

**UC:** Projetos

**Assunto:**

Projeto Pizzaria – Frontend\_02 (NextJS)

Sistema Pizzaria Web

51 – Cadastrando Categoria

next-server (v15.0.3)

```
c:\Users\Dev\Desktop\Pizzaria\frontend>npm run dev
```

```
> frontend@0.1.0 dev
> next dev
  ▲ Next.js 15.0.3
  - Local:          http://localhost:3000

  Starting...
  Ready in 3.3s
```

Sujeito Pizza - A melhor pizzaria

localhost:3000/das...

# SujeitoPizza

Categoria Produto [→]

## Nova Categoria

Nome da categoria, ex: Pizzas

Cadastrar

C:\ npm run dev

```
c:\Users\Dev\Desktop\Pizzaria\backend>npm run dev

> backend@1.0.0 dev
> ts-node-dev --transpile-only src/server.ts

[INFO] 19:18:51 ts-node-dev ver. 2.0.0 (using ts-node ver. 10.9.2, typescript ver. 5.6.3)
Servidor online!!!!
```

Sujeito Pizza - A melhor pizzaria

localhost:3000/das...

# SujeitoPizza

Categoria Produto [→]

## Nova Categoria

Nome da categoria, ex: Pizzas

Cadastrar

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C + Query #1 categories [all] +

Filter ENTITIES 217 +

public

- \_prisma\_migrations
- categories
- items
- orders
- products
- users

information\_schema

pg\_catalog

	id	equals	Enter Value
1	00d8f79f-5536-429a-bd59-7a82f18700ac	bebidas	2024-11-21 23:37:54.506
2	80de4305-34d1-418c-a215-455a38a57fa4	sobremesas	2024-11-22 00:08:08.794
3	e21422b1-f695-4426-aa06-9d23c2f3502b	pizzas	2024-11-21 23:39:55.683

The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under **FRONTEND**, including **.next**, **node\_modules**, **public**, **src** (with **app**, **dashboard**, **category**), and files like **page.tsx**, **styles.module.scss**, **components**, **product**, **layout.tsx**, **page.tsx**, **fonts**, **signup**, **favicon.ico**, **globals.scss**, **layout.tsx**, **page.module.scss**, and **page.tsx**.
- Search:** Shows a search bar with the query **frontend**.
- Outline:** Shows the outline of the current file, **page.tsx**.
- Timeline:** Shows the timeline of the project.
- Editor:** Displays the **page.tsx** file content:

```
src > app > dashboard > category > page.tsx > ...
1 import styles from './styles.module.scss'
2 import { Button } from "@/app/dashboard/components/button"
3 import { api } from '@/services/api'
4 
5 export default function Category(){
6 
7     async function handleRegisterCategory(){
8         "use server"
9         console.log("TESTEEE")
10    }
11 
12 
13     return(
14         <main className={styles.container}>
15             <h1>Nova Categoria</h1>
16 
17             <form
18                 className={styles.form}
19                 action={handleRegisterCategory}
20             >
21                 <input
22                     type="text"
23                     name="name"
24                     placeholder="Nome da categoria, ex:
Pizzas"
```

**Browser Preview:**

The browser window displays the application **SujeitoPizza** with the title **Nova Categoria**. It features a text input field labeled "Nome da categoria, ex: Pizzas" and a green button labeled "Cadastrar".

The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under **FRONTEND**, including **.next**, **node\_modules**, **public**, and **src** (with **app**, **dashboard**, and **category** subfolders). The file **page.tsx** is currently selected.
- Search:** A search bar at the top contains the text **frontend**.
- Editor:** The **page.tsx** file content is displayed:

```
src > app > dashboard > category > page.tsx > Category > handleRegisterCategory
1 import styles from './styles.module.scss'
2 import { Button } from "@/app/dashboard/components/button"
3 import { api } from '@/services/api'
4
5 export default function Category(){
6
7     async function handleRegisterCategory(){
8         "use server"
9         console.log("TESTEEE")
10    }
11
12
13     return(
14         <main className={styles.container}>
15             <h1>Nova Categoria</h1>
16
17             <form
18                 className={styles.form}
19                 action={handleRegisterCategory}
20             >
21                 <input
22                     type="text"
23                     name="name"
24                     placeholder="Nome da categoria, ex:
Pizzas"
```
- Bottom Status Bar:** Shows file statistics (**Ln 9, Col 9 (22 selected)**), encoding (**UTF-8**), and Prettier status.

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:3000/das...
- Content:** The page displays the heading **Nova Categoria** and a text input field with placeholder text **Nome da categoria, ex: Pizzas**. Below the input is a green button labeled **Cadastrar**.
- Header:** The header includes the **SujeitoPizza** logo and navigation links for **Categoria** and **Produto**.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "page.tsx" is the active file.
- Content:** The code is a Next.js page component for creating a category. It includes imports for styles, a button component, and an API service. The logic handles form submission using the "use server" directive, logs the category name to the console, and returns a form with an input field for the category name.
- Status Bar:** Shows character counts (134), spaces (2), and file type (TypeScript JSX).

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:3000/das...
- Page Content:** "SujeitoPizza" logo, navigation links ("Categoria", "Produto"), and a "Nova Categoria" section with a text input field and a green "Cadastrar" button.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (Left):**

- Explorer:** Shows the project structure. The **FRONTEND** folder is expanded, showing **.next**, **node\_modules**, **public**, and **src**. The **src** folder contains **app**, **dashboard**, **category**, and **layout.tsx**.
- File:** **page.tsx** is the active file. It contains TypeScript code for a category creation page. The code includes imports for styles, a button component, and an API service. It defines a **Category** function that handles form data submission using the "use server" hook. The form itself is rendered with an input field for the category name.

**Browser Preview (Right):**

The browser window displays the application "Sujeito Pizza - A melhor pizzaria". The current page is "Nova Categoria". A text input field contains the value "sorvete". Below it is a green button labeled "Cadastrar".

```
import styles from './styles.module.scss'
import { Button } from '@/app/dashboard/components/button'
import { api } from '@/services/api'

export default function Category(){

    async function handleRegisterCategory(formData: FormData){
        "use server"

        const name = formData.get("name")

        if(name === "") return;

        console.log(name)
    }

    return(
        <main className={styles.container}>
            <h1>Nova Categoria</h1>

            <form
                className={styles.form}
                action={handleRegisterCategory}>
                <input
                    type="text"
                    name="name"
                    placeholder="Nome da categoria, ex: Pizzas"
                    required
                    className={styles.input}>
            </form>
        </main>
    )
}
```

The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (Left):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "page.tsx" is the active file.
- Content:** The code is a Next.js page component for creating a category. It imports styles from "styles.module.scss", a button component, and an API service. It defines an "Category" function that handles form data submission using the "use server" hook. The form contains a single input field for the category name.

```
import styles from './styles.module.scss'
import { Button } from "@/app/dashboard/components/button"
import { api } from '@/services/api'

export default function Category(){

    async function handleRegisterCategory(formData: FormData){
        "use server"

        const name = formData.get("name")

        if(name === "") return;

        console.log(name)
    }

    return(
        <main className={styles.container}>
            <h1>Nova Categoria</h1>

            <form
                className={styles.form}
                action={handleRegisterCategory}
            >

                <input
                    type="text"
                    name="name"
                    placeholder="Nome da categoria, ex: Pizzas"
                    required
                    className={styles.input}
                >
            </form>
        </main>
    )
}
```

**Browser Preview (Right):**

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:3000/das...
- Header:** SujeitoPizza
- Navigation:** Categoria | Produto | [→]
- Section:** Nova Categoria
- Form:** Input field labeled "Nome da categoria, ex: Pizzas" and a green "Cadastrar" button.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "page.tsx" is the active file.
- Content:** The code is a Next.js page component for creating a category. It includes imports for styles, a button component, and an API service. The logic handles form submission using the "use server" directive, logs the category name to the console, and returns a form for input.

```
src > app > dashboard > category > page.tsx > Category > handleRegisterCategory
1 import styles from './styles.module.scss'
2 import { Button } from "@/app/dashboard/components/button"
3 import { api } from '@/services/api'
4
5 export default function Category(){
6
7     async function handleRegisterCategory(formData: FormData){
8         "use server"
9
10        const name = formData.get("name")
11
12        if(name === "") return;
13
14        console.log(name);
15
16    }
17
18
19    return(
20        <main className={styles.container}>
21            <h1>Nova Categoria</h1>
22
23            <form
24                className={styles.form}
25                action={handleRegisterCategory}
26            >
27
28                <input
29                    type="text"
30                    name="name"
31                    placeholder="Nome da categoria, ex: Pizzas"
32                    required
33                    className={styles.input}
34            </form>
35        </main>
36    )
37}
```

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:3000/das...
- Header:** "SujeitoPizza" logo and navigation links for "Categoria" and "Produto".
- Section:** "Nova Categoria"
- Form:** An input field labeled "Nome da categoria, ex: Pizzas" and a green "Cadastrar" button.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (Left):**

- Explorer:** Shows the project structure. The **FRONTEND** folder is expanded, showing **.next**, **node\_modules**, **public**, and **src**. The **src** folder contains **app**, **dashboard**, and **category**. The **category** folder is selected, showing files: **page.tsx** (selected), **styles.module.scss**, **components**, **product**, **layout.tsx**, **page.tsx**, **fonts**, **signup**, **favicon.ico**, **globals.scss**, **layout.tsx**, **page.module.scss**, **page.tsx**, **lib**, **services**, **middleware.ts**, **.gitignore**, **next-env.d.ts**, **next.config.ts**, **OUTLINE**, and **TIMELINE**.
- Editor:** The **page.tsx** file is open. The code defines a **Category** component that handles the registration of a new category. It uses the **use server** hook to handle the form submission. The component displays a "Nova Categoria" heading and a form for entering a category name. The **handleRegisterCategory** function logs the category name to the console.

**Browser Preview (Right):**

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:3000/dashboard/category
- Content:** The page title is **SujeitoPizza**. The main heading is **Nova Categoria**. There is an input field containing the value **sorvete**. A green button labeled **Cadastrar** is visible.

At the bottom of the screen, there are status indicators: Ln 15, Col 1, Spaces: 2, UTF-8, CRLF, TypeScript JSX, Prettier.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure. The **FRONTEND** folder is expanded, showing **.next**, **node\_modules**, **public**, and **src**. The **src** folder contains **app**, **dashboard**, and **category**. The **category** folder contains **page.tsx** and **styles.module.scss**.
- File:** **page.tsx**
- Content:** The code defines a **Category** component. It includes imports for styles, a button component, and an API service. The component has a **handleRegisterCategory** function that logs the category name to the console. It then returns a main container with a heading "Nova Categoria" and a form for registering a new category. The form has a text input field for the category name.

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:3000/dashboard/category
- Header:** SujeitoPizza
- Navigation:** Categoria | Produto | [→]
- Section:** Nova Categoria
- Form:** Nome da categoria, ex: Pizzas
- Button:** Cadastrar

```
next-server (v15.0.3)
More info: https://sass-lang.com/d/legacy-js-api
Import trace for requested module:
./src/app/dashboard/components/button/styles.module.scss
./src/app/dashboard/components/button/index.tsx
./src/app/dashboard/components/header/styles.module.scss
Deprecation The legacy JS API is deprecated and will be removed in Dart Sass 2.0.0.
More info: https://sass-lang.com/d/legacy-js-api
Import trace for requested module:
./src/app/dashboard/components/header/styles.module.scss
./src/app/dashboard/components/header/index.tsx
./src/app/dashboard/components/orders/styles.module.scss
Deprecation The legacy JS API is deprecated and will be removed in Dart Sass 2.0.0.
More info: https://sass-lang.com/d/legacy-js-api
Import trace for requested module:
./src/app/dashboard/components/orders/styles.module.scss
./src/app/dashboard/components/orders/index.tsx
./src/app/dashboard/page.tsx
./src/app/page.module.scss
Deprecation The legacy JS API is deprecated and will be removed in Dart Sass 2.0.0.
More info: https://sass-lang.com/d/legacy-js-api
Import trace for requested module:
./src/app/page.module.scss
./src/app/page.tsx
GET /dashboard 200 in 713ms
POST / 303 in 895ms
true
GET /dashboard 200 in 29ms
true
GET /dashboard/category 200 in 28ms
true
sorvete
POST /dashboard/category 200 in 15ms
```

Sujeito Pizza - A melhor pizzaria

localhost:300...

# SujeitoPizza

Categoria Produto

## Nova Categoria

Cadastrar

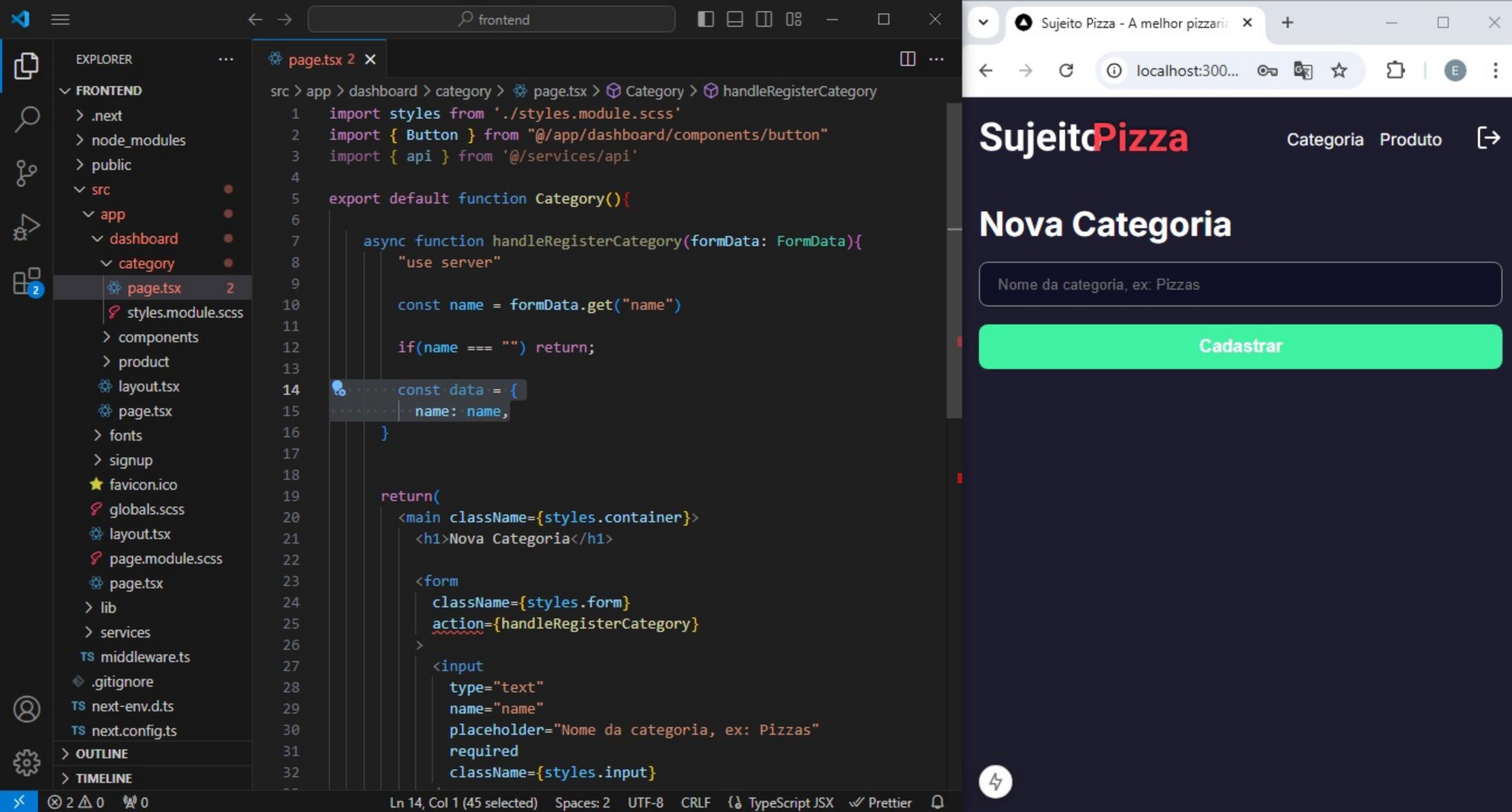
The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "page.tsx" is the active file.
- Content:** The code is a Next.js page component named "Category". It includes imports for styles, a button component, and an API service. It defines an "handleRegisterCategory" function using the "use server" hook. This function logs the category name to the console and returns a form for creating a new category. The form has a text input field for the category name.
- Bottom Status Bar:** Shows file statistics (Ln 14, Col 9 (18 selected)), code settings (Spaces: 2, UTF-8, CRLF), and tooling status (TypeScript JSX, Prettier).

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:300...
- Page Content:** Displays the "Nova Categoria" (New Category) form. It has a text input field labeled "Nome da categoria, ex: Pizzas" and a green "Cadastrar" (Register) button.
- Header:** Shows the "SujeitoPizza" logo and navigation links for "Categoria" and "Produto".



The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "page.tsx" is the active file.
- Content:** The code is for a category creation page. It includes an asynchronous function "handleRegisterCategory" that sends a POST request to "/category".
- Editor Status:** Line 18 has a yellow lightbulb icon, indicating a potential issue or suggestion.
- Bottom Bar:** Includes file navigation icons (X, ▲, △, A), status bar (Ln 18, Col 1 (102 selected)), and toolbars for Spaces, UTF-8, CRLF, TypeScript JSX, Prettier, and a bell icon.

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:300...
- Page Content:**
  - Header:** SujeitoPizza (Logo) | Categoria | Produto | [Entrar]
  - Section:** Nova Categoria
  - Form:** Nome da categoria, ex: Pizzas (Input field)
  - Button:** Cadastrar (Green button)

```
next-server (v15.0.3)

More info: https://sass-lang.com/d/legacy-js-api

Import trace for requested module:
./src/app/dashboard/components/button/styles.module.scss
./src/app/dashboard/components/button/index.tsx

./src/app/dashboard/components/header/styles.module.scss
Deprecation The legacy JS API is deprecated and will be removed in Dart Sass 2.0.0.

More info: https://sass-lang.com/d/legacy-js-api

Import trace for requested module:
./src/app/dashboard/components/header/styles.module.scss
./src/app/dashboard/components/header/index.tsx
  ./src/app/dashboard/category/styles.module.scss
Deprecation The legacy JS API is deprecated and will be removed in Dart Sass 2.0.0.

More info: https://sass-lang.com/d/legacy-js-api

Import trace for requested module:
./src/app/dashboard/category/styles.module.scss
./src/app/dashboard/category/page.tsx

./src/app/dashboard/components/button/styles.module.scss
Deprecation The legacy JS API is deprecated and will be removed in Dart Sass 2.0.0.

More info: https://sass-lang.com/d/legacy-js-api

Import trace for requested module:
./src/app/dashboard/components/button/styles.module.scss
./src/app/dashboard/components/button/index.tsx

./src/app/dashboard/components/header/styles.module.scss
Deprecation The legacy JS API is deprecated and will be removed in Dart Sass 2.0.0.

More info: https://sass-lang.com/d/legacy-js-api

Import trace for requested module:
./src/app/dashboard/components/header/styles.module.scss
./src/app/dashboard/components/header/index.tsx
GET /favicon.ico 200 in 861ms
```

Sujeito Pizza - A melhor pizzaria

localhost:3000/das...

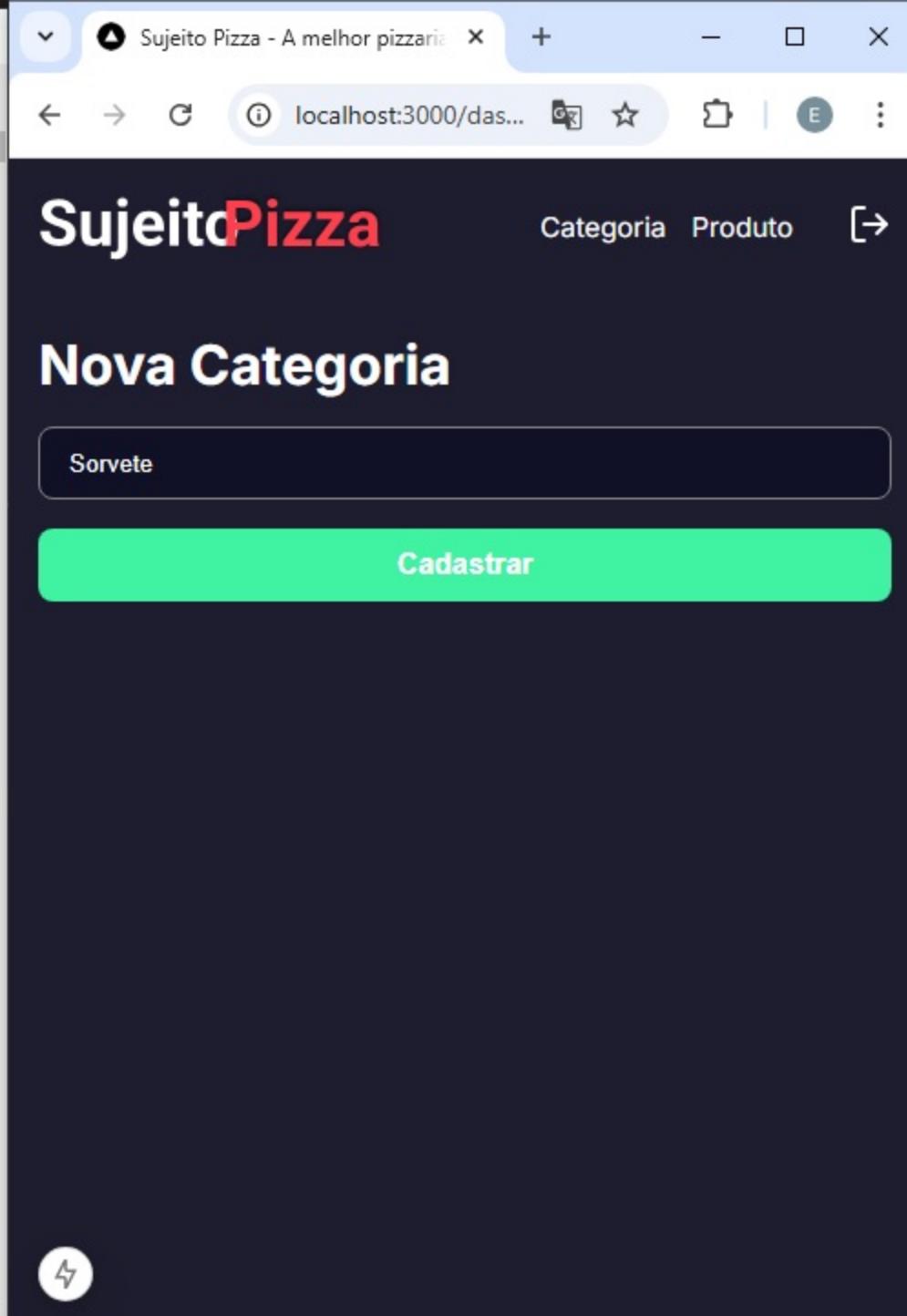
# SujeitoPizza

Categoria Produto [→]

## Nova Categoria

Sorvete

Cadastrar



```
next-server (v15.0.3)

./src/app/dashboard/components/header/styles.module.scss
./src/app/dashboard/components/header/index.tsx
GET /favicon.ico 200 in 861ms
true
Error [AxiosError]: Request failed with status code 401
  at async handleRegisterCategory (src\app\dashboard\category\page.tsx:18:6)
  at async (webpack://next/dist/src/server/app-render/action-handler.ts:881:24)
  at async handleAction (webpack://next/dist/src/server/app-render/action-handler.ts:617:4)
  at async renderToHTMLOrFlightImpl (webpack://next/dist/src/server/app-render/app-render.tsx:1310:34)
  at async doRender (node_modules\next\src\server\base-server.ts:2666:21)
  at async responseGenerator (node_modules\next\src\server\base-server.ts:3027:21)
  at async DevServer.renderToResponseWithComponentsImpl (node_modules\next\src\server\base-server.ts:3039:23)
  at async DevServer.renderPageComponent (node_modules\next\src\server\base-server.ts:3597:15)
  at async DevServer.renderToResponseImpl (node_modules\next\src\server\base-server.ts:3659:23)
  at async DevServer.pipeImpl (node_modules\next\src\server\base-server.ts:1698:20)
  at async NextNodeServer.handleCatchallRenderRequest (node_modules\next\src\server\next-server.ts:1034:6)
  at async DevServer.handleRequestImpl (node_modules\next\src\server\base-server.ts:1462:8)
  at async (node_modules\next\src\server\dev\next-dev-server.ts:514:13)
  at async Span.traceAsyncFn (node_modules\next\src\trace\trace.ts:143:13)
  at async DevServer.handleRequest (node_modules\next\src\server\dev\next-dev-server.ts:512:19)
  at async invokeRender (node_modules\next\src\server\lib\router-server.ts:284:10)
  at async handleRequest (node_modules\next\src\server\lib\router-server.ts:530:15)
  at async requestHandlerImpl (node_modules\next\src\server\lib\router-server.ts:576:6)
  at async Server.requestListener (node_modules\next\src\server\lib\start-server.ts:146:6)
16 |
17 |
18     await api.post("/category", data, )
19     ^
20     .catch((err) => {
21         console.log(err);
code: 'ERR_BAD_REQUEST',
config: [Object],
request: [ClientRequest],
response: [Object],
status: 401,
constructor: [Function],
toJSON: [Function: toJSON]
}
POST /dashboard/category 200 in 735ms
```

Sujeito Pizza - A melhor pizzaria

SujeitoPizza

Categoria Produto

## Nova Categoria

Nome da categoria, ex: Pizzas

Cadastrar

The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- File Explorer:** Shows the project structure under "FRONTEND".
- Search Bar:** Contains the text "frontend".
- Code Editor Area:** Displays the file "page.tsx" with the following content:

```
 3 import { api } from '@/services/api'
 4 import { redirect } from 'next/navigation'
 5 import { getCookieServer } from '@/lib/cookieServer'
 6
 7 export default function Category(){
 8
 9     async function handleRegisterCategory(formData: FormData){
10         "use server"
11
12         const name = formData.get("name")
13
14         if(name === "") return;
15
16         const data = {
17             name: name,
18         }
19
20         const token = await getCookieServer();
21
22         await api.post("/category", data, {
23             headers: {
24                 Authorization: `Bearer ${token}`
25             }
26         })
27         .catch((err) => {
28             console.log(err);
29         })
30
31     }
32
33 }
```
- Bottom Status Bar:** Shows "2 selections (365 characters selected)", "Spaces: 2", "UTF-8", "CRLF", "TypeScript JSX", "Prettier", and a bell icon.

