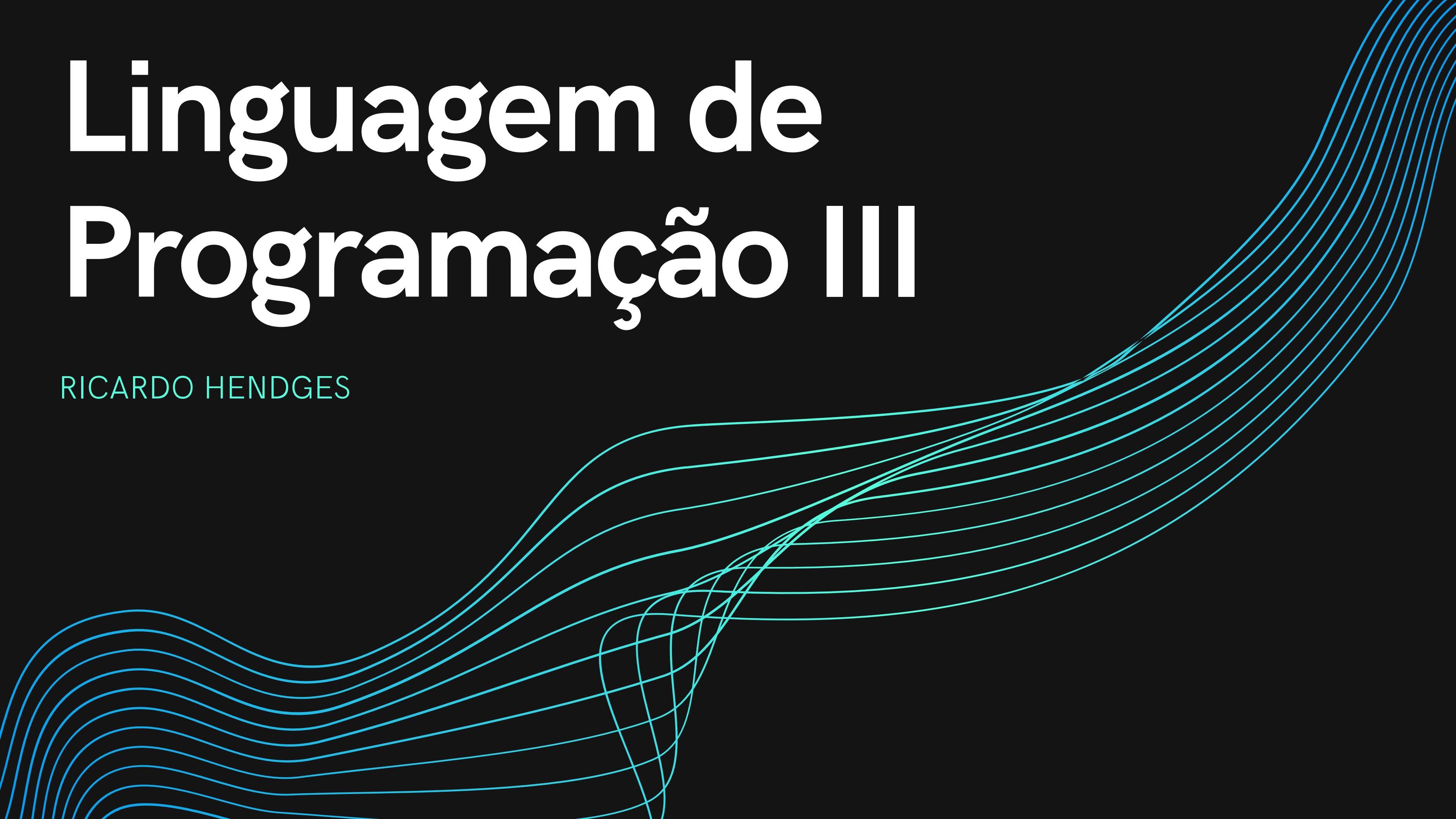


Linguagem de Programação III

RICARDO HENDGES



Quem sou eu?

