

# Desenvolvimento Front-end com Frameworks

Fundamentos de React

# Agenda

**Etapa 2:** Declaração de variáveis e funções.

- Revisão de Sintaxe.
- Construção de Componentes Básicos.



# Revisão de Sintaxe

## Desestruturação

A sintaxe de **desestruturação** é uma expressão JavaScript que permite descompactar (ou retirar) valores de arrays ou propriedades de objetos, em variáveis distintas.

1

```
let a, b, rest;  
[a, b] = [10, 20];  
  
console.log(a);  
// Expected output: 10  
  
console.log(b);  
// Expected output: 20
```

2

```
const x = [1, 2, 3, 4, 5];  
const [y, z] = x;  
  
console.log(y); // 1  
console.log(z); // 2
```

3

```
const obj = { a: 1, b: 2 };  
const { a, b } = obj;  
// is equivalent to:  
// const a = obj.a;  
// const b = obj.b;
```

Developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Expressions and operators / Destructuring assignment

Get real-time assistance with your coding queries. Try AI Help now!

mdn web docs

ReferencesGuidesPlusCurriculumNEWBlogPlayAI HelpBETAThemeLog inSign up for free

References > JavaScript > Reference > Expressions and operators > Destructuring assignmentEnglish (US)

Filter

Decrement (--)  
delete  
Destructuring assignment  
Division (/)  
Division assignment (/=  
Equality (==)  
Exponentiation (\*\*)  
Exponentiation assignment (\*\*=  
function expression  
function\* expression  
Greater than (>)

# Destructuring assignment

The **destructuring assignment** syntax is a JavaScript expression that makes it possible to unpack values from arrays, or properties from objects, into distinct variables.


## Try it

JavaScript Demo: Expressions - Destructuring assignment

```
1 let a, b, rest;
2 [a, b] = [10, 20];
3
4 console.log(a);
5 // Expected output: 10
6
7 console.log(b);
8 // Expected output: 20
9
```

In this article

Try it  
Syntax  
Description  
Examples  
Specifications  
Browser comp  
See also



Avaliador.jsx X

src > Avaliador.jsx > Avaliador

```
1  import { FaStar } from "react-icons/fa";
2
3  export default function Avaliador({ avaliacao, quantidade, titulo }) {
4    function gerarAvaliacao() {
5      const retorno = [];
6      if (avaliacao <= quantidade) {
7        for (let i = 0; i < quantidade; i++) {
8          if (i < avaliacao) {
9            retorno.push(<FaStar color="red" />);
10           } else {
11             retorno.push(<FaStar color="grey" />);
12           }
13         }
14       } else {
15         throw Error("A quantidade tem que ser maior ou igual a avaliação");
16       }
17       return retorno;
18     }
19
20     return (
21       <div>
22         {gerarAvaliacao()} {titulo}
23       </div>
24     );
25   }
```

Aqui usamos a  
desestruturação para obter  
somente os itens de props  
que nos interessam



## Operador de Propagação - spread

Em ES6 e posterior, você pode usar o operador de propagação para incluir os elementos de um array dentro de outro ou copiar propriedades de objetos.

1

```
let a = [1, 2, 3];  
let b = [0, ...a, 4]; // b == [0, 1, 2, 3, 4]
```

2

```
let original = [1,2,3];  
let copy = [...original];  
copy[0] = 0; // Modifying the copy does not change the original  
original[0] // => 1
```

Avaliador.jsx

App.js x



Preview x



src &gt; App.js &gt; App

```
1 import './styles.css';
2 import { FaStar } from 'react-icons/fa';
3 import Avaliador from './Avaliador';
4
5 export default function App() {
6   return (
7     <div className="App">
8       <div className="container">
9         <h4>Componentes Básicos</h4>
10      </div>
11      <div className="container">
12        <p>
13          <Avaliador
14            style={{ backgroundColor: "yellow" }}
15            quantidade="5"
16            avaliacao="5"
17            titulo="Homem Aranha"
18          />
19        </p>
20        <p>
21          <Avaliador quantidade="5" avaliacao="3" titulo="Batman" />
22        </p>
23        <p>
24          <Avaliador quantidade="5" avaliacao="4" titulo="Barbie" />
25        </p>
26      </div>
27    </div>
28  );
29}
```