**Browser Preview:**

- Title Bar:** "Sujeito Pizza - A melhor pizzaria" and "localhost:3000/das...".
- Header:** "SujeitoPizza" logo, "Categoria", "Produto", and a search bar placeholder "Nome da categoria, ex: Pizzas".
- Content:** "Nova Categoria" heading, a text input field, and a green "Cadastrar" button.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "page.tsx" is the active file.
- Content:** The code is a TypeScript function for handling category registration. It imports "api", "redirect", and "getCookieServer". It uses "use server" for the function body. It checks if the name is empty and returns if true. It creates a "data" object with "name". It gets a token from "getCookieServer". It performs a POST request to "/category" with the "data" object and a header "Authorization: Bearer \${token}". It catches any errors and logs them. Finally, it logs the response.

```
import { api } from '@/services/api'
import { redirect } from 'next/navigation'
import { getCookieServer } from '@/lib/cookieServer'

export default function Category() {
  async function handleRegisterCategory(formData: FormData) {
    "use server"

    const name = formData.get("name")

    if(name === "") return;

    const data = {
      name: name,
    }

    const token = await getCookieServer()

    const response = await api.post("/category", data, {
      headers: {
        Authorization: `Bearer ${token}`
      }
    })
    .catch((err) => {
      console.log(err);
    })
  }

  console.log(response);
}
```

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:3000/das...
- Header:** SujeitoPizza
- Navigation:** Categoria | Produto | [ ]
- Section:** Nova Categoria
- Form:** Nome da categoria, ex: Pizzas
- Button:** Cadastrar

```
next-server (v15.0.3) Sujeito Pizza - A melhor pizzaria + - □ X
[Symbol(kHasPrimitive)]: false,
[Symbol(asyncId)]: 244677,
[Symbol(triggerId)]: 244675,
[Symbol(kResourceStore)]: undefined,
[Symbol(kResourceStore)]: undefined,
[Symbol(kResourceStore)]: undefined,
[Symbol(kResourceStore)]: undefined,
[Symbol(kResourceStore)]: [Object],
[Symbol(kResourceStore)]: [Object],
[Symbol(kResourceStore)]: undefined,
[Symbol(kResourceStore)]: undefined,
[Symbol(kResourceStore)]: undefined,
[Symbol(kResourceStore)]: undefined,
[Symbol(kBuffer)]: null,
[Symbol(kBufferCb)]: null,
[Symbol(kBufferGen)]: null,
[Symbol(shapeMode)]: true,
[Symbol(kCapture)]: false,
[Symbol(kSetNoDelay)]: true,
[Symbol(kSetKeepAlive)]: true,
[Symbol(kSetKeepAliveInitialDelay)]: 1,
[Symbol(kBytesRead)]: 0,
[Symbol(kBytesWritten)]: 0
},
[Symbol(kOutHeaders)]: [Object: null prototype] {
  accept: [Array],
  'content-type': [Array],
  authorization: [Array],
  'user-agent': [Array],
  'content-length': [Array],
  'accept-encoding': [Array],
  host: [Array]
},
[Symbol(errorred)]: null,
[Symbol(kHighWaterMark)]: 16384,
[Symbol(kRejectNonStandardBodyWrites)]: false,
[Symbol(kUniqueHeaders)]: null
},
data: { id: 'f44fb594-d1c4-425f-b1bd-87541a6637a1', name: 'Sorvete' }
}
POST /dashboard/category 200 in 142ms
```

# SujeitoPizza

Categoria Produto [→]

## Nova Categoria

Nome da categoria, ex: Pizzas

Cadastrar

next-server (v15.0.3)

```
Import trace for requested module:  
.src/app/dashboard/components/button/styles.module.scss  
.src/app/dashboard/components/button/index.tsx  
  
.src/app/dashboard/components/header/styles.module.scss  
Deprecation The legacy JS API is deprecated and will be removed in Dart Sass 2.0.0.  
More info: https://sass-lang.com/d/legacy-js-api  
  
Import trace for requested module:  
.src/app/dashboard/components/header/styles.module.scss  
.src/app/dashboard/components/header/index.tsx  
GET /favicon.ico 200 in 750ms  
true  
{  
  status: 200,  
  statusText: 'OK',  
  headers: Object [AxiosHeaders] {  
    'x-powered-by': 'Express',  
    'access-control-allow-origin': '*',  
    'content-type': 'application/json; charset=utf-8',  
    'content-length': '62',  
    etag: 'W/"3e-9zb9L6bMhmMKOTRftv/sIvsMKQ0"',  
    date: 'Thu, 28 Nov 2024 00:36:47 GMT',  
    connection: 'keep-alive',  
    'keep-alive': 'timeout=5'  
  },  
  config: {  
    transitional: {  
      silentJSONParsing: true,  
      forcedJSONParsing: true,  
      clarifyTimeoutError: false  
    },  
    adapter: [ 'xhr', 'http', 'fetch' ],  
    transformRequest: [ [Function: transformRequest] ],  
    transformResponse: [ [Function: transformResponse] ],  
    timeout: 0,  
    xsrfCookieName: 'XSRF-TOKEN',  
    xsrfHeaderName: 'X-XSRF-TOKEN',  
    maxContentLength: -1,  
    maxBodyLength: -1,  
    env: { FormData: [Function], Blob: [class Blob] },  
    validateStatus: [Function: validateStatus],
```

Sujeito Pizza - A melhor pizzaria

localhost:3000/das...

# SujeitoPizza

Categoria Produto

## Nova Categoria

Nome da categoria, ex: Pizzas

Cadastrar

```
next-server (v15.0.3)

[Symbol(kResourceStore)]: [Object]
},
[Symbol(kBuffer)]: null,
[Symbol(kBufferCb)]: null,
[Symbol(kBufferGen)]: null,
[Symbol(shapeMode)]: true,
[Symbol(kCapture)]: false,
[Symbol(kSetNoDelay)]: true,
[Symbol(kSetKeepAlive)]: true,
[Symbol(kSetKeepAliveInitialDelay)]: 1,
[Symbol(kBytesRead)]: 0,
[Symbol(kBytesWritten)]: 0
},
[Symbol(kOutHeaders)]: [Object: null prototype] {
  accept: [Array],
  'content-type': [Array],
  authorization: [Array],
  'user-agent': [Array],
  'content-length': [Array],
  'accept-encoding': [Array],
  host: [Array]
},
[Symbol(errored)]: null,
[Symbol(kHighWaterMark)]: 16384,
[Symbol(kRejectNonStandardBodyWrites)]: false,
[Symbol(kUniqueHeaders)]: null
},
data: { id: 'f44fb594-d1c4-425f-b1bd-87541a6637a1', name: 'Sorvete' }
}
POST /dashboard/category 200 in 142ms
```

Sujeito Pizza - A melhor pizzaria

localhost:3000/das...

# SujeitoPizza

Categoria Produto [→]

## Nova Categoria

Nome da categoria, ex: Pizzas

Cadastrar

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C +

Query #1 categories [all] x +

id equals Enter Value

	id	name	created_at	updated_at
1	00d8f79f-5536-429a-bd59-7a82f18700ac	bebidas	2024-11-21 23:37:54.506	2024-11-21 23:37:54....
2	80de4305-34d1-418c-a215-455a38a57fa4	sobremesas	2024-11-22 00:08:08.794	2024-11-22 00:08:08....
3	e21422b1-f695-4426-aa06-9d23c2f3502b	pizzas	2024-11-21 23:39:55.683	2024-11-21 23:39:55....

ENTITIES 217 +

- public
  - \_prisma\_migrations
  - categories
  - items
  - orders
  - products
  - users
- information\_schema
- pg\_catalog

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria ▾ C + ↻ Query #1 categories [all] +

Filter ↻ id equals Enter Value + ⚡

ENTITIES 217 +

public

- \_prisma\_migrations
- categories
- items
- orders
- products
- users

information\_schema

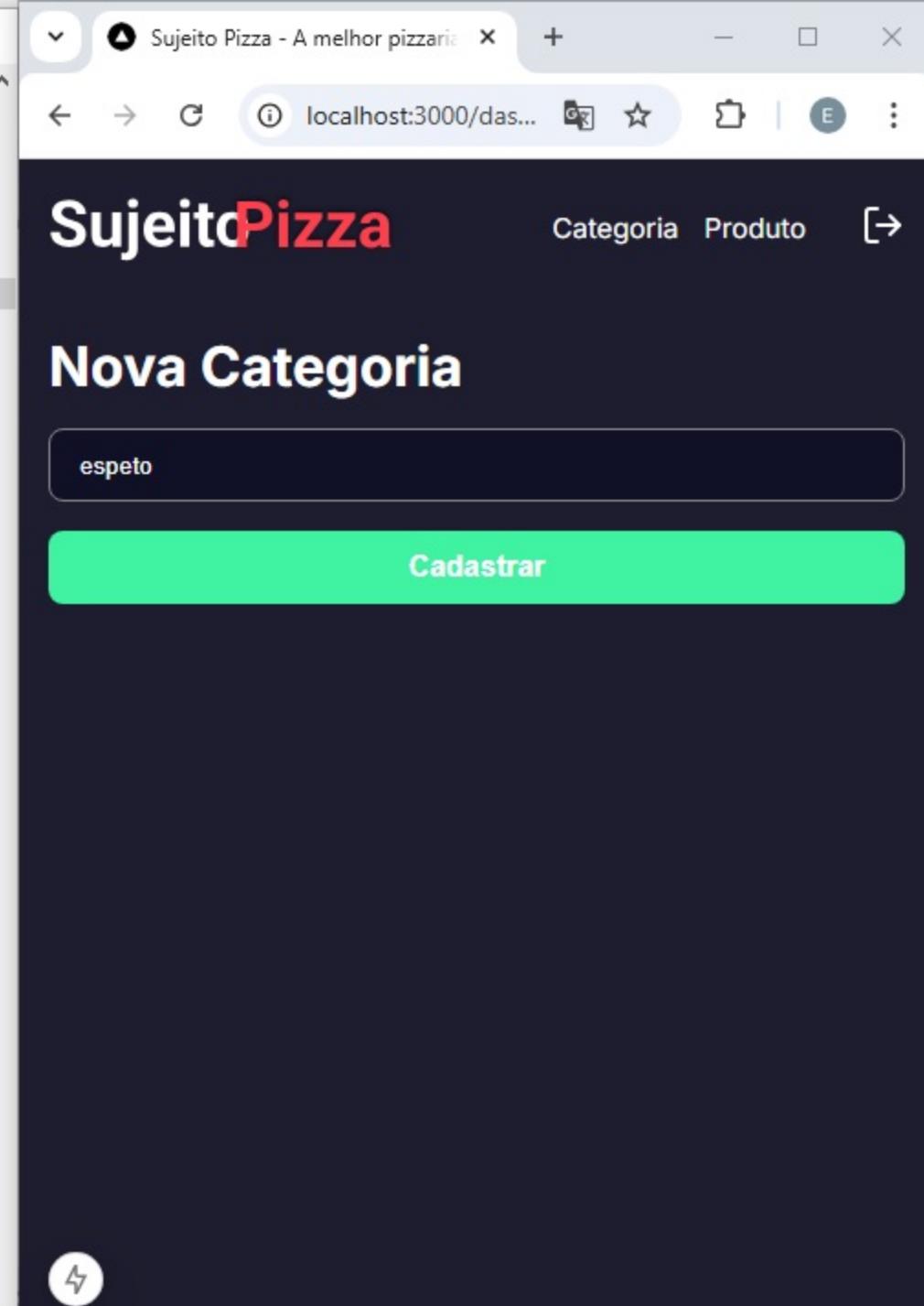
pg\_catalog

	id	name	created_at	updated_at
1	00d8f79f-5536-429a-bd59-7a82f18700ac	bebidas	2024-11-21 23:37:54.506	2024-11-21 23:37:54...
2	80de4305-34d1-418c-a215-455a38a57fa4	sobremesas	2024-11-22 00:08:08.794	2024-11-22 00:08:08...
3	e21422b1-f695-4426-aa06-9d23c2f3502b	pizzas	2024-11-21 23:39:55.683	2024-11-21 23:39:55...
4	f44fb594-d1c4-425f-b1bd-87541a6637a1	Sorvete	2024-11-28 00:36:47.872	2024-11-28 00:36:47...

Refresh Table (ctrl+r or F5)

```
next-server (v15.0.3)

  [Symbol(kResourceStore)]: [Object]
},  
  [Symbol(kBuffer)]: null,  
  [Symbol(kBufferCb)]: null,  
  [Symbol(kBufferGen)]: null,  
  [Symbol(shapeMode)]: true,  
  [Symbol(kCapture)]: false,  
  [Symbol(kSetNoDelay)]: true,  
  [Symbol(kSetKeepAlive)]: true,  
  [Symbol(kSetKeepAliveInitialDelay)]: 1,  
  [Symbol(kBytesRead)]: 0,  
  [Symbol(kBytesWritten)]: 0
},  
[Symbol(kOutHeaders)]: [Object: null prototype] {  
  accept: [Array],  
  'content-type': [Array],  
  authorization: [Array],  
  'user-agent': [Array],  
  'content-length': [Array],  
  'accept-encoding': [Array],  
  host: [Array]
},  
[Symbol(errored)]: null,  
[Symbol(kHighWaterMark)]: 16384,  
[Symbol(kRejectNonStandardBodyWrites)]: false,  
[Symbol(kUniqueHeaders)]: null
},  
data: { id: 'f44fb594-d1c4-425f-b1bd-87541a6637a1', name: 'Sorvete' }
}  
POST /dashboard/category 200 in 142ms
```



```
next-server (v15.0.3)

[Symbol(kResourceStore)]: undefined,
[Symbol(kResourceStore)]: undefined,
[Symbol(kResourceStore)]: undefined,
[Symbol(kResourceStore)]: [Object]
},
[Symbol(kBuffer)]: null,
[Symbol(kBufferCb)]: null,
[Symbol(kBufferGen)]: null,
[Symbol(shapeMode)]: true,
[Symbol(kCapture)]: false,
[Symbol(kSetNoDelay)]: true,
[Symbol(kSetKeepAlive)]: true,
[Symbol(kSetKeepAliveInitialDelay)]: 1,
[Symbol(kBytesRead)]: 0,
[Symbol(kBytesWritten)]: 0
},
[Symbol(kOutHeaders)]: [Object: null prototype] {
accept: [Array],
'content-type': [Array],
authorization: [Array],
'user-agent': [Array],
'content-length': [Array],
'accept-encoding': [Array],
host: [Array]
},
[Symbol(errorred)]: null,
[Symbol(kHighWaterMark)]: 16384,
[Symbol(kRejectNonStandardBodyWrites)]: false,
[Symbol(kUniqueHeaders)]: null
},
data: { id: 'df84bb27-a449-4c28-81eb-8c9908d69d12', name: 'espeto' }
}
POST /dashboard/category 200 in 50ms
```



```
next-server (v15.0.3)

[Symbol(kResourceStore)]: [Object]
},
[Symbol(kBuffer)]: null,
[Symbol(kBufferCb)]: null,
[Symbol(kBufferGen)]: null,
[Symbol(shapeMode)]: true,
[Symbol(kCapture)]: false,
[Symbol(kSetNoDelay)]: true,
[Symbol(kSetKeepAlive)]: true,
[Symbol(kSetKeepAliveInitialDelay)]: 1,
[Symbol(kBytesRead)]: 0,
[Symbol(kBytesWritten)]: 0
},
[Symbol(kOutHeaders)]: [Object: null prototype] {
  accept: [Array],
  'content-type': [Array],
  authorization: [Array],
  'user-agent': [Array],
  'content-length': [Array],
  'accept-encoding': [Array],
  host: [Array]
},
[Symbol(errorred)]: null,
[Symbol(kHighWaterMark)]: 16384,
[Symbol(kRejectNonStandardBodyWrites)]: false,
[Symbol(kUniqueHeaders)]: null
},
data: { id: 'df84bb27-a449-4c28-81eb-8c9908d69d12', name: 'espeto' }
}
POST /dashboard/category 200 in 50ms
```

# Sujeito Pizza - A melhor pizzaria

localhost:3000/das...

## Nova Categoria

Cadastrar

```
next-server (v15.0.3)

  [Symbol(kResourceStore)]: [Object]
},
[Symbol(kBuffer)]: null,
[Symbol(kBufferCb)]: null,
[Symbol(kBufferGen)]: null,
[Symbol(shapeMode)]: true,
[Symbol(kCapture)]: false,
[Symbol(kSetNoDelay)]: true,
[Symbol(kSetKeepAlive)]: true,
[Symbol(kSetKeepAliveInitialDelay)]: 1,
[Symbol(kBytesRead)]: 0,
[Symbol(kBytesWritten)]: 0
},
[Symbol(kOutHeaders)]: [Object: null prototype] {
  accept: [Array],
  'content-type': [Array],
  authorization: [Array],
  'user-agent': [Array],
  'content-length': [Array],
  'accept-encoding': [Array],
  host: [Array]
},
[Symbol(errored)]: null,
[Symbol(kHighWaterMark)]: 16384,
[Symbol(kRejectNonStandardBodyWrites)]: false,
[Symbol(kUniqueHeaders)]: null
},
data: { id: '1e4ffc20-04b2-46db-96f0-ceea3ca2975a', name: 'hamburger' }
}
POST /dashboard/category 200 in 39ms
```

Sujeito Pizza - A melhor pizzaria

localhost:3000/das...

SujeitoPizza

Categoria Produto [→]

## Nova Categoria

Nome da categoria, ex: Pizzas

Cadastrar

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C + Query #1 categories [all] +

Filter ENTITIES 217 +

public

- \_prisma\_migrations
- categories
- items
- orders
- products
- users

information\_schema

pg\_catalog

	id	equals	Enter Value	
1	00d8f79f-5536-429a-bd59-7a82f18700ac	bebidas	2024-11-21 23:37:54.506	2024-11-21 23:37:54...
2	80de4305-34d1-418c-a215-455a38a57fa4	sobremesas	2024-11-22 00:08:08.794	2024-11-22 00:08:08...
3	e21422b1-f695-4426-aa06-9d23c2f3502b	pizzas	2024-11-21 23:39:55.683	2024-11-21 23:39:55...
4	f44fb594-d1c4-425f-b1bd-87541a6637a1	Sorvete	2024-11-28 00:36:47.872	2024-11-28 00:36:47...

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

**pizzaria** C + Query #1 categories [all] x +

Filter Y

ENTITIES 217 +

- public
  - \_prisma\_migrations
  - categories
  - items
  - orders
  - products
  - users
- information\_schema
- pg\_catalog

	id	equals	Enter Value	
1	00d8f79f-5536-429a-bd59-7a82f18700ac	bebidas	2024-11-21 23:37:54.506	2024-11-21 23:37:54...
2	1e4ffc20-04b2-46db-96f0-ceea3ca2975a	hamburger	2024-11-28 00:42:45.771	2024-11-28 00:42:45...
3	80de4305-34d1-418c-a215-455a38a57fa4	sobremesas	2024-11-22 00:08:08.794	2024-11-22 00:08:08...
4	df84bb27-a449-4c28-81eb-8c9908d69d12	espeto	2024-11-28 00:41:32.767	2024-11-28 00:41:32...
5	e21422b1-f695-4426-aa06-9d23c2f3502b	pizzas	2024-11-21 23:39:55.683	2024-11-21 23:39:55...
6	f44fb594-d1c4-425f-b1bd-87541a6637a1	Sorvete	2024-11-28 00:36:47.872	2024-11-28 00:36:47...

Refresh Table (ctrl+r or F5)

The image shows a developer's workspace with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- File Explorer:** Shows the project structure under "FRONTEND".
- Search Bar:** Contains the text "frontend".
- Code Editor Area:** Displays the file "page.tsx" with the following content:

```
src > app > dashboard > category > page.tsx > Category > handleRegisterCategory
3 import { api } from '@/services/api'
4 import { redirect } from 'next/navigation'
5 import { getCookieServer } from '@/lib/cookieServer'
6
7 export default function Category(){
8
9     async function handleRegisterCategory(formData: FormData){
10         "use server"
11
12         const name = formData.get("name")
13
14         if(name === "") return;
15
16         const data = {
17             name: name,
18         }
19
20         const token = await getCookieServer();
21
22         const response = await api.post("/category", data, {
23             headers:{
24                 Authorization: `Bearer ${token}`
25             }
26         })
27         .catch((err) => {
28             console.log(err);
29         })
30
31         console.log(response);
32
33     }
34 }
```

**Browser Preview:**

- Title Bar:** "Sujeito Pizza - A melhor pizzaria" and "localhost:3000/das...".
- Header:** "SujeitoPizza" logo, "Categoria" and "Produto" navigation links, and a "Cadastrar" button.
- Form:** An input field labeled "Nome da categoria, ex: Pizzas" and a green "Cadastrar" button.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "page.tsx" is the active file.
- Content:** The code handles form submission to register a category. It uses `FormData` to get the name, `getCookieServer` to get a token, and `api.post` to send the data to the "/category" endpoint. It includes error handling and redirection to the dashboard.

```
const name = formData.get("name");
if(name === "") return;

const data = {
  name: name,
}

const token = await getCookieServer();

await api.post("/category", data, {
  headers: {
    Authorization: `Bearer ${token}`
  }
})
.catch((err) => {
  console.log(err);
  return;
})

redirect("/dashboard")

return(
  <main className={styles.container}>
    <h1>Nova Categoria</h1>

    <form
      className={styles.form}
      action={handleRegisterCategory}
    >
```

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:3000/das...
- Header:** SujeitoPizza
- Navigation:** Categoria | Produto | [ ]
- Section:** Nova Categoria
- Form:** Nome da categoria, ex: Pizzas
- Button:** Cadastrar

The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure. The **FRONTEND** folder is expanded, showing **.next**, **node\_modules**, **public**, and **src**. The **src** folder contains **app**, **dashboard**, and **category**. The **category** folder is expanded, showing **page.tsx** (selected), **styles.module.scss**, **components**, **product**, **layout.tsx**, and **page.tsx**.
- Search:** Shows results for **frontend** across the project.
- File:** Shows files like **favicon.ico**, **globals.scss**, **layout.tsx**, **page.module.scss**, **page.tsx**, **lib**, **services**, **middleware.ts**, **.gitignore**, **next-env.d.ts**, and **next.config.ts**.
- Outline:** Shows the outline of the current file (**page.tsx**).
- Timeline:** Shows the timeline of the current file.

**Browser Preview:**

The browser window displays the application "Sujeito Pizza - A melhor pizzaria". The URL is [localhost:3000/das...](http://localhost:3000/das...). The page title is "SujeitoPizza".

The page content is "Nova Categoria". It features a sidebar with "acompanhamentos" and a large green button labeled "Cadastrar".

The code in the editor is as follows:

```
src > app > dashboard > category > page.tsx > Category > handleRegisterCategory
  1 import styles from './styles.module.scss'
  2 import { Button } from '@/app/dashboard/components/button'
  3 import { api } from '@/services/api'
  4 import { redirect } from 'next/navigation'
  5 import { getCookieServer } from '@/lib/cookieServer'

  6
  7 export default function Category(){
  8
  9   async function handleRegisterCategory(formData: FormData){
 10     "use server"
 11
 12     const name = formData.get("name")
 13
 14     if(name === "") return;
 15
 16     const data = {
 17       name: name,
 18     }
 19
 20     const token = await getCookieServer();
 21
 22     await api.post("/category", data, {
 23       headers:{
 24         Authorization: `Bearer ${token}`
 25       }
 26     })
 27     .catch((err) => {
 28       console.log(err);
 29       return;
 30     })
 31
 32     redirect("/dashboard")
 33   }
 34
 35 }
```

The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "page.tsx" is the active file.
- Content:** The code is a TypeScript function named "Category". It handles a POST request to "/category" using the "api" service and "getCookieServer". It logs errors to the console and redirects to "/dashboard".

```
import styles from './styles.module.scss'
import { Button } from '@/app/dashboard/components/button'
import { api } from '@/services/api'
import { redirect } from 'next/navigation'
import { getCookieServer } from '@/lib/cookieServer'

export default function Category(){
    async function handleRegisterCategory(formData: FormData){
        "use server"

        const name = formData.get("name")

        if(name === "") return;

        const data = {
            name: name,
        }

        const token = await getCookieServer();

        await api.post("/category", data, {
            headers:{
                Authorization: `Bearer ${token}`
            }
        })
        .catch((err) => {
            console.log(err);
            return;
        })
    }

    redirect("/dashboard")
}
```

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:3000/das...
- Header:** SujeitoPizza
- Navigation:** Categoria | Produto | [Entrar]
- Section:** Últimos pedidos
- Items:** Mesa 10, Mesa 13

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C +

Query #1 categories [all] x +

id equals Enter Value

ENTITIES 217 +

public

- \_prisma\_migrations
- categories
- items
- orders
- products
- users

information\_schema

pg\_catalog

	id	name	created_at	updated_at
1	00d8f79f-5536-429a-bd59-7a82f18700ac	bebidas	2024-11-21 23:37:54.506	2024-11-21 23:37:54...
2	1e4ffc20-04b2-46db-96f0-ceea3ca2975a	hamburger	2024-11-28 00:42:45.771	2024-11-28 00:42:45...
3	80de4305-34d1-418c-a215-455a38a57fa4	sobremesas	2024-11-22 00:08:08.794	2024-11-22 00:08:08...
4	df84bb27-a449-4c28-81eb-8c9908d69d12	espeto	2024-11-28 00:41:32.767	2024-11-28 00:41:32...
5	e21422b1-f695-4426-aa06-9d23c2f3502b	pizzas	2024-11-21 23:39:55.683	2024-11-21 23:39:55...
6	f44fb594-d1c4-425f-b1bd-87541a6637a1	Sorvete	2024-11-28 00:36:47.872	2024-11-28 00:36:47...

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C +

Query #1 categories [all] x +

id equals Enter Value

ENTITIES 217 +

public

- \_prisma\_migrations
- categories
- items
- orders
- products
- users

information\_schema

pg\_catalog

	id	name	created_at	updated_at
1	00d8f79f-5536-429a-bd59-7a82f18700ac	bebidas	2024-11-21 23:37:54.506	2024-11-21 23:37:54...
2	1e4ffc20-04b2-46db-96f0-ceea3ca2975a	hamburger	2024-11-28 00:42:45.771	2024-11-28 00:42:45...
3	3c1fd0e0-f3ad-458f-98c0-f10798bad20d	acompanhamentos	2024-11-28 00:49:20.371	2024-11-28 00:49:20...
4	80de4305-34d1-418c-a215-455a38a57fa4	sobremesas	2024-11-22 00:08:08.794	2024-11-22 00:08:08...
5	df84bb27-a449-4c28-81eb-8c9908d69d12	espeto	2024-11-28 00:41:32.767	2024-11-28 00:41:32...
6	e21422b1-f695-4426-aa06-9d23c2f3502b	pizzas	2024-11-21 23:39:55.683	2024-11-21 23:39:55...
7	f44fb594-d1c4-425f-b1bd-87541a6637a1	Sorvete	2024-11-28 00:36:47.872	2024-11-28 00:36:47...

Refresh Table (ctrl+r or F5)

**UC:** Projetos

**Assunto:**

Projeto Pizzaria – Frontend\_02 (NextJS)

Sistema Pizzaria Web

52 – Página Cadastro Produtos

The image shows a developer's workspace with two main windows: a code editor and a web browser.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure for a Next.js application named "frontend".
- Frontend Folder:** Contains ".next", "node\_modules", "public", "src", "app", "dashboard", "category", "components", "product", and "lib".
- product Folder:** Contains "components", "page.tsx", "layout.tsx", and "page.module.scss".
- Global Configuration:** Contains "globals.css", "layout.tsx", "page.module.scss", and "page.tsx".
- Services:** Contains "middleware.ts", ".gitignore", "next-env.d.ts", and "next.config.ts".
- Outline and Timeline:** Shows "OUTLINE" and "TIMELINE" sections.

**Bottom Status Bar:**

- Icons for file operations (New, Open, Save, etc.)
- File count: 0
- Line count: 0
- Character count: 0

**Code Snippets:**

- Show All Commands: `Ctrl + Shift + P`
- Go to File: `Ctrl + P`
- Find in Files: `Ctrl + Shift + F`
- Toggle Full Screen: `F11`
- Show Settings: `Ctrl + ,`

**Browser:**

- Title:** Sujeito Pizza - A melhor pizzaria
- Address:** localhost:3000/das...
- Content:** The page displays the header "SujeitoPizza" with "Categoria" and "Produto" navigation links, and a main section titled "Página Novo Produto".

The image shows a developer's workspace with two main windows: a code editor and a web browser.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure for a Next.js application named "frontend".
  - FRONTEND**: .next, node\_modules, public, src (selected)
  - src**: app (selected), dashboard, category, components, product, components
  - product**: components (selected), form (selected), page.tsx, layout.tsx, page.tsx, fonts, signup, favicon.ico, globals.scss, layout.tsx, page.module.scss, page.tsx, lib, services, middleware.ts, .gitignore, next-env.d.ts
  - OUTLINE**, **TIMELINE**
- Search Bar:** frontend
- Bottom Status Bar:** Prettier

**Web Browser:**

- Title:** Sujeito Pizza - A melhor pizzaria
- Address Bar:** localhost:3000/das...
- Content:** The page displays the header "SujeitoPizza" with "Categoria" and "Produto" navigation links, and a main section titled "Página Novo Produto".

The image shows a developer's environment with two main windows: a code editor and a web browser.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
  - src
    - app
      - dashboard
        - category
        - components
      - product
        - components
          - form
  - index.tsx (selected file)
  - page.tsx
  - layout.tsx
  - page.tsx
  - fonts
  - signup
  - favicon.ico
  - globals.scss
  - layout.tsx
  - page.module.scss
  - page.tsx
  - lib
  - services
  - middleware.ts
  - .gitignore

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:3000/dashboard/novo
- Content:** The page displays the header "SujeitoPizza" and "Categoria Produto" navigation links, followed by the main title "Página Novo Produto".

The image shows a developer's workspace with two main windows: a code editor and a web browser.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure:
  - FRONTEND
  - .next
  - node\_modules
  - public
  - src
    - app
      - dashboard
        - category
        - components
      - product
        - components
          - form
            - index.tsx
    - fonts
    - signup
    - favicon.ico
    - globals.scss
    - layout.tsx
    - page.module.scss
    - page.tsx
    - lib
    - services
    - middleware.ts
- Search:** Shows the current search term: "frontend".
- Editor:** Displays the file "styles.module.scss" with the following content:

```
src > app > dashboard > product > components > form > styles.module.scss
```

**Browser:**

- Title:** Sujeito Pizza - A melhor pizzaria
- Address:** localhost:3000/das...
- Content:** The page displays the header "SujeitoPizza" and the sub-header "Página Novo Produto".