RICARDO HENDGES

Nascimento: Modelo - 04/05/1994.

Idade: 28 Anos.

Pai da Ellie!!!

Formação:

Sistemas de Informação - Horus Faculdades (2015)

Pós - Desenvolvimento de Jogos Digitais - Unyleya (2017)

Pós - Machine Learning - IGTI (2020)

Cursando:

Pós - Arquiteto de Soluções - PUC Minas



Quem sou eu?

RICARDO HENDGES

Experiencia:

Grupo Dass - Expedição (2010 - 2012)

JB Software - Estagiário (2012 - 2013)

H2J - Dev. Delphi (2013 - 2017)

Grupo Dass - Dev. Pleno Delphi (2017 - 2019)

Horus - Professor (2019 - Atual)

Questor - Dev. Sr. Delphi (2019)

G&AR - Dev. Games (2019 - 2020)

Grupo Dass - Dev. Web (2020 - 2022)

Grupo Dass - Arquiteto de Soluções (2022 - Atual)

Contatos



WHATSAPP

(49) 98847-6877



FACEBOOK

Ricardo Hendges



LINKEDIN

Ricardo Hendges



EMAIL

ricardohe04@gmail.com



node.js

Node.js é uma plataforma de código aberto baseada em JavaScript para desenvolvimento de aplicações server-side. Ele permite que os desenvolvedores escrevam aplicativos para a web usando JavaScript, construído sobre o motor JavaScript V8 do Google Chrome.

Node.js é uma escolha popular para aplicativos web em tempo real, como chats, jogos e aplicativos de mensagens, devido à sua capacidade de lidar com muitas conexões simultâneas com uma baixa latência. Ele também é amplamente utilizado para construir aplicativos de back-end, APIs e microserviços.

Instale o Node.js em seu computador.

LPIII-2023

`mkdir LPIII2023` - Criação de uma pasta para controle de atividades, estudos e trabalhos da matéria.

`github` - Criação de um novo projeto publico para inicio dos estudos. (backend)

* `git clone <link>` - Dentro do repositório criado (LPIII2022)

