



UNINOVE

IOT (Internet of Things)



Conteúdo da Aula

- Node.js
- Servidor Web (Apache2 e NodeJs)
- Express (Framework)
- React
- Bootstrap
- Comunicação com a Placa
- Interface WEB
- Exemplos de possibilidades



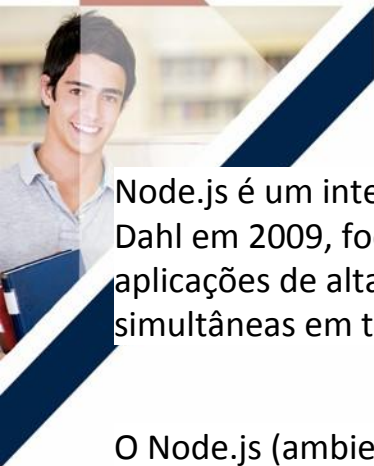
UNINOVE

Mestrando Prof. Rafael Rojas

E-mail: rafael.rojas@uni9.pro.br



About Node.js



Node.js é um interpretador de JavaScript assíncrono com código aberto orientado a eventos, criado por Ryan Dahl em 2009, focado em migrar a programação do Javascript do cliente (frontend) para os servidores, criando aplicações de alta escalabilidade (como um servidor web), manipulando milhares de conexões/eventos simultâneas em tempo real numa única máquina física.

O Node.js (ambiente de execução Javascript no servidor) foi implementado baseado no interpretador V8 JavaScript Engine (interpretador de JavaScript em C++ com código aberto do Google, utilizado no Chrome), com desenvolvimento mantido pela fundação Node.js em parceria com a Linux Foundation.

No exemplo "olá mundo" a seguir, muitas conexões podem ser tratadas simultaneamente. Em cada conexão, o retorno de chamada é acionado, mas se não houver trabalho a ser feito, o Node.js entrará em suspensão.

Alguns comandos

`npm init -y`

`npm start`

`Ctrl^C`

Exemplo de código Node.js

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

```
const hostname = '127.0.0.1';  
const port = 3000;
```

```
const server = http.createServer((req, res) => {  
  res.statusCode = 200;  
  res.setHeader('Content-Type', 'text/plain');  
  res.end('Hello World');  
});
```

```
server.listen(port, hostname, () => {  
  console.log(`Server running at http://${hostname}:${port}/`);  
});
```

Vantagens de uso

- **Flexibilidade**

O Node.js possui o gerenciador de pacotes reusáveis NPM (Node Package Manager), o maior repositório de softwares, dando ao interpretador um potencial a ser utilizada em qualquer situação. O pacote mais conhecido é o Express.js, um framework completo para desenvolvimento de aplicações Web.

- **Leveza**

Um ambiente Node.js não exige muitos recursos computacionais tradicionais. Se utilizado em conjunto com a ferramentas Docker, o ganho na velocidade de deploy e replicação de máquinas pode ser muito significativo e, em ambientes escaláveis, de micro-serviços e serverless, representa menos custo e mais eficiência.[3] Um exemplo de uso é uma aplicação de conversa/bate papo (webchat).

- **Suporte**

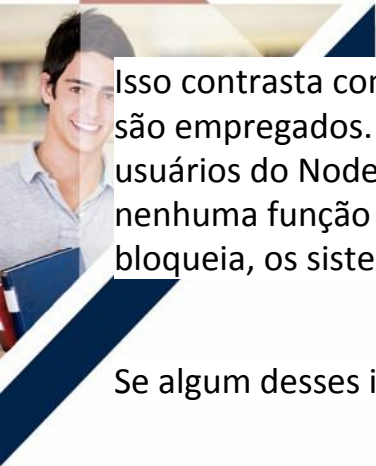
Conta com suporte das principais empresas de serviços de armazenamento na nuvem, como a AWS, Google Cloud e, Microsoft Azure, com suporte nativo ao Node.JS.

- **Produtividade**

O gerenciador NPM fornece pacotes reusáveis disponível gratuitamente para uso em grupo;

Mesma linguagem no frontend e backend, que pode representar ganhos de reuso de código e criação de equipes multidisciplinares, reaproveitando recursos;

A possibilidade de deploys e interações mais rápidas, e resolução de problemas "on the fly", também permitem a criação de soluções próprias e inovadoras, como a Uber faz para resolver alguns problemas.



Isso contrasta com o modelo de concorrência mais comum de hoje, no qual os threads do sistema operacional são empregados. A rede baseada em thread é relativamente ineficiente e muito difícil de usar. Além disso, os usuários do Node.js estão livres de preocupações de bloquear o processo, pois não há bloqueios. Quase nenhuma função no Node.js executa E / S diretamente, portanto, o processo nunca bloqueia. Como nada bloqueia, os sistemas escaláveis são muito razoáveis para se desenvolver no Node.js.

Se algum desses idiomas não estiver familiarizado, há um artigo completo sobre Bloqueio x Não-bloqueio.

Mais informações em <https://nodejs.org/>

Resumindo, o NodeJs é um interpretador JS que não depende do navegador.



Servidor Web

Já mencionamos servidores web anteriormente e existem opções online como [os 10 melhores hosts em 2020](#).

Uma boa opção é o Apache que hoje possui versões para inúmeros SO, mas o NodeJs também pode ser usado como servidor web, pois existem diversos módulos de servidor web, conexão à banco de dados e etc.