The image shows a developer's environment with two main windows: a code editor and a web browser.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- Search:** A search bar at the top.
- File List:** Shows files like `index.tsx`, `styles.module.scss`, etc.
- Code View:** Displays the `index.tsx` file content:

```
1  "use client"
2
3  export function Form(){
4      return(
5          <div>
6              <h1>Formulário</h1>
7          </div>
8      )
9  }
10
```

**Browser Preview:**

- Title Bar:** "Sujeito Pizza - A melhor pizzaria"
- Address Bar:** "localhost:3000/das..."
- Page Content:** "SujeitoPizza" logo, "Categoria" and "Produto" buttons, and a heading "Página Novo Produto".

**Bottom Status Bar:** Shows file paths, line counts (Ln 1, Col 1 (128 selected)), and other development tools like Prettier.

The image shows a developer's workspace with a code editor and a browser preview side-by-side.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure:
  - .next
  - node\_modules
  - public
  - src
    - app
      - dashboard
      - category
      - components
    - product
      - components
      - form
      - index.tsx**
      - styles.module.scss**
  - page.tsx
  - layout.tsx
  - page.tsx
  - fonts
  - signup
  - favicon.ico
  - globals.scss
  - layout.tsx
  - page.module.scss
  - page.tsx
  - lib
  - services
  - middleware.ts
- Search:** frontend
- Status Bar:** Ln 4, Col 1 (73 selected) Spaces: 4 UTF-8 CRLF TypeScript JSX Prettier

**Browser Preview:**

The browser shows a dark-themed website for "Sujeito Pizza". The header includes the logo "SujeitoPizza" and navigation links for "Categoria" and "Produto". The main content area displays the heading "Página Novo Produto".

```
1 export default function Product(){
2     return(
3         <main>
4             <h1>Página Novo Produto</h1>
5         </main>
6     )
7 }
8 }
```

The screenshot shows a developer's environment with a code editor, browser preview, and file navigation interface.

**Code Editor:**

- Explorer:** Shows the project structure under `FRONTEND`, including `.next`, `node_modules`, `public`, and `src` (with `app`, `dashboard`, `category`, `components`, `product`, `components`, `form`, `index.tsx`, and `styles.module.scss`).
- File List:** Shows files like `index.tsx`, `page.tsx` (selected), `styles.module.scss`, `layout.tsx`, `page.tsx`, `fonts`, `signup`, `favicon.ico`, `globals.scss`, `layout.tsx`, `page.module.scss`, `page.tsx`, `lib`, `services`, `middleware.ts`, `OUTLINE`, and `TIMELINE`.
- Code View:** Displays the `page.tsx` file content:

```
1 import { Form } from "./components/form"
2
3 export default function Product() {
4   return (
5     <Form/>
6   )
7 }
```

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:30...
- Content:** Shows the header "SujeitoPizza" and the page title "Formulário".

**Bottom Status Bar:**

- Ln 1, Col 1 (118 selected)
- Spaces: 4
- UTF-8
- CRLF
- TypeScript JSX
- Prettier

The image shows a developer's environment with a code editor and a browser preview side-by-side.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** index.tsx (selected), page.tsx, styles.module.scss
- Content:** index.tsx code (Lines 1-10).

```
1 "use client"
2
3 export function Form(){
4     return(
5         <div>
6             <h1>Formulário</h1>
7         </div>
8     )
9 }
```
- Bottom Status Bar:** 2 selections (100 characters selected) | Spaces: 4 | UTF-8 | CRLF | TypeScript JSX | Prettier

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:30...
- Content:** The page displays the title "SujeitoPizza" and the heading "Formulário".
- Navigation:** Back, Forward, Stop, Refresh, Home, Download, Error.

The screenshot displays a developer's environment with an IDE and a web browser side-by-side.

**IDE (Left):**

- EXPLORER:** Shows the project structure:
  - FRONTEND
  - .next
  - node\_modules
  - public
  - src
    - app
      - dashboard
        - category
        - components
      - product
        - components
          - form
        - index.tsx (selected)
        - styles.module.scss
      - page.tsx
      - layout.tsx
      - page.tsx
      - fonts
      - signup
- OUTLINE
- TIMELINE

**BROWSER (Right):**

- Title Bar:** localhost:3000/dashboard/prod
- Error Message:** 1 of 2 errors  
Console Error
- Text:** [ Server ] React.jsx: type is invalid -- expected a string (for built-in components) or a class/function (for composite components) but got: undefined. You likely forgot to export your component from the file it's defined in, or you might have mixed up default and named imports.
- Source:** src\app\dashboard\product\page.tsx  
(5:10) @ Form

```
3 | export default function Product(){  
4 |     return(  
5 |         <Form/>  
6 |     )  
7 | }
```

The image shows a developer's environment with a code editor and a browser preview side-by-side.

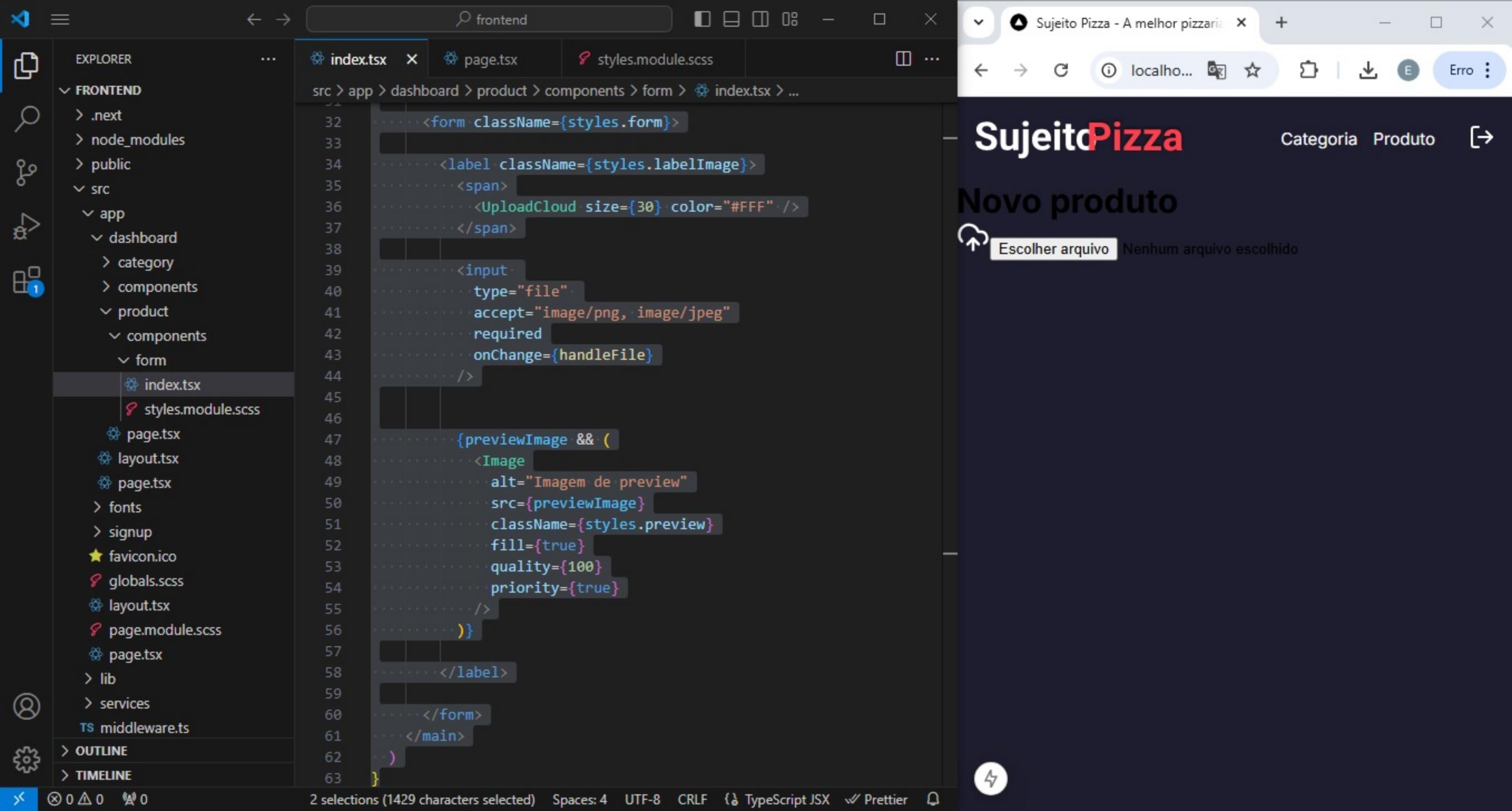
**Code Editor (VS Code):**

- File Explorer:** Shows the project structure under "FRONTEND".
- Current File:** "index.tsx" (selected), part of the "product" component.
- Content:** TypeScript code for a file input handling component.

```
1  "use client"
2
3  import { ChangeEvent, useState } from 'react'
4  import styles from './styles.module.scss'
5  import { UploadCloud } from 'lucide-react'
6  import Image from 'next/image'
7
8  export function Form(){
9    const [image, setImage] = useState<File>()
10   const [previewImage, setPreviewImage] = useState("")
11
12   function handleFile(e: ChangeEvent<HTMLInputElement>){
13     if(e.target.files && e.target.files[0]){
14       const image = e.target.files[0];
15
16       if(image.type !== "image/jpeg" && image.type !== "image/png"){
17         console.log("FORMATO PROIBIDO!!!!")
18         return;
19       }
20
21       setImage(image);
22       setPreviewImage(URL.createObjectURL(image))
23
24     }
25   }
26
27
28   return(
29     <main className={styles.container}>
30       <h1>Novo produto</h1>
31   
```

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- Page Content:** "Novo produto" with a file upload input field labeled "Escolher arquivo" which says "Nenhum arquivo escolhido".



The image shows a developer's environment with a code editor and a browser preview side-by-side.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "styles.module.scss" is the active file, located at `src > app > dashboard > product > components > form > styles.module.scss`.
- Content:** The code defines CSS rules for ".container", "h1", ".form", and ".labelImage".

```
.container{  max-width: 720px;  margin: 20px auto;  padding: 0 16px;  display: flex;  flex-direction: column;}h1{  color: var(--white);} .form{  display: flex;  flex-direction: column;  margin: 16px 0;} .labelImage{  width: 100%;  height: 280px;  position: relative;}
```

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** "SujeitoPizza" (in red), "Categoria", "Produto", and a search icon.
- Section:** "Novo produto"
- Form:** An input field with a placeholder "Escolher arquivo" and a note "Nenhum arquivo escolhido".

**Bottom Status Bar:**

- Ln 1, Col 1 (376 selected)
- Spaces: 4
- UTF-8
- CRLF
- SCSS
- Prettier

**UC:** Projetos

**Assunto:**

Projeto Pizzaria – Frontend\_02 (NextJS)

Sistema Pizzaria Web

53 – Preview Produto

The image shows a developer's environment with a code editor and a browser preview side-by-side.

**Code Editor (Left):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "styles.module.scss" is the active file.
- Content:** CSS styles for a form component. The selected code block is:

```
.form{  
  display: flex;  
  flex-direction: column;  
  margin: 16px 0;  
  
.select{  
  width: 100%;  
  height: 40px;  
  border-radius: 8px;  
  background-color: var(--dark-900);  
  color: var(--white);  
  padding: 0 8px;  
  border: 1px solid var(--gray-100);  
}  
  
.textarea{  
  width: 100%;  
  min-height: 120px;  
  resize: none;  
  padding: 8px;  
  color: var(--white);  
}  
  
}  
  
.labelImage{  
  width: 100%;  
  height: 280px;  
  position: relative;  
  background-color: var(--dark-900);  
  border-radius: 8px;
```

**Browser Preview (Right):**

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** SujeitoPizza, Categoria, Produto, [Entrar]
- Section:** Novo produto
- Content:** An empty form with a placeholder "Novo produto" and a "Enviar" button.

The screenshot shows a development setup with two main windows:

- Code Editor (Left):** An IDE window displaying a SCSS file (`styles.module.scss`) from a Next.js application. The file contains CSS rules for a form component, including styles for a label image, input field, and a span element with a hover state. The code editor has syntax highlighting and a dark theme.
- Browser and Error Panel (Right):** A browser window titled "Sujeito Pizza - A melhor pizzaria" showing a build error. The error message is "Failed to compile" and points to line 19 of the same `styles.module.scss` file. The error details state: "Syntax error: Selector 'select' is not pure (pure selectors must contain at least one local class or id)". The browser interface includes tabs, a search bar, and various developer tools.

Bottom status bar:

- Ln 82, Col 6 (815 selected)
- Spaces: 4
- UTF-8
- CRLF
- SCSS
- Prettier

The screenshot shows a development environment with two main windows. On the left is a code editor (VS Code) displaying a SCSS file named `styles.module.scss`. The file contains CSS rules for a `.input` class and a `.preview` class. The right window is a browser-based interface showing a "Build Error". The error message is "Failed to compile" and points to a syntax error in the `select` selector at line 19 of the same `styles.module.scss` file. The error message states: "Syntax error: Selector 'select' is not pure (pure selectors must contain at least one local class or id)". The code editor shows the file path as `src/app/dashboard/product/components/form/styles.module.scss`. The browser title bar says "Sujeito Pizza - A melhor pizzaria".

frontend

EXPLORER

FRONTEND

.next  
node\_modules  
public  
src  
app  
dashboard  
category  
components  
product  
components  
form  
index.tsx  
styles.module.scss  
page.tsx  
layout.tsx  
page.tsx  
fonts  
signup  
favicon.ico  
globals.scss  
layout.tsx  
page.module.scss  
page.tsx  
lib  
services  
middleware.ts

OUTLINE

TIMELINE

styles.module.scss src\app\dashboard\product\components\form

Ln 82, Col 6 (311 selected) Spaces: 4 UTF-8 CRLF { SCSS Prettier

styles.module.scss src\app\dashboard\product\components\form

styles.module.scss src\app\dashboard\product\components\form .input

67 .preview{  
68 width: 100%;  
69 height: 100%;  
70 border-radius: 8px;  
71 object-fit: cover;  
72 }  
73  
74 .input{  
75 border: 0;  
76 height: 40px;  
77 background-color: var(--dark-900);  
78 border: 1px solid var(--gray-100);  
79 border-radius: 8px;  
80 padding: 0 8px;  
81 color: var(--white);  
82 }

Build Error

Failed to compile

./src/app/dashboard/product/components/form/styles.module.scss:19:3

Syntax error: Selector "select" is not pure  
(pure selectors must contain at least one  
local class or id)

17 }  
18  
19 select {  
20 width: 100%;  
21 height: 40px;

This error occurred during the build process and can only be dismissed by fixing the error.

The image shows a developer's workspace with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- Search:** Search bar at the top.
- File List:** Shows files like index.tsx, page.tsx, styles.module.scss, etc.
- Editor:** Displays the content of index.tsx. The code includes a dropdown menu for categories (Pizzas, Massas) and two input fields for product name and price.

```
<Form>
  <label>
    quality={100}
    priority={true}
  </label>
  <select name="category">
    <option key={1} value={1}>
      Pizzas
    </option>
    <option key={1} value={1}>
      Massas
    </option>
  </select>
  <input type="text" name="name" placeholder="Digite o nome do produto..." required className={styles.input}>
  <input type="text" name="price" placeholder="Preço do produto..." required className={styles.input}>
</Form>
```

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** Categoria, Produto, [→]
- Section:** Novo produto
- Form Fields:** Category dropdown (Pizzas selected), Product name input, Price input, Description input.
- UI Elements:** Save button (cloud icon).

The image shows a developer's environment with a code editor and a browser preview side-by-side.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** index.tsx (selected), page.tsx, styles.module.scss.
- Search:** frontend
- Output:** 1 error
- Bottom Status Bar:** Ln 85, Col 1 (232 selected) Spaces: 4 UTF-8 CRLF TypeScript JSX Prettier

```
<input  
  type="text"  
  name="price"  
  placeholder="Preço do produto..."  
  required  
  className={styles.input}  
>  
  
<textarea  
  className={styles.input}  
  placeholder="Digite a descrição do produto..."  
  required  
  name="description"></textarea>  
  
<Button name="Cadastrar produto" />  
  
</form>  
</main>  
)  
}
```

**Browser Preview:**

The browser shows a dark-themed landing page for "Sujeito Pizza".

- Header:** SujeitoPizza (with a red "Pizza" logo), Categoria, Produto, [→]
- Title:** Novo produto
- Form Fields:** Input field for price, Textarea for product description, and a "Cadastrar produto" button.
- Bottom Buttons:** A large blue "Cadastrar" button with a white arrow icon, and a smaller "Cancelar" button with a red "X" icon.
- Side Panel:** A sidebar titled "Pizzas" with input fields for "Nome do produto", "Preço do produto", and "Descrição do produto".

The image shows a developer's environment with a code editor and a browser preview side-by-side.

**Code Editor (left):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx", "page.tsx", "styles.module.scss" (selected).
- Content:** The "styles.module.scss" file contains SCSS code for a form component. It includes styles for a container, an h1 title, a form, a select input, and a text area.

```
.container{  
  max-width: 720px;  
  margin: 20px auto;  
  padding: 0 16px;  
  display: flex;  
  flex-direction: column;  
  
  h1{  
    color: var(--white);  
  }  
  
  .form{  
    display: flex;  
    flex-direction: column;  
    margin: 16px 0;  
    gap: 16px;  
  
    select {  
      width: 100%;  
      height: 40px;  
      border-radius: 8px;  
      background-color: var(--dark-900);  
      color: var(--white);  
      padding: 0 8px;  
      border: 1px solid var(--gray-100);  
    }  
  
    textarea{  
      width: 100%;  
      min-height: 120px;  
      resize: none;  
    }  
  }  
}
```

**Browser Preview (right):**

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** Categoria, Produto, [→]
- Section:** Novo produto
- Form Fields:** Pizzas, Digite o nome do produto..., Preço do produto... (with a lightning bolt icon).

**UC:** Projetos

**Assunto:**

Projeto Pizzaria – Frontend\_02 (NextJS)

Sistema Pizzaria Web

54 – Cadastrando Produto

The image shows a developer's environment with a code editor and a browser preview side-by-side.

**Code Editor (left):**

- EXPLORER:** Shows the project structure:
  - FRONTEND
  - .next
  - node\_modules
  - public
  - src
    - app
      - dashboard
        - category
        - components
      - product
        - components
          - form
        - index.tsx
        - styles.module.scss
- File List:** index.tsx, page.tsx (selected), styles.module.scss
- Content:** A snippet of TypeScript code for a product page. It imports Form, api, and getCookieServer from their respective modules. It defines an asynchronous function Product that fetches categories from the API using the token obtained from getCookieServer. The response is logged to the console, and a Form component is returned.

```
import { Form } from './components/form'
import { api } from '@/services/api'
import { getCookieServer } from '@/lib/cookieServer'

export default async function Product(){
  const token = await getCookieServer();

  const response = await api.get("/category", {
    headers: {
      Authorization: `Bearer ${token}`
    }
  });

  console.log(response.data)
  return(
    <Form/>
  )
}
```

**Browser Preview (right):**

The browser shows the application "SujeitoPizza" with a title "Novo produto". The page has a dark theme and contains a form for adding a new product. The form fields are:

- Pizzas
- Digite o nome do produto...
- Preço do produto...
- Digite a descrição do produto...

A green button at the bottom right says "Cadastrar produto".

```
next-server (v15.0.3) - X
More info: https://sass-lang.com/d/legacy-js-api

Import trace for requested module:
./src/app/dashboard/components/header/styles.module.scss
./src/app/dashboard/components/header/index.tsx

./src/app/dashboard/product/components/form/styles.module.scss
Deprecation The legacy JS API is deprecated and will be removed in Dart Sass 2.0.0.
More info: https://sass-lang.com/d/legacy-js-api

Import trace for requested module:
./src/app/dashboard/product/components/form/styles.module.scss
./src/app/dashboard/product/components/form/index.tsx
[
  { id: '00d8f79f-5536-429a-bd59-7a82f18700ac', name: 'bebidas' },
  { id: 'e21422b1-f695-4426-aa06-9d23c2f3502b', name: 'pizzas' },
  { id: '80de4305-34d1-418c-a215-455a38a57fa4', name: 'sobremesas' },
  { id: 'df84bb27-a449-4c28-81eb-8c9908d69d12', name: 'espeto' },
  { id: '1e4ffc20-04b2-46db-96f0-ceea3ca2975a', name: 'hamburger' },
  {
    id: '3c1fd0e0-f3ad-458f-98c0-f10798bad20d',
    name: 'acompanhamentos'
  },
  { id: 'f44fb594-d1c4-425f-b1bd-87541a6637a1', name: 'sorvete' }
]
GET /dashboard/product?category=6&name=A%C3%A7ai+300ml&price=12&description=Pote+de+A%C3%A7ai+300ml 200 in 4088ms
  ./src/app/dashboard/components/button/styles.module.scss
Deprecation The legacy JS API is deprecated and will be removed in Dart Sass 2.0.0.
```

The image shows a developer's environment with a code editor and a browser preview side-by-side.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- File List:** Includes files like `index.tsx`, `page.tsx` (highlighted with 1 error), and `styles.module.scss`.
- Content:** The `page.tsx` file contains the following TypeScript code:

```
src > app > dashboard > product > page.tsx > Product
1 import { Form } from './components/form'
2 import { api } from '@/services/api'
3 import { getCookieServer } from '@/lib/cookieServer'
4
5 export default async function Product(){
6     const token = await getCookieServer();
7
8     const response = await api.get("/category", {
9         headers: {
10             Authorization: `Bearer ${token}`
11         }
12     })
13
14     console.log(response.data)
15     return(
16         <Form categories={response.data}/>
17     )
18 }
```

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** SujeitoPizza
- Section:** Novo produto
- Form Fields:**
  - Pizzas: Placeholder "Digite o nome do produto..."
  - Preço do produto: Placeholder "Digite o preço do produto..."
  - Descrição: Placeholder "Digite a descrição do produto..."
- Buttons:** Cadastrar produto

The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- Search:** A search bar at the top.
- File List:** Includes files like index.tsx, page.tsx (1 error), styles.module.scss, layout.tsx, and fonts.
- Editor:** The "page.tsx" file is open, showing TypeScript code for a product creation page. It imports Form, api, and getCookieServer. It uses await to get a token and categories from the API, logs the response, and returns a Form component.
- Bottom Status Bar:** Shows line 13, column 1 (32 selected), spaces: 2, CRLF, TypeScript JSX, Prettier, and a bell icon.

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** SujeitoPizza, Categoria, Produto, and a right arrow.
- Section:** Novo produto
- Form:** A form for adding a new product. It has fields for "Nome do produto" (Digite o nome do produto...), "Preço do produto" (Digite o preço do produto...), and "Descrição do produto" (Digite a descrição do produto...). The "Nome do produto" field is currently active.
- Buttons:** A green "Cadastrar produto" button at the bottom.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- File List:** Includes files like `index.tsx`, `page.tsx` (highlighted with 1 error), and `styles.module.scss`.
- Content Area:** Displays the `page.tsx` file content:

```
src > app > dashboard > product > page.tsx > Product
1 import { Form } from './components/form'
2 import { api } from '@/services/api'
3 import { getCookieServer } from '@/lib/cookieServer'
4
5 export default async function Product(){
6     const token = await getCookieServer();
7
8     const response = await api.get("/category", {
9         headers: {
10             Authorization: `Bearer ${token}`
11         }
12     })
13
14     return(
15         <Form categories={response.data}/>
16     )
17 }
```

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** SujeitoPizza
- Page Content:** Novo produto
- Form Fields:**
  - Pizzas: Digite o nome do produto...
  - Preço do produto...
  - Digite a descrição do produto...
- Buttons:** Cadastrar produto

The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** index.tsx (selected), page.tsx, styles.module.scss.
- Content:** Code for a "Form" component. It includes imports for react, styled-components, and lucide-react. It defines two interfaces: "CategoryProps" and "Props". The "Props" interface has a "categories" field of type "CategoryProps[]". The "Form" component takes "categories" as a prop and handles file uploads, checking for allowed image formats (jpeg/png).

**Browser Preview:**

The browser window displays a dark-themed web application for "Sujeito Pizza".

- Header:** SujeitoPizza - A melhor pizzaria.
- Section:** Novo produto
- Form Fields:** Pizzas, Nome do produto..., Preço do produto..., Descrição do produto...
- Buttons:** Cadastrar produto

The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** index.tsx, page.tsx, styles.module.scss
- Path:** src > app > dashboard > product > components > form > index.tsx
- Content:** TypeScript code for a product creation form. It includes an image preview component and a dropdown menu for categories like Pizzas and Massas.

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** SujeitoPizza, Categoria, Produto, [Entrar]
- Section:** Novo produto
- Form Fields:**
  - Pizzas (selected)
  - Digite o nome do produto...
  - Preço do produto...
  - Digite a descrição do produto...
- Buttons:** Cadastrar produto

The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** index.tsx, page.tsx, styles.module.scss
- Path:** src > app > dashboard > product > components > form > index.tsx
- Content:** TypeScript code for a product form component. It includes an  component with attributes like alt="Imagen de preview", src={previewImage}, and className={styles.preview}. It also includes a  element for categories and an  element for product name.

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** Categoria, Produto, [Entrar]
- Section:** Novo produto
- Form Fields:**
  - Pizzas: dropdown menu
  - Nome do produto: input field placeholder "Digite o nome do produto..."
  - Preço do produto: input field placeholder "Digite o preço do produto..."
  - Descrição do produto: input field placeholder "Digite a descrição do produto..."
- Buttons:** Cadastrar produto (green button)

The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** index.tsx (selected), page.tsx, styles.module.scss.
- Content:** The code is for a product form component. It includes an  tag with attributes like alt="Imagen de preview", src={previewImage}, and className={styles.preview}. It also includes a  tag for categories and an  tag for product name.

```
index.tsx
src > app > dashboard > product > components > form > index.tsx > Form
54      >
55
56
57      {previewImage && (
58          <Image
59              alt="Imagen de preview"
60              src={previewImage}
61              className={styles.preview}
62              fill={true}
63              quality={100}
64              priority={true}
65          />
66      )}
67
68      </label>
69
70      <select name="category">
71          {categories.map( (category, index) =>(
72              <option key={category.id} value={index}>
73                  {category.name}
74              </option>
75          )))
76      </select>
77
78      <input
79          type="text"
80          name="name"
81          placeholder="Digite o nome do produto..."
82          required
83          className={styles.input}
84      />
```

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** SujeitoPizza (Logo), Categoria, Produto, [Entrar]
- Section:** Novo produto
- Form Fields:** Category dropdown (bebidas selected), Product Name input field.
- Buttons:** Cadastrar produto (green button at the bottom right).

The image shows a developer's workspace with two main windows: a code editor and a browser preview.

**Code Editor (Left):**

- Explorer:** Shows the project structure:
  - FRONTEND
  - .next
  - node\_modules
  - public
  - src
    - app
      - dashboard
      - category
      - components
    - product
      - components
      - form
- File List:** index.tsx, page.tsx, styles.module.scss
- Content:** A snippet of TypeScript code for a form component. It includes an  element with attributes like alt="Imagen de preview", src={previewImage}, and className={styles.preview}. It also includes a  element for categories and an  element for product names.

Bottom status bar: Ln 79, Col 23, Spaces: 4, UTF-8, CRLF, TypeScript JSX, Prettier

**Browser Preview (Right):**

The browser window displays a dark-themed application for "Sujeito Pizza". The title bar says "Sujeito Pizza - A melhor pizzaria". The main content area has a heading "Nova Categoria" and a text input field labeled "Nome da categoria, ex: Pizzas". Below the input is a green button labeled "Cadastrar".

The image shows a developer's workspace with two main windows: a code editor and a browser preview.

**Code Editor (Left):**

- Explorer:** Shows the project structure under "FRONTEND".
- File List:** Includes files like index.tsx, page.tsx, styles.module.scss, and various component and utility files.
- Current File:** index.tsx (highlighted).
- Content:** Displays TypeScript code for a form component, including an  component and a  element for categories.

**Browser Preview (Right):**

- Title:** Sujeito Pizza - A melhor pizzaria
- Page Content:** "Nova Categoria" (New Category) page.
- Form Fields:** An input field with placeholder "esfihas".
- Buttons:** A green "Cadastrar" (Register) button.