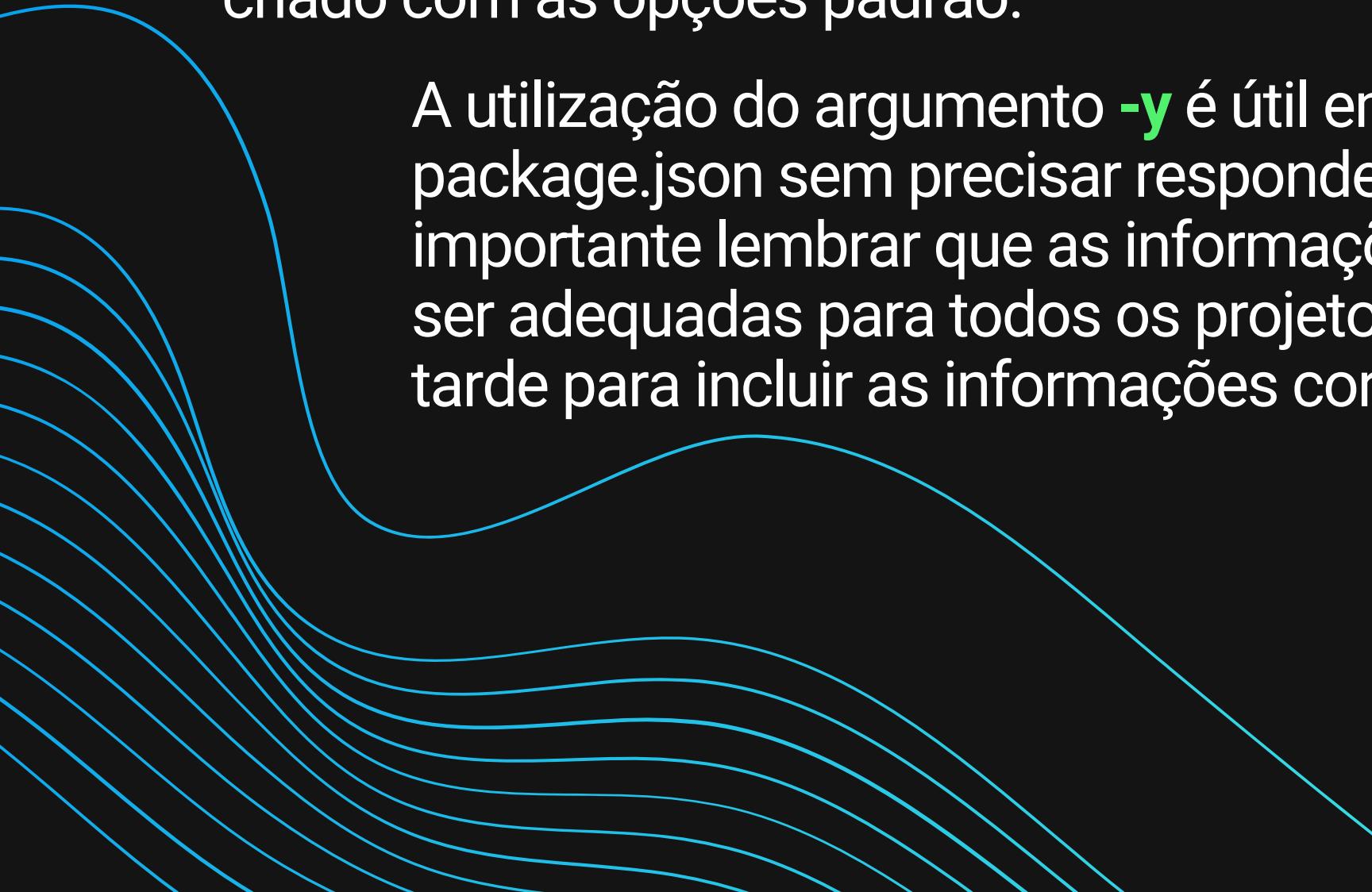
npm init

-y somente para nao vir perguntas. . .

O comando **npm init -y** é utilizado para criar um arquivo package.json em um projeto Node.js. O arquivo package.json é um arquivo de configuração que contém informações sobre o projeto, incluindo informações sobre as dependências do projeto, scripts para execução, entre outras informações.

O comando **npm init -y** inicializa o processo de criação do arquivo package.json, respondendo automaticamente "yes" a todas as perguntas que são feitas durante o processo de inicialização. Isso significa que as opções padrão serão aceitas para todas as perguntas e o arquivo package.json será criado com as opções padrão.

A utilização do argumento **-y** é útil em situações onde você deseja criar rapidamente o arquivo package.json sem precisar responder a todas as perguntas interativamente. No entanto, é importante lembrar que as informações padrão incluídas no arquivo package.json podem não ser adequadas para todos os projetos, e você pode precisar editar manualmente o arquivo mais tarde para incluir as informações corretas.



node.js

Para desenvolver uma aplicação back-end com Node.js, precisamos ter conhecimento básico de JavaScript e de algumas bibliotecas do Node.js.

Express

Express é uma biblioteca de roteamento e middleware para Node.js que facilita o desenvolvimento de aplicações web back-end. Quando usado com Node.js, o Express permite que você crie rotas, manipule requisições HTTP e forneça respostas para o cliente de uma maneira clara e concisa.

