

Conteúdo

Módulo 1: Introdução ao Node.JS

- O que é o Express.JS
- Criando servidor HTTP com Express.JS
- Atividade sobre Rotas com Express.JS
- Prática - Criando Landing Page com Express.JS

Tendo sua versão inicial lançada no ano de 2010, o Express.js (ou somente Express) é um Framework para o desenvolvimento de aplicações JavaScript com o [Node.js](#).

De código aberto, sobre a licença MIT, o Express.js foi desenvolvido para otimizar a construção de aplicações web e APIs, tornando-se um dos Frameworks mais populares da internet e que utiliza o Node para execução do javascript como linguagem de back-end.

Relembrando Framework

Em palavras mais simples, o framework é um facilitador no desenvolvimento de diversas aplicações e, sem dúvida, sua utilização poupa tempo e custos para quem utiliza, pois de forma mais básica, é um conjunto de bibliotecas utilizadas para criar uma base, onde as aplicações são construídas, um otimizador de recursos.

Possui como principal objetivo resolver problemas recorrentes com uma abordagem mais genérica. Ele permite ao desenvolvedor focar nos “problemas” da aplicação, não na arquitetura e configurações.

Um pouco sobre o Node.js

O Node.js por sua vez, é uma plataforma open source que permite a execução de código JavaScript a nível front-end e back-end.

Em palavras mais simples, o Node.js é uma forma de executar o JavaScript do lado do servidor de uma aplicação.

Teve seu lançamento em 2009 sobre a licença MIT e é utilizado por diversas grandes empresas como LinkedIn, Groupon, PayPal, entre outras.

O Node.js possui aplicabilidade em diversos meios, dentre eles podemos citar:

- Criação de aplicações de chats e mensagens instantâneas;
- Criação de API's escaláveis;
- Aplicações web que funcionam em real-time;
- Aplicações CLI (Client Line Interface), entre outros.

De volta ao Express.js

O Express.js é um Framework rápido e um dos mais utilizados em conjunto com o Node.js, facilitando no desenvolvimento de aplicações back-end e até, em conjunto com sistemas de templates, aplicações full-stack.

Escrito em JavaScript, o Express.js é utilizado por diversas empresas ao redor do mundo, dentre elas a Fox Sports, PayPal, IBM, Uber, entre outras.

Muito popular tanto em grandes empresas quanto na comunidade, o Express facilita a criação de aplicações utilizando o Node em conjunto com o JavaScript, tornando este ecossistema ainda mais poderoso.

Características do Express.js

O Express é um framework incrível e possui diversas características que facilitam o desenvolvimento de nossas aplicações. Dentre suas principais características, podemos citar:

- Possui um sistema de rotas completo;
- Possibilita o tratamento de exceções dentro da aplicação;
- Permite a integração de vários sistemas de templates que facilitam a criação de páginas web para suas aplicações;
- Gerencia diferentes requisições HTTP com seus mais diversos verbos;
- Feito para a criação rápida de aplicações utilizando um conjunto pequeno de arquivos e pastas;

Podemos concluir que...

O Express.js é um ótimo Framework para o desenvolvimento de aplicações utilizando o Node. No [site do Express.js](https://expressjs.com/) podemos acessar toda a sua documentação.

<https://expressjs.com/>

Criando um servidor HTTP com Express

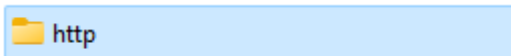
O Express é um framework utilizado em conjunto com o Node.js. Ele possui características que facilitam o desenvolvimento de nossas aplicações web e APIs, como um sistema de rotas completo, tratamento de exceções, gerenciamento de requisições HTTP e muito mais.

O Express é um framework utilizado em conjunto com o Node.js. Ele possui características que facilitam o desenvolvimento de nossas aplicações web e APIs, como um sistema de rotas completo, tratamento de exceções, gerenciamento de requisições HTTP e muito mais.

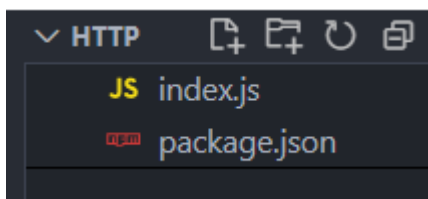
Instalação do Express

Primeiramente, você deve ter o ambiente Node.js instalado.