At the bottom of the browser window, there are navigation icons for back, forward, search, and refresh, along with a "Erro" (Error) button.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** index.tsx, page.tsx, styles.module.scss.
- Path:** src > app > dashboard > product > components > form > index.tsx.
- Content:** TypeScript code for a form component. It includes an  component with props like alt="Imagen de preview", src={previewImage}, and className={styles.preview}. It also includes a  element for categories and an  element for product name.

```
index.tsx
src > app > dashboard > product > components > form > index.tsx
54      >
55
56
57      {previewImage && (
58        <Image
59          alt="Imagen de preview"
60          src={previewImage}
61          className={styles.preview}
62          fill={true}
63          quality={100}
64          priority={true}
65        />
66      )}
67
68    </label>
69
70    <select name="category">
71      {categories.map( (category, index) =>(
72        <option key={category.id} value={index}>
73          {category.name}
74        </option>
75      )))
76    </select>
77
78    <input
79      type="text"
80      name="name"
81      placeholder="Digite o nome do produto..."
82      required
83      className={styles.input}
84    />
85
```

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** Categoria, Produto, [ ]
- Section:** Novo produto
- Form Fields:** Placeholder: "Digite o nome do produto..." (Required).
- Category Sidebar:** Shows a dropdown menu with categories: bebidas, pizzas, sobremesas, espeto, hamburger, acompanhamentos, sorvete, esfihas.
- Buttons:** Cadastrar produto (green button), upload icon.

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C +

Query #1 categories [all] products [all] +

ENTITIES 217 +

public

- \_prisma\_migrations
- categories
- items
- orders
- products
- users

information\_schema

pg\_catalog

Filter Y

id equals Enter Value + 🔍

	id	name	created_at	updated_at
1	00d8f79f-5536-429a-bd59-7a82f18700ac	bebidas	2024-11-21 23:37:54.506	2024-11-21 23:37:54...
2	1e4ffc20-04b2-46db-96f0-ceea3ca2975a	hamburger	2024-11-28 00:42:45.771	2024-11-28 00:42:45...
3	3c1fd0e0-f3ad-458f-98c0-f10798bad20d	acompanhamentos	2024-11-28 00:49:20.371	2024-11-28 00:49:20...
4	80de4305-34d1-418c-a215-455a38a57fa4	sobremesas	2024-11-22 00:08:08.794	2024-11-22 00:08:08...
5	df84bb27-a449-4c28-81eb-8c9908d69d12	espeto	2024-11-28 00:41:32.767	2024-11-28 00:41:32...
6	e21422b1-f695-4426-aa06-9d23c2f3502b	pizzas	2024-11-21 23:39:55.683	2024-11-21 23:39:55...
7	f44fb594-d1c4-425f-b1bd-87541a6637a1	sorvete	2024-11-28 00:36:47.872	2024-11-28 00:36:47...

pizzaria

Query #1 categories [all] products [all]

ENTITIES 217

public

- \_prisma\_migrations
- categories
- items
- orders
- products
- users

information\_schema

pg\_catalog

id text equals Enter Value

	id	name	created_at	updated_at
1	00d8f79f-5536-429a-bd59-7a82f18700ac	bebidas	2024-11-21 23:37:54.506	2024-11-21 23:37:54...
2	1e4ffc20-04b2-46db-96f0-ceea3ca2975a	hamburger	2024-11-28 00:42:45.771	2024-11-28 00:42:45...
3	3c1fd0e0-f3ad-458f-98c0-f10798bad20d	acompanhamentos	2024-11-28 00:49:20.371	2024-11-28 00:49:20...
4	80de4305-34d1-418c-a215-455a38a57fa4	sobremesas	2024-11-22 00:08:08.794	2024-11-22 00:08:08...
5	df84bb27-a449-4c28-81eb-8c9908d69d12	espeto	2024-11-28 00:41:32.767	2024-11-28 00:41:32...
6	e01d585e-820b-42bc-aa94-0cd8d869d600	esfihas	2024-11-29 00:26:07.825	2024-11-29 00:26:07...
7	e21422b1-f695-4426-aa06-9d23c2f3502b	pizzas	2024-11-21 23:39:55.683	2024-11-21 23:39:55...
8	f44fb594-d1c4-425f-b1bd-87541a6637a1	sorvete	2024-11-28 00:36:47.872	2024-11-28 00:36:47...

Refresh Table (ctrl+r or F5)

frontend

index.tsx 1 X page.tsx styles.module.scss

```
src > app > dashboard > product > components > form > index.tsx > Form
  27
  28   console.log(`FORMATO PROIBIDO!!!!`)
  29   return;
  30 }
  31
  32   setImage(image);
  33   setPreviewImage(URL.createObjectURL(image))
  34 }
  35
  36
  37
  38 return(
  39   <main className={styles.container}>
  40     <h1>Novo produto</h1>
  41
  42     <form className={styles.form} action={handleRegisterProduct}>
  43
  44       <label className={styles.labelImage}>
  45         <span>
  46           <UploadCloud size={30} color="#FFF" />
  47         </span>
  48
  49       <input
  50         type="file"
  51         accept="image/png, image/jpeg"
  52         required
  53         onChange={handleFile}
  54       />
  55
  56
  57       {previewImage && (
  58         <Image
  59           alt="Tela com a preview"
```

Ln 42, Col 67 (30 selected) Spaces: 4 CRLF TypeScript JSX Prettier

localhost:3000/dashboard/prod

localhost:300... Error

1 of 1 error

Unhandled Runtime Error

ReferenceError: handleRegisterProduct is not defined

Source

```
src\app\dashboard\product\components\form\index.tsx
(42:45) @ handleRegisterProduct
```

40 | <h1>Novo produto</h1>
41 |
42 | <form className={styles.form} action={handleRegisterProduct}>
43 |
44 | <label className={styles.labelImage}>
45 | <span>

Call Stack

> Next.js

The image shows a developer's workspace with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** index.tsx (selected), page.tsx, styles.module.scss.
- Content:** Code for a "Form" component. It includes an interface "Props" with a "categories" array, useState hooks for image and previewImage, and an async function "handleRegisterProduct" that logs category, name, price, and description. It also includes a file handling function "handleFile" that checks if the file type is either jpeg or png.
- Bottom:** Status bar showing: Ln 22, Col 1 (401 selected) Spaces: 4 UTF-8 CRLF TypeScript JSX ✓ Prettier

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** SujeitoPizza, Categoria, Produto, [→]
- Section:** Novo produto
- Form Fields:** Category (dropdown menu showing "bebidas"), Name (text input placeholder "Digite o nome do produto..."), Price (text input placeholder "Preço do produto..."), Description (text input placeholder "Digite a descrição do produto...").
- Buttons:** Cadastrar produto (green button at the bottom right).

frontend

EXPLORER

FRONTEND

- .next
- node\_modules
- public
- src
  - app
    - dashboard
    - category
    - components
  - product
    - components
    - form
      - index.tsx
      - page.tsx
      - styles.module.scss
- layout.tsx
- page.tsx
- fonts
- signup
- favicon.ico
- globals.scss
- layout.tsx
- page.module.scss
- page.tsx
- lib
- services
- middleware.ts

OUTLINE

TIMELINE

index.tsx page.tsx styles.module.scss

```
src > app > dashboard > product > components > form > index.tsx >
14 interface Props {
15   categories: CategoryProps[]
16 }
17
18 export function Form({ categories }: Props) {
19   const [image, setImage] = useState<File>()
20   const [previewImage, setPreviewImage] = useState<string>('')
21
22   async function handleRegisterProduct(formData: FormData) {
23     const category = formData.get("category")
24     const name = formData.get("name")
25     const price = formData.get("price")
26     const description = formData.get("description")
27
28     if (!name || !category || !price || !description) {
29       return;
30     }
31
32     console.log({
33       category, name, price, description
34     })
35   }
36
37 }
38
39 function handleFile(e: ChangeEvent<HTMLInputElement>){
40   if(e.target.files && e.target.files[0]){
41     const image = e.target.files[0];
42
43     if(image.type !== "image/jpeg" && image.type !== "image/png"){
44       console.log("FORMATO PROIBIDO!!!")
45       return;
46     }
47   }
48 }
```

Ln 22, Col 1 (401 selected) Spaces: 4 UTF-8 CRLF TypeScript JSX Prettier

Abrir

Área de Trabalho > arquivos

Organizar Nova pasta

açaí

Nome: açaí Arquivos personalizados

Digite o nome do produto...

Preço do produto...

Digite a descrição do produto...

Cadastrar produto

The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under `FRONTEND`, including `.next`, `node_modules`, `public`, and `src` folders. `src` contains `app`, `dashboard`, `category`, `components`, `product`, `components`, `form`, `index.tsx`, `styles.module.scss`, `page.tsx`, `layout.tsx`, and `fonts`.
- Search:** A search bar at the top right of the editor window.
- Editor:** The `index.tsx` file is open, showing TypeScript code for a product registration form. It includes imports for `Props` and `CategoryProps`, state management for image, and an asynchronous function `handleRegisterProduct` that logs form data and handles file uploads.
- Bottom Status Bar:** Displays file information (Ln 32, Col 1), code style settings (Spaces: 4, CRLF), and build status (TypeScript JSX, Prettier).

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** Shows the brand name "SujeitoPizza" and navigation links for "Categoria" and "Produto".
- Section:** "Novo produto" (New Product).
- Image:** A bowl of acai bowl topped with granola, banana, and strawberries, with a banana in the background.
- Form Fields:** Input fields for "bebidas" (drinks) containing "Açaí 300ml" and "12".
- Text Area:** A text area labeled "Teste Descrição" (Test Description).
- Buttons:** A green "Cadastrar produto" (Register Product) button.

Sujeito Pizza

## Categoria Produto [→]

## **Novo produto**



bebidas

Açaí 300ml

12

Teste Descricão

- ⚠ The resource [http://localhost:3000/\\_next/static/media/logo.77bc50ee.svg](http://localhost:3000/_next/static/media/logo.77bc50ee.svg) was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate `as` value and it is preloaded intentionally. [product:1](#)
  - ⚠ The resource [http://localhost:3000/\\_next/static/media/logo.77bc50ee.svg](http://localhost:3000/_next/static/media/logo.77bc50ee.svg) was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate `as` value and it is preloaded intentionally. [product:1](#)

▶ Object c:\Users\Dev\Desktop...s\form\index.tsx:33

  - ⚠ The resource [http://localhost:3000/\\_next/static/media/logo.77bc50ee.svg](http://localhost:3000/_next/static/media/logo.77bc50ee.svg) was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate `as` value and it is preloaded intentionally. [product:1](#) 

Sujeito Pizza - A melhor pizzaria

localhost:3000/dashboard/product

Elements Console Sources Network Performance Memory Application Security Lighthouse Recorder > 3 1 E Erro

# SujeitoPizza

Categoria Produto [→]

## Novo produto



bebidas

Açaí 300ml

12

Teste Descrição

Cadastrar produto

The resource [http://localhost:3000/\\_next/static/media/logo.77bc50ee.svg](http://localhost:3000/_next/static/media/logo.77bc50ee.svg) was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally. product:1

The resource [http://localhost:3000/\\_next/static/media/logo.77bc50ee.svg](http://localhost:3000/_next/static/media/logo.77bc50ee.svg) was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally. product:1

Object

- category: "0"
- description: "Teste Descrição"
- name: "Açaí 300ml"
- price: "12"
- [[Prototype]]: Object

The resource [http://localhost:3000/\\_next/static/media/logo.77bc50ee.svg](http://localhost:3000/_next/static/media/logo.77bc50ee.svg) was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally. product:1

frontend

EXPLORER

FRONTEND

- > .next
- > node\_modules
- > public
- > src
  - > app
    - > dashboard
    - > category
    - > components
    - > product
      - > components
      - > form
  - > fonts
  - > signup
  - ★ favicon.ico
  - ✖ globals.scss
  - ✖ layout.tsx
  - ✖ page.module.scss
  - ✖ page.tsx
  - > lib
  - > services
  - TS middleware.ts

OUTLINE

TIMELINE

index.tsx 1 X page.tsx styles.module.scss

```
6 import Image from 'next/image'
7 import { Button } from '@/app/dashboard/components/button'
8
9 interface CategoryProps{
10   id: string;
11   name: string;
12 }
13
14 interface Props {
15   categories: CategoryProps[]
16 }
17
18 export function Form({ categories }: Props ) {
19   const [image, setImage] = useState<File>()
20   const [previewImage, setPreviewImage] = useState<>("")
21
22   async function handleRegisterProduct(formData: FormData) {
23
24     const categoryIndex = formData.get("category")
25     const name = formData.get("name")
26     const price = formData.get("price")
27     const description = formData.get("description")
28
29     if(!name || !categoryIndex || !price || !description){
30       return;
31     }
32
33     console.log({
34       category, name, price, description
35     })
36
37   }
38 }
```

2 selections (26 characters selected) Spaces: 4 UTF-8 CRLF ⓘ TypeScript JSX ✓ Prettier

Sujeito Pizza - A melhor pizzaria

localhost

Categoria Produto

# SujeitoPizza

## Novo produto



bebidas

Açaí 300ml

12

Teste Descrição

Cadastrar produto

The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** index.tsx (1 change), page.tsx, styles.module.scss.
- Content:** Code for a "Form" component. It includes an interface "Props" with a "categories" prop, an export function "Form" that handles state for image selection, and an "async" function "handleRegisterProduct" that logs form data to the console. The "category" field is underlined in red, indicating a potential error or warning.
- Status Bar:** Shows file statistics: Line 33, Col 1 (70 selected), Spaces: 4, UTF-8, CRLF, TypeScript JSX, Prettier.

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** SujeitoPizza, Categoria, Produto, [→]
- Section:** Novo produto
- Image:** Acai bowl with fruit and granola.
- Form Fields:**
  - bebidas
  - Açaí 300ml
  - 12
  - Teste Descrição
- Buttons:** Cadastrar produto

Screenshot showing a development environment with code editor and browser preview.

The code editor (left) displays `index.tsx` from a project structure:

```
index.tsx
page.tsx
styles.module.scss
```

File content:

```
12 }
13
14 interface Props {
15   categories: CategoryProps[]
16 }
17
18 export function Form({ categories }: Props) {
19   const [image, setImage] = useState<File>()
20   const [previewImage, setPreviewImage] = useState("")
21
22   async function handleRegisterProduct(formData: FormData) {
23
24     const categoryIndex = formData.get("category")
25     const name = formData.get("name")
26     const price = formData.get("price")
27     const description = formData.get("description")
28
29     if(!name || !categoryIndex || !price || !description){
30       return;
31     }
32
33     console.log(categories[Number(categoryIndex)])
34
35   }
36
37   function handleFile(e: ChangeEvent<HTMLInputElement>){
38     if(e.target.files && e.target.files[0]){
39       const image = e.target.files[0];
40
41       if(image.type !== "image/jpeg" && image.type !== "image/png"){
42         console.log("FORMATO PROIBIDO!!!")
43         return;
44       }
45     }
46   }
47 }
```

The browser preview (right) shows a product registration form for "Sujeito Pizza".

Page Title: Sujeito Pizza - A melhor pizzaria

Header: Categoria Produto

Section: Novo produto

Image: Acai bowl with banana, granola, and strawberry.

Form Fields:

- bebidas
- Açaí 300ml
- 12
- Teste Descrição

Buttons:

- Cadastrar produto

The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** index.tsx (selected), page.tsx, styles.module.scss.
- Content:** Code for a "Form" component. It includes an interface "Props" with a "categories" array, an export function "Form" that handles state for image and preview image, and an asynchronous function "handleRegisterProduct" that logs the selected category index and handles file uploads. It also includes validation logic for name, category, price, and description.
- Bottom Status Bar:** Ln 33, Col 1 (50 selected) | Spaces: 4 | UTF-8 | CRLF | TypeScript JSX | Prettier.

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localho... (highlighted)
- Page Content:** "SujeitoPizza" logo, "Novo produto" heading, and a form for product registration. The form fields include a dropdown menu set to "bebidas", a text input placeholder "Digite o nome do produto...", a text input placeholder "Preço do produto...", and a text input placeholder "Digite a descrição do produto...".
- Buttons:** "Cadastrar produto" button at the bottom right.

frontend

index.tsx page.tsx styles.module.scss

FRONTEND .next node\_modules public src app dashboard category components product components form index.tsx styles.module.scss page.tsx layout.tsx page.tsx fonts signup favicon.ico globals.scss layout.tsx page.module.scss page.tsx lib services middleware.ts OUTLINE TIMELINE

12 }  
13  
14 interface Props {  
15 categories: CategoryProps[]  
16 }  
17  
18 export function Form({ categories }: Props) {  
19 const [image, setImage] = useState<File>()  
20 const [previewImage, setPreviewImage] = useState<string>("")  
21  
22 async function handleRegisterProduct(formData: FormData) {  
23 const categoryIndex = formData.get("category")  
24 const name = formData.get("name")  
25 const price = formData.get("price")  
26 const description = formData.get("description")  
27  
28 if (!name || !categoryIndex || !price || !description)  
29 return;  
30  
31 }  
32  
33 console.log(categories[Number(categoryIndex)])  
34  
35 }  
36  
37 function handleFile(e: ChangeEvent<HTMLInputElement>){  
38 if(e.target.files && e.target.files[0]){  
39 const image = e.target.files[0];  
40 const image: File  
41  
42 if(image.type !== "image/jpeg" && image.type !== "image/png"){  
43 console.log("FORMATO PROIBIDO!!!")  
44 return;  
45 }  
46 }  
47 }  
48  
49 const [name, setName] = useState("")  
50 const [category, setCategory] = useState("")  
51 const [price, setPrice] = useState("")  
52 const [description, setDescription] = useState("")  
53  
54 const [image, setImage] = useState<File>()  
55  
56 const [previewImage, setPreviewImage] = useState<string>("")  
57  
58 const [categoryIndex, setCategoryIndex] = useState(0)  
59  
60 const [loading, setLoading] = useState(false)  
61  
62 const [error, setError] = useState("")  
63  
64 const [success, setSuccess] = useState("")  
65  
66 const [categories, setCategories] = useState<CategoryProps[]>([  
67 {  
68 id: 1,  
69 name: "Açaí",  
70 image: "açaí.jpg"  
71 },  
72 {  
73 id: 2,  
74 name: "Sorvete",  
75 image: "sorvete1.jpg"  
76 }  
77 ])  
78  
79 const [form, setForm] = useState({  
80 name: "",  
81 category: "",  
82 price: "",  
83 description: ""  
84 })  
85  
86 const [image, setImage] = useState<File>()  
87  
88 const [previewImage, setPreviewImage] = useState<string>("")  
89  
90 const [categoryIndex, setCategoryIndex] = useState(0)  
91  
92 const [loading, setLoading] = useState(false)  
93  
94 const [error, setError] = useState("")  
95  
96 const [success, setSuccess] = useState("")  
97  
98 const [categories, setCategories] = useState<CategoryProps[]>([  
99 {  
100 id: 1,  
101 name: "Açaí",  
102 image: "açaí.jpg"  
103 },  
104 {  
105 id: 2,  
106 name: "Sorvete",  
107 image: "sorvete1.jpg"  
108 }  
109 ])  
110  
111 const handleRegisterProduct = async () => {  
112 try {  
113 const response = await fetch(`http://localhost:3001/products`, {  
114 method: "POST",  
115 headers: {  
116 "Content-Type": "application/json"  
117 },  
118 body: JSON.stringify(form)  
119 })  
120 const data = await response.json()  
121 if (data.error) {  
122 setError(data.error)  
123 } else {  
124 setSuccess("Produto cadastrado com sucesso!")  
125 }  
126 } catch (error) {  
127 setError("Erro ao cadastrar produto")  
128 }  
129 }  
130  
131 const handleImageChange = (e) => {  
132 const file = e.target.files[0]  
133 if (file) {  
134 setImage(file)  
135 const reader = new FileReader()  
136 reader.onload = () => {  
137 setPreviewImage(reader.result)  
138 }  
139 reader.readAsDataURL(file)  
140 }  
141 }  
142  
143 const handleCategoryChange = (e) => {  
144 const category = e.target.value  
145 setCategoryIndex(Number(category))  
146 }  
147  
148 const handleNameChange = (e) => {  
149 const name = e.target.value  
150 setName(name)  
151 }  
152  
153 const handlePriceChange = (e) => {  
154 const price = e.target.value  
155 setPrice(price)  
156 }  
157  
158 const handleDescriptionChange = (e) => {  
159 const description = e.target.value  
160 setDescription(description)  
161 }  
162  
163 const handleCategorySelect = (e) => {  
164 const category = e.target.value  
165 setCategoryIndex(Number(category))  
166 }  
167  
168 const handleImageUpload = () => {  
169 handleImageChange()  
170 }  
171  
172 const handleRegisterProduct = () => {  
173 handleRegisterProduct()  
174 }  
175  
176 const handleSuccess = () => {  
177 setSuccess("")  
178 }  
179  
180 const handleImageError = () => {  
181 setError("")  
182 }  
183  
184 const handleImagePreview = () => {  
185 setPreviewImage("")  
186 }  
187  
188 const handleImageReset = () => {  
189 setImage(null)  
190 }  
191  
192 const handleImageSelect = () => {  
193 handleImageSelect()  
194 }  
195  
196 const handleImageDelete = () => {  
197 handleImageDelete()  
198 }  
199  
200 const handleImagePreview = () => {  
201 setPreviewImage("")  
202 }  
203  
204 const handleImageReset = () => {  
205 setImage(null)  
206 }  
207  
208 const handleImageSelect = () => {  
209 handleImageSelect()  
210 }  
211  
212 const handleImageDelete = () => {  
213 handleImageDelete()  
214 }  
215  
216 const handleImagePreview = () => {  
217 setPreviewImage("")  
218 }  
219  
220 const handleImageReset = () => {  
221 setImage(null)  
222 }  
223  
224 const handleImageSelect = () => {  
225 handleImageSelect()  
226 }  
227  
228 const handleImageDelete = () => {  
229 handleImageDelete()  
230 }  
231  
232 const handleImagePreview = () => {  
233 setPreviewImage("")  
234 }  
235  
236 const handleImageReset = () => {  
237 setImage(null)  
238 }  
239  
240 const handleImageSelect = () => {  
241 handleImageSelect()  
242 }  
243  
244 const handleImageDelete = () => {  
245 handleImageDelete()  
246 }  
247  
248 const handleImagePreview = () => {  
249 setPreviewImage("")  
250 }  
251  
252 const handleImageReset = () => {  
253 setImage(null)  
254 }  
255  
256 const handleImageSelect = () => {  
257 handleImageSelect()  
258 }  
259  
260 const handleImageDelete = () => {  
261 handleImageDelete()  
262 }  
263  
264 const handleImagePreview = () => {  
265 setPreviewImage("")  
266 }  
267  
268 const handleImageReset = () => {  
269 setImage(null)  
270 }  
271  
272 const handleImageSelect = () => {  
273 handleImageSelect()  
274 }  
275  
276 const handleImageDelete = () => {  
277 handleImageDelete()  
278 }  
279  
280 const handleImagePreview = () => {  
281 setPreviewImage("")  
282 }  
283  
284 const handleImageReset = () => {  
285 setImage(null)  
286 }  
287  
288 const handleImageSelect = () => {  
289 handleImageSelect()  
290 }  
291  
292 const handleImageDelete = () => {  
293 handleImageDelete()  
294 }  
295  
296 const handleImagePreview = () => {  
297 setPreviewImage("")  
298 }  
299  
300 const handleImageReset = () => {  
301 setImage(null)  
302 }  
303  
304 const handleImageSelect = () => {  
305 handleImageSelect()  
306 }  
307  
308 const handleImageDelete = () => {  
309 handleImageDelete()  
310 }  
311  
312 const handleImagePreview = () => {  
313 setPreviewImage("")  
314 }  
315  
316 const handleImageReset = () => {  
317 setImage(null)  
318 }  
319  
320 const handleImageSelect = () => {  
321 handleImageSelect()  
322 }  
323  
324 const handleImageDelete = () => {  
325 handleImageDelete()  
326 }  
327  
328 const handleImagePreview = () => {  
329 setPreviewImage("")  
330 }  
331  
332 const handleImageReset = () => {  
333 setImage(null)  
334 }  
335  
336 const handleImageSelect = () => {  
337 handleImageSelect()  
338 }  
339  
340 const handleImageDelete = () => {  
341 handleImageDelete()  
342 }  
343  
344 const handleImagePreview = () => {  
345 setPreviewImage("")  
346 }  
347  
348 const handleImageReset = () => {  
349 setImage(null)  
350 }  
351  
352 const handleImageSelect = () => {  
353 handleImageSelect()  
354 }  
355  
356 const handleImageDelete = () => {  
357 handleImageDelete()  
358 }  
359  
360 const handleImagePreview = () => {  
361 setPreviewImage("")  
362 }  
363  
364 const handleImageReset = () => {  
365 setImage(null)  
366 }  
367  
368 const handleImageSelect = () => {  
369 handleImageSelect()  
370 }  
371  
372 const handleImageDelete = () => {  
373 handleImageDelete()  
374 }  
375  
376 const handleImagePreview = () => {  
377 setPreviewImage("")  
378 }  
379  
380 const handleImageReset = () => {  
381 setImage(null)  
382 }  
383  
384 const handleImageSelect = () => {  
385 handleImageSelect()  
386 }  
387  
388 const handleImageDelete = () => {  
389 handleImageDelete()  
390 }  
391  
392 const handleImagePreview = () => {  
393 setPreviewImage("")  
394 }  
395  
396 const handleImageReset = () => {  
397 setImage(null)  
398 }  
399  
400 const handleImageSelect = () => {  
401 handleImageSelect()  
402 }  
403  
404 const handleImageDelete = () => {  
405 handleImageDelete()  
406 }  
407  
408 const handleImagePreview = () => {  
409 setPreviewImage("")  
410 }  
411  
412 const handleImageReset = () => {  
413 setImage(null)  
414 }  
415  
416 const handleImageSelect = () => {  
417 handleImageSelect()  
418 }  
419  
420 const handleImageDelete = () => {  
421 handleImageDelete()  
422 }  
423  
424 const handleImagePreview = () => {  
425 setPreviewImage("")  
426 }  
427  
428 const handleImageReset = () => {  
429 setImage(null)  
430 }  
431  
432 const handleImageSelect = () => {  
433 handleImageSelect()  
434 }  
435  
436 const handleImageDelete = () => {  
437 handleImageDelete()  
438 }  
439  
440 const handleImagePreview = () => {  
441 setPreviewImage("")  
442 }  
443  
444 const handleImageReset = () => {  
445 setImage(null)  
446 }  
447  
448 const handleImageSelect = () => {  
449 handleImageSelect()  
450 }  
451  
452 const handleImageDelete = () => {  
453 handleImageDelete()  
454 }  
455  
456 const handleImagePreview = () => {  
457 setPreviewImage("")  
458 }  
459  
460 const handleImageReset = () => {  
461 setImage(null)  
462 }  
463  
464 const handleImageSelect = () => {  
465 handleImageSelect()  
466 }  
467  
468 const handleImageDelete = () => {  
469 handleImageDelete()  
470 }  
471  
472 const handleImagePreview = () => {  
473 setPreviewImage("")  
474 }  
475  
476 const handleImageReset = () => {  
477 setImage(null)  
478 }  
479  
480 const handleImageSelect = () => {  
481 handleImageSelect()  
482 }  
483  
484 const handleImageDelete = () => {  
485 handleImageDelete()  
486 }  
487  
488 const handleImagePreview = () => {  
489 setPreviewImage("")  
490 }  
491  
492 const handleImageReset = () => {  
493 setImage(null)  
494 }  
495  
496 const handleImageSelect = () => {  
497 handleImageSelect()  
498 }  
499  
500 const handleImageDelete = () => {  
501 handleImageDelete()  
502 }  
503  
504 const handleImagePreview = () => {  
505 setPreviewImage("")  
506 }  
507  
508 const handleImageReset = () => {  
509 setImage(null)  
510 }  
511  
512 const handleImageSelect = () => {  
513 handleImageSelect()  
514 }  
515  
516 const handleImageDelete = () => {  
517 handleImageDelete()  
518 }  
519  
520 const handleImagePreview = () => {  
521 setPreviewImage("")  
522 }  
523  
524 const handleImageReset = () => {  
525 setImage(null)  
526 }  
527  
528 const handleImageSelect = () => {  
529 handleImageSelect()  
530 }  
531  
532 const handleImageDelete = () => {  
533 handleImageDelete()  
534 }  
535  
536 const handleImagePreview = () => {  
537 setPreviewImage("")  
538 }  
539  
540 const handleImageReset = () => {  
541 setImage(null)  
542 }  
543  
544 const handleImageSelect = () => {  
545 handleImageSelect()  
546 }  
547  
548 const handleImageDelete = () => {  
549 handleImageDelete()  
550 }  
551  
552 const handleImagePreview = () => {  
553 setPreviewImage("")  
554 }  
555  
556 const handleImageReset = () => {  
557 setImage(null)  
558 }  
559  
560 const handleImageSelect = () => {  
561 handleImageSelect()  
562 }  
563  
564 const handleImageDelete = () => {  
565 handleImageDelete()  
566 }  
567  
568 const handleImagePreview = () => {  
569 setPreviewImage("")  
570 }  
571  
572 const handleImageReset = () => {  
573 setImage(null)  
574 }  
575  
576 const handleImageSelect = () => {  
577 handleImageSelect()  
578 }  
579  
580 const handleImageDelete = () => {  
581 handleImageDelete()  
582 }  
583  
584 const handleImagePreview = () => {  
585 setPreviewImage("")  
586 }  
587  
588 const handleImageReset = () => {  
589 setImage(null)  
590 }  
591  
592 const handleImageSelect = () => {  
593 handleImageSelect()  
594 }  
595  
596 const handleImageDelete = () => {  
597 handleImageDelete()  
598 }  
599  
600 const handleImagePreview = () => {  
601 setPreviewImage("")  
602 }  
603  
604 const handleImageReset = () => {  
605 setImage(null)  
606 }  
607  
608 const handleImageSelect = () => {  
609 handleImageSelect()  
610 }  
611  
612 const handleImageDelete = () => {  
613 handleImageDelete()  
614 }  
615  
616 const handleImagePreview = () => {  
617 setPreviewImage("")  
618 }  
619  
620 const handleImageReset = () => {  
621 setImage(null)  
622 }  
623  
624 const handleImageSelect = () => {  
625 handleImageSelect()  
626 }  
627  
628 const handleImageDelete = () => {  
629 handleImageDelete()  
630 }  
631  
632 const handleImagePreview = () => {  
633 setPreviewImage("")  
634 }  
635  
636 const handleImageReset = () => {  
637 setImage(null)  
638 }  
639  
640 const handleImageSelect = () => {  
641 handleImageSelect()  
642 }  
643  
644 const handleImageDelete = () => {  
645 handleImageDelete()  
646 }  
647  
648 const handleImagePreview = () => {  
649 setPreviewImage("")  
650 }  
651  
652 const handleImageReset = () => {  
653 setImage(null)  
654 }  
655  
656 const handleImageSelect = () => {  
657 handleImageSelect()  
658 }  
659  
660 const handleImageDelete = () => {  
661 handleImageDelete()  
662 }  
663  
664 const handleImagePreview = () => {  
665 setPreviewImage("")  
666 }  
667  
668 const handleImageReset = () => {  
669 setImage(null)  
670 }  
671  
672 const handleImageSelect = () => {  
673 handleImageSelect()  
674 }  
675  
676 const handleImageDelete = () => {  
677 handleImageDelete()  
678 }  
679  
680 const handleImagePreview = () => {  
681 setPreviewImage("")  
682 }  
683  
684 const handleImageReset = () => {  
685 setImage(null)  
686 }  
687  
688 const handleImageSelect = () => {  
689 handleImageSelect()  
690 }  
691  
692 const handleImageDelete = () => {  
693 handleImageDelete()  
694 }  
695  
696 const handleImagePreview = () => {  
697 setPreviewImage("")  
698 }  
699  
700 const handleImageReset = () => {  
701 setImage(null)  
702 }  
703  
704 const handleImageSelect = () => {  
705 handleImageSelect()  
706 }  
707  
708 const handleImageDelete = () => {  
709 handleImageDelete()  
710 }  
711  
712 const handleImagePreview = () => {  
713 setPreviewImage("")  
714 }  
715  
716 const handleImageReset = () => {  
717 setImage(null)  
718 }  
719  
720 const handleImageSelect = () => {  
721 handleImageSelect()  
722 }  
723  
724 const handleImageDelete = () => {  
725 handleImageDelete()  
726 }  
727  
728 const handleImagePreview = () => {  
729 setPreviewImage("")  
730 }  
731  
732 const handleImageReset = () => {  
733 setImage(null)  
734 }  
735  
736 const handleImageSelect = () => {  
737 handleImageSelect()  
738 }  
739  
740 const handleImageDelete = () => {  
741 handleImageDelete()  
742 }  
743  
744 const handleImagePreview = () => {  
745 setPreviewImage("")  
746 }  
747  
748 const handleImageReset = () => {  
749 setImage(null)  
750 }  
751  
752 const handleImageSelect = () => {  
753 handleImageSelect()  
754 }  
755  
756 const handleImageDelete = () => {  
757 handleImageDelete()  
758 }  
759  
760 const handleImagePreview = () => {  
761 setPreviewImage("")  
762 }  
763  
764 const handleImageReset = () => {  
765 setImage(null)  
766 }  
767  
768 const handleImageSelect = () => {  
769 handleImageSelect()  
770 }  
771  
772 const handleImageDelete = () => {  
773 handleImageDelete()  
774 }  
775  
776 const handleImagePreview = () => {  
777 setPreviewImage("")  
778 }  
779  
780 const handleImageReset = () => {  
781 setImage(null)  
782 }  
783  
784 const handleImageSelect = () => {  
785 handleImageSelect()  
786 }  
787  
788 const handleImageDelete = () => {  
789 handleImageDelete()  
790 }  
791  
792 const handleImagePreview = () => {  
793 setPreviewImage("")  
794 }  
795  
796 const handleImageReset = () => {  
797 setImage(null)  
798 }  
799  
800 const handleImageSelect = () => {  
801 handleImageSelect()  
802 }  
803  
804 const handleImageDelete = () => {  
805 handleImageDelete()  
806 }  
807  
808 const handleImagePreview = () => {  
809 setPreviewImage("")  
810 }  
811  
812 const handleImageReset = () => {  
813 setImage(null)  
814 }  
815  
816 const handleImageSelect = () => {  
817 handleImageSelect()  
818 }  
819  
820 const handleImageDelete = () => {  
821 handleImageDelete()  
822 }  
823  
824 const handleImagePreview = () => {  
825 setPreviewImage("")  
826 }  
827  
828 const handleImageReset = () => {  
829 setImage(null)  
830 }  
831  
832 const handleImageSelect = () => {  
833 handleImageSelect()  
834 }  
835  
836 const handleImageDelete = () => {  
837 handleImageDelete()  
838 }  
839  
840 const handleImagePreview = () => {  
841 setPreviewImage("")  
842 }  
843  
844 const handleImageReset = () => {  
845 setImage(null)  
846 }  
847  
848 const handleImageSelect = () => {  
849 handleImageSelect()  
850 }  
851  
852 const handleImageDelete = () => {  
853 handleImageDelete()  
854 }  
855  
856 const handleImagePreview = () => {  
857 setPreviewImage("")  
858 }  
859  
860 const handleImageReset = () => {  
861 setImage(null)  
862 }  
863  
864 const handleImageSelect = () => {  
865 handleImageSelect()  
866 }  
867  
868 const handleImageDelete = () => {  
869 handleImageDelete()  
870 }  
871  
872 const handleImagePreview = () => {  
873 setPreviewImage("")  
874 }  
875  
876 const handleImageReset = () => {  
877 setImage(null)  
878 }  
879  
880 const handleImageSelect = () => {  
881 handleImageSelect()  
882 }  
883  
884 const handleImageDelete = () => {  
885 handleImageDelete()  
886 }  
887  
888 const handleImagePreview = () => {  
889 setPreviewImage("")  
890 }  
891  
892 const handleImageReset = () => {  
893 setImage(null)  
894 }  
895  
896 const handleImageSelect = () => {  
897 handleImageSelect()  
898 }  
899  
900 const handleImageDelete = () => {  
901 handleImageDelete()  
902 }  
903  
904 const handleImagePreview = () => {  
905 setPreviewImage("")  
906 }  
907  
908 const handleImageReset = () => {  
909 setImage(null)  
910 }  
911  
912 const handleImageSelect = () => {  
913 handleImageSelect()  
914 }  
915  
916 const handleImageDelete = () => {  
917 handleImageDelete()  
918 }  
919  
920 const handleImagePreview = () => {  
921 setPreviewImage("")  
922 }  
923  
924 const handleImageReset = () => {  
925 setImage(null)  
926 }  
927  
928 const handleImageSelect = () => {  
929 handleImageSelect()  
930 }  
931  
932 const handleImageDelete = () => {  
933 handleImageDelete()  
934 }  
935  
936 const handleImagePreview = () => {  
937 setPreviewImage("")  
938 }  
939  
940 const handleImageReset = () => {  
941 setImage(null)  
942 }  
943  
944 const handleImageSelect = () => {  
945 handleImageSelect()  
946 }  
947  
948 const handleImageDelete = () => {  
949 handleImageDelete()  
950 }  
951  
952 const handleImagePreview = () => {  
953 setPreviewImage("")  
954 }  
955  
956 const handleImageReset = () => {  
957 setImage(null)  
958 }  
959  
960 const handleImageSelect = () => {  
961 handleImageSelect()  
962 }  
963  
964 const handleImageDelete = () => {  
965 handleImageDelete()  
966 }  
967  
968 const handleImagePreview = () => {  
969 setPreviewImage("")  
970 }  
971  
972 const handleImageReset = () => {  
973 setImage(null)  
974 }  
975  
976 const handleImageSelect = () => {  
977 handleImageSelect()  
978 }  
979  
980 const handleImageDelete = () => {  
981 handleImageDelete()  
982 }  
983  
984 const handleImagePreview = () => {  
985 setPreviewImage("")  
986 }  
987  
988 const handleImageReset = () => {  
989 setImage(null)  
990 }  
991  
992 const handleImageSelect = () => {  
993 handleImageSelect()  
994 }  
995  
996 const handleImageDelete = () => {  
997 handleImageDelete()  
998 }  
999  
1000 const handleImagePreview = () => {  
1001 setPreviewImage("")  
1002 }  
1003  
1004 const handleImageReset = () => {  
1005 setImage(null)  
1006 }  
1007  
1008 const handleImageSelect = () => {  
1009 handleImageSelect()  
1010 }  
1011  
1012 const handleImageDelete = () => {  
1013 handleImageDelete()  
1014 }  
1015  
1016 const handleImagePreview = () => {  
1017 setPreviewImage("")  
1018 }  
1019  
1020 const handleImageReset = () => {  
1021 setImage(null)  
1022 }  
1023  
1024 const handleImageSelect = () => {  
1025 handleImageSelect()  
1026 }  
1027  
1028 const handleImageDelete = () => {  
1029 handleImageDelete()  
1030 }  
1031  
1032 const handleImagePreview = () => {  
1033 setPreviewImage("")  
1034 }  
1035  
1036 const handleImageReset = () => {  
1037 setImage(null)  
1038 }  
1039  
1040 const handleImageSelect = () => {  
1041 handleImageSelect()  
1042 }  
1043  
1044 const handleImageDelete = () => {  
1045 handleImageDelete()  
1046 }  
1047  
1048 const handleImagePreview = () => {  
1049 setPreviewImage("")  
1050 }