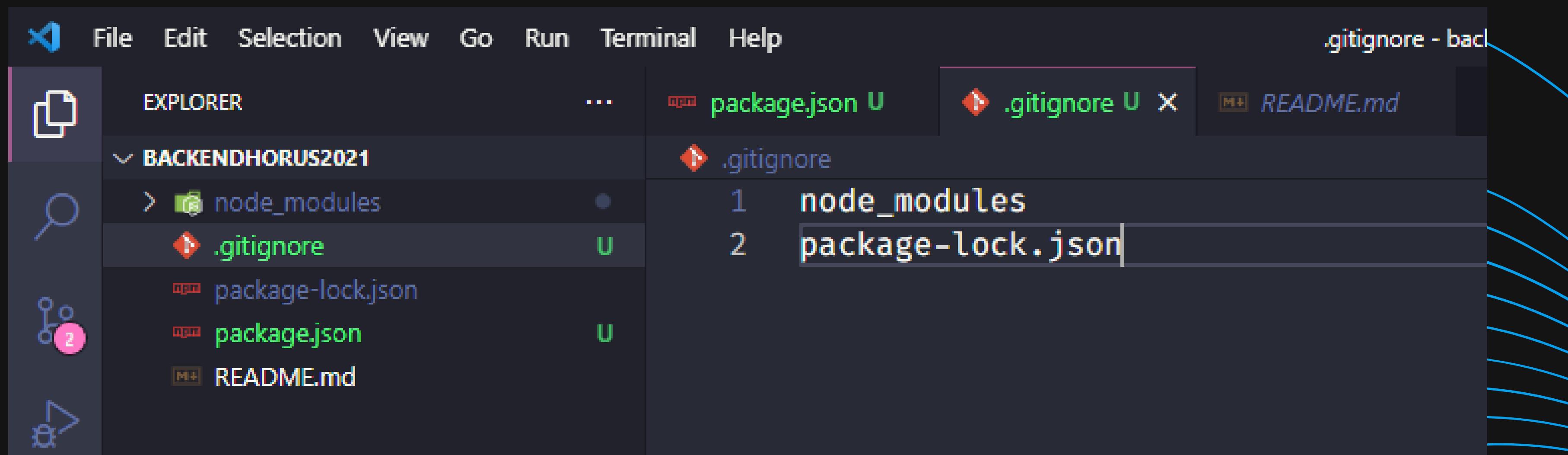
Express é uma das ferramentas mais populares e amplamente utilizadas para o desenvolvimento de aplicações back-end com Node.js. Ele oferece uma série de recursos para tornar o desenvolvimento mais fácil e rápido, além de ser altamente personalizável e escalável de acordo com as necessidades do seu projeto.

Instale o Express no projeto - *npm i express*

```
{  
  "name": "back-23",  
  "version": "1.0.0",  
  "description": "backend - 2023",  
  "main": "index.js",  
  ▷ Debug  
  "scripts": {  
    "test": "echo \"Error: no test specified\" && exit 1"  
  },  
  "repository": {  
    "type": "git",  
    "url": "git+https://github.com/ricardohendges/back-23.git"  
  },  
  "author": "",  
  "license": "ISC",  
  "bugs": {  
    "url": "https://github.com/ricardohendges/back-23/issues"  
  },  
  "homepage": "https://github.com/ricardohendges/back-23#readme",  
  "dependencies": {  
    "express": "^4.18.2"  
  }  
}
```

gitignore

Você pode criar um arquivo `.gitignore` no diretório-raiz do seu repositório para dizer ao Git quais arquivos e diretórios devem ser ignorados ao fazer um commit. Para compartilhar as regras de ignorar com outros usuários que clonarem o repositório, faça o commit do arquivo `.gitignore` no seu repositório.



Iniciando projeto

Para todos fontes do projeto teremos a pasta `src` dentro do projeto.