Logo, vamos criar um novo projeto na pasta raiz 'http', e dentro dela vamos iniciar o projeto node.js pelo comando `npm install`. Portanto, a estrutura do projeto ficará da seguinte forma:



E abram no visual studio code:

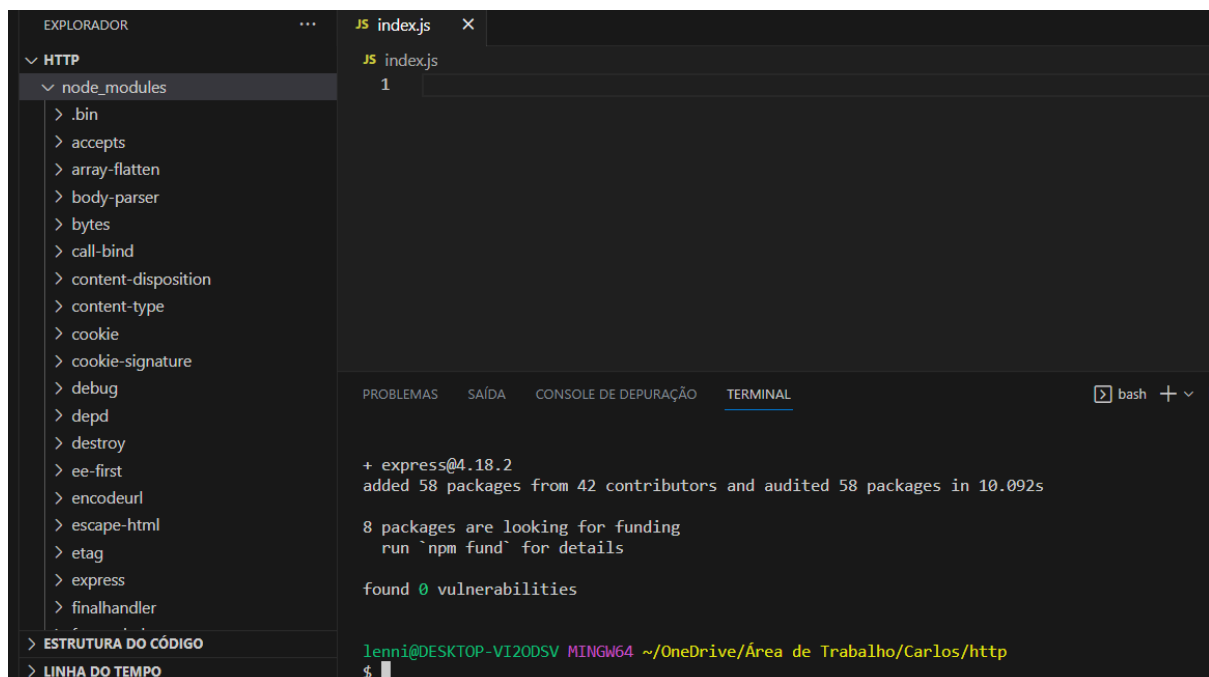


Módulos do Node instalados com a opção `--save` são incluídas na lista `dependencies` no arquivo `package.json`. Posteriormente, executando `npm install` no diretório `app` irá automaticamente instalar os módulos na lista de dependências.

Agora precisamos instalar o Express, para isso vamos utilizar o comando: `npm install --save express`. Onde a flag `--save` é utilizada para salvar as dependências do nosso projeto no arquivo `package.json`. Ou seja, após a instalação o arquivo `package.json` ficará similar ao exemplo abaixo:

```
{
  "name": "http-server",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "express": "^4.17.1"
  }
}
```

Ao instalarmos o Express, teremos instalado dentro da pasta node_modules outras libs que são dependências necessárias para o Express funcionar ok?