The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** index.tsx (selected), page.tsx, styles.module.scss.
- Content:** Code for a "Form" component. It includes an interface "Props" with a "categories" array, a function "handleRegisterProduct" that logs the category index, and a "handleFile" function that checks if the file type is allowed (image/jpeg or image/png).

```
index.tsx
interface Props {
    categories: CategoryProps[]
}

export function Form({ categories }: Props) {
    const [image, setImage] = useState<File>()
    const [previewImage, setPreviewImage] = useState("")

    async function handleRegisterProduct(formData: FormData) {
        const categoryIndex = formData.get("category")
        const name = formData.get("name")
        const price = formData.get("price")
        const description = formData.get("description")

        if(!name || !categoryIndex || !price || !description){
            return;
        }

        console.log(categories[Number(categoryIndex)])
    }

    function handleFile(e: ChangeEvent<HTMLInputElement>){
        if(e.target.files && e.target.files[0]){
            const image = e.target.files[0];

            if(image.type !== "image/jpeg" && image.type !== "image/png"){
                console.log("FORMATO PROIBIDO!!!")
                return;
            }
        }
    }
}
```

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** SujeitoPizza, Categoria, Produto, [Entrar]
- Section:** Novo produto
- Image:** A close-up image of a pizza slice with toppings.
- Form Fields:** Name (sorvete), Price (sfdsaf), Description (234), Category (dadasa).
- Buttons:** Cadastrar produto (green button at the bottom right).

A screenshot of a web browser window. The address bar shows the URL "Sujeito Pizza - A melhor pizzaria". The page content includes a logo for "Sujeito Pizza", a navigation bar with links for "Categoria", "Produto", and "[→]", and a large image of a pizza slice. Below the image is a form with dropdown menus for "bebidas" and "Categorias", and input fields for "Nome do produto...", "Preço do produto...", and "Descrição do produto...". At the bottom is a green button labeled "Cadastrar produto".

Sujeito Pizza - A melhor pizzaria

localhost:3000/dashboard/product

Elements Console Sources Network Performance Memory Application Security Lighthouse Recorder ▶ 3 1 ⚙ X

Novo produto



bebidas

Digite o nome do produto...

Preço do produto...

Digite a descrição do produto...

Cadastrar produto

Console

top ▾ Filter

Default levels ▾ 1 Issue: 1 ⚙

- The resource [http://localhost:3000/\\_next/static/media/logo.77bc50ee.svg](http://localhost:3000/_next/static/media/logo.77bc50ee.svg) was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally. product:1
- The resource [http://localhost:3000/\\_next/static/media/logo.77bc50ee.svg](http://localhost:3000/_next/static/media/logo.77bc50ee.svg) was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally. product:1
- ▶ Object c:\Users\Dev\Desktop...s\form\index.tsx:33
- The resource [http://localhost:3000/\\_next/static/media/logo.77bc50ee.svg](http://localhost:3000/_next/static/media/logo.77bc50ee.svg) was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally. product:1

Console AI assistance

Sujeito Pizza - A melhor pizzaria

localhost:3000/dashboard/product

Elements Console Sources Network Performance Memory Application Security Lighthouse Recorder » 3 1 E : X

top Filter Default levels 1 Issue: 1

The resource [http://localhost:3000/\\_next/static/media/logo.77bc50ee.svg](http://localhost:3000/_next/static/media/logo.77bc50ee.svg) was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally. product:1

The resource [http://localhost:3000/\\_next/static/media/logo.77bc50ee.svg](http://localhost:3000/_next/static/media/logo.77bc50ee.svg) was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally. product:1

Object 1  
id: "f44fb594-d1c4-425f-b1bd-87541a6637a1"  
name: "sorvete"  
[[Prototype]]: Object

c:\Users\Dev\Desktop\s\form\index.tsx:33

The resource [http://localhost:3000/\\_next/static/media/logo.77bc50ee.svg](http://localhost:3000/_next/static/media/logo.77bc50ee.svg) was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally. product:1

>

# SujeitoPizza

## Categoria Produto [→]

### Novo produto



bebidas

Digite o nome do produto...

Preço do produto...

Digite a descrição do produto...

Cadastrar produto

The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** index.tsx (selected), page.tsx, styles.module.scss.
- Content:** Code for a "Form" component. It includes an interface "Props" with a "categories" array, an export function "Form" that handles state for image selection, and an "async" function "handleRegisterProduct" that logs the selected category index and checks file types before returning.

```
index.tsx
interface Props {
    categories: CategoryProps[]
}

export function Form({ categories }: Props) {
    const [image, setImage] = useState<File>()
    const [previewImage, setPreviewImage] = useState("")

    async function handleRegisterProduct(formData: FormData) {
        const categoryIndex = formData.get("category")
        const name = formData.get("name")
        const price = formData.get("price")
        const description = formData.get("description")

        if(!name || !categoryIndex || !price || !description){
            return;
        }

        console.log(categories[Number(categoryIndex)])
    }

    function handleFile(e: ChangeEvent<HTMLInputElement>){
        if(e.target.files && e.target.files[0]){
            const image = e.target.files[0];

            if(image.type !== "image/jpeg" && image.type !== "image/png"){
                console.log("FORMATO PROIBIDO!!!")
                return;
            }
        }
    }
}
```

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** SujeitoPizza, Categoria, Produto, [arrow].
- Section:** Novo produto
- Image:** A close-up photo of a pizza slice with toppings.
- Form Fields:**bebidas, Digite o nome do produto..., Preço do produto..., Digite a descrição do produto...
- Button:** Cadastrar produto

Sujeito Pizza - A melhor pizzaria

frontend

EXPLORER

FRONTEND

- .next
- node\_modules
- public
- src
  - app
    - dashboard
      - category
      - components
    - product
      - components
        - form
      - index.tsx
      - page.module.scss
      - page.tsx
  - fonts
  - signup
  - favicon.ico
  - globals.scss
  - layout.tsx
  - page.module.scss
  - page.tsx
  - lib
  - services
  - middleware.ts
- OUTLINE
- TIMELINE

index.tsx

page.tsx

styles.module.scss

12 }

13

14 interface Props {

15     categories: CategoryProps[]

16 }

17

18 export function Form({ categories }: Props) {

19     const [image, setImage] = useState<File>()

20     const [previewImage, setPreviewImage] = useState("")

21

22     async function handleRegisterProduct(formData: FormData) {

23

24         const categoryIndex = formData.get("category")

25         const name = formData.get("name")

26         const price = formData.get("price")

27         const description = formData.get("description")

28

29         if (!name || !categoryIndex || !price || !description || !image) {

30             return;

31         }

32

33         const data = new FormData();

34

35         data.append("name", name)

36         data.append("price", price)

37         data.append("description", description)

38         data.append("category\_id", categories[Number(categoryIndex)].id)

39         data.append("file", image)

40 }

41

42

43     function handleFile(e: ChangeEvent<HTMLInputElement>){

2 selections (256 characters selected) Spaces: 4 CRLF ↵ TypeScript JSX ✓ Prettier

# SujeitoPizza

## Novo produto



bebidas

Digite o nome do produto...

Preço do produto...

Digite a descrição do produto...

Cadastrar produto

The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** index.tsx (selected), page.tsx, styles.module.scss.
- Content:** Code for a Next.js application component. It includes imports for react, styles, UploadCloud, Image, Button, and api. It defines interfaces CategoryProps and Props, and an export function Form. The Form function handles file selection, state management for preview image, and an asynchronous handleRegisterProduct function that validates form data and returns if any fields are missing.

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** SujeitoPizza, Categoria, Produto, [Entrar]
- Section:** Novo produto
- Image:** A close-up image of a pizza slice with toppings.
- Form Fields:**bebidas, Digite o nome do produto..., Preço do produto..., Digite a descrição do produto...
- Button:** Cadastrar produto

Sujeito Pizza - A melhor pizzaria

frontend

EXPLORER

FRONTEND

- .next
- node\_modules
- public
- src
  - app
    - dashboard
      - category
      - components
    - product
      - components
        - form
      - index.tsx
      - page.tsx
      - styles.module.scss
  - fonts
  - signup
  - favicon.ico
  - globals.scss
  - layout.tsx
  - page.module.scss
  - page.tsx
  - lib
  - services
- middleware.ts

OUTLINE

TIMELINE

index.tsx

page.tsx

styles.module.scss

```
  export function Form({ categories }: Props) {
    const [image, setImage] = useState<File>()
    const [previewImage, setPreviewImage] = useState("")

    async function handleRegisterProduct(formData: FormData) {
        const categoryIndex = formData.get("category")
        const name = formData.get("name")
        const price = formData.get("price")
        const description = formData.get("description")

        if(!name || !categoryIndex || !price || !description || !image){
            return;
        }

        const data = new FormData()

        data.append("name", name)
        data.append("price", price)
        data.append("description", description)
        data.append("category_id", categories[Number(categoryIndex)].id)
        data.append("file", image)

        await api.post("/product", data, {
            headers: {
                Authorization: `Bearer ${localStorage.getItem("token")}`
            }
        })
    }

    function handleFile(e: ChangeEvent<HTMLInputElement>){
        if(e.target.files && e.target.files[0]){
            const reader = new FileReader()
            reader.onload = () => setPreviewImage(reader.result)
            reader.readAsDataURL(e.target.files[0])
        }
    }
}
```

Ln 42, Col 1 (113 selected) Spaces: 4 UTF-8 CRLF ↵ TypeScript JSX ✓ Prettier

SujeitoPizza

Categoria Produto

Novo produto



bebidas

Digite o nome do produto...

Preço do produto...

Digite a descrição do produto...

Cadastrar produto

The screenshot shows a VS Code interface with the following details:

- Explorer View:** Shows the project structure under "FRONTEND". The "src" folder contains "app", "category", "components", "fonts", "layout", "page", "services", and "lib". "app" contains "dashboard", "category", "components", and "product". "product" contains "components" and "form". "form" contains "index.tsx" (selected), "page.tsx", "layout.tsx", "page.tsx", "fonts", "signup", "favicon.ico", "globals.scss", "layout.tsx", "page.module.scss", "page.tsx", "lib", "services", and "middleware.ts".
- Search Bar:** Contains the text "frontend".
- Code Editor:** Displays the content of "index.tsx". The code uses TypeScript and React, importing components like ChangeEvent, useState, and UploadCloud from various packages. It defines interfaces for CategoryProps and Props, and exports a Form function. The Form function handles file selection and preview, and performs an asynchronous registration process.
- Status Bar:** Shows "Ln 9, Col 1 (52 selected)" and other file-related information.

Sujeito Pizza - A melhor pizzaria

Novo produto

bebidas

Digite o nome do produto...

Preço do produto...

Digite a descrição do produto...

Cadastrar produto

The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** index.tsx (selected), page.tsx, styles.module.scss.
- Content:** Code for handling product registration. It uses useState for previewImage, sets it to an empty string initially. Then, it checks if name, categoryIndex, price, and description are provided. If so, it creates a new FormData object, appends the required fields, and the file (image). Finally, it uses axios to post the data to "/product" with an Authorization header containing a Bearer token.
- Bottom Status Bar:** 2 selections (44 characters selected), Spaces: 4, UTF-8, CRLF, TypeScript JSX, Prettier.

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** Categoria, Produto, [arrow].
- Section:** Novo produto.
- Image:** A close-up photo of a pizza slice with toppings.
- Form Fields:**bebidas, Digite o nome do produto..., Preço do produto..., Digite a descrição do produto... (all are empty).
- Button:** Cadastrar produto (in green).

The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code is for a product registration form. It uses useState for previewImage and handles form data submission via API post to "/product".

```
const [previewImage, setPreviewImage] = useState("")  
const categoryIndex = formData.get("category")  
const name = formData.get("name")  
const price = formData.get("price")  
const description = formData.get("description")  
  
if(!name || !categoryIndex || !price || !description || !image){  
    return;  
}  
  
const data = new FormData();  
  
data.append("name", name)  
data.append("price", price)  
data.append("description", description)  
data.append("category_id", categories[Number(categoryIndex)].id)  
data.append("file", image)  
  
const token = getCookieClient();  
  
await api.post("/product", data, {  
    headers:{  
        Authorization: `Bearer ${token}`  
    }  
})  
  
console.log("CADASTRADO COM SUCESSO!")
```

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** Categoria Produto
- Section:** Novo produto
- Image:** A close-up image of a pizza slice.
- Form Fields:**
  - bebidas
  - Digite o nome do produto...
  - Preço do produto...
  - Digite a descrição do produto...
- Button:** Cadastrar produto

The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** index.tsx (selected), page.tsx, styles.module.scss.
- Content:** TypeScript code for handling product registration. It includes form validation, creating a FormData object, setting headers with an Authorization token, and logging success.
- Bottom Status Bar:** Shows file path (app > dashboard > product > components > form > index.tsx), line 47, column 45, and other development details.

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** SujeitoPizza (with a red "P") and navigation links: Categoria, Produto, [Entrar].
- Section:** Novo produto
- Form Fields:** Name (placeholder: "Digite o nome do produto..."), Price (placeholder: "Preço do produto..."), Description (placeholder: "Digite a descrição do produto...").
- Buttons:** Cadastrar produto (green button).

frontend

index.tsx page.tsx styles.module.scss

app > dashboard > product > components > form > index.tsx > Form

```
25
26     const categoryIndex = formData.get("category")
27     const name = formData.get("name")
28     const price = formData.get("price")
29     const description = formData.get("description")
30
31     if(!name || !categoryIndex || !price || !description) {
32         return;
33     }
34
35     const data = new FormData();
36
37     data.append("name", name)
38     data.append("price", price)
39     data.append("description", description)
40     data.append("category_id", categories[Number(categoryIndex)])
41     data.append("file", image)
42
43     const token = getCookieClient();
44
45     await api.post("/product", data, {
46         headers: {
47             "Authorization": `Bearer ${token}`
48         }
49     })
50
51     console.log("CADASTRADO COM SUCESSO!")
52
53 }
54
55 function handleFile(e: ChangeEvent<HTMLInputElement>){
56     if(e.target.files && e.target.files[0]){

Nome: hamburger Arquivos personalizados Abrir Cancelar
```

Digite o nome do produto...

Preço do produto...

Digite a descrição do produto...

Cadastrar produto

frontend

index.tsx page.tsx styles.module.scss

FRONTEND .next node\_modules public src app dashboard category components product components form index.tsx styles.module.scss page.tsx layout.tsx page.tsx fonts signup favicon.ico globals.scss layout.tsx page.module.scss page.tsx lib services middleware.ts

EXPLORER OUTLINE TIMELINE

Ln 47, Col 45 Spaces: 4 UTF-8 CRLF TypeScript JSX Prettier

Sujeito Pizza - A melhor pizzaria

SujeitoPizza Categoria Produto

Novo produto

hamburger

Dev Burger Salada

40

Burgão Dev de Salada com 1 hambúrguer de 160g

Cadastrar produto

```
const categoryIndex = formData.get("category")
const name = formData.get("name")
const price = formData.get("price")
const description = formData.get("description")

if(!name || !categoryIndex || !price || !description || !image){
    return;
}

const data = new FormData();

data.append("name", name)
data.append("price", price)
data.append("description", description)
data.append("category_id", categories[Number(categoryIndex)].id)
data.append("file", image)

const token = getCookieClient();

await api.post("/product", data, {
    headers:{
        Authorization: `Bearer ${token}`
    }
})

console.log("CADASTRADO COM SUCESSO!")

function handleFile(e: ChangeEvent<HTMLInputElement>){
    if(e.target.files && e.target.files[0]){

```

frontend

index.tsx page.tsx styles.module.scss

FRONTEND .next node\_modules public src app dashboard category components product components form index.tsx styles.module.scss page.tsx layout.tsx page.tsx fonts signup favicon.ico globals.scss layout.tsx page.module.scss page.tsx lib services middleware.ts

OUTLINE TIMELINE

Ln 47, Col 45 Spaces: 4 UTF-8 CRLF TypeScript JSX Prettier

Sujeito Pizza - A melhor pizzaria

Categoria Produto

## Novo produto



bebidas

Digite o nome do produto...

Preço do produto...

Digite a descrição do produto...

Cadastrar produto

Sujeito Pizza

Categoria Produto →

## **Novo produto**



behindas

Digite o nome do produto

Digite a descrição do produto...

## Cadastrar produto

Elements Console Sources Network Performance Memory Application Security Lighthouse Recorder > ▲ 4 📈 1 ⚙️ ⋮ X

⚠ The resource [http://localhost:3000/\\_next/static/media/logo.77bc50ee.svg](http://localhost:3000/_next/static/media/logo.77bc50ee.svg) was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally.

⚠ The resource [http://localhost:3000/\\_next/static/media/logo.77bc50ee.svg](http://localhost:3000/_next/static/media/logo.77bc50ee.svg) was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally.

CADASTRADO COM SUCESSO

c:\Users\Dev\Desktop..s\form\index.tsx:51

**⚠** The resource [http://localhost:3000/\\_next/static/media/logo.77bc50ee.svg](http://localhost:3000/_next/static/media/logo.77bc50ee.svg) was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally.

**⚠** The resource [http://localhost:3000/\\_next/static/media/logo.77bc50ee.svg](http://localhost:3000/_next/static/media/logo.77bc50ee.svg) was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally.