Dentro dessa pasta, teremos o arquivo `index.js` e o seguinte código dentro dele.

```
const express = require('express');
const app = express();
const port = 3000;

app.get('/', (req, res) => { res.send('Hello World!'); });

app.listen(port, () => {
  console.log(`Aplicação rodando na porta ${port}`);
});
```

run

Dentro do arquivo `package.json` na sessão `script` deve ser adicionado o seguinte código para executar nosso projeto utilizando `npm`.

```
"dev": "node ./src/index.js"
```

Após isso execute o comando "`npm run dev`"
Abra seu navegador na seguinte URL "`localhost:3000`"



Organizando Projeto e suas Rotas

Crie as seguintes pastas dentro de src

controllers
routes
services

Services

Arquivo responsável pela carga de dados e execução de regra de negócio, formatação dos campos e retorno para quem chama a API.

```
const getAluno = async (params) => {
  let aluno = {}
  if (params.name === "Ricardo") {
    aluno.id = 16105
    aluno.nome = "Ricardo"
    aluno.sobrenome = "Hendges"
    aluno.notas = []
    aluno.notas.push(10)
    aluno.notas.push(9)
  } else {
    throw "Aluno não cadastrado!"
  }
  return aluno
}

module.exports.getAluno = getAluno
```

Controllers

Arquivo responsável pela pelo tratamento de parâmetros de entrada(req) e também pelo retorno da chamada(res)

```
const alunoService = require('../services/alunos')

const getAlunos = async (req, res, next) => {
  try {
    const retorno = await alunoService.getAlunos(req.params)
    res.status(201).json(retorno)
  } catch (err) {
    res.status(500).send(err)
  }
}

module.exports.getAlunos = getAlunos
```

Routes

Arquivos que registram quais ENDPOINTS a API terá.

```
const alunoController = require('../controllers/aluno');

module.exports = (app) => {
  app.get('/aluno/:name', alunoController.getAlunos)
}
```

src/routes/index.js

```
const Alunos = require('./aluno');

module.exports = (app) => {
  Alunos(app)
}
```

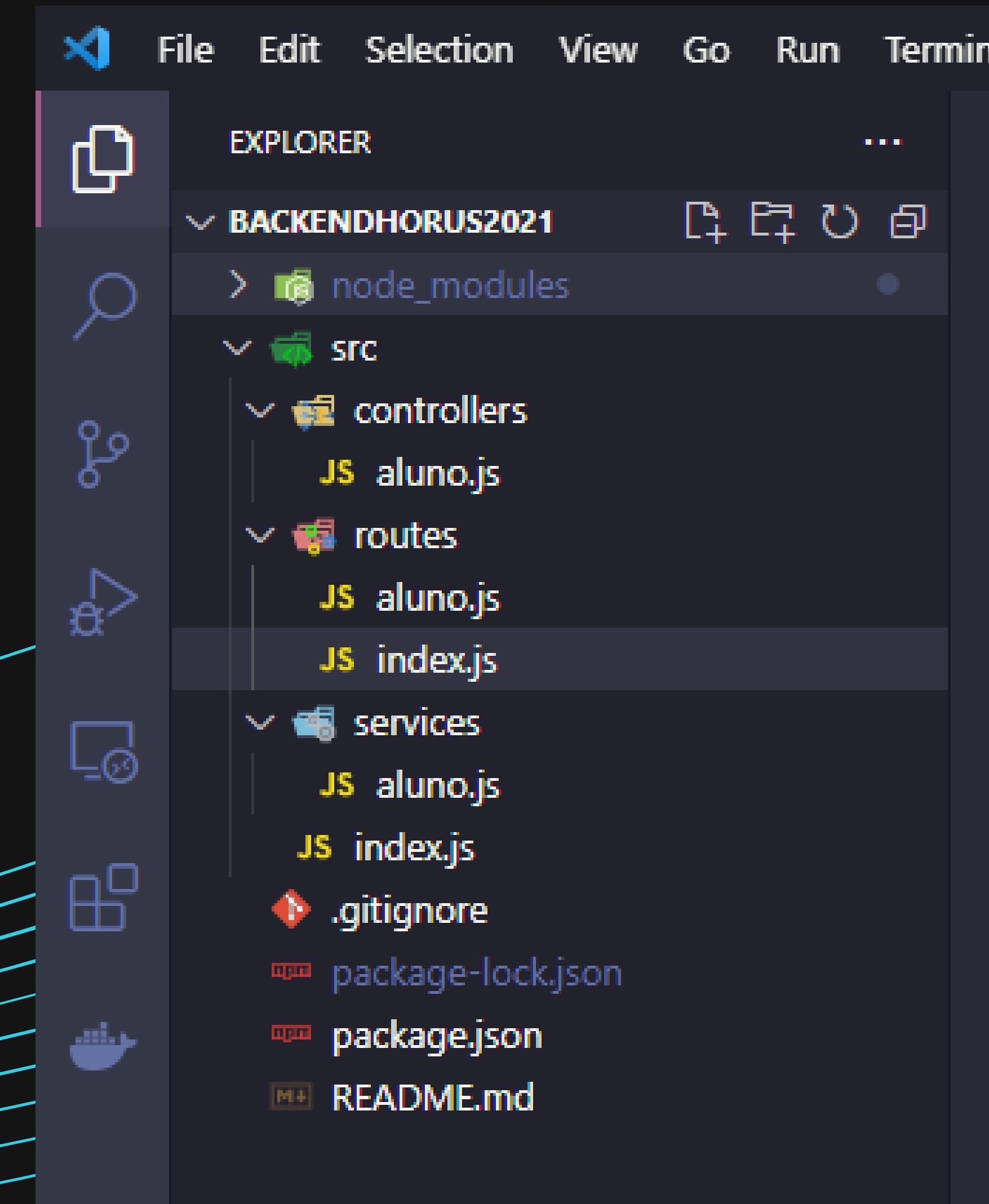
Captura todas rotas para exportar para o aplicativo.