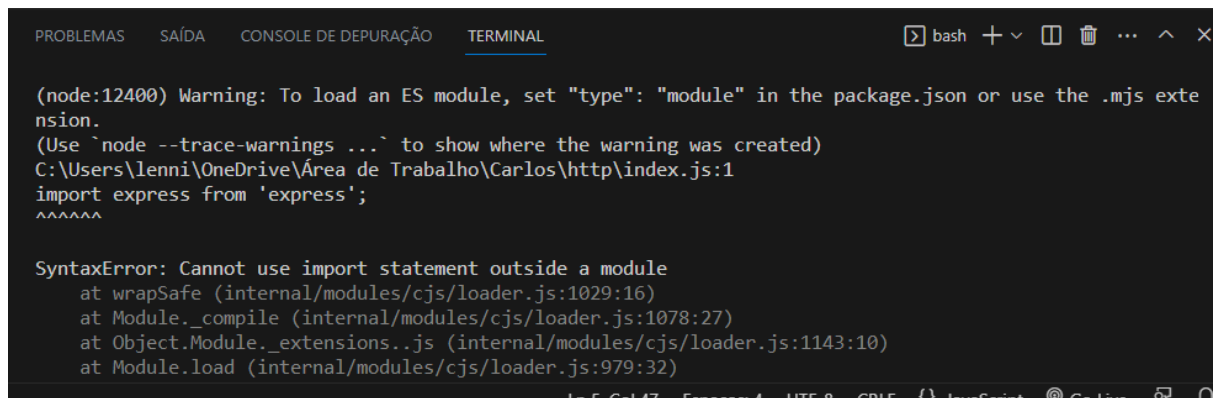
Criando servidor HTTP com Express

Após a instalação do Express, podemos efetivamente colocar a mão na massa.

Para isso, vamos importar o Express em nosso arquivo index.js.

```
import express from 'express';
```

Perceba que utilizamos o padrão do ES6 para importar o Express, portanto, não podemos esquecer de adicionar o atributo `"type": "module"` no arquivo `package.json`, caso contrário teremos um erro de importação.

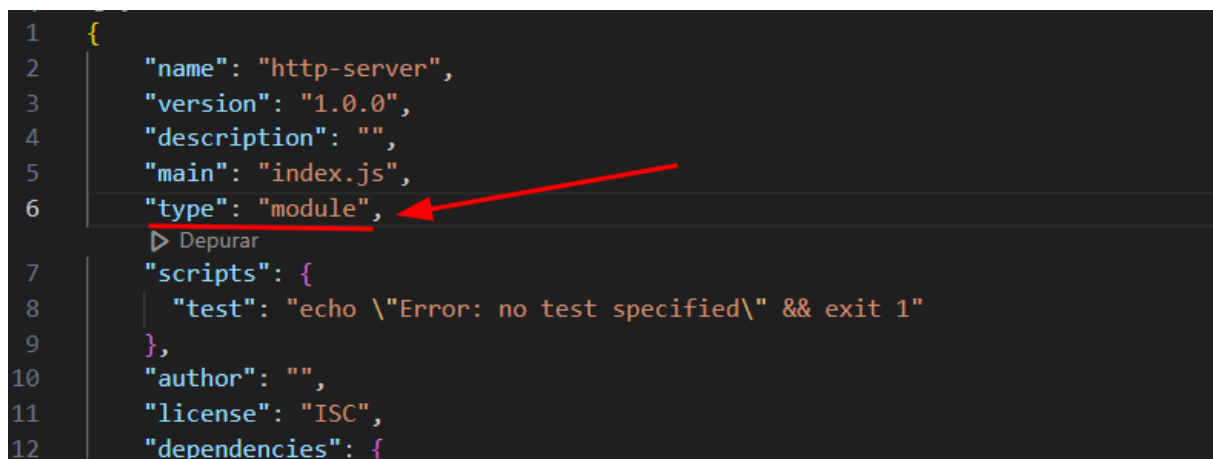


```
PROBLEMAS  SAÍDA  CONSOLE DE DEPURAÇÃO  TERMINAL

(node:12400) Warning: To load an ES module, set "type": "module" in the package.json or use the .mjs extension.
(Use `node --trace-warnings ...` to show where the warning was created)
C:\Users\lenni\OneDrive\Área de Trabalho\Carlos\http\index.js:1
import express from 'express';
^^^^^^

SyntaxError: Cannot use import statement outside a module
    at wrapSafe (internal/modules/cjs/loader.js:1029:16)
    at Module._compile (internal/modules/cjs/loader.js:1078:27)
    at Object.Module._extensions..js (internal/modules/cjs/loader.js:1143:10)
    at Module.load (internal/modules/cjs/loader.js:979:32)
```

