

Engenharia De Software

AULA 2 - GERÊNCIA DE DEPENDÊNCIA

Inatel

Conteúdo da aula

- Instruções para o projeto
- NPM / Yarn
 - Comparação com o Maven
 - Como usar
 - Entendendo a estrutura do projeto
- Outras linguagens

Orientações para o projeto

- **Regras:**

- Grupo de até 3 pessoas
- Tema é livre.
- Linguagem da sua preferência (Orientada a Objetos)
- CRUD (Obrigatório listar e salvar dados)
- Permanência dos dados (Arquivo ou Banco de Dados)

IMPORTANTE:

**Entregar o tema, equipe e ferramenta
escolhida até o dia 14/04**

Orientações para o projeto

- **É obrigatório conter em todo projeto:**
 - Ferramenta de controle de versão
Exemplo: GitHub, Bitbucket, GitLab, etc
 - Gerenciador de dependências
Exemplo: Maven, NPM, Nuget, etc
 - Cobertura de Teste e testes de unidade
 - No mínimo 1 teste mock
 - CI
Exemplo: Travis, Jenkins ou Github Actions.
 - No mínimo 2 padrões de projeto (Implementados por você!)
 - Arquitetura MVC ou outra

Orientações para o projeto

- **O README do GitHub deve conter:**
 - Descrição do projeto
 - Funcionalidades do projeto
 - Bibliotecas utilizadas
 - Ferramentas utilizadas
 - CI Badge
 - Como utilizar

YARN / NPM

- **Comparação com o Maven**

- Automação da Build e gerenciamento de dependência.

- **Maven Repository:** <https://mvnrepository.com>

- Arquivo **pom.xml**

- Exemplo XML do Gson:

```
<!-- https://mvnrepository.com/artifact/com.google.code.gson/gson -->
<dependency>
  <groupId>com.google.code.gson</groupId>
  <artifactId>gson</artifactId>
  <version>2.8.6</version>
</dependency>
```

YARN / NPM

- **Comparação com o Maven**
 - Gerenciamento de dependência.
 - **NPM:** <https://www.npmjs.com>
 - **Yarn:** <https://yarnpkg.com>
 - Arquivo **package.json**
 - Exemplo:

```
yarn add react-router-dom
```

```
npm install --save react-router-dom
```

YARN / NPM

- Diferença entre YARN e NPM

NPM		YARN
<code>npm init</code>		<code>yarn init</code>
<code>npm install ...</code>		<code>yarn add ...</code>
<code>npm update ...</code>		<code>yarn upgrade ...</code>
<code>npm remove ...</code>		<code>yarn remove ...</code>

YARN / NPM

- Como usar o YARN / NPM
 - Instalação : Manual de Instalação
 - Abrir o terminal de comando
 - Criar uma aplicação **Node js**

```
C:\Users\Usuario>npm -v  
6.14.9  
  
C:\Users\Usuario>yarn -v  
1.22.5
```

```
$ yarn init
```

```
$ npm init
```

YARN / NPM

- Usando o comando **YARN**
- Configurando o arquivo **package.json**

```
Vanessa Swerts@Swerts96 MINGW64 /e/Desktop/Projetos/Monitoria/C214 - L2/aula03-nodejs
$ yarn init
yarn init v1.22.5
question name (aula03-nodejs):
question version (1.0.0):
question description: Projeto criado para a Aula 3 - Gerencia de Dependencias
question entry point (index.js):
question repository url:
question author: Vanessa Swerts
question license (MIT):
question private:
success Saved package.json
Done in 76.23s.
```