<https://j22vs4.csb.app/>

## Componentes Básicos

★★★★★ Homem Aranha

★★★★☆ Batman

★★★★☆ Barbie



```
Avaliador.jsx x App.js
src > Avaliador.jsx > Avaliador
3 export default function Avaliador({ avaliacao, quantidade, titulo, style }) {
4   function gerarAvaliacao() {
8     if (1 < avaliacao) {
9       retorno.push(<FaStar color="red" />);
10    } else {
11      retorno.push(<FaStar color="grey" />);
12    }
13  }
14  } else {
15    throw Error("A quantidade tem que ser maior ou igual a");
16  }
17  return retorno;
18 }
19
20 return (
21   <div style={{ margin: "10px", padding: "10px", ...style }}>
22     {gerarAvaliacao()} {titulo}
23   </div>
24 );
25 }
```

Aqui style é um objeto que recebe margin, padding e todos os outros atributos copiados do objeto passado em props

# Construção de Componentes Básicos



# Faker

Generate massive amounts of fake (but realistic) data for testing and development.

[Get Started](#)[View on GitHub](#)

## Products

Generate Prices, Product Names, Adjectives, and Descriptions.



## Finance

Create stubbed out Account Details, Transactions, and Crypto Addresses.



## Addresses

Generate valid Addresses, Zip Codes, Street Names, States, and Countries!

Person | Faker

fakerjs.dev/api/person.html

Faker

Location

Lorem

Music

Number

Person

Phone

Science

String

System

Vehicle

Word

About

Search

CtrlK

Guide

API

Try it

About

v9.7.0

# Person

Module to generate people's personal information such as names and job titles. Prior to Faker 8.0.0, this module was known as `faker.name`.

## Overview

To generate a full name, use `fullName`. Note that this is not the same as simply concatenating `firstName` and `lastName`, as the full name may contain a prefix, suffix, or both. Additionally, different supported locales will have differing name patterns. For example, the last name may appear before the first name, or there may be a double or hyphenated first or last name.

You can also generate the parts of a name separately, using `prefix`, `firstName`, `middleName`, `lastName`, and `suffix`. Not all locales support all of these parts.

On this page

bio

firstName

fullName

gender

jobArea

jobDescriptor

jobTitle

jobType

lastName

middleName

prefix

sex

sexType

codesandbox.io/dashboard/recent

Google Lens

## Create

Popular

Browser

Server

Frontend

Backend

Workspace

react



React

CodeSandbox

8.2M



React (TS)

CodeSandbox

235.5k



Next.js

CodeSandbox

107.4k



Preact

CodeSandbox

19.5k



React + Tailwind

CodeSandbox

14.5k



React (JS)

CodeSandbox

14.5k



React Native with Expo

CodeSandbox

9.0k



AI Code Completion

Codelum

4.5k



Create-T3-app

CodeSandbox

3.0k



Gatsby

CodeSandbox

1.5k



Docusaurus

CodeSandbox

1.1k



Storybook (React)

CodeSandbox

386



React + Chakra

CodeSandbox

374



Hono + Next.js

CodeSandbox

202



React (Rsbuild + TS)

CodeSandbox

163



React Router (Remix)

CodeSandbox

136

1

1

The image shows a VS Code interface with a dark theme. On the left, the Explorer sidebar is open, showing a project structure with 'Sandbox Info', 'Nodebox', 'public', 'src', 'App.js', 'index.js', 'styles.css', and 'package.json'. Below this, the 'Dependencies' panel is open, displaying a search bar and a list of installed packages: 'react ^19.0.0', 'react-dom ^19.0.0', 'react-scripts ^5.0.0', and '@faker-js/faker 9.7.0'. A red box highlights the 'Dependencies' panel. The main editor area shows the 'package.json' file, which contains the following JSON content:

```
{
  "name": "react",
  "version": "1.0.0",
  "description": "",
  "keywords": [],
  "main": "src/index.tsx",
  "dependencies": {
    "react": "^19.0.0",
    "react-dom": "^19.0.0",
    "react-scripts": "^5.0.0",
    "@faker-js/faker": "9.7.0"
  },
  "devDependencies": {
    "@types/react": "19.0.0",
    "@types/react-dom": "19.0.0",
    "loader-utils": "3.2.1",
    "typescript": "5.7.2"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test --env=jsdom",
    "eject": "react-scripts eject"
  }
}
```

A red box highlights the 'dependencies' section of the 'package.json' file, which lists the same packages as the Dependencies panel: 'react', 'react-dom', 'react-scripts', and '@faker-js/faker'.