Vai ocasionar este erro ao tentar rodar o projeto então antes precisamos add no `package.json` esse comando:



```
1  {
2    "name": "http-server",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "type": "module",
7    "scripts": {
8      "test": "echo \\\"Error: no test specified\\\" && exit 1"
9    },
10   "author": "",
11   "license": "ISC",
12   "dependencies": {
```

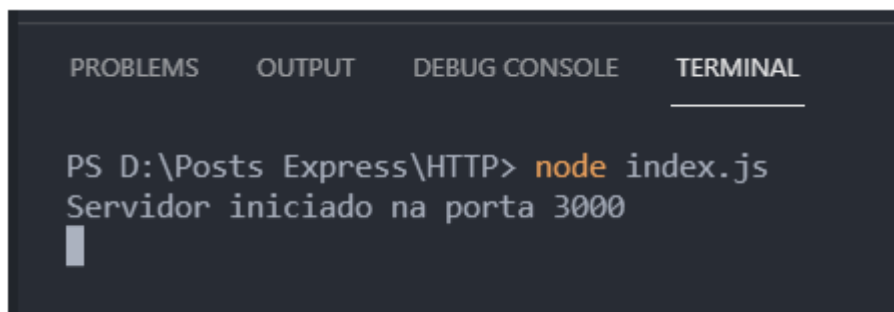
Agora vamos declarar a constante `app` e utilizar o método `listen()` para subir nosso servidor, vamos utilizar a porta 3000. Teremos o seguinte código:

```
import express from 'express';

const app = express();

app.listen(3000, () =>
  console.log('Servidor iniciado na porta 3000')
);
```

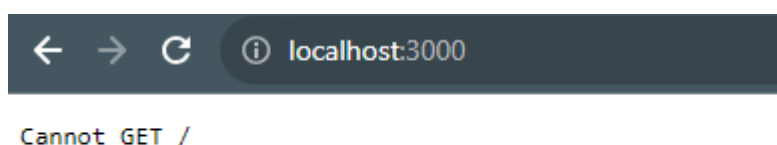
Neste caso passamos como parâmetro a porta (3000) e uma arrow function para exibir um log com a mensagem de que tudo está ok.



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS D:\Posts Express\HTTP> node index.js
Servidor iniciado na porta 3000
```

Porém ao acessar nosso servidor pela porta 3000 (<http://localhost:3000/>), vamos obter a mensagem “Cannot GET /”, afinal, não configuramos nenhuma rota para a requisição feita pelo navegador.



Para configurar essa rota, vamos utilizar o método `get()`, que recebe dois parâmetros, a rota e o que vamos retornar nesta requisição (request e response).

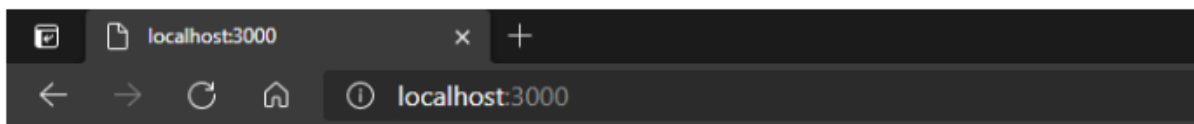

```
import express from 'express';

const app = express();

app.listen(3000, () =>
  console.log('Servidor iniciado na porta 3000')
);