Extrutura até o momento

Temos pasta **SRC** para todos fontes.

Dentro dela temos as pastas **controllers**, **routes** e **services** para controlar os endpoints criados na API.

Paralelo a elas temos o arquivo **index.js** responsável por levantar a aplicação.



Identificando as rotas criadas

Para que o arquivo index.js identifique que foram criadas as rotas de consulta de alunos devemos primeiro carregar elas na aplicação.

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

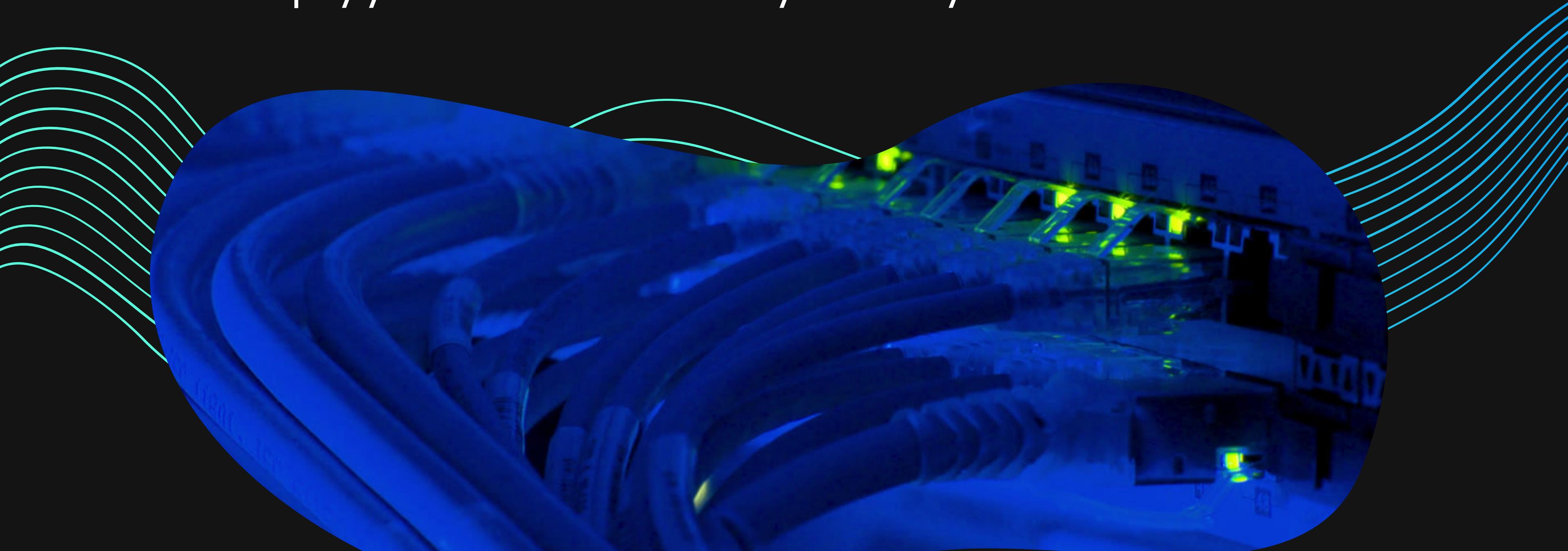
require('./routes')(app)          You, 5 minutes ago • rotas
app.get('/', (req, res) => res.status(200).send('Hello World'))