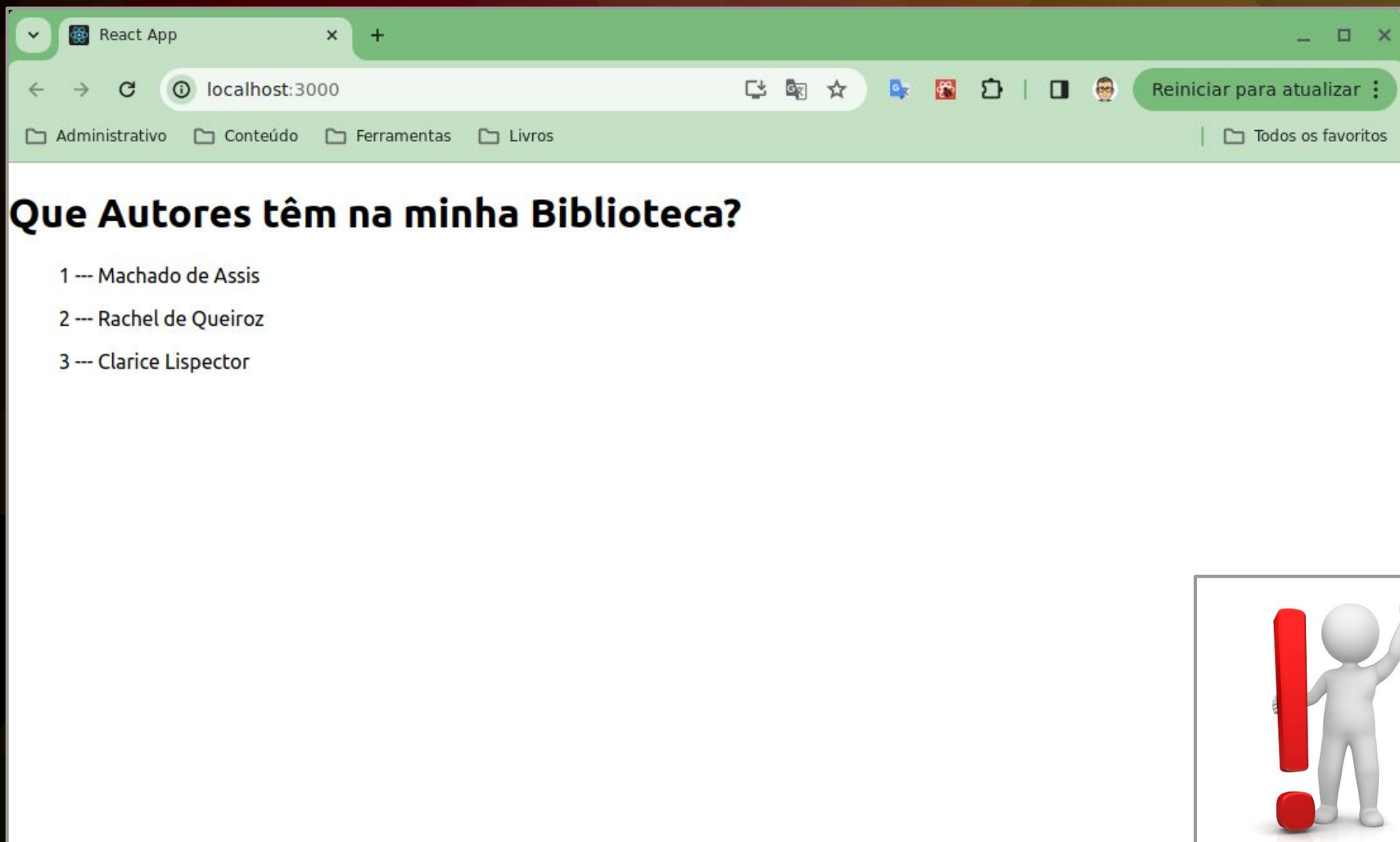


2

Tela1.jsx U X JS App.js M

src &gt; Tela1.jsx &gt; ...

```
1  export default function Tela1() {
2    const autores = [
3      { id: 1, nome: 'Machado de Assis' },
4      { id: 2, nome: 'Rachel de Queiroz' },
5      { id: 3, nome: 'Clarice Lispector' }
6    ];
7    return (
8      <div>
9        <h1>Que Autores têm na minha Biblioteca?</h1>
10       <ul>
11         {autores.map((item) => <p>{item.id} --- {item.nome}</p>)}
12       </ul>
13     </div>
14   );
15 }
```



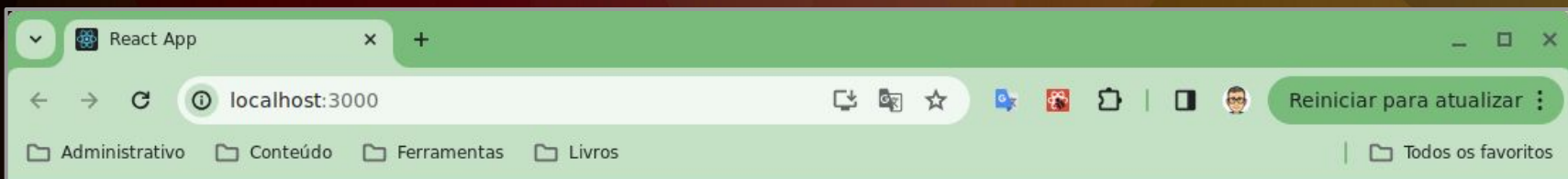
3

Tela2.jsx U X

JS App.js M

src &gt; Tela2.jsx &gt; ...

```
1  import { faker } from '@faker-js/faker';
2
3  export default function Tela2() {
4
5      const autores = new Array(10).fill()
6          .map((value, index) => {
7              let autor = new Object();
8              autor.id = faker.string.uuid();
9              autor.nome = faker.person.fullName();
10             return autor;
11         });
12
13     return (
14         <div>
15             <h1>Que Autores têm na minha Biblioteca?</h1>