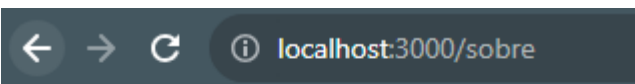
app.get('/', (req, res) =>
  res.send('<h1 style="color: blue">CRIANDO UM SERVIDOR COM EXPRESS.JS</h1>')
);
```

Ao reiniciar o servidor, portanto, retornará o seguinte conteúdo conforme o exemplo acima:

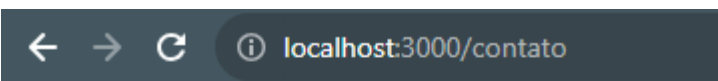


CRIANDO UM SERVIDOR COM EXPRESS.JS

Bacana então agora vamos praticar? Crie você 3 rotas novas, para ficar assim :



PÁGINA SOBRE



PÁGINA CONTATO

PÁGINA EMPRESA

Beleza? Parabéns por ter conseguido.

Vamos praticar? Criaremos agora uma página com bootstrap para entendermos como funciona a rota e criação de página com express.

Para criar este projeto, precisaremos de:

- NodeJS + NPM;
- Um editor de código de sua preferência. Recomendo utilizar o Visual Studio Code.

Vamos começar!

De início, precisamos criar nossa pasta que conterá todos os arquivos do projeto. Utilizando um terminal, crie a pasta chamada `node_website` com o comando:

Dentro da nossa pasta, criaremos dois arquivos:

- `package.json` (conterá nossas dependências);
- `server.js` (conterá nossos scripts para criação do servidor).

Crie também uma pasta chamada `node-example-website` e, dentro desta pasta, crie outra pasta chamada `express`. Essa pasta `express` conterá nosso arquivo `index.html`.

Ao código!

No nosso arquivo `package.json`, precisamos adicionar o nome do nosso projeto, versão, os scripts e as dependências. O seu arquivo deverá ficar assim:

```
{  
  "name": "node-website",
```

```
"version": "1.0.0",  
"description": "",  
"scripts": {  
  "start": "node server.js"  
},  
"dependencies": {  
  "express": "^4.17.1"  
}  
}
```

para isso basta criar o comando :

npm init -y

Assim, já criamos nosso arquivo de dependências.

Agora, precisamos criar nosso servidor. No nosso arquivo server.js, iremos:

- Fazer a requisição dos pacotes que utilizaremos;
- Criaremos o servidor na porta que escolhermos utilizar.

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

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

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

```
const app = express();
```

```
app.use(express.json());
```

```
app.use(express.static('express'));
```

```
// URL padrão do site
```

UniSENAI

```
app.use('/', (req, res) => {
```

```
  res.sendFile(path.join(__dirname+'/index.html'));
```

```
});
```

```
const server = http.createServer(app);
```

```
const port = 3000;
```

```
server.listen(port);
```

```
console.debug('Server inicializado na porta ' + port);
```

```
JS server.js > app.use('/') callback
1  const http = require('http');
2  const express = require('express');
3  const path = require('path');
4
5  const app = express();
6  app.use(express.json());
7  app.use(express.static('express'));
8
9  // URL padrão do site
10 app.use('/', (req, res) => {
11   res.sendFile(path.join(__dirname + '/index.html'));
12 });
13
14 const server = http.createServer(app);
15 const port = 3000;
16 server.listen(port);
17
18 console.debug('Server inicializado na porta ' + port);
```

Neste projeto, estamos utilizando a porta 3000.

Agora, precisamos instalar as outras dependências e módulos do node no nosso projeto. Dentro da pasta node-example-website, rode o seguinte comando:

```
npm install
```

Esse comando vai instalar as dependências definidas no nosso package.json.

No nosso arquivo index.html, utilizaremos um exemplo do W3CSS, chamado Cafe Template. Você pode utilizar o template de sua preferência ou criar o seu próprio!

<https://www.w3schools.com/w3css/default.asp>

o w3 css é um concorrente do bootstrap muito utilizado pelo mundo e tem alguns exemplos interessantes pegamos o do café aqui :

https://www.w3schools.com/w3css/w3css_templates.asp

UniSENAI

```
<!DOCTYPE html>

<html>

<title>W3.CSS Template</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width,
initial-scale=1">

<link rel="stylesheet"
href="https://www.w3schools.com/w3css/4/w3.css">

<link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Inconsolata">

<style>

body, html {

    height: 100%;

    font-family: "Inconsolata", sans-serif;

}

.bgimg {

    background-position: center;

    background-size: cover;
```

```
background-image:
url("https://www.w3schools.com/w3images/coffeehouse.jpg");

min-height: 75%;