YARN / NPM

- Usando o comando **NPM**
- Configurando o arquivo **package.json**

```
Vanessa Swerts@Swerts96 MINGW64 /e/Desktop/Projetos/Monitoria/C214 - L2/aula03
$ npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

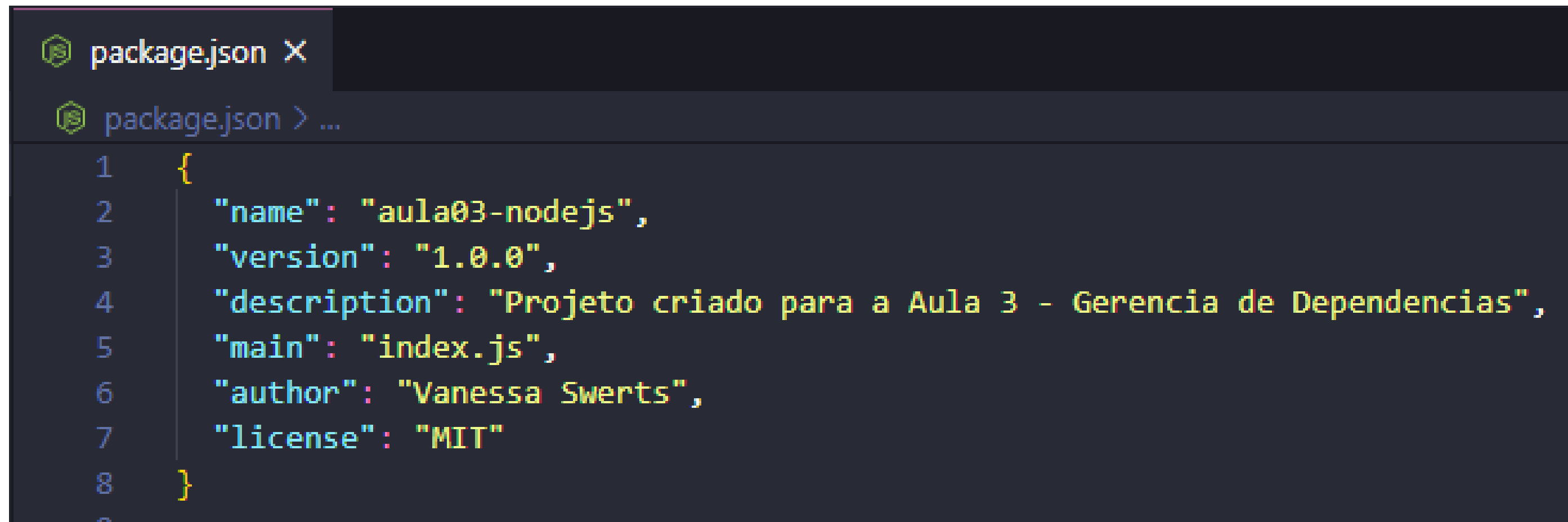
Press ^C at any time to quit.
package name: (aula03)
version: (1.0.0)
description: Projeto criado para a Aula 03 - Gerencia Dedependencia
entry point: (index.js)
test command:
git repository:
keywords:
author: Vanessa Swerts
license: (ISC)
About to write to E:\Desktop\Projetos\Monitoria\C214 - L2\aula03\package.json:
```

```
{
  "name": "aula03",
  "version": "1.0.0",
  "description": "Projeto criado para a Aula 03 - Gerencia Dedependencia",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Vanessa Swerts",
  "license": "ISC"
}
```

Is this OK? (yes)

YARN / NPM

- Como usar o YARN / NPM
 - Arquivo **package.json**



```
package.json X
package.json > ...
1  {
2    "name": "aula03-nodejs",
3    "version": "1.0.0",
4    "description": "Projeto criado para a Aula 3 - Gerencia de Dependencias",
5    "main": "index.js",
6    "author": "Vanessa Swerts",
7    "license": "MIT"
8  }
```

YARN / NPM

- **Como instalar dependências?**

- **Faker.js**

A dependência Faker é uma biblioteca que fornece dados falsos, vamos utiliza-la para criar um exemplo bem simples.

```
$ yarn add faker
```

```
$ npm i faker
```