16             <ul>
17                 {autores.map((item) => <p>ID: {item.id} / Nome: {item.nome}</p>)}
18             </ul>
19         </div>
20     );
21 }
```



## Que Autores têm na minha Biblioteca?

ID: 4e631508-6147-45bd-ac1a-dec6548896cc / Nome: Alfredo O'Kon  
ID: 23f21b2f-ef69-4ff8-8199-3c729ba96654 / Nome: Kent Weimann  
ID: 2a9619c3-725b-432a-ad29-5dac6f81df4d / Nome: Kim Cormier  
ID: eaf58627-0fd5-4891-9860-2b609934e598 / Nome: Albert Mertz  
ID: 82e1e28d-0eb5-45e5-b588-b0481e85d003 / Nome: Olivia Hilll  
ID: 2ba31d32-2dad-426b-bc66-a3eb9a6934f0 / Nome: Mr. Carroll Stracke  
ID: 50d0c537-fe02-4664-a8c5-4a1dcafbe782 / Nome: Lillie Volkman  
ID: 28c82f41-cad5-4fbf-92de-ea3471635b51 / Nome: Mr. Rodney Mraz  
ID: 702f5050-c18c-4ed7-a4cd-7c1f6286186b / Nome: Darnell Heller  
ID: 1b47184b-887a-40a6-9366-25ec6d1b0f87 / Nome: Marco Kirlin



4

```
Tela3.jsx U JS App.js M # App.css M X
src > # App.css > ...
1  .container {
2    background-color: lightblue;
3    border: 1px solid black;
4    margin: 5px;
5    padding: 5px;
6    display: inline-table;
7  }
```