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C + Query #1 categories [all] products [all] x

Filter ENTITIES 217 +

public

- \_prisma\_migrations
- categories
- items
- orders
- products
- users

information\_schema

pg\_catalog

id equals Enter Value

	id	name	price	description	banner	created_at	updated_at
1	1cd5c28d-6392-4d30-9b2d-91278fccef7f0	Coca-Cola 2L	10	Coca-Cola gelad...	b98497f70a...	2024-11-25 23:47:24.095	2024-11-25 23:47:24.095
2	7f66d79a-ceb2-4175-b4a6-dc5208a9f50e	Pizza de calabresa	46	Pizza de calabre...	8a652b025a...	2024-11-25 22:06:31.016	2024-11-25 22:06:31.016
3	ca5c64fa-e4dd-4b67-9c33-2c4961d4b0b1	Coca lata	5	Coca-Cola lata 3...	765f54ffa7f0...	2024-11-25 22:17:10.368	2024-11-25 22:17:10.368

localhost:5432/pizzaria postgres Structure # Unknown 1 hour ago 1 > C +

localhost:5432/pizzaria - Beekeeper Studio

pizzaria		C	+	Query #1	categories [all]	products [all] x	+										
				id	equals	Enter Value											
				id	text	name	text	price	text	description	text	banner	text	created_at	timestamp(3)	update_at	time
ENTITIES 217		+		1	1cd5c28d-6392-4d30-9b2d-91278fcf7f0	Coca-Cola 2L	10	Coca-Cola gelad...	b98497f70a...	2024-11-25 23:47:24.095	2024-11-25 2						
public		+		2	7f66d79a-ceb2-4175-b4a6-dc5208a9f50e	Pizza de calabresa	46	Pizza de calabre...	8a652b025a...	2024-11-25 22:06:31.016	2024-11-25 2						
-> _prisma_migrations		+		3	96eb4477-824b-4137-8251-dbce1d72f755	Dev Burger Salada	40	Burgão Dev de S...	c83f6cc7a8b...	2024-11-29 01:06:24.05	2024-11-29 0						
-> categories		+		4	ca5c64fa-e4dd-4b67-9c33-2c4961d4b0b1	Coca lata	5	Coca-Cola lata 3...	765f54ffa7f0...	2024-11-25 22:17:10.368	2024-11-25 2						
-> items		+															
-> orders		+															
-> products		+															
-> users		+															
> information_schema		+															
> pg_catalog		+															

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

**pizzaria** C + Query #1 categories [all] products [all] x +

Filter

ENTITIES 217 +

public

- \_prisma\_migrations
- categories
- items
- orders
- products
- users

information\_schema

pg\_catalog

	name	price	description	banner	created_at	update_at	category_id
1	Coca-Cola 2L	10	Coca-Cola gelad...	b98497f70a...	2024-11-25 23:47:24.095	2024-11-25 23:47:24.0...	00d8f79f-5536-429a-bd59-7a82f18700ac
2	Pizza de calabresa	46	Pizza de calabre...	8a652b025a...	2024-11-25 22:06:31.016	2024-11-25 22:06:31.0...	e21422b1-f695-4426-aa06-9d23c2f3502b
3	Burgão Dev Burger Salada	40	Burgão Dev de S...	c83f6cc7a8b...	2024-11-29 01:06:24.05	2024-11-29 01:06:24.05	1e4ffc20-04b2-46db-96f0-ceea3ca2975a
4	Coca lata	5	Coca-Cola lata 3...	765f54ffa7f0...	2024-11-25 22:17:10.368	2024-11-25 22:17:10.3...	00d8f79f-5536-429a-bd59-7a82f18700ac

00d8f79f-5536-429a-bd59-7a82f18700ac View record in categories

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

**pizzaria** C + **Query #1** products [all] +

**ENTITIES** 217 +

- public
  - \_prisma\_migrations
  - categories
  - items
  - orders
  - products
  - users
- information\_schema
- pg\_catalog

	id	equals	Enter Value					
1	ba9f50e	Pizza de calabre...	46	Pizza de cal...	8a652b025a...	2024-11-25 ...	2024-11-25 22:06:31.016	e21422b1-f695-4426-aa06-9d23c2f3502b
2	d4b0...	Coca lata	5	Coca-Cola la...	765f54ffa7f...	2024-11-25 ...	2024-11-25 22:17:10.368	00d8f79f-5536-429a-bd59-7a82f18700ac
3	fcef7f0	Coca-Cola 2L	10	Coca-Cola g...	b98497f70a...	2024-11-25 ...	2024-11-25 23:47:24.095	00d8f79f-5536-429a-bd59-7a82f18700ac
4	d72f7...	Dev Burger Sala...	40	Burgão Dev ...	c83f6cc7a8...	2024-11-29 ...	2024-11-29 01:06:24.05	1e4ffc20-04b2-46db-96f0-ceea3ca2975a

**1e4ffc20-04b** View record in categories

pizzaria

Filter

ENTITIES 217

- public
  - \_prisma\_migrations
  - categories
  - items
  - orders
  - products
  - users
- information\_schema
- pg\_catalog

Query #1 products [all] categories [1e4ffc20-04b2-46db-96f0-ceea3ca2975a]

id equals 1e4ffc20-04b2-46db-96f0-ceea3ca2975a

	id	name	created_at...	update_at
1	1e4ffc20-04b2-46db-96f0-ceea3ca2975a	hamburger	2024-11-28 ...	2024-11-28 00:42:45.7...

1e4ffc20-04b2-46db-96f0-ceea3ca2975a

The image shows a developer's environment with two main windows: a code editor in the foreground and a browser preview in the background.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** index.tsx (selected), page.tsx, styles.module.scss.
- Content:** TypeScript code for handling product registration. It includes logic to validate form data, create a FormData object, append data, and make a POST request to the "/product" endpoint using an Authorization header.

```
const categoryIndex = formData.get("category")
const name = formData.get("name")
const price = formData.get("price")
const description = formData.get("description")

if(!name || !categoryIndex || !price || !description || !image){
    return;
}

const data = new FormData()

data.append("name", name)
data.append("price", price)
data.append("description", description)
data.append("category_id", categories[Number(categoryIndex)].id)
data.append("file", image)

const token = getCookieClient()

await api.post("/product", data, {
    headers: {
        Authorization: `Bearer ${token}`
    }
}).catch((err) => {
    console.log(err);
})
```

**Browser Preview (Microsoft Edge):**

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** Categoria Produto [ ]
- Section:** Novo produto
- Image:** A large, appetizing burger with lettuce, cheese, and meat.
- Form Fields:**bebidas  
Digite o nome do produto...  
Preço do produto...  
Digite a descrição do produto...
- Button:** Cadastrar produto

frontend

index.tsx page.tsx styles.module.scss

FRONTEND .next node\_modules public src app dashboard category components product components form index.tsx styles.module.scss page.tsx layout.tsx page.tsx fonts signup favicon.ico globals.scss layout.tsx page.module.scss page.tsx lib services middleware.ts

OUTLINE TIMELINE

Ln 47, Col 39 (3 selected) Spaces: 4 UTF-8 CRLF TypeScript JSX Prettier

const categoryIndex = formData.get("category")  
const name = formData.get("name")  
const price = formData.get("price")  
const description = formData.get("description")  
  
if(!name || !categoryIndex || !price || !description || !image){  
 return;  
}  
  
const data = new FormData();  
  
data.append("name", name)  
data.append("price", price)  
data.append("description", description)  
data.append("category\_id", categories[Number(categoryIndex)].id)  
data.append("file", image)  
  
const token = getCookieClient();  
  
await api.post("/product", data, {  
 headers:{  
 Authorization: `Bearer 123`  
 }  
}.catch((err) => {  
 console.log(err);  
})  
  
console.log("CADASTRADO COM SUCESSO!")

Sujeito Pizza - A melhor pizzaria

Categoria Produto [ ]

# SujeitoPizza

## Novo produto



bebidas

Digite o nome do produto...

Preço do produto...

Digite a descrição do produto...

Cadastrar produto

The image shows a developer's workspace with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** index.tsx (selected), page.tsx, styles.module.scss.
- Content:** TypeScript code for handling product registration. It includes logic to validate form data, create a FormData object, and make a POST request to the "/product" endpoint using an Authorization header.

```
const categoryIndex = formData.get("category")
const name = formData.get("name")
const price = formData.get("price")
const description = formData.get("description")

if(!name || !categoryIndex || !price || !description || !image){
    return;
}

const data = new FormData()

data.append("name", name)
data.append("price", price)
data.append("description", description)
data.append("category_id", categories[Number(categoryIndex)].id)
data.append("file", image)

const token = getCookieClient()

await api.post("/product", data, {
    headers:{
        Authorization: `Bearer 123`
    }
})
.catch((err) => {
    console.log(err);
})

console.log("CADASTRADO COM SUCESSO!")
```

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** SujeitoPizza
- Section:** Novo produto
- Form Fields:** Name (placeholder: bebidas), Price (placeholder: Digite o nome do produto...), Description (placeholder: Preço do produto...).
- Buttons:** Cadastrar produto

frontend

index.tsx page.tsx styles.module.scss

FRONTEND .next node\_modules public src app dashboard category components product components form index.tsx styles.module.scss page.tsx layout.tsx page.tsx fonts signup favicon.ico globals.scss layout.tsx page.module.scss page.tsx lib services middleware.ts

OUTLINE TIMELINE

Ln 47, Col 39 (3 selected) Spaces: 4 UTF-8 CRLF TypeScript JSX Prettier

25  
26     const categoryIndex = formData.get("category")  
27     const name = formData.get("name")  
28     const price = formData.get("price")  
29     const description = formData.get("description")  
30  
31     if(!name || !categoryIndex || !price || !description)  
32     |     return;  
33     }  
34  
35     const data = new FormData();  
36  
37     data.append("name", name)  
38     data.append("price", price)  
39     data.append("description", description)  
40     data.append("category\_id", categories[Number(categoryIndex)])  
41     data.append("file", image)  
42  
43     const token = getCookieClient();  
44  
45     await api.post("/product", data, {  
46         headers:{  
47             |     Authorization: `Bearer 123`  
48         }  
49     })  
50     .catch((err) => {  
51         console.log(err);  
52     })  
53  
54     console.log("CADASTRADO COM SUCESSO!")  
55  
56 }

Abrir

Área de Trabalho > arquivos

Organizar Nova pasta

açaí hamburger sorvete 1

Nome: hamburger Arquivos personalizados

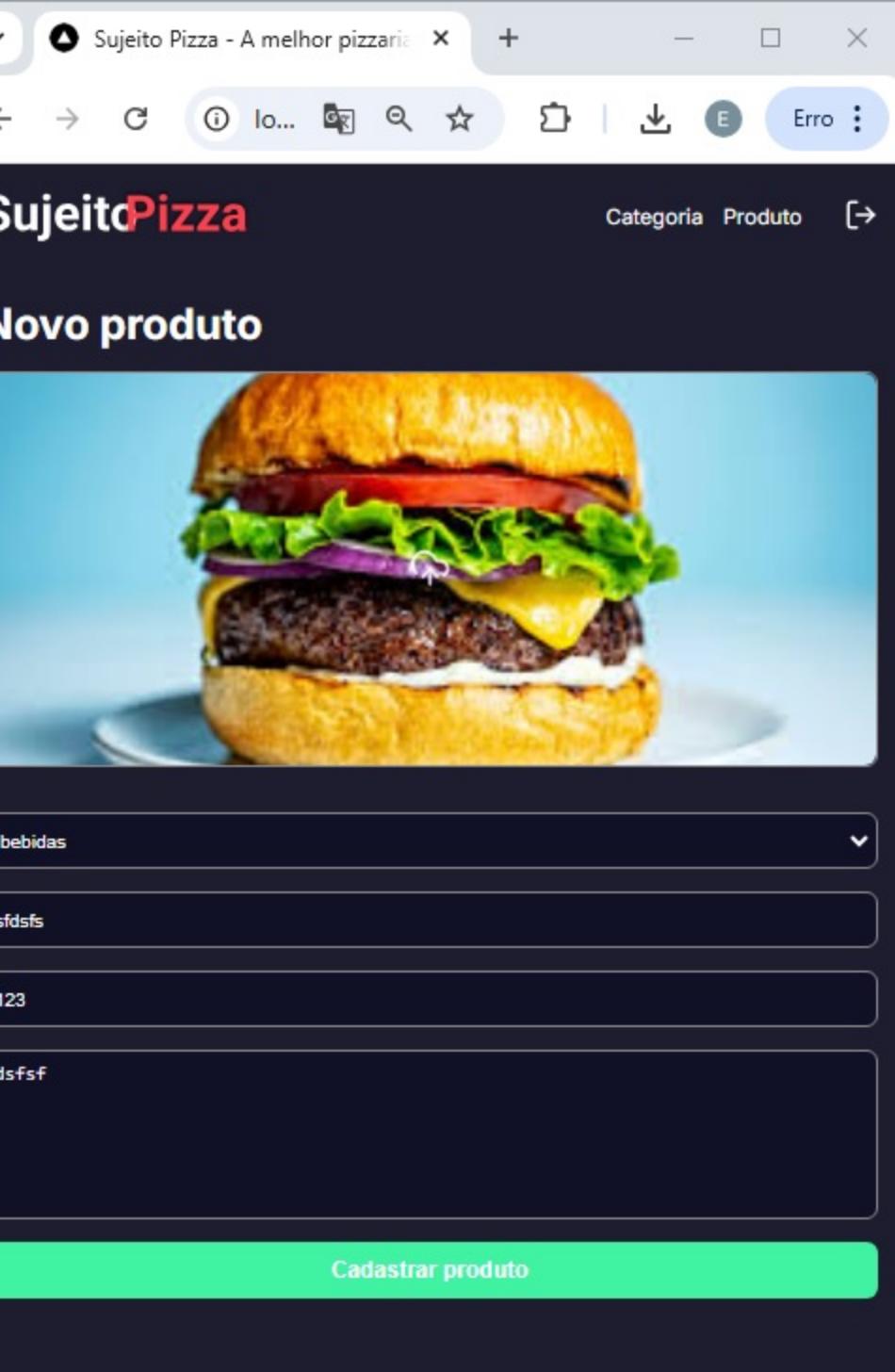
Abrir Cancelar

Digite o nome do produto...

Preço do produto...

Digite a descrição do produto...

Cadastrar produto



The screenshot shows a development environment with a code editor and a browser window.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** index.tsx (selected), page.tsx, styles.module.scss.
- Content:** Code for handling product registration. It checks for required fields (name, categoryIndex, price, description, image). It creates a FormData object, appends fields, and sends a POST request to "/product" with an Authorization header (Bearer 123). It logs errors and success messages.

```
const categoryIndex = formData.get("category")
const name = formData.get("name")
const price = formData.get("price")
const description = formData.get("description")

if(!name || !categoryIndex || !price || !description || !image){
    return;
}

const data = new FormData();

data.append("name", name)
data.append("price", price)
data.append("description", description)
data.append("category_id", categories[Number(categoryIndex)].id)
data.append("file", image)

const token = getCookieClient();

await api.post("/product", data, {
    headers:{
        Authorization: `Bearer 123`
    }
})
.catch((err) => {
    console.log(err);
})

console.log("CADASTRADO COM SUCESSO!")
```

**Browser:** Screenshot of a pizza delivery website titled "Sujeito Pizza - A melhor pizzaria".

- Header:** SujeitoPizza, Categoria, Produto, [Entrar]
- Section:** Novo produto
- Image:** A large, appetizing burger with lettuce, cheese, and meat.
- Input Fields:** bebidas, sfdsfs, 123, dsfsf.
- Button:** Cadastrar produto

frontend

index.tsx page.tsx styles.module.scss

FRONTEND

.next node\_modules public src app dashboard category components product components form index.tsx styles.module.scss page.tsx layout.tsx page.tsx fonts signup favicon.ico globals.scss layout.tsx page.module.scss page.tsx lib services middleware.ts

OUTLINE

TIMELINE

Ln 47, Col 39 (3 selected) Spaces: 4 UTF-8 CRLF TypeScript JSX Prettier

```
const categoryIndex = formData.get("category")
const name = formData.get("name")
const price = formData.get("price")
const description = formData.get("description")

if(!name || !categoryIndex || !price || !description || !image){
    return;
}

const data = new FormData()

data.append("name", name)
data.append("price", price)
data.append("description", description)
data.append("category_id", categories[Number(categoryIndex)].id)
data.append("file", image)

const token = getCookieClient()

await api.post("/product", data, {
    headers:{
        Authorization: `Bearer 123`
    }
})
.catch((err) => {
    console.log(err);
})

console.log("CADASTRADO COM SUCESSO!")
```

Sujeito Pizza - A melhor pizzaria

Categoria Produto [ ]

# SujeitoPizza

## Novo produto



bebidas

Digite o nome do produto...

Preço do produto...

Digite a descrição do produto...

Cadastrar produto

Sujeito Pizza - A melhor pizzaria

localhost:3000/dashboard/product

Elements Console Sources Network Performance Memory Application Security Lighthouse

Default levels | 1 Issue: 1

The resource [http://localhost:3000/\\_next/static/media/logo.77bc50ee.svg](http://localhost:3000/_next/static/media/logo.77bc50ee.svg) was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally. product:1

The resource [http://localhost:3000/\\_next/static/media/logo.77bc50ee.svg](http://localhost:3000/_next/static/media/logo.77bc50ee.svg) was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally. product:1

Failed to load resource: the server responded with a status of 401 (Unauthorized) :3333/product:1

AxiosError

CADASTRADO COM SUCESSO!

The resource [http://localhost:3000/\\_next/static/media/logo.77bc50ee.svg](http://localhost:3000/_next/static/media/logo.77bc50ee.svg) was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally. product:1

The resource [http://localhost:3000/\\_next/static/media/logo.77bc50ee.svg](http://localhost:3000/_next/static/media/logo.77bc50ee.svg) was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally. product:1

bebidas

Digite o nome do produto...

Preço do produto...

Digite a descrição do produto...

Cadastrar produto

Console AI assistance

Sujeito Pizza - A melhor pizzaria

localhost:3000/dashboard/product

Elements Console Sources Network Performance Memory Application Security Lighthouse

Default levels | 1 Issue: 1

The resource [http://localhost:3000/\\_next/static/media/logo.77bc50ee.svg](http://localhost:3000/_next/static/media/logo.77bc50ee.svg) was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally. product:1

The resource [http://localhost:3000/\\_next/static/media/logo.77bc50ee.svg](http://localhost:3000/_next/static/media/logo.77bc50ee.svg) was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally. product:1

Failed to load resource: the server responded with a status of 401 (Unauthorized) :3333/product:1

c:\Users\Dev\Desktop\s\form\index.tsx:51

AxiosError

- code: "ERR\_BAD\_REQUEST"
- config: {transitional: {}, adapter: Array(3), transformRequest: Array(1), transformResponse: Array(1), timeout: 0, ...} message: "Request failed with status code 401" name: "AxiosError"
- request: XMLHttpRequest {onreadystatechange: null, readyState: 4, timeout: 0, withCredentials: false, upload: XMLHttpRequestUpload, ...}
- response: {data: '', status: 401, statusText: 'Unauthorized', headers: AxiosHeaders, config: {}, ...} status: 401 stack: "AxiosError: Request failed with status code 401\n at settle (webpack-internal:///app-pages-browser/.node\_modules/axios/lib/core/[[Prototype]]: Error

CADASTRADO COM SUCESSO!

c:\Users\Dev\Desktop\s\form\index.tsx:54

The resource [http://localhost:3000/\\_next/static/media/logo.77bc50ee.svg](http://localhost:3000/_next/static/media/logo.77bc50ee.svg) was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally. product:1

The resource [http://localhost:3000/\\_next/static/media/logo.77bc50ee.svg](http://localhost:3000/_next/static/media/logo.77bc50ee.svg) was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally. product:1

Digite o nome do produto...

Digite o preço do produto...

Digite a descrição do produto...

Cadastrar produto

Console AI assistance

Sujeito Pizza - A melhor pizzaria

localhost:3000/dashboard/product

Elements Console Sources Network Performance Memory Application Security Lighthouse

Default levels | 1 Issue: 1

**SujeitoPizza** Categoria Produto [→]

## Novo produto



bebidas

Digite o nome do produto...

Preço do produto...

Digite a descrição do produto...

Cadastrar produto

The resource [http://localhost:3000/\\_next/static/media/logo.77bc50ee.svg](http://localhost:3000/_next/static/media/logo.77bc50ee.svg) was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally. product:1

The resource [http://localhost:3000/\\_next/static/media/logo.77bc50ee.svg](http://localhost:3000/_next/static/media/logo.77bc50ee.svg) was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally. product:1

Failed to load resource: the server responded with a status of 401 (Unauthorized) :3333/product:1

```
c:\Users\Dev\Desktop...s\form\index.tsx:51
```

AxiosError {  
 code: "ERR\_BAD\_REQUEST"  
 config: {transitional: {}, adapter: Array(3), transformRequest: Array(1), transformResponse: Array(1), timeout: 0, ...}  
 message: "Request failed with status code 401"  
 name: "AxiosError"  
 request: XMLHttpRequest {onreadystatechange: null, readyState: 4, timeout: 0, withCredentials: false, upload: XMLHttpRequestUpload, ...}  
 response:  
 config: {transitional: {}, adapter: Array(3), transformRequest: Array(1), transformResponse: Array(1), timeout: 0, ...}  
 data: ""  
 headers: AxiosHeaders {content-length: '0'}  
 request: XMLHttpRequest {onreadystatechange: null, readyState: 4, timeout: 0, withCredentials: false, upload: XMLHttpRequestUpload, ...}  
 status: 401  
 statusText: "Unauthorized"  
 [[Prototype]]: Object  
 status: 401  
 stack: "AxiosError: Object failed with status code 401\n at settle (webpack-internal:///app-pages-browser/.node\_modules/axios/lib/core/[[Prototype]]: Error  
 c:\Users\Dev\Desktop...s\form\index.tsx:54

CADASTRADO COM SUCESSO! c:\Users\Dev\Desktop...s\form\index.tsx:54

The resource [http://localhost:3000/\\_next/static/media/logo.77bc50ee.svg](http://localhost:3000/_next/static/media/logo.77bc50ee.svg) was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally. product:1

The resource [http://localhost:3000/\\_next/static/media/logo.77bc50ee.svg](http://localhost:3000/_next/static/media/logo.77bc50ee.svg) was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally. product:1

Sujeito Pizza - A melhor pizzaria

localhost:3000/dashboard/product

Elements Console Sources Network Performance Memory Application Security Lighthouse

Default levels | 1 Issue:

The resource [http://localhost:3000/\\_next/static/media/logo.77bc50ee.svg](http://localhost:3000/_next/static/media/logo.77bc50ee.svg) was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally.

The resource [http://localhost:3000/\\_next/static/media/logo.77bc50ee.svg](http://localhost:3000/_next/static/media/logo.77bc50ee.svg) was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally.

Failed to load resource: the server responded with a status of 401 (Unauthorized)

AxiosError

code: "ERR\_BAD\_REQUEST"

config: {transitional: {}, adapter: Array(3), transformRequest: Array(1), transformResponse: Array(1), timeout: 0, ...}

message: "Request failed with status code 401"

name: "AxiosError"

request: XMLHttpRequest {onreadystatechange: null, readyState: 4, timeout: 0, withCredentials: false, upload: XMLHttpRequestUpload, ...}

response:

config:

adapter: ['xhr', 'http', 'fetch']  
baseURL: "http://localhost:3333"

data: FormData {}

env: {FormData: f, Blob: f}

headers: AxiosHeaders

Accept: "application/json, text/plain, \*/\*"  
Authorization: "Bearer 123"  
Content-Type: "application/x-www-form-urlencoded"

clear: (...)  
concat: (...)  
constructor: (...)  
delete: (...)  
get: (...)  
getAccept: (...)  
getAcceptEncoding: (...)  
getAuthorization: (...)  
getContentLength: (...)  
getContentType: (...)  
getUserAgent: (...)  
has: (...)

bebidas

Digite o nome do produto...

Preço do produto...

Digite a descrição do produto...

Cadastrar produto



frontend

index.tsx page.tsx styles.module.scss

FRONTEND .next node\_modules public src app dashboard category components product components form index.tsx styles.module.scss page.tsx layout.tsx page.tsx fonts signup favicon.ico globals.scss layout.tsx page.module.scss page.tsx lib services middleware.ts

OUTLINE TIMELINE

2 selections (23 characters selected) Spaces: 4 UTF-8 CRLF TypeScript JSX Prettier

Sujeito Pizza - A melhor pizzaria

SujeitoPizza

Categoria Produto

Novo produto

bebidas

Digite o nome do produto...

Preço do produto...

Digite a descrição do produto...

Cadastrar produto

```
const categoryIndex = formData.get("category")
const name = formData.get("name")
const price = formData.get("price")
const description = formData.get("description")

if(!name || !categoryIndex || !price || !description || !image){
    return;
}

const data = new FormData()

data.append("name", name)
data.append("price", price)
data.append("description", description)
data.append("category_id", categories[Number(categoryIndex)].id)
data.append("file", image)

const token = getCookieClient()

await api.post("/product", data, {
    headers: {
        Authorization: `Bearer ${token}`
    }
})
.catch((err) => {
    console.log(err);
    return;
})

console.log("CADASTRADO COM SUCESSO!")
```

The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** index.tsx (selected), page.tsx, styles.module.scss.
- Content:** TypeScript code for product registration. It includes form validation, creating a FormData object, setting headers with a Bearer token, and handling the API post request.

```
const categoryIndex = formData.get("category")
const name = formData.get("name")
const price = formData.get("price")
const description = formData.get("description")

if(!name || !categoryIndex || !price || !description || !image){
    return;
}

const data = new FormData()

data.append("name", name)
data.append("price", price)
data.append("description", description)
data.append("category_id", categories[Number(categoryIndex)].id)
data.append("file", image)

const token = getCookieClient()

await api.post("/product", data, {
    headers:{
        Authorization: `Bearer ${token}`
    }
})
.catch((err) => {
    console.log(err);
    return;
})

console.log("CADASTRADO COM SUCESSO!")
```

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** Categoria Produto [ ]
- Section:** Novo produto
- Form Fields:**  (placeholder: Dígite o nome do produto...), ,
- Buttons:** Cadastrar produto

frontend

EXPLORER

FRONTEND

- > .next
- > node\_modules
- > public
- src
  - app
    - dashboard
      - > category
      - > components
    - product
      - > components
        - form
          - index.tsx
          - styles.module.scss

index.tsx

page.tsx

styles.module.scss

25

26     const categoryIndex = formData.get("category")

27     const name = formData.get("name")

28     const price = formData.get("price")

29     const description = formData.get("description")

30

31     if(!name || !categoryIndex || !price || !description) {

32         return;

33     }

34

35     const data = new FormData();

36

37     data.append("name", name)

38     data.append("price", price)

39     data.append("description", description)

40     data.append("category\_id", categories[Number(categoryIndex)])

41     data.append("file", image)

42

43     const token = getCookieClient();

44

45     await api.post("/product", data, {

46         headers:{

47             Authorization: `Bearer \${token}`

48         }

49     })

50     .catch((err) => {

51         console.log(err);

52         return;

53     })

54

55     console.log("CADASTRADO COM SUCESSO!")

56

Organizar

Nova pasta

açaí

hamburger

sorvete 1

Nome: açaí

Arquivos personalizados

Abrir

Cancelar

Digite o nome do produto...

Preço do produto...

Digite a descrição do produto...