YARN / NPM

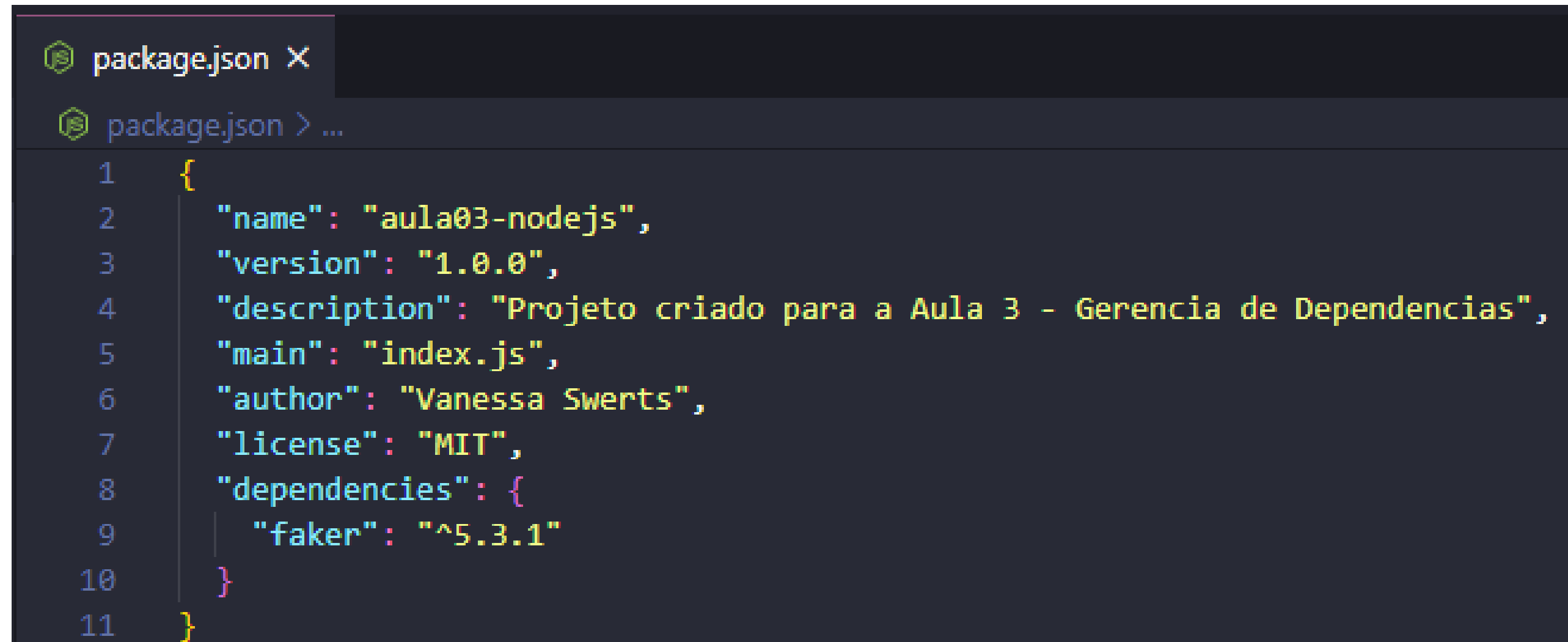
- Como instalar dependências?
- Instalação do Faker.js

```
Vanessa Swerts@Swerts96 MINGW64 /e/Desktop/Projetos/Monitoria/C214 - L2/aula03-nodejs
$ yarn add faker
yarn add v1.22.5
info No lockfile found.
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
[4/4] Building fresh packages...

success Saved lockfile.
success Saved 1 new dependency.
info Direct dependencies
└─ faker@5.3.1
info All dependencies
└─ faker@5.3.1
Done in 15.37s.
```

YARN / NPM

- Como instalar dependências?
- Arquivo **package.json** com a dependência do **Faker.js**