5

```
JS App.js M Tela3.jsx U X
src > Tela3.jsx > ...
1  import { faker } from '@faker-js/faker';
2
3  export default function Tela2() {
4
5    const funcionarios = new Array(10).fill()
6      .map((value, index) => {
7        let funcionario = new Object();
8        funcionario.id = faker.string.uuid();
9        funcionario.nome = faker.person.fullName();
10       funcionario.cargo = faker.person.jobTitle();
11       funcionario.departamento = faker.person.jobArea();
12       return funcionario;
13     });
14 }
```

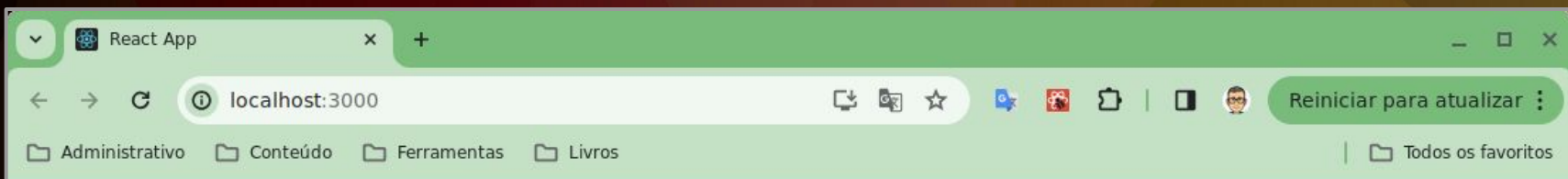
JS App.js M

Tela3.jsx U X

src &gt; Tela3.jsx &gt; ...

```
3  export default function Tela2() {
15    return (
16      <div>
17        <h1>Quais São os Funcionários da Filial RJ?</h1>
18        <ul>
19          {funcionarios.map((item) =>
20            <div key={item.id} className="container">
21              <h3>{item.nome}</h3>
22              <p>Cargo: {item.cargo} / Departamento: {item.departamento}</p>
23            </div>
24          )}
25        </ul>
26      </div>
27    );
28  }
```





## Quais São os Funcionários da Filial RJ?

**Neal Windler**

Cargo: Corporate Markets Coordinator / Departamento: Mobility

**Oliver Homenick**

Cargo: International Assurance Planner / Departamento: Optimization

**Elmer Waelchi**

Cargo: Chief Research Designer / Departamento: Infrastructure

**Charlotte Fadel**

Cargo: Future Factors Strategist / Departamento: Operations

**Jennie Windler-Schoen**

Cargo: Direct Group Orchestrator / Departamento: Accountability

**Reginald Reinger**

Cargo: Future Web Executive / Departamento: Account

**Mr. Elbert Hilll-Flatley**

Cargo: Investor Quality Executive / Departamento: Optimization

**Justin Morar DVM**

Cargo: Regional Quality Orchestrator / Departamento: C

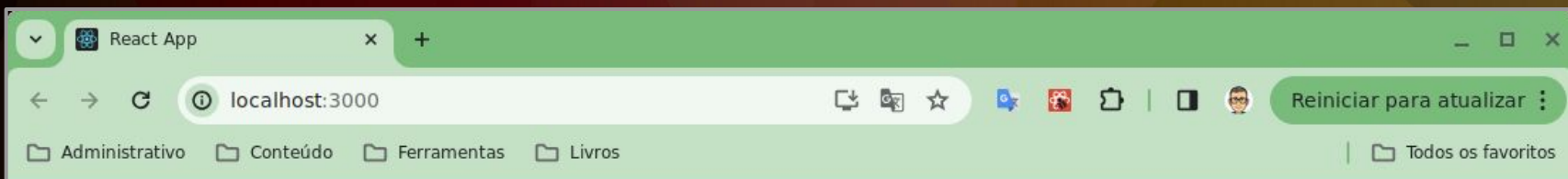


7

Tela4.jsx U x

src &gt; Tela4.jsx &gt; ...

```
1  import { faker } from '@faker-js/faker';
2
3  export default function Tela4() {
4
5      const posts = new Array(10).fill()
6          .map((value, index) => {
7              let post = new Object();
8              post.id = faker.string.uuid();
9              post.title = faker.lorem.words(5);
10             post.body = faker.lorem.sentences(8);
11             return post;
12         });
13
14     return (
15         <div>
16             <h1>Quais São os Posts de Hoje?</h1>
17             <ul>
18                 {posts.map((item) =>
19                     <div key={item.id} className="container">
20                         <h3>{item.title}</h3>
21                         <p>{item.body}</p>
22                     </div>
23                 )}
24             </ul>
25         </div>
26     );
27 }
```