Cadastrar produto

2 selections (23 characters selected) Spaces: 4 UTF-8 CRLF TypeScript JSX Prettier

frontend

index.tsx page.tsx styles.module.scss

```
const categoryIndex = formData.get("category")
const name = formData.get("name")
const price = formData.get("price")
const description = formData.get("description")

if(!name || !categoryIndex || !price || !description || !image){
    return;
}

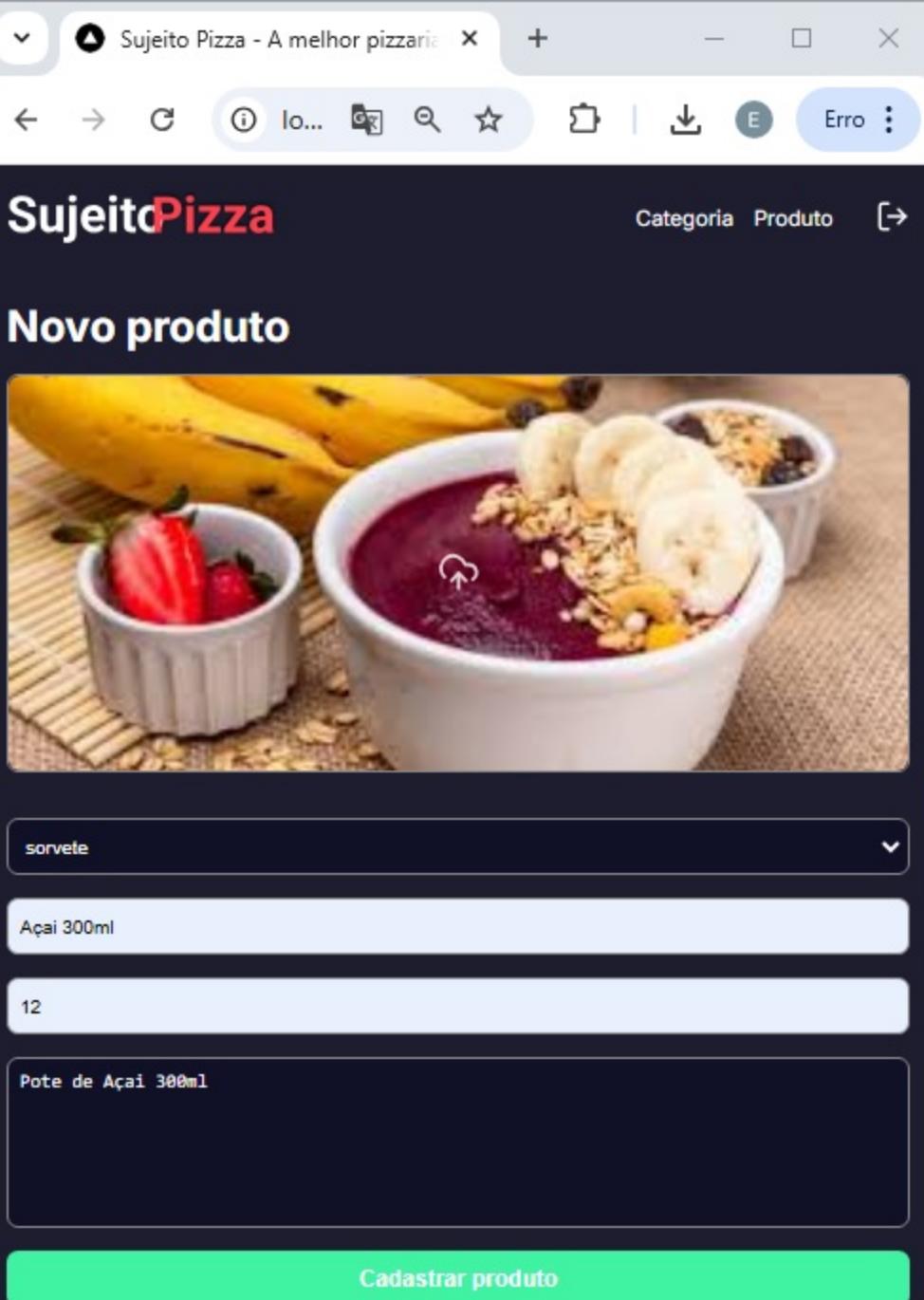
const data = new FormData()

data.append("name", name)
data.append("price", price)
data.append("description", description)
data.append("category_id", categories[Number(categoryIndex)].id)
data.append("file", image)

const token = getCookieClient()

await api.post("/product", data, {
    headers:{
        Authorization: `Bearer ${token}`
    }
})
.catch((err) => {
    console.log(err);
    return;
})

console.log("CADASTRADO COM SUCESSO!")
```



frontend

EXPLORER

FRONTEND

- > .next
- > node\_modules
- > public
- src
  - app
    - dashboard
    - > category
    - > components
    - product
      - components
        - form
      - index.tsx
      - styles.module.scss
    - page.tsx
    - layout.tsx
    - page.tsx
    - fonts
    - signup
    - favicon.ico
    - globals.scss
    - layout.tsx
    - page.module.scss
    - page.tsx
    - lib
    - services
    - middleware.ts

OUTLINE

TIMELINE

index.tsx page.tsx styles.module.scss

```
const categoryIndex = formData.get("category")
const name = formData.get("name")
const price = formData.get("price")
const description = formData.get("description")

if(!name || !categoryIndex || !price || !description || !image){
    return;
}

const data = new FormData();

data.append("name", name)
data.append("price", price)
data.append("description", description)
data.append("category_id", categories[Number(categoryIndex)].id)
data.append("file", image)

const token = getCookieClient()

await api.post("/product", data, {
    headers:{
        Authorization: `Bearer ${token}`
    }
})
.catch((err) => {
    console.log(err);
    return;
})

console.log("CADASTRADO COM SUCESSO!")
```

2 selections (23 characters selected) Spaces: 4 UTF-8 CRLF TypeScript JSX Prettier

Sujeito Pizza - A melhor pizzaria

Categoria Produto [ ]

# SujeitoPizza

## Novo produto



bebidas

Digite o nome do produto...

Preço do produto...

Digite a descrição do produto...

Cadastrar produto

Sujeito Pizza

Categoria Produto →

## **Novo produto**



behindas

Digite o nome do produto...

Digite a descrição do produto...

## Cadastrar produto

Elements Console Sources Network Performance Memory Application Security Lighthouse Recorder > ▲ 4 📈 1 ⚙️ ⋮ X

**⚠** The resource [http://localhost:3000/\\_next/static/media/logo.77bc50ee.svg](http://localhost:3000/_next/static/media/logo.77bc50ee.svg) was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally.

⚠ The resource [http://localhost:3000/\\_next/static/media/logo.77bc50ee.svg](http://localhost:3000/_next/static/media/logo.77bc50ee.svg) was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally. [product:1](#)

CADASTRADO COM SUCESSO

c:\Users\Dev\Desktop...s\form\index.tsx:55

⚠ The resource [http://localhost:3000/\\_next/static/media/logo.77bc50ee.svg](http://localhost:3000/_next/static/media/logo.77bc50ee.svg) was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally.

**⚠** The resource [http://localhost:3000/\\_next/static/media/logo.77bc50ee.svg](http://localhost:3000/_next/static/media/logo.77bc50ee.svg) was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally.

8

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C +

Query #1 products [all] categories [1e4ffc20-04b2-4c]

Filter Y

ENTITIES 217 +

- public
  - \_prisma\_migrations
  - categories
  - items
  - orders
  - products
  - users
- information\_schema
- pg\_catalog

	id	equals	Enter Value					
1	3a9f50e	Pizza de calabre...	46	Pizza de cal...	8a652b025a...	2024-11-25 ...	2024-11-25 22:06:31.016	e21422b1-f695-4426-aa06-9d23c2f3502b
2	d4b0...	Coca lata	5	Coca-Cola la...	765f54ffa7f...	2024-11-25 ...	2024-11-25 22:17:10.368	00d8f79f-5536-429a-bd59-7a82f18700ac
3	fcef7f0	Coca-Cola 2L	10	Coca-Cola g...	b98497f70a...	2024-11-25 ...	2024-11-25 23:47:24.095	00d8f79f-5536-429a-bd59-7a82f18700ac
4	d72f7...	Dev Burger Sala...	40	Burgão Dev ...	c83f6cc7a8...	2024-11-29 ...	2024-11-29 01:06:24.05	1e4ffc20-04b2-46db-96f0-ceea3ca2975a



pizzaria

Filter

ENTITIES 217

- public
  - \_prisma\_migrations
  - categories
  - items
  - orders
  - products
  - users
- information\_schema
- pg\_catalog

Query #1 products [all] categories [1e4ffc20-04b2-4e]

id equals Enter Value

	name	price	description	banner	created_at	update_at	category_id
1	a9f50e Pizza de calabre...	46	Pizza de cal...	8a652b025a...	2024-11-25 ...	2024-11-25 22:06:31.016	e21422b1-f695-4426-aa06-9d23c2f3502b
2	d4b0... Coca lata	5	Coca-Cola la...	765f54ffa7f...	2024-11-25 ...	2024-11-25 22:17:10.368	00d8f79f-5536-429a-bd59-7a82f18700ac
3	fcef7f0 Coca-Cola 2L	10	Coca-Cola g...	b98497f70a...	2024-11-25 ...	2024-11-25 23:47:24.095	00d8f79f-5536-429a-bd59-7a82f18700ac
4	d72f7... Dev Burger Sala...	40	Burgão Dev ...	c83f6cc7a8...	2024-11-29 ...	2024-11-29 01:06:24.05	1e4ffc20-04b2-46db-96f0-ceea3ca2975a
5	53b8ed Açaí 300ml	12	Pote de Açaí...	d3c76275c1...	2024-11-29 ...	2024-11-29 01:29:50.485	f44fb594-d1c4-425f-b1bd-87541a6637a1

localhost:5432/pizzaria postgres Structure ↗ # Unknown just now 1 > C + ⚙

**UC:** Projetos

**Assunto:**

Projeto Pizzaria – Frontend\_02 (NextJS)

Sistema Pizzaria Web

55 – Configurando Alertas

```
C:\ Administrador: Prompt de Comando - □ X ^A

More info: https://sass-lang.com/d/legacy-js-api

Import trace for requested module:
./src/app/dashboard/components/header/styles.module.scss
./src/app/dashboard/components/header/index.tsx

./src/app/dashboard/components/orders/styles.module.scss
Deprecation The legacy JS API is deprecated and will be removed in Dart Sass 2.0.0.

More info: https://sass-lang.com/d/legacy-js-api

Import trace for requested module:
./src/app/dashboard/components/orders/styles.module.scss
./src/app/dashboard/components/orders/index.tsx
./src/app/dashboard/page.tsx

./src/app/dashboard/product/components/form/styles.module.scss
Deprecation The legacy JS API is deprecated and will be removed in Dart Sass 2.0.0.

More info: https://sass-lang.com/d/legacy-js-api

Import trace for requested module:
./src/app/dashboard/product/components/form/styles.module.scss
./src/app/dashboard/product/components/form/index.tsx
GET /favicon.ico 200 in 650ms
Deseja finalizar o arquivo em lotes (S/N)? ^C

c:\Users\Dev\Desktop\Pizzaria\frontend>
```



Administrador: Prompt de Comando

```
c:\Users\Dev\Desktop\Pizzaria\frontend>npm install sonner
```

next-server (v15.0.3)

```
c:\Users\Dev\Desktop\Pizzaria\frontend>npm install sonner
added 1 package, and audited 65 packages in 4s

11 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

c:\Users\Dev\Desktop\Pizzaria\frontend>npm run dev
> frontend@0.1.0 dev
> next dev

  ▲ Next.js 15.0.3
  - Local:          http://localhost:3000

  [ Starting...
  [ Ready in 7.7s
```

The image shows a developer's environment with two main windows: a code editor and a browser.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure. The `src/app/layout.tsx` file is currently selected.
- File List:** Includes `index.tsx`, `layout.tsx`, `page.tsx`, and `styles.module.scss`.
- Content:** The `layout.tsx` file contains TypeScript code for a Next.js layout component. It imports metadata from `next`, a font from `next/font/google`, and a toaster from `sonner`. It defines a `RootLayout` function that returns an `html` element with a `body` class and a `Toaster` component.

**Browser:**

- Title:** Sobre - Sujeito Pizza - A melhor pizzaria
- Header:** Shows navigation icons and a search bar.
- Page Content:** Displays the `SujeitoPizza` logo and a heading `Novo produto`. Below it is a large image of a smoothie bowl with granola, banana, and fruit.
- Form:** A product creation form with fields for `bebidas` (selected), `Nome do produto`, `Preço do produto`, and `Descrição do produto`. A green `Cadastrar produto` button is at the bottom.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** index.tsx (selected), layout.tsx, page.tsx, styles.module.scss.
- Content:** Code for a "Form" component. It includes imports for react, styles, UploadCloud, Image, Button, api, getCookieClient, and toast. It defines interfaces CategoryProps and Props, and an export function Form. The function handles file selection, previewing, and product registration. It also includes validation logic for name, category, price, and description.
- Bottom Status Bar:** Ln 10, Col 1 (30 selected) | Spaces: 4 | UTF-8 | CRLF | TypeScript JSX | Prettier

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** SujeitoPizza (highlighted in red), Categoria, Produto, [Entrar]
- Section:** Novo produto
- Image:** A bowl of smoothie bowl with granola, banana, and fruit, with a small strawberry cup.
- Input Fields:** bebidas, Digite o nome do produto..., Preço do produto..., Digite a descrição do produto...
- Button:** Cadastrar produto

Sujeito Pizza - A melhor pizzaria

frontend

index.tsx layout.tsx page.tsx styles.module.scss ...

src > app > dashboard > product > components > form > index.tsx > Form > handleRegister

```
38     data.append("name", name)
39     data.append("price", price)
40     data.append("description", description)
41     data.append("category_id", categories[Number(categoryIndex)].id)
42     data.append("file", image)

43     const token = getCookieClient();

44     await api.post("/product", data, {
45         headers: {
46             Authorization: `Bearer ${token}`
47         }
48     })
49     .catch((err) => {
50         console.log(err);
51         return;
52     })

53     console.log("CADASTRADO COM SUCESSO!")

54 }

55

56 function handleFile(e: ChangeEvent<HTMLInputElement>){
57     if(e.target.files && e.target.files[0]){
58         const image = e.target.files[0];

59         if(image.type !== "image/jpeg" && image.type !== "image/png"){
60             console.log("FORMATO PROIBIDO!!!")
61             return;
62         }

63         setImage(image);
64     }
65 }
```

Ln 56, Col 4 (39 selected) Spaces: 4 UTF-8 CRLF TypeScript JSX Prettier

EXPLORER FRONTEND .next node\_modules public src app dashboard category components product components form index.tsx styles.module.scss page.tsx layout.tsx page.tsx fonts signup favicon.ico globals.scss layout.tsx page.module.scss page.tsx lib services middleware.ts OUTLINE TIMELINE

1

SujeitoPizza

Categoria Produto [→]

Novo produto



bebidas

Digite o nome do produto...

Preço do produto...

Digite a descrição do produto...

Cadastrar produto

The image shows a developer's environment with a code editor and a browser preview side-by-side.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** index.tsx (selected), layout.tsx, page.tsx, styles.module.scss.
- Content:** TypeScript code for handling product registration. It includes sending data to the API, catching errors, displaying success toast messages, and handling file uploads.

```
index.tsx
data.append("name", name)
data.append("price", price)
data.append("description", description)
data.append("category_id", categories[Number(categoryIndex)].id)
data.append("file", image)

const token = getCookieClient();

await api.post("/product", data, {
  headers: {
    Authorization: `Bearer ${token}`
  }
}).catch((err) => {
  console.log(err);
  return;
})

toast.success("Produto Registrado com Sucesso")

function handleFile(e: ChangeEvent<HTMLInputElement>){
  if(e.target.files && e.target.files[0]){
    const image = e.target.files[0];

    if(image.type !== "image/jpeg" && image.type !== "image/png"){
      console.log("FORMATO PROIBIDO!!!")
      return;
    }

    setImage(image);
  }
}
```

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** Categoria, Produto, [Entrar]
- Section:** Novo produto
- Image:** Acai bowl with fruit and granola.
- Form Fields:**bebidas, Digite o nome do produto..., Preço do produto..., Digite a descrição do produto...
- Button:** Cadastrar produto

The image shows a developer's environment with a code editor and a browser preview side-by-side.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** index.tsx (selected), layout.tsx, page.tsx, styles.module.scss.
- Content:** TypeScript code for product registration. It includes validation logic, setting up a FormData object, sending a POST request to "/product", and handling errors with toast messages.

```
const categoryIndex = formData.get("category")
const name = formData.get("name")
const price = formData.get("price")
const description = formData.get("description")

if(!name || !categoryIndex || !price || !description || !image){
    toast.warning("Preencha todos os campos")
    return;
}

const data = new FormData();

data.append("name", name)
data.append("price", price)
data.append("description", description)
data.append("category_id", categories[Number(categoryIndex)].id)
data.append("file", image)

const token = getCookieClient()

await api.post("/product", data, {
    headers: {
        Authorization: `Bearer ${token}`
    }
})
.catch((err) => {
    console.log(err);
    toast.warning("Falha ao Cadastrar este Produto")
    return;
})

toast.success("Produto Registrado com Sucesso")
```

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** SujeitoPizza (highlighted), Categoria, Produto, [→]
- Section:** Novo produto
- Image:** A bowl of purple smoothie with banana, granola, and strawberries.
- Form Fields:**bebidas  
Digite o nome do produto...  
Preço do produto...  
Digite a descrição do produto...
- Buttons:** Cadastrar produto

frontend

EXPLORER

FRONTEND

- > .next
- > node\_modules
- > public
- src
  - app
    - dashboard
      - > category
    - components
      - > button
      - > header
        - index.tsx
        - styles.module.scss
  - > orders
  - > product
  - layout.tsx
  - page.tsx
  - > fonts
  - > signup
  - favicon.ico
  - globals.scss
  - layout.tsx
  - page.module.scss
  - page.tsx
  - > lib
  - > services
  - TS middleware.ts
- > OUTLINE
- > TIMELINE

index.tsx ...\\form index.tsx ...\\header X layout.tsx page.tsx ...

src > app > dashboard > components > header > index.tsx > Header > handleLogout

```
1 "use client"
2
3 import Link from 'next/link'
4 import styles from './styles.module.scss'
5 import Image from 'next/image'
6 import logoImg from '/public/logo.svg'
7 import { LogOutIcon } from 'lucide-react'
8 import { deleteCookie } from 'cookies-next'
9 import { useRouter } from 'next/navigation'
10 import { toast } from 'sonner'

11 export function Header(){
12     const router = useRouter();
13
14     async function handleLogout(){
15         deleteCookie("session", { path: "/" })
16         toast.success("Logout Realizado com Sucesso!")
17         router.replace("/")
18     }
19
20
21     return(
22         <header className={styles.headerContainer}>
23             <div className={styles.headerContent}>
24                 <Link href="/dashboard">
25                     <Image
26                         alt="Logo Sujeito Pizza"
27                         src={logoImg}
28                         width={190}
29                         height={60}
30                         priority={true}
31                         quality={100}
32                     />
33             </div>
34         </header>
35     )
36 }
```

Spaces: 2 UTF-8 CRLF TypeScript JSX Prettier

Sujeito Pizza - A melhor pizzaria

SujeitoPizza

Categoria Produto [→]

## Novo produto



bebidas

Digite o nome do produto...

Preço do produto...

Digite a descrição do produto...

Cadastrar produto

The image shows a developer's environment with a code editor and a browser preview side-by-side.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code is for a "Header" component. It imports various Next.js and custom modules. It includes a logout function using `useRouter` and `deleteCookie`. The component itself contains an image of the logo and a link to the dashboard.

```
1  "use client"
2
3  import Link from 'next/link'
4  import styles from './styles.module.scss'
5  import Image from 'next/image'
6  import logoImg from '/public/logo.svg'
7  import { LogOutIcon } from 'lucide-react'
8  import { deleteCookie } from 'cookies-next'
9  import { useRouter } from 'next/navigation'
10 import { toast } from 'sonner'
11
12 export function Header(){
13   const router = useRouter();
14
15   async function handleLogout(){
16     deleteCookie("session", { path: "/" })
17     toast.success("Logout Realizado com Sucesso!")
18     router.replace("/")
19   }
20
21   return(
22     <header className={styles.headerContainer}>
23       <div className={styles.headerContent}>
24         <Link href="/dashboard">
25           <Image
26             alt="Logo Sujeito Pizza"
27             src={logoImg}
28             width={190}
29             height={60}
30             priority={true}
31             quality={100}
32           />
33       </div>
34     </header>
35   )
36 }
```

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** Shows the "SujeitoPizza" logo and a "Logout" button.
- Section:** "Novo produto"
- Form:** A form for adding a new product. It has fields for "bebidas", "Digite o nome do produto...", "Preço do produto...", and "Digite a descrição do produto...".
- Buttons:** A green "Cadastrar produto" button at the bottom.

## Novo produto



bebidas

Digite o nome do produto...

Preço do produto...

Digite a descrição do produto...

Cadastrar produto

# SujeitoPizza

Digite seu email...\*\*\*\*\*Acessar

[Não possui uma conta? Cadastre-se](#)

Logout Realizado com Sucesso!

# Sujeito**Pizza**

Não possui uma conta? [Cadastre-se](#)

Sujeito Pizza

Categoria Produto [→]

[→]

**Últimos pedidos** 

Mesa 1

Mesa 1

## Novo produto



bebidas

Digite o nome do produto...

Preço do produto...

Digite a descrição do produto...

**Cadastrar produto**

Abrir

Área de Trabalho > arquivos

Pesquisar em arquivos

Organizar Nova pasta

açaí

hamburger duplo

hamburger

sorvete 1

Categoria Produto [→]

Nome: hamburger duplo Arquivos personalizados

Abrir Cancelar

Digite o nome do produto...

Preço do produto...

Digite a descrição do produto...

Cadastrar produto

## Novo produto



hamburger

Burger Completo Duplo

45

Hambuger Duplo Top

Cadastrar produto

## Novo produto



bebidas

Digite o nome do produto...

Preço do produto...

Digite a descrição do produto...

Cadastrar produto

Produto Registrado com Sucesso

The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (Left):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** Displays the following TypeScript code:

```
1  "use client"
2
3  import { ChangeEvent, useState } from 'react'
4  import styles from './styles.module.scss'
5  import { UploadCloud } from 'lucide-react'
6  import Image from 'next/image'
7  import { Button } from '@/app/dashboard/components/button'
8  import { api } from '@/services/api'
9  import { getCookieClient } from '@/lib/cookieClient'
10 import { toast } from 'sonner'
11 import { useRouter } from 'next/navigation'
12
13 interface CategoryProps{
14     id: string;
15     name: string;
16 }
17
18 interface Props {
19     categories: CategoryProps[]
}
20
21
22 export function Form({ categories }: Props ){
23     const router = useRouter();
24     const [image, setImage] = useState<File>()
25     const [previewImage, setPreviewImage] = useState("")
26
27     async function handleRegisterProduct(formData: FormData) {
28
29         const categoryIndex = formData.get("category")
30         const name = formData.get("name")
31         const price = formData.get("price")
32         const description = formData.get("description")
```

**Browser Preview (Right):**

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** SujeitoPizza
- Navigation:** Categoria | Produto | [Entrar]
- Section:** Últimos pedidos
- Items:** Mesa 10, Mesa 13

The image shows a developer's environment with a code editor and a browser preview side-by-side.

**Code Editor (VS Code):**

- File Structure:** The left sidebar shows a project structure under "FRONTEND".
  - src:** .next, node\_modules, public.
  - app:** dashboard, category, components, button, header, index.tsx, styles.module.scss, orders, product, components, form, index.tsx (selected), styles.module.scss, page.tsx, layout.tsx, page.tsx, fonts, signup, favicon.ico, globals.scss, layout.tsx.
- index.tsx Content:** The selected file contains TypeScript code for a form component. It includes an axios POST request to handle product registration, handling errors, displaying success and warning toast messages, and navigating back to the dashboard. It also includes a function to handle file uploads, validate image formats, set state, and update preview URLs. The code uses hooks like useState and useEffect.

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** SujeitoPizza
- Navigation:** Categoria, Produto, [Entrar]
- Section:** Últimos pedidos
- Content:** Mesa 10, Mesa 13

At the bottom, there are status bars for the code editor and browser, and a system tray with icons for battery, signal, and notifications.

```
51     Authorization: `Bearer ${token}`
52   }
53 }
54 .catch((err) => {
55   console.log(err);
56   toast.warning("Falha ao Cadastrar este Produto")
57   return;
58 })
59
60 toast.success("Produto Registrado com Sucesso")
61 router.push("/dashboard")
62
63 }
64
65 function handleFile(e: ChangeEvent<HTMLInputElement>){
66   if(e.target.files && e.target.files[0]){
67     const image = e.target.files[0];
68
69     if(image.type !== "image/jpeg" && image.type !== "image/png"){
70       console.log("FORMATO PROIBIDO!!!")
71       return;
72     }
73
74     setImage(image);
75     setPreviewImage(URL.createObjectURL(image))
76   }
77 }
78
79
80
81 return(
82   <main className={styles.container}>
```

The image shows a developer's workspace with two main windows: a code editor and a web browser.

**Code Editor (left):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file, showing TypeScript code for a file upload handling function.
- Bottom Status Bar:** Shows file statistics: Ln 70, Col 1 (42 selected), Spaces: 4, UTF-8, CRLF, TypeScript JSX, Prettier.

```
Authorization: `Bearer ${token}`  
}  
).catch((err) => {  
    console.log(err);  
    toast.warning("Falha ao Cadastrar este Produto")  
    return;  
})  
  
toast.success("Produto Registrado com Sucesso")  
router.push("/dashboard")  
  
function handleFile(e: ChangeEvent<HTMLInputElement>){  
    if(e.target.files && e.target.files[0]){  
        const image = e.target.files[0];  
  
        if(image.type !== "image/jpeg" && image.type !== "image/png"){  
            console.log("FORMATO PROIBIDO!!!")  
            return;  
        }  
  
        setImage(image);  
        setPreviewImage(URL.createObjectURL(image))  
    }  
}  
  
return(  
    <main className={styles.container}>
```

**Browser (right):**

- Title:** Sujeito Pizza - A melhor pizzaria
- Content:** Displays the "Últimos pedidos" (Last orders) section, showing "Mesa 10" and "Mesa 13".
- Header:** Categories: Categoria, Produto.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (Left):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** index.tsx (selected), index.tsx ...\\form, index.tsx ...\\header, layout.tsx, page.tsx, ...
- Content:** Code in index.tsx (Line 51 to Line 82). It handles file uploads, sends POST requests to an API endpoint, and displays success or error messages via toast notifications.

```
51     Authorization: `Bearer ${token}`  
52   }  
53 }  
54 .catch((err) => {  
55   console.log(err);  
56   toast.warning("Falha ao Cadastrar este Produto")  
57   return;  
58 })  
59  
60 toast.success("Produto Registrado com Sucesso")  
61 router.push("/dashboard")  
62  
63 }  
64  
65 function handleFile(e: ChangeEvent<HTMLInputElement>){  
66   if(e.target.files && e.target.files[0]){  
67     const image = e.target.files[0];  
68  
69     if(image.type !== "image/jpeg" && image.type !== "image/png"){  
70       toast.warning("Formato não Permitido!")  
71       return;  
72     }  
73  
74     setImage(image);  
75     setPreviewImage(URL.createObjectURL(image))  
76   }  
77 }  
78  
79 }  
80  
81 return(  
82   <main className={styles.container}>
```

**Browser Preview (Right):**

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** SujeitoPizza
- Navigation:** Categoria, Produto, [ ]
- Section:** Últimos pedidos
- Items:** Mesa 10, Mesa 13

The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file, showing TypeScript code for a product creation form. It includes logic for handling file uploads, validating file types, and displaying success/failure messages via toast notifications.
- Bottom Status Bar:** Shows file statistics (Ln 70, Col 1 (47 selected)), code settings (Spaces: 4, CRLF), and build status (TypeScript JSX, Prettier).

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** "SujeitoPizza" logo, navigation links for "Categoria" and "Produto", and a search bar.
- Section:** "Novo produto" (New Product).
- Form Fields:** A large input area with a placeholder "Digite o nome do produto..." and a "Cadastrar produto" button at the bottom.
- Side Panel:** A dropdown menu showing categories: "bebidas" (beverages) and "lanches" (sandwiches). Below the dropdown is a text input field with placeholder "Digite o nome do produto...".

Abrir

Área de Trabalho > arquivos

Pesquisar em arquivos

Organizar Nova pasta

açaí hamburger duplo hamburger logo sorvete 1 video 1

Nome: video 1 Todos os arquivos

Abrir Cancelar

Digite o nome do produto...

Preço do produto...

Digite a descrição do produto...

Cadastrar produto

Categoria Produto [→]

Ero :

Arquivo Novo Abri... Salvar Salvar como Imprimir Excluir Desfazer Refazer Recortar Copiar Colar Cortar Copiar

## Novo produto



bebidas

Digite o nome do produto...

Preço do produto...

Digite a descrição do produto...

Cadastrar produto

Formato não Permitido!

## Novo produto



bebidas

Digite o nome do produto...

Preço do produto...

Digite a descrição do produto...

Cadastrar produto

Abrir

Área de Trabalho > arquivos

Pesquisar em arquivos

Organizar Nova pasta

açaí

hamburger duplo

hamburger triplo

hamburger

sorvete 1

Categoria Produto [→]

Nome: hamburger triplo Arquivos personalizados

Abrir Cancelar

Digite o nome do produto...

Preço do produto...

Digite a descrição do produto...