app.listen(3000)
```

Testando rotas

<http://localhost:3000/aluno/Ellie>

<http://localhost:3000/aluno/Ricardo>

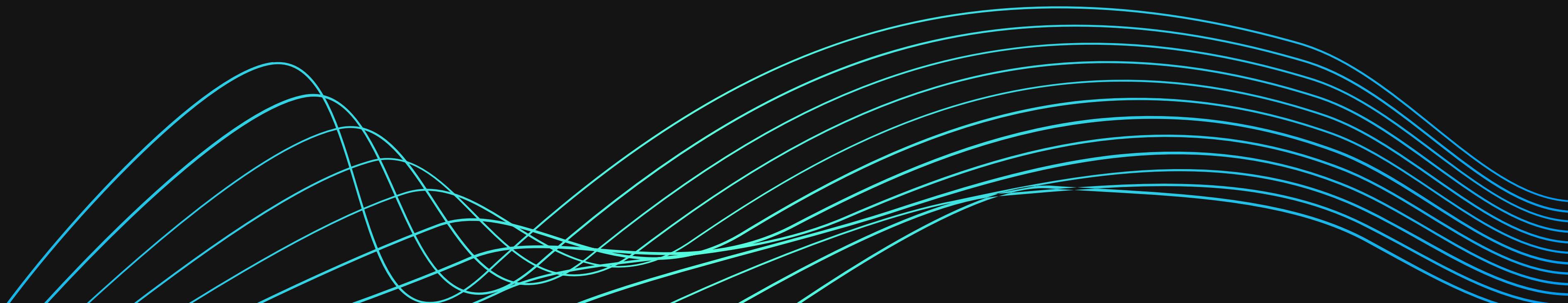


Nodemon

npm install --save-dev nodemon

Nodemon é uma ferramenta que ajuda a desenvolver aplicativos baseados em node.js, reiniciando automaticamente o aplicativo de node.js quando mudanças de arquivo no diretório são detectadas.

O nodemon não requer nenhuma mudança adicional em seu código ou método de desenvolvimento. Nodemon é um invólucro substituto para o node. Para usar o nodemon, substitua a palavra node na linha de comando ao executar seu script.



```
"scripts": {  
  "test": "echo \"Error: no test specified\" && exit 1",  
  "dev": "nodemon ./src/index.js",  
  "prd": "node ./src/index.js"  
},
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Projetos\Estudos\LPV2021\backendhorus2021> npm run dev

> backendhorus2021@1.0.0 dev
> nodemon ./src/index.js

[nodemon] 2.0.12
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): **.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node ./src/index.js`
[nodemon] restarting due to changes...
[nodemon] restarting due to changes...
[nodemon] starting `node ./src/index.js`