}
```

```
.menu {
```

```
display: none;
```

```
}
```

```
</style>
```

```
<body>
```

```
<!-- Links (sit on top) -->
```

```
<div class="w3-top">
```

```
<div class="w3-row w3-padding w3-black">
```

```
<div class="w3-col s3">
```

```
<a href="#" class="w3-button w3-block w3-black">HOME</a>
```

```
</div>
```

```
<div class="w3-col s3">
```

```
<a href="#about" class="w3-button w3-block
w3-black">ABOUT</a>
```

```
</div>
```

UniSENAI

```
<div class="w3-col s3">
```

```
  <a href="#menu" class="w3-button w3-block  
w3-black">MENU</a>
```

```
</div>
```

```
<div class="w3-col s3">
```

```
  <a href="#where" class="w3-button w3-block  
w3-black">WHERE</a>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<!-- Header with image -->
```

```
<header class="bgimg w3-display-container w3-grayscale-min"  
id="home">
```

```
  <div class="w3-display-bottomleft w3-center w3-padding-large  
w3-hide-small">
```

```
    <span class="w3-tag">Open from 6am to 5pm</span>
```

```
  </div>
```

```
<div class="w3-display-middle w3-center">
```

```
  <span class="w3-text-white"  
style="font-size:90px">the<br>Cafe</span>
```

```
</div>
```



```
<div class="w3-display-bottomright w3-center  
w3-padding-large">
```

```
<span class="w3-text-white">15 Adr street, 5015</span>
```

```
</div>
```

```
</header>
```

```
<!-- Add a background color and large text to the whole page  
-->
```

```
<div class="w3-sand w3-grayscale w3-large">
```

```
<!-- About Container -->
```

```
<div class="w3-container" id="about">
```

```
<div class="w3-content" style="max-width:700px">
```

```
<h5 class="w3-center w3-padding-64"><span class="w3-tag  
w3-wide">ABOUT THE CAFE</span></h5>
```

```
<p>The Cafe was founded in blabla by Mr. Smith in lorem  
ipsum dolor sit amet, consectetur adipiscing elit, sed do  
eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut  
enim ad minim veniam, quis nostrud exercitation ullamco  
laboris nisi ut aliquip ex ea commodo consequat.</p>
```

```
<p>In addition to our full espresso and brew bar menu, we  
serve fresh made-to-order breakfast and lunch sandwiches, as  
well as a selection of sides and salads and other good  
stuff.</p>
```

```
<div class="w3-panel w3-leftbar w3-light-grey">
```

```
<p><i>"Use products from nature for what it's worth -  
but never too early, nor too late." Fresh is the new  
sweet.</i></p>
```

```
<p>Chef, Coffeeist and Owner: Liam Brown</p>
```

```
</div>
```

```

```

```
<p><strong>Opening hours:</strong> everyday from 6am to  
5pm.</p>
```

```
<p><strong>Address:</strong> 15 Adr street, 5015, NY</p>
```

```
</div>
```

```
</div>
```

```
<!-- Menu Container -->
```

```
<div class="w3-container" id="menu">
```

```
<div class="w3-content" style="max-width:700px">
```

```
<h5 class="w3-center w3-padding-48"><span class="w3-tag  
w3-wide">THE MENU</span></h5>
```

```
<div class="w3-row w3-center w3-card w3-padding">
```

```
<a href="javascript:void(0)" onclick="openMenu(event,  
'Eat');" id="myLink">
```

```
<div class="w3-col s6 tablink">Eat</div>
```

```
</a>
```

```
<a href="javascript:void(0)" onclick="openMenu(event, 'Drinks');">
```

```
<div class="w3-col s6 tablink">Drink</div>
```

```
</a>
```

```
</div>
```

```
<div id="Eat" class="w3-container menu w3-padding-48 w3-card">
```

```
<h5>Bread Basket</h5>
```

```
<p class="w3-text-grey">Assortment of fresh baked fruit breads and muffins 5.50</p><br>
```

```
<h5>Honey Almond Granola with Fruits</h5>
```

```
<p class="w3-text-grey">Natural cereal of honey toasted oats, raisins, almonds and dates 7.00</p><br>
```

```
<h5>Belgian Waffle</h5>
```

```
<p class="w3-text-grey">Vanilla flavored batter with malted flour 7.50</p><br>
```

