recuperar dados do banco de dados. 55
4.5. Implementação de operações CRUD em uma aplicação Node.js. 58
4.6. Resumo. 63
4.7. Atividade de Fixação. 64
Solução das Atividades de Fixação. 65
Os Autores. ..... **Erro! Indicador não definido.**

INTRODUÇÃO

Bem-vindo ao **Desenvolvimento de Aplicações com Node.js**! Nesta obra, você vai aprender os conceitos básicos do Node.js, uma plataforma de desenvolvimento back-end baseada no JavaScript. O Node.js permite criar aplicações web escaláveis, rápidas e eficientes, sendo uma das tecnologias mais populares e inovadoras do mercado atual. Ele é usado por grandes empresas como Netflix, PayPal, Uber, LinkedIn, entre outras. Com o Node.js, você pode criar desde simples páginas web até complexos sistemas de streaming, chat, jogos e muito mais.

Nesta obra, você vai acompanhar passo a passo como criar a sua primeira aplicação back-end com o Node.js. Você vai aprender como configurar o ambiente Node.js, como criar um servidor simples, como usar módulos para organizar o seu código, como lidar com operações assíncronas, eventos, e muito mais.

Ao final desta obra, você terá uma base sólida para continuar a sua jornada no mundo do Node.js. Você poderá explorar outras funcionalidades e bibliotecas do Node.js, bem como integrar o seu back-end com outros serviços e tecnologias. Esperamos que você aproveite este material e se divirta aprendendo o Node.js!

TERMINOLOGIA BÁSICA

Node.js: Plataforma de desenvolvimento back-end baseada em JavaScript, usada para criar aplicações web escaláveis e eficientes.

JavaScript: Linguagem de programação principal do Node.js, essencial para o desenvolvimento de aplicações web2.

Express.js: Framework para Node.js que facilita a criação de APIs RESTful e servidores HTTP.

CRUD: Acrônimo para Create, Read, Update, Delete, que são as operações básicas de manipulação de dados em um banco de dados3.

Sequelize: ORM (Object-Relational Mapping) para Node.js que facilita a interação com bancos de dados SQL.

Assíncrono: Tipo de programação onde as operações não bloqueiam a execução do código, permitindo que outras tarefas sejam realizadas simultaneamente.

Promises: Objeto usado para representar a eventual conclusão (ou falha) de uma operação assíncrona e seu valor resultante.

Async/Await: Sintaxe que simplifica o trabalho com Promises, tornando o código assíncrono mais legível e fácil de escrever.

Middleware: Funções que têm acesso ao objeto de requisição (req), ao objeto de resposta (res) e à próxima função middleware no ciclo de solicitação/resposta de uma aplicação Express.js.

1. Desenvolvimento Back-End com Node.Js

Habilidades:

- **JavaScript:** O JavaScript é a linguagem de programação principal do Node.js, portanto, é importante que desenvolva bons conhecimentos e habilidades de programação JavaScript.
- **Instalação do Node.js:** Você irá aprender a instalar o Node.Js da forma correta e testar se tudo correu bem.
- **Primeira aplicação Node.Js:** Você irá executar seus primeiros códigos no interpretador de comandos e no editor de códigos Visual Studio Code.
- **Primeira aplicação cliente x servidor:** Você irá aprender a criar um ambiente cliente e servidor, a realizar uma requisição HTTP e executar uma aplicação back-end em um navegador de internet.

1.1. O que é Node.js e por que é usado para o desenvolvimento back-end.

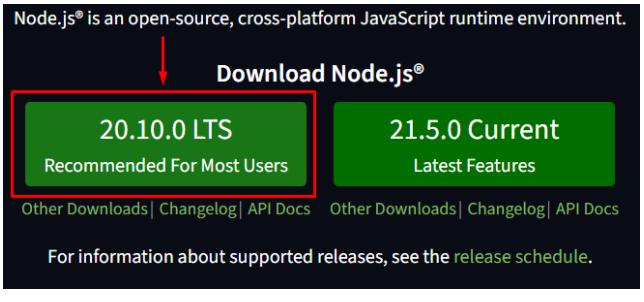
Uma aplicação back-end é a parte de um aplicativo que fica escondida do usuário. Ela é responsável por lidar com as tarefas que são necessárias para que o aplicativo funcione, mas que o usuário não vê.

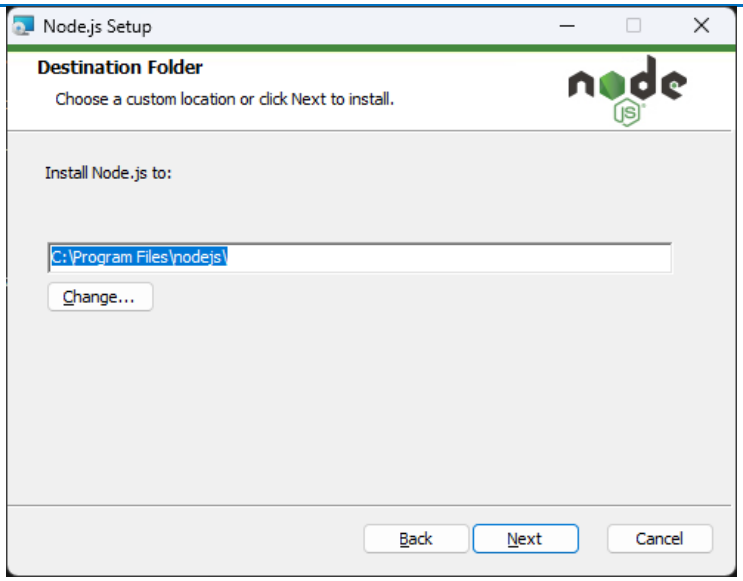
O Node.js é uma plataforma que ajuda a criar aplicativos back-end para a web, sendo uma ótima opção para criar aplicativos em tempo real, como bate-papos, jogos e aplicativos de streaming. Além disso, o código é escrito em JavaScript simples, o que significa menos tempo gasto para lidar com mudanças de código entre navegador e servidor web, não sendo necessária uma mudança na linguagem.