## Quais São os Posts de Hoje?

### **cinis cenaculum carcer aureus censura**

Asporto quo accendo terra volup qui vicinus aegre harum surculus. Bellum sordeo audeo sumo voluptas sophismata barba quibusdam. Contabesco caterva defaeco qui vicissitudo tero admoveo. Colo candidus strues custodia cognatus trans tamisium. Utroque cunctatio clam. Amplexus porro comprehendendo turbo ullam virga denuo conculco. Ulciscor coepi eveniet caterva praesentium delectus speculum. Suscipit auditor libero sto color terga.

### **desparatus congregatio sufficio iure vesco**

Placeat ultio umerus. Crapula brevis defungo reiciendis aer saepe solium claro. Optio tribuo cupressus fuga nisi denuo copia. Quas studio conatus succedo aqua velut. Vespillo arbor celo temeritas cometes corroboro constans vesica. Adflicto audio. Vito laborum vinco annus virgo deorsum. Calcar utpote careo absorbeo constans.

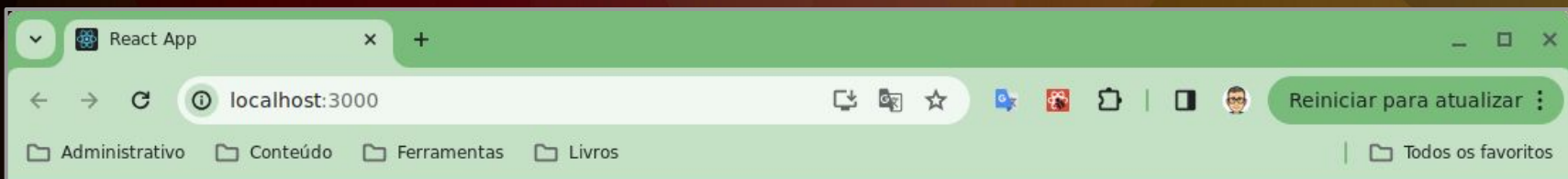
### **clementia terror ventus dolorum dapifer**

Cursus aqua demulceo sonitus decens civis. Stultus velociter ut dens peior. Sulum vigilo commemoro tondeo tamen deliber tamquam. Corona super depopulo trepide crepusculum derideo. Una timidus arx. Repellendus volaticus tabernus claro. Vel votum. Absens cribro adnuo.



8

```
Tela5.jsx U x
src > Tela5.jsx > ...
1  import { faker } from '@faker-js/faker';
2
3  export default function Tela5() {
4
5      const musics = new Array(10).fill()
6        .map((value, index) => {
7          let music = new Object();
8          music.id = faker.string.uuid();
9          music.title = faker.music.songName();
10         music.genre = faker.music.genre();
11         return music;
12       });
13
14     return (
15       <div>
16         <h1>Qual é a "Play List" de Hoje?</h1>
17         <ul>
18           {musics.map((item) =>
19             <div key={item.id} className="container">
20               <h3>{item.title} - {item.genre}</h3>
21             </div>
22           )}
23         </ul>
24       </div>
25     );
26   }
```



## Qual é a "Play List" de Hoje?

**Don't Let the Stars Get in Your Eyes - Rock**

**Paper Planes - Hip Hop**

**Dardanella - Jazz**

**Da Doo Ron Ron (When He Walked Me Home) - Latin**

**Shadow Dancing - Rap**

**Bridge Over Troubled Water - Rap**

**96 Tears - Non Music**

**Stagger Lee - Electronic**

**Chattanooga Choo Choo - Latin**

**Smells Like Teen Spirit - Blues**