```
<h5>Scrambled eggs</h5>
```

UniSENAI

```
<p class="w3-text-grey">Scrambled eggs, roasted red  
pepper and garlic, with green onions 7.50</p><br>
```

```
<h5>Blueberry Pancakes</h5>
```

```
<p class="w3-text-grey">With syrup, butter and lots of  
berries 8.50</p>
```

```
</div>
```

```
<div id="Drinks" class="w3-container menu w3-padding-48  
w3-card">
```

```
<h5>Coffee</h5>
```

```
<p class="w3-text-grey">Regular coffee 2.50</p><br>
```

```
<h5>Chocolato</h5>
```

```
<p class="w3-text-grey">Chocolate espresso with milk  
4.50</p><br>
```

```
<h5>Corretto</h5>
```

```
<p class="w3-text-grey">Whiskey and coffee 5.00</p><br>
```

```
<h5>Iced tea</h5>
```

```
<p class="w3-text-grey">Hot tea, except not hot  
3.00</p><br>
```

```
<h5>Soda</h5>
```

```
<p class="w3-text-grey">Coke, Sprite, Fanta, etc.  
2.50</p>
```

```
</div>
```

```

```

```
</div>
```

```
</div>
```

```
<!-- Contact/Area Container -->
```

```
<div class="w3-container" id="where"  
style="padding-bottom:32px;">
```

```
<div class="w3-content" style="max-width:700px">
```

```
<h5 class="w3-center w3-padding-48"><span class="w3-tag  
w3-wide">WHERE TO FIND US</span></h5>
```

```
<p>Find us at some address at some place.</p>
```

```

```

```
<p><span class="w3-tag">FYI!</span> We offer full-service  
catering for any event, large or small. We understand your  
needs and we will cater the food to satisfy the biggerst  
criteria of them all, both look and taste.</p>
```

```
<p><strong>Reserve</strong> a table, ask for today's
special or just send us a message:</p>
```

```
<form action="/action_page.php" target="_blank">
```

```
<p><input class="w3-input w3-padding-16 w3-border"
type="text" placeholder="Name" required name="Name"></p>
```

```
<p><input class="w3-input w3-padding-16 w3-border"
type="number" placeholder="How many people" required
name="People"></p>
```

```
<p><input class="w3-input w3-padding-16 w3-border"
type="datetime-local" placeholder="Date and time" required
name="date" value="2017-11-16T20:00"></p>
```

```
<p><input class="w3-input w3-padding-16 w3-border"
type="text" placeholder="Message \ Special requirements"
required name="Message"></p>
```

```
<p><button class="w3-button w3-black" type="submit">SEND
MESSAGE</button></p>
```

```
</form>
```

```
</div>
```

```
</div>
```

```
<!-- End page content -->
```

```
</div>
```

```
<!-- Footer -->
```

```
<footer class="w3-center w3-light-grey w3-padding-48 w3-large">
```

```
  <p>Powered by <a href="https://www.w3schools.com/w3css/default.asp" title="W3.CSS" target="_blank" class="w3-hover-text-green">w3.css</a></p>
```

```
</footer>
```

```
<script>
```

```
// Tabbed Menu
```

```
function openMenu(evt, menuName) {
```

```
  var i, x, tablinks;
```

```
  x = document.getElementsByClassName("menu");
```

```
  for (i = 0; i < x.length; i++) {
```

```
    x[i].style.display = "none";
```

```
  }
```

```
  tablinks = document.getElementsByClassName("tablink");
```

```
  for (i = 0; i < x.length; i++) {
```

```
    tablinks[i].className = tablinks[i].className.replace("w3-dark-grey", "");
```

```
  }
```

```
  document.getElementById(menuName).style.display = "block";
```

```
    evt.currentTarget.firstChild.className += "w3-dark-grey";
```

```
}
```

```
document.getElementById("myLink").click();
```

```
</script>
```

```
</body>
```

```
</html>
```

Você também pode adicionar o css e js em arquivos separados!

Para inicializar o servidor, utilizamos o comando:

```
npm start
```

Para testar, abra o seu browser e digite:

localhost:3000

E veja o seu site rodando!

UniSENAI