The screenshot shows a code editor with a dark theme. The top bar shows the file name 'package.json' with a close button. Below the top bar, the file path 'package.json > ...' is visible. The main area displays the content of the package.json file, which is a JSON object with the following properties: 'name', 'version', 'description', 'main', 'author', 'license', and 'dependencies'. The 'dependencies' property is an object containing 'faker' with a version range of '^5.3.1'. The code is syntax-highlighted, with strings in yellow, keywords in blue, and punctuation in pink. Line numbers 1 through 11 are visible on the left side of the editor.

```
1  {
2    "name": "aula03-nodejs",
3    "version": "1.0.0",
4    "description": "Projeto criado para a Aula 3 - Gerencia de Dependencias",
5    "main": "index.js",
6    "author": "Vanessa Swerts",
7    "license": "MIT",
8    "dependencies": {
9      "faker": "^5.3.1"
10   }
11 }
```

YARN / NPM

- Como instalar dependências?
- No arquivo index.js

```
JS index.js X
JS index.js > ...
1  const faker = require('faker')
2
3  const randomName = faker.name.findName()
4  console.log("name", randomName)
5
6  const randomVehicle = faker.vehicle.vehicle();
7  console.log('vehicle', randomVehicle)
8
```


YARN / NPM

- Como instalar dependências?
- Adicionando um script no **package.json**



```
package.json X
package.json > ...
1  {
2    "name": "aula03-nodejs",
3    "version": "1.0.0",
4    "description": "Projeto criado para a Aula 3 - Gerencia de Dependencias",
5    "main": "index.js",
6    "author": "Vanessa Swerts",
7    "license": "MIT",
8    "scripts": {
9      "start": "node index.js"
10   },
11   "dependencies": {
12     "faker": "^5.3.1"
13   }
14 }
```

YARN / NPM

- Como instalar dependências?
- Executando a aplicação

```
Vanessa Swerts@Swerts96 MINGW64 /e/Desktop/Projetos/Monitoria/C214 - L2/aula03-nodejs
$ yarn start
yarn run v1.22.5
$ node index.js
name Josephine Lemke
vehicle Dodge Golf
Done in 2.33s.
```

```
Vanessa Swerts@Swerts96 MINGW64 /e/Desktop/Projetos/Monitoria/C214 - L2/aula03-nodejs
$ npm start

> aula03-nodejs@1.0.0 start E:\Desktop\Projetos\Monitoria\C214 - L2\aula03-nodejs
> node index.js

name Geneva Block
vehicle Nissan Volt
```

YARN / NPM

- Como usar o YARN / NPM
 - Como criar uma aplicação **React** ?

```
npx create-react-app my-app
```

```
npm init react-app my-app
```

```
yarn create react-app my-app
```

YARN / NPM

```
🌟 Done in 12.06s.
```

```
Success! Created my-app at ~/my-app
```

```
Inside that directory, you can run several commands:
```

```
yarn start
```

```
Starts the development server.
```

```
yarn build
```

```
Bundles the app into static files for production.
```

```
yarn test
```

```
Starts the test runner.
```

```
yarn eject
```

```
Removes this tool and copies build dependencies, configuration files  
and scripts into the app directory. If you do this, you can't go back!
```

```
We suggest that you begin by typing:
```

```
cd my-app
```

```
yarn start
```

```
Happy hacking!
```

YARN / NPM

- Entendendo a estrutura do projeto

```
my-app
├── README.md
├── node_modules
├── package.json
├── .gitignore
├── public
│   ├── favicon.ico
│   ├── index.html
│   ├── logo192.png
│   ├── logo512.png
│   ├── manifest.json
│   └── robots.txt
└── src
    ├── App.css
    ├── App.js
    ├── App.test.js
    ├── index.css
    ├── index.js
    ├── logo.svg
    ├── serviceWorker.js
    └── setupTests.js
```

YARN / NPM

- **Como instalar dependências?**

- React Router Dom

```
yarn add react-router-dom
```

```
npm install --save react-router-dom
```

- Dependência de Desenvolvimento

```
yarn add --dev react-test-renderer
```

Documentação: <https://jestjs.io/docs/en/tutorial-react>

Outras Linguagens

- **Android Nativo:** Gradle
- **Flutter:** pub.dev
- **Python:** Py.pi
- **C#:** Nuget
- **Ruby:** RubyGems
- **PHP:** Composer