O Node.js é muito popular e tem muitas pessoas dispostas a ajudar. Ele também tem muitas ferramentas disponíveis para ajudar os desenvolvedores a criar aplicativos web. Algumas das habilidades mais importantes para um desenvolvedor Node.js incluem saber JavaScript, saber como usar a API do Node.js, saber como usar frameworks de aplicativos da Web, saber como lidar com programação assíncrona, saber sobre segurança na Web, saber sobre linguagens de modelagem do lado do servidor, saber sobre bancos de dados e saber sobre ferramentas de gerenciamento de pacotes.

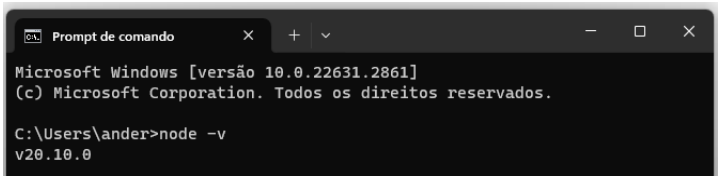
1.2. Configuração do ambiente de desenvolvimento Node.js.

Aqui estão os passos para ajudá-lo a instalar e configurar o seu ambiente de desenvolvimento Node.js.

Tema(s):	Configuração do ambiente de desenvolvimento Node.js
Nível:	Fácil
Estudo de caso:	Instalação do Node.JS
Descrição:	Configurar o ambiente de desenvolvimento Node.js é uma tarefa importante para começar a trabalhar com Node.js.
Objetivos:	∉ Instalar o ambiente de desenvolvimento Node.JS. ∉ Testar a instalação. ∉ Criar uma familiaridade com o ambiente de desenvolvimento Node.JS
Orientações:	1. **Download:** usuários do Windows podem fazer o download do Node.js no site oficial lendo o QR Code abaixo. Opte pela versão *"LTS Recommended For Most Users".* Download: https://nodejs.org/en  2. **Instalação:** execute o arquivo baixado e prossiga com a instalação. Se atente em manter o local padrão de instalação "C:\Program Files\nodejs\" para evitar ter problemas com os próximos tópicos da obra.



3. **Teste a instalação:** Após instalar o Node.js, execute o Prompt de Comandos, você poderá encontrá-lo pesquisando na barra do menu iniciar do Windows, ou pressionando as teclas *WinKey + R* e digitando o comando *CMD.* Com o prompt aberto, digite **`node –v`**, se o Node.js estiver instalado corretamente, em sua tela deverá aparecer a versão atual da instalação.



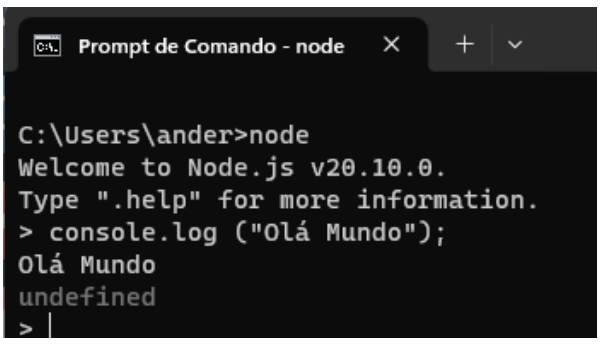
Além da instalação convencional do Node.JS em seu computador, você também pode utilizar compiladores on-line que não precisam de instalação e podem ser acessados de celulares. Existem vários compiladores e recomendamos o JDoodle, disponível devido a sua popularidade e facilidade de uso.

https://www.jdoodle.com/execute-nodejs-online/

1.3. Primeira aplicação back-end simples com Node.js.

A maneira mais simples de escrever aplicações Node.js é pelo console. Essa maneira é útil para pequenas aplicações ou para aplicações que você deseja testar rapidamente.

Para isso, basta abrir um terminal e iniciar o interpretador Node.js. Em seguida, você pode digitar o código da sua aplicação diretamente no console.

Tema(s):	Primeira aplicação back-end simples com Node.js.
Nível:	Fácil
Estudo de caso:	Aplicação "Olá Mundo" no console.
Descrição:	Aprender a escrever uma aplicação simples de back-end usando o prompt de comando do Windows e o interpretador de códigos do Node.js.
Objetivos:	∉ Utilizar o Prompt de Comandos (CMD) do Windows ∉ Utilizar o console do Node.js. ∉ Criar um código simples para uma aplicação Node.js. ∉ Criar uma familiaridade com o ambiente de desenvolvimento Node.JS
Orientações:	1. Execute o Prompt de Comandos, você poderá encontrá-lo pesquisando na barra do menu iniciar do Windows, ou pressionando as teclas *WinKey + R* e digitando o comando *CMD.* 2. Com o prompt em execução digite `node` para executar o console do interpretador Node.js 3. No prompt do node, digite `console.log ("Olá Mundo");` e pressione a tecla Enter. O resultado deverá ser conforme a imagem;  4. Você pode continuar digitando mais mensagens utilizando a função console.log(), por exemplo, você pode agora informar no prompt