A definição de servidor web é: **Servidor web** é um *software* responsável por aceitar pedidos em HTTP de clientes, geralmente os navegadores, e servi-los com respostas em HTTP, incluindo opcionalmente dados, que geralmente são páginas *web*, tais como documentos em HTML com objetos embutidos (imagens, etc.) ou um computador que executa um programa que provê a funcionalidade descrita anteriormente. O mais popular, e mais utilizado no mundo, é o **servidor Apache** (*software livre*). A Microsoft possui a sua própria solução denominada **IIS** (*Internet Information Services*).

Express

- **Aplicativos da Web**

O Express é um framework para aplicativo da web do Node.js mínimo e flexível que fornece um conjunto robusto de recursos para aplicativos web e móvel.

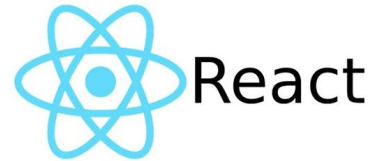
- **APIs**

Com uma miríade de métodos utilitários HTTP e middleware a seu dispor, criar uma API robusta é rápido e fácil.

- **Desempenho**

O Express fornece uma camada fina de recursos fundamentais para aplicativos da web, sem obscurecer os recursos do Node.js que você conhece e ama.

React



Biblioteca para a criação de interfaces.

Muito utilizado para a criação de SPAs (Single-Page Applications)

Bootstrap

O Bootstrap é uma ferramenta gratuita para desenvolvimento HTML, CSS e JS. Crie protótipos rapidamente ou aplicações completas com nossas variáveis e mixins Sass, sistemas de grid responsivo, componentes pré-construídos e poderosos plugins com jQuery.

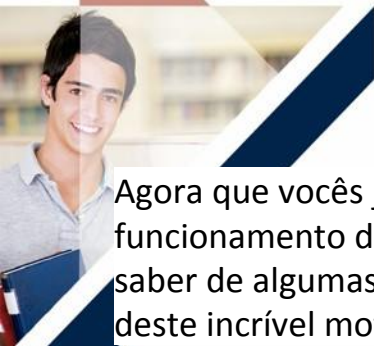
Exemplos de páginas feitas em bootstrap:

https://themes.getbootstrap.com/preview/?theme_id=47394

https://themes.getbootstrap.com/preview/?theme_id=1651

Para mais informações: <https://getbootstrap.com.br/>

Comunicação com a Placa



Agora que vocês já entenderam o funcionamento do Node.js é importante saber de algumas das possibilidades deste incrível motor!

Com isso, vamos instalar os seguintes módulos no nosso node.js.

Obs. Todos os módulos instalados vão ficar registrados no arquivo package.json como mostrado no exemplo abaixo;

```
1 {  
2   "name": "main",  
3   "version": "1.0.0",  
4   "description": "Arduino na Web",  
5   "main": "main.js",  
6   "scripts": {  
7     "test": "echo \\\"Error: no test specified\\\" && exit 1",  
8     "start": "node main.js"  
9   },  
10  "author": "Rafael Rojas",  
11  "license": "ISC",  
12  "dependencies": {  
13    "chart.js": "^2.9.3",  
14    "express": "^4.17.1",  
15    "mariadb": "^2.3.1",  
16    "mysql": "^2.18.1",  
17    "mysql2": "^2.1.0",  
18    "pg": "^8.2.1",  
19    "pg-hstore": "^2.3.3",  
20    "sequelize": "^5.21.10",  
21    "serialport": "^9.0.0",  
22    "socket.io": "^2.3.0",  
23    "sqlite3": "^4.2.0"  
24  }  
25 }
```

Nosso caso

No nosso arquivo .json temos as seguintes instalações:

"chart.js": "^2.9.3", → Módulo para gerar as classes dos gráficos que vai ler o potenciômetro

"express": "^4.17.1", → Módulo para simplificar a comunicação com a interface WEB

"mariadb": "^2.3.1", → Módulo de comunicação com banco de dados MariaDB

"mysql": "^2.18.1", → Módulo de comunicação com banco de dados MySQL

"mysql2": "^2.1.0", → Módulo de comunicação com banco de dados MySQL2

"sqlite3": "^4.2.0", → Módulo de comunicação com banco de dados SqlLite3

"pg": "^8.2.1", → Módulo nativo para comunicação do JS

"pg-hstore": "^2.3.3", → Um pacote de nós para serializar e desserializar dados JSON para o formato hstore

"sequelize": "^5.21.10", → Módulo para otimizar a comunicação com os bancos de dados

"serialport": "^9.0.0", → Módulo de comunicação entre o JS e as portas seriais de algum dispositivo (ex Arduino)

"socket.io": "^2.3.0", → Módulo de comunicação com dispositivos de Input e Output

Arquivos

No nosso projeto, usaremos 5 arquivos:

main.js → Página que será iniciada primeiramente quando demos o npm start, esse arquivo que controla todos outros arquivos.

btn.js → Página que é responsável pela ação do botão de acender o LED

grafico.js → Página com os dados recebidos do potenciômetro para geração do gráfico. Aqui que definimos toda estética do gráfico também.

bd.js → Página que recebe os dados do LED e do potenciômetro e guarda em um BD

index.html → Página de apresentação gráfica das informações, responsáveis por enviar e receber os dados da placa Arduino.

Apresentando a Página

Agora, vamos analisar o código e ver como ele funciona.

Após isso vamos trabalhar com o código e pensar nos nossos futuros projetos.