Cadastrar produto

## Novo produto



hamburger

Burger Triplo Dev

50

Hamburger Triplo com Cheddar

Cadastrar produto

# SujeitoPizza

[Categoria](#) [Produto](#)

## Últimos pedidos

Mesa 10

Mesa 13

Produto Registrado com Sucesso



File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C + Query #1 products [all] categories [1e4ffc20-04b2-4e]

Filter +

ENTITIES 217 +

public

- \_prisma\_migrations
- categories
- items
- orders
- products
- users

information\_schema

pg\_catalog

	id	equals	Enter Value					
1	a9f50e	Pizza de calabre...	46	Pizza de cal...	8a652b025a...	2024-11-25 ...	2024-11-25 22:06:31.016	e21422b1-f695-4426-aa06-9d23c2f3502b
2	d4b0...	Coca lata	5	Coca-Cola la...	765f54ffa7f...	2024-11-25 ...	2024-11-25 22:17:10.368	00d8f79f-5536-429a-bd59-7a82f18700ac
3	fcef7f0	Coca-Cola 2L	10	Coca-Cola g...	b98497f70a...	2024-11-25 ...	2024-11-25 23:47:24.095	00d8f79f-5536-429a-bd59-7a82f18700ac
4	d72f7...	Dev Burger Sala...	40	Burgão Dev ...	c83f6cc7a8...	2024-11-29 ...	2024-11-29 01:06:24.05	1e4ffc20-04b2-46db-96f0-ceea3ca2975a
5	53b8ed	Açaí 300ml	12	Pote de Açaí...	d3c76275c1...	2024-11-29 ...	2024-11-29 01:29:50.485	f44fb594-d1c4-425f-b1bd-87541a6637a1

localhost:5432/pizzaria - Beekeeper Studio

The screenshot shows the Beekeeper Studio interface with the following details:

- File Edit View Ultimate Tools Help** (top menu bar)
- localhost:5432/pizzaria - Beekeeper Studio** (title bar)
- pizzaria** (selected database dropdown)
- Query #1** (query tab)
- products [all]** (entity tab)
- categories [1e4ffc20-04b2-4c]** (entity tab)
- ENTITIES 217** (left sidebar)
- public** (expanded schema)
- \_prisma\_migrations**
- categories**
- items**
- orders**
- products**
- users**
- information\_schema**
- pg\_catalog**

**Query #1 Results:**

	<b>id</b>	<b>equals</b>	Enter Value							
1	7f66d79a-ceb2-4175-b4a6-dc5208a9f50e	Pizza de calabresa	46	Pizza de cal...	8a652b025a...	2024-11-25 ...	2024-11-25 22:06:31.016	e21422b1-		
2	ca5c64fa-e4dd-4b67-9c33-2c4961d4b0...	Coca lata	5	Coca-Cola la...	765f54ffa7f...	2024-11-25 ...	2024-11-25 22:17:10.368	00d8f79f-5		
3	1cd5c28d-6392-4d30-9b2d-91278fccef7f0	Coca-Cola 2L	10	Coca-Cola g...	b98497f70a...	2024-11-25 ...	2024-11-25 23:47:24.095	00d8f79f-5		
4	96eb4477-824b-4137-8251-dbce1d72f7...	Dev Burger Salada	40	Burgão Dev ...	c83f6cc7a8...	2024-11-29 ...	2024-11-29 01:06:24.05	1e4ffc20-0		
5	dc850cd8-a417-4719-9494-ddff3f53b8ed	Açaí 300ml	12	Pote de Açaí...	d3c76275c1...	2024-11-29 ...	2024-11-29 01:29:50.485	f44fb594-c		
6	0cd97aa3-1583-458e-b0ef-549fd1f40e37	Burger Completo Duplo	45	Hambuger D...	040cfab507...	2024-11-29 ...	2024-11-29 01:57:17.362	1e4ffc20-0		
7	6ad3ded6-115b-468c-89eb-25de1a9459...	Burger Triplo Dev	50	Hamburger T...	9b5565412c...	2024-11-29 ...	2024-11-29 02:14:03.556	1e4ffc20-0		

# SujeitoPizza

[Categoria](#) [Produto](#)

## Últimos pedidos

Mesa 10

Mesa 13



# SujeitoPizza

Digite seu email...\*\*\*\*\*Acessar

[Não possui uma conta? Cadastre-se](#)



**UC:** Projetos

**Assunto:**

Projeto Pizzaria – Frontend\_02 (NextJS)

Sistema Pizzaria Web

56 – Buscando Pedidos

# SujeitoPizza

[Categoria](#) [Produto](#) [\[→\]](#)

## Últimos pedidos

Mesa 10

Mesa 13

```
npm run dev
```

```
c:\Users\Dev\Desktop\Pizzaria\backend>npm run dev
```

```
> backend@1.0.0 dev
> ts-node-dev --transpile-only src/server.ts
```

```
[INFO] 23:19:26 ts-node-dev ver. 2.0.0 (using ts-node ver. 10.9.2, typescript ver. 5.6.3)
Servidor online!!!!
```

next-server (v15.0.3) - □ X

```
c:\Users\Dev\Desktop\Pizzaria\frontend>npm run dev

> frontend@0.1.0 dev
> next dev

▲ Next.js 15.0.3
- Local:          http://localhost:3000

[ Starting...
[ Ready in 9s
```

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C + Query #1 products [all] categories [1e4ffc20-04b2-4c... orders [all] +

Filter ENTITIES 217 +

public

- \_prisma\_migrations
- categories
- items
- orders **orders**
- products
- users

information\_schema

pg\_catalog

	id	equals	Enter Value																
	id	text		table	int4(32,0)	status	bool	draft	bool	name	text	created_at...	update_at	timestamp(3)					
1	0d0fef8f-9947-4e0a-94d2-dc2722b9cee0		55			false		false		(NULL)		2024-11-26 ...		2024-11-26 01:30:26.0...					
2	c9019c83-9a02-4152-a475-21f07e1e39...		9			true		false		(NULL)		2024-11-26 ...		2024-11-26 15:36:31.8...					



Search.. Ctrl + P

Invite

Pizzaria ▾



Sujeito Programador ▾

Run

POST ▾ http://localhost:3333/session

Send ▾

200 OK

216 ms

351 B

Just Now ▾

Params

Body ▾

Auth

Headers 4

Scripts

Docs

Preview

Headers 8

Cookies

Tests 0 / 0

→ Mock

Console

JSON ▾

```
1 {  
2   "email": "almir@teste.com",  
3   "password": "123123"  
4 }
```

Preview ▾

```
1 {  
2   "id": "6467d8c9-295d-4ab-8ec0-e036875247d9",  
3   "name": "Almir Ante",  
4   "email": "almir@teste.com",  
5   "token":  
6     "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyYW1lijoiQWxtaXgQW50ZSIiImVtYW1sIjoiYWxtaXJAdGVzdGUuY29tIiwiaWF0IjoxNzMyODQ3MDU2LCJleHAiOjE3MzU0MzkwNTYsInN1YiI6IjY0Njd0KGM5LT15NWQtNDhhYi04ZWwLNUwMZY4NzUyNDdkOSJ9.hlyfiui7-3eqQxHsWx21S71WEmVZ5x4a1RmwEkpw4xA"  
7 }
```



Base Environment

Add Cookies

Add Certificates

Filter

order

product

category

users

POST Criar usuário

POST Logar usuário ▾

GET Detalhes do usuário

master



Beautify JSON

\$.store.books[\*].author



Preferences

Online

Made with ❤ by Kong



Search.. Ctrl + P

Invite

Pizzaria ▾



Sujeito Programador ▾

Run

POST ▾ http://localhost:3333/session

Send ▾

200 OK

216 ms

351 B

1 Minute Ago ▾

Base Environment

Add Cookies

Add Certificates

Filter

order

product

category

users

POST Criar usuário

POST Logar usuário

GET Detalhes do usuário

Params

Body ▾

Auth

Headers ▾

Scripts

Docs

Preview

Headers ▾

Cookies

Tests 0 / 0

→ Mock

Console

JSON ▾

```
1 {  
2   "email": "almir@teste.com",  
3   "password": "123123"  
4 }
```

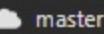
Preview ▾

```
1 {  
2   "id": "6467d8c9-295d-48ab-8ec0-e036875247d9",  
3   "name": "Almir Ante",  
4   "email": "almir@teste.com",  
5   "token":  
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyYW1lijoiQwxtaxIgQW50ZSIiMvTyWL  
sijoiYwxtaXAdGVzdGUuY29tIiwiawF0IjoxNzMyODQ3MDU2LCJleHAiOjE3MzU0MzkwNTYs  
InN1YiI6IjY0NjdkOGM5LTISNWQtNDhhYi04ZWwLWUwMzY4NzUyNDdkOSJ9.hlyfiui7-  
3eqQxHsWx21S71WEmVZ5"  
6 }
```

Search with Google

Copy

Select All



master



Beautify JSON

\$.store.books[\*].author



Preferences

Online

Made with ❤ by Kong



Search.. Ctrl + P

Invite

Pizzaria ▾



Sujeito Programador ▾

Run

POST ▾ http://localhost:3333/order

Send ▾

200 OK

148 ms

174 B

2 Days Ago ▾

Params

Body

Auth

Headers

Scripts

Docs

Preview

Headers

Cookies

Tests 0 / 0

→ Mock

Console

Bearer Token ▾

ENABLED



TOKEN

5LTi5NWQtNDhhYi04ZWMwLWUwMzY4NzUyNDdkOSJ9.hlyfuiu

PREFIX

Bearer

Preview ▾

```
1 {
2   "id": "c9019c83-9a02-4152-a475-21f07e1e39d2",
3   "table": 9,
4   "status": false,
5   "draft": true,
6   "name": null,
7   "created_at": "2024-11-26T15:36:31.834Z",
8   "update_at": "2024-11-26T15:36:31.834Z"
9 }
```

Base Environment

Add Cookies

Add Certificates

Filter

order

PUT Finalizar Pedido (Finish Order)

GET Detalhes de um Pedido (Deta...)

GET Listar Pedidos (List Orders)

PUT Enviar Pedido (Send Order)

POST Adicionar um Item ao Pedido

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa ▾

DEL Deletar uma ORDER - Pedido

product

category

users

master



\$.store.books[\*].author

Preferences

Online

Made with ❤ by Kong



Search.. Ctrl + P

Invite

Pizzaria ▾



Sujeito Programador ▾

Run

POST ▾ http://localhost:3333/order

Send ▾

200 OK

148 ms

174 B

2 Days Ago ▾

Params

Body ▾

Auth ▾

Headers ▾

Scripts

Docs

Preview

Headers ▾

Cookies

Tests 0 / 0

→ Mock

Console

JSON ▾

```
1 {  
2   "table": 35,  
3   "name": "Jorge"  
4 }
```

Preview ▾

```
1 {  
2   "id": "c9019c83-9a02-4152-a475-21f07e1e39d2",  
3   "table": 9,  
4   "status": false,  
5   "draft": true,  
6   "name": null,  
7   "created_at": "2024-11-26T15:36:31.834Z",  
8   "update_at": "2024-11-26T15:36:31.834Z"  
9 }
```

Filter

order

PUT Finalizar Pedido (Finish Order)

GET Detalhes de um Pedido (Deta...)

GET Listar Pedidos (List Orders)

PUT Enviar Pedido (Send Order)

POST Adicionar um Item ao Pedido

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa ▾

DEL Deletar uma ORDER - Pedido

product

category

users

master



Beautify JSON

\$.store.books[\*].author



Preferences

Online

Made with ❤ by Kong



Search.. Ctrl + P

Invite

Pizzaria ▾



Sujeito Programador ▾

Run

POST ▾ http://localhost:3333/order

Send ▾

200 OK

66 ms

178 B

Just Now ▾

Params

Body

Auth

Headers

Scripts

Docs

Preview

Headers

Cookies

Tests 0 / 0

→ Mock

Console

JSON ▾

```
1 {  
2   "table": 35,  
3   "name": "Jorge"  
4 }
```

Preview ▾

```
1 {  
2   "id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",  
3   "table": 35,  
4   "status": false,  
5   "draft": true,  
6   "name": "Jorge",  
7   "created_at": "2024-11-29T02:28:31.944Z",  
8   "update_at": "2024-11-29T02:28:31.944Z"  
9 }
```

Filter

order

PUT Finalizar Pedido (Finish Order)

GET Detalhes de um Pedido (Deta...)

GET Listar Pedidos (List Orders)

PUT Enviar Pedido (Send Order)

POST Adicionar um Item ao Pedido

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa ▾

DEL Deletar uma ORDER - Pedido

product

category

users

master



Beautify JSON

\$.store.books[\*].author



Preferences

Online

Made with ❤ by Kong



Search.. Ctrl + P

Invite

Pizzaria



Sujeito Programador ▾

Run

POST ▾ http://localhost:3333/order

Send ▾

200 OK

66 ms

178 B

1 Minute Ago ▾

Params

Body

Auth

Headers

Scripts

Docs

Preview

Headers

Cookies

Tests 0 / 0

→ Mock

Console

JSON ▾

```

1 ▶ {
2   "table": 35,
3   "name": "Jorge"
4 }
```

Preview ▾

```

1 ▶ {
2   "id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",
3   "table": 35,
4   "status": false,
5   "draft": true,
6   "name": "Jorge",
7   "created_at": "2024-11-29T02:28:28.000Z",
8   "update_at": "2024-11-29T02:28:28.000Z"
9 }
```

Search with Google

Copy

Select All

Base Environment

Add Cookies

Add Certificates

Filter

order

PUT Finalizar Pedido (Finish Order)

GET Detalhes de um Pedido (Deta...)

GET Listar Pedidos (List Orders)

PUT Enviar Pedido (Send Order)

POST Adicionar um Item ao Pedido

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa

DEL Deletar uma ORDER - Pedido

product

category

users

master



Beautify JSON

\$.store.books[\*].author



Preferences

Online

Made with ❤ by Kong



Search.. Ctrl + P

Invite

Pizzaria



Sujeito Programador ▾

Run

POST ▾ http://localhost:3333/order/add

Send ▾

200 OK

115 ms

237 B

2 Days Ago ▾

Params

Body

Auth

Headers

Scripts

Docs

Preview

Headers

Cookies

Tests 0 / 0

→ Mock

Console

JSON ▾

```
1 ▶ {  
2   "order_id": "0d0fef8f-9947-4e0a-94d2-dc2722b9cee0",  
3   "product_id": "7f66d79a-ceb2-4175-b4a6-dc5208a9f50e",  
4   "amount": 3  
5 }
```

Preview ▾

```
1 ▶ {  
2   "id": "0d94f740-43ba-4b43-9663-001465523545",  
3   "amount": 3,  
4   "created_at": "2024-11-26T17:09:39.369Z",  
5   "update_at": "2024-11-26T17:09:39.369Z",  
6   "order_id": "0d0fef8f-9947-4e0a-94d2-dc2722b9cee0",  
7   "product_id": "7f66d79a-ceb2-4175-b4a6-dc5208a9f50e"  
8 }
```

Base Environment

Add Cookies

Add Certificates

Filter

order

PUT Finalizar Pedido (Finish Order)

GET Detalhes de um Pedido (Deta...)

GET Listar Pedidos (List Orders)

PUT Enviar Pedido (Send Order)

POST Adicionar um Item ao Pedido ▾

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa

DEL Deletar uma ORDER - Pedido

product

category

users

master



Beautify JSON

\$.store.books[\*].author



Preferences

Online

Made with ❤ by Kong



Search.. Ctrl + P

Invite

Pizzaria ▾

Sujeito Programador ▾

Run

POST ▾ http://localhost:3333/order/add

Send ▾

200 OK

115 ms

237 B

2 Days Ago ▾

Params

Body

Auth

Headers

4

Scripts

Docs

Preview

Headers

8

Cookies

Tests

0 / 0

→ Mock

Console

JSON ▾

```
1 {  
2   "order_id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",  
3   "product_id": "7f66d79a-ceb2-4175-b4a6-dc5208a9f50e",  
4   "amount": 3  
5 }
```

Preview ▾

```
1 {  
2   "id": "0d94f740-43ba-4b43-9663-001465523545",  
3   "amount": 3,  
4   "created_at": "2024-11-26T17:09:39.369Z",  
5   "update_at": "2024-11-26T17:09:39.369Z",  
6   "order_id": "0d0fef8f-9947-4e0a-94d2-dc2722b9cee0",  
7   "product_id": "7f66d79a-ceb2-4175-b4a6-dc5208a9f50e"  
8 }
```

Base Environment  
Add Cookies  
Add Certificates

Filter

order

PUT Finalizar Pedido (Finish Order)

GET Detalhes de um Pedido (Deta...)

GET Listar Pedidos (List Orders)

PUT Enviar Pedido (Send Order)

POST Adicionar um Item ao Pedido

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa

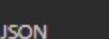
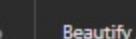
DEL Deletar uma ORDER - Pedido

product

category

users

master



Beautify JSON

\$.store.books[\*].author



Preferences

Online

Made with ❤ by Kong



Search.. Ctrl + P

Invite

Pizzaria ▾



Sujeito Programador ▾

Run

GET ▾ http://localhost:3333/category/product

Send ▾

200 OK

9.37 ms

331 B

3 Days Ago ▾

Params 1

Body

Auth 1

Headers 3

Scripts

Docs

Preview

Headers 8

Cookies

Tests 0 / 0

→ Mock

Console

Bearer Token ▾

ENABLED



TOKEN

nN1Yi6jY0NjdkOGM5LTi5NWQtNDhhYi04ZWMwLWUwMzY4I

Preview ▾

```
1 [  
2 {  
3   "id": "7f66d79a-ceb2-4175-b4a6-dc5208a9f50e",  
4   "name": "Pizza de calabresa",  
5   "price": "46",  
6   "description": "Pizza de calabresa saborosa",  
7   "banner": "8a652b025acaa467024b0a74085c90f9-pizza de calabresa.jpg",  
8   "created_at": "2024-11-25T22:06:31.016Z",  
9   "update_at": "2024-11-25T22:06:31.016Z",  
10  "category_id": "e21422b1-f695-4426-aa06-9d23c2f3502b"  
11 }  
12 ]
```

PREFIX

Bearer

Filter

POST Adicionar um Item ao Pedido

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa

DEL Deletar uma ORDER - Pedido

product

POST Cadastrar Produto

GET Listar Produto por Categoria ▾

category

users

POST Criar usuário

POST Logar usuário

GET Detalhes do usuário

master



\$store.books[\*].author



Preferences

Online

Made with ❤ by Kong

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C +

Query #1 products [all] categories [1e4ffc20-04b2-46db-96f0-ceea3ca2975a] orders [all]

Filter Y

ENTITIES 217 +

public

- \_prisma\_migrations
- categories
- items
- orders
- products
- users

information\_schema

pg\_catalog

id text equals 1e4ffc20-04b2-46db-96f0-ceea3ca2975a

1 1e4ffc20-04b2-46db-96f0-ceea3ca2975a hamburger 2024-11-28 ... 2024-11-28 00:42:45.7...

1e4ffc20-04b2-46db-96f0-ceea3ca2975a

Set as NULL

Copy Ctrl+C

Copy as TSV for Excel

Copy as JSON

Copy as Markdown

Copy as SQL

Paste Ctrl+V

Clone row Ctrl+D

Delete row Delete

localhost:5432/pizzaria postgres Structure # Unknown 1 hour ago 1 > C +



Search.. Ctrl + P

Invite

Pizzaria ▾



Sujeito Programador ▾

Run

GET ▾ http://localhost:3333/category/product

Send ▾

200 OK

9.37 ms

331 B

3 Days Ago ▾

Base Environment

Add Cookies

Add Certificates

Filter

POST Adicionar um Item ao Pedido

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa

DEL Deletar uma ORDER - Pedido

product

POST Cadastrar Produto

GET Listar Produto por Categoria

category

users

POST Criar usuário

POST Logar usuário

GET Detalhes do usuário

master



Preview

Headers ▾

Cookies

Tests 0 / 0

→ Mock

Console

URL PREVIEW

http://localhost:3333/category/product?category\_id=1e4ffc20-04b2-46db-96f0-ceea3ca2975a

QUERY PARAMETERS

Import from URL Bulk Edit

+ Add

Delete all

Description

category\_id

1e4ffc20-04b2-46db-96f0-ceea: ▾

PATH PARAMETERS

Path parameters are url path segments that start with a colon ':' e.g. ':id' X

Preview ▾

```
1 [  
2 {  
3   "id": "7f66d79a-ceb2-4175-b4a6-dc5208a9f50e",  
4   "name": "Pizza de calabresa",  
5   "price": "46",  
6   "description": "Pizza de calabresa saborosa",  
7   "banner": "8a652b025acaa467024b0a74085c90f9-pizza de calabresa.jpg",  
8   "created_at": "2024-11-25T22:06:31.016Z",  
9   "update_at": "2024-11-25T22:06:31.016Z",  
10  "category_id": "e21422b1-f695-4426-aa06-9d23c2f3502b"  
11 }  
12 ]
```

\$.store.books[\*].author



Preferences

Online

Made with ❤ by Kong



Search.. Ctrl + P

Invite

Pizzaria ▾



Sujeito Programador ▾

Run

GET ▾ http://localhost:3333/category/product

Send ▾

200 OK

112 ms

987 B

Just Now ▾

Base Environment

Add Cookies

Add Certificates

Filter

POST Adicionar um Item ao Pedido

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa

DEL Deletar uma ORDER - Pedido

product

POST Cadastrar Produto

GET Listar Produto por Categoria

category

users

POST Criar usuário

POST Logar usuário

GET Detalhes do usuário

master



Params 1

Body

Auth 1

Headers 3

Scripts

Docs

URL PREVIEW

http://localhost:3333/category/product?category\_id=1e4ffc20-04b2-46db-96f0-ceea3ca2975a

QUERY PARAMETERS

Import from URL Bulk Edit

+ Add Delete all Description

category\_id

1e4ffc20-04b2-46db-96f0-ceea3ca2975a

PATH PARAMETERS

Path parameters are url path segments that start with a colon ':' e.g. ':id'

X

Preview

Headers 8

Cookies

Tests 0 / 0

→ Mock

Console

Preview ▾

```

1 [ 
2 { 
3   "id": "96eb4477-824b-4137-8251-dbce1d72f755",
4   "name": "Dev Burger Salada",
5   "price": "40",
6   "description": "Burgão Dev de Salada com 1 hambúrguer de 160g",
7   "banner": "c83f6cc7a8b3dea100c14366e0808e99-hamburger.jpg",
8   "created_at": "2024-11-29T01:06:24.050Z",
9   "update_at": "2024-11-29T01:06:24.050Z",
10  "category_id": "1e4ffc20-04b2-46db-96f0-ceea3ca2975a"
11 },
12 { 
13   "id": "0cd97aa3-1583-458e-b0ef-549fd1f40e37",
14   "name": "Burger Completo Duplo",
15   "price": "45",
16   "description": "Hambúrguer Duplo Top",
17   "banner": "040cfab507866ce68a0ce7fdc60e8cc8-hamburger duplo.jpg",
18   "created_at": "2024-11-29T01:57:17.362Z",
19   "update_at": "2024-11-29T01:57:17.362Z",
20   "category_id": "1e4ffc20-04b2-46db-96f0-ceea3ca2975a"
21 },
22 { 
23   "id": "6ad3ded6-115b-468c-89eb-25de1a945944",
24   "name": "Burger Triplo Dev",
25   "price": "50",
26   "description": "Hambúrguer Triplo com Cheddar",
27   "banner": "9b5565412cabf8912fd21c9c26cfaca6-hamburger triplo.jpg",
28   "created_at": "2024-11-29T02:14:03.556Z",
29   "update_at": "2024-11-29T02:14:03.556Z",

```

\$store.books[\*].author

Preferences

Online

Made with ❤ by Kong



Search.. Ctrl + P

Invite

Pizzaria ▾



Sujeito Programador ▾

Run

GET ▾ http://localhost:3333/category/product

Send ▾

200 OK

112 ms

987 B

2 Minutes Ago ▾

Params 1

Body

Auth 1

Headers 3

Scripts

Docs

Preview

Headers 8

Cookies

Tests 0 / 0

→ Mock

Console

No Body ▾

Preview ▾

```

1  [
2  {
3      "id": "96eb4477-824b-4137-8251-dbce1d72f755",
4      "name": "Dev Burger Salada",
5      "price": "40",
6      "description": "Burgão Dev de Salada com 1 hambúrguer de 160g",
7      "banner": "c83f6cc7a8b3dea100c14366e0808e99-hamburger.jpg",
8      "created_at": "2024-11-29T01:06:24.050Z",
9      "update_at": "2024-11-29T01:06:24.050Z",
10     "category_id": "1e4ffc20-04b2-46db-96f0-ceea3ca2975a"
11 },
12 {
13     "id": "0cd97aa3-1583-458e-b0ef-549fd1f40e37",
14     "name": "Burger Completo Duplo",
15     "price": "45",
16     "description": "Hambúrguer Duplo Top",
17     "banner": "040cfab507866ce68a0ce7fdc60e8cc8-hamburger duplo.jpg",
18     "created_at": "2024-11-29T01:57:17.362Z",
19     "update_at": "2024-11-29T01:57:17.362Z",
20     "category_id": "1e4ffc20-04b2-46db-96f0-ceea3ca2975a"
21 },
22 {
23     "id": "6ad3ded6-115b-468c-89eb-25de1a945944",
24     "name": "Burger Triplo Dev",
25     "price": "50",
26     "description": "Hambúrguer Triplo com 3 hambúrgueres de 160g",
27     "banner": "9b5565412cabf8912fd21c9c26-hamburger triplo.jpg",
28     "created_at": "2024-11-29T02:14:03.556Z",
29     "update_at": "2024-11-29T02:14:03.556Z"

```

Search with Google

Copy

Select All

\$.store.books[\*].author



Enter a URL and send to get a response

Select a body type from above to send data in the body of a request

Introduction to Insomnia ▾

Base Environment

Add Cookies

Add Certificates

Filter

POST Adicionar um Item ao Pedido

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa

DEL Deletar uma ORDER - Pedido

product

POST Cadastrar Produto

GET Listar Produto por Categoria

category

users

POST Criar usuário

POST Logar usuário

GET Detalhes do usuário

master



Preferences

Online

Made with ❤ by Kong



Search.. Ctrl + P

Invite

Pizzaria ▾



Sujeito Programador ▾

Run

POST ▾ http://localhost:3333/order/add

Send ▾

200 OK

115 ms

237 B

2 Days Ago ▾

Params

Body ▾

Auth ▾

Headers ▾

Scripts

Docs

Preview

Headers ▾

Cookies

Tests 0 / 0

→ Mock

Console

JSON ▾

```
1 <pre>{
2   "order_id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",
3   "product_id": "6ad3ded6-115b-468c-89eb-25de1a945944",
4   "amount": 3
5 }</pre>
```

Preview ▾

```
1 <pre>{
2   "id": "0d94f740-43ba-4b43-9663-001465523545",
3   "amount": 3,
4   "created_at": "2024-11-26T17:09:39.369Z",
5   "update_at": "2024-11-26T17:09:39.369Z",
6   "order_id": "0d0fef8f-9947-4e0a-94d2-dc2722b9cee0",
7   "product_id": "7f66d79a-ceb2-4175-b4a6-dc5208a9f50e"
8 }</pre>
```

Base Environment

Add Cookies

Add Certificates

Filter

order

PUT Finalizar Pedido (Finish Order)

GET Detalhes de um Pedido (Deta...)

GET Listar Pedidos (List Orders)

PUT Enviar Pedido (Send Order)

POST Adicionar um Item ao Pedido

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa

DEL Deletar uma ORDER - Pedido

product

POST Cadastrar Produto

GET Listar Produto por Categoria

master



Beautify JSON

\$.store.books[\*].author



Preferences

Online

Made with ❤ by Kong



Search.. Ctrl + P

Invite

Pizzaria



Sujeito Programador ▾

Run

POST ▾ http://localhost:3333/order/add

Send ▾

200 OK

115 ms

237 B

2 Days Ago ▾

Params

Body

Auth

Headers

Scripts

Docs

Preview

Headers

Cookies

Tests 0 / 0

→ Mock

Console

JSON ▾

```

1 {
2   "order_id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",
3   "product_id": "6ad3ded6-115b-468c-89eb-25de1a945944",
4   "amount": 2
5 }
```

```

1 {
2   "id": "0d94f740-43ba-4b43-9663-001465523545",
3   "amount": 3,
4   "created_at": "2024-11-26T17:09:39.369Z",
5   "update_at": "2024-11-26T17:09:39.369Z",
6   "order_id": "0d0fefef8f-9947-4e0a-94d2-dc2722b9ce0",
7   "product_id": "7f66d79a-ceb2-4175-b4a6-dc5208a9f50e"
8 }
```

Filter

order

PUT Finalizar Pedido (Finish Order)

GET Detalhes de um Pedido (Deta...)

GET Listar Pedidos (List Orders)

PUT Enviar Pedido (Send Order)

POST Adicionar um Item ao Pedido ▾

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa

DEL Deletar uma ORDER - Pedido

product

POST Cadastrar Produto

GET Listar Produto por Categoria

master



Beautify JSON

\$.store.books[\*].author



Preferences

Online

Made with ❤ by Kong



Search.. Ctrl + P

Invite

Pizzaria ▾



Sujeito Programador ▾

Run

POST ▾ http://localhost:3333/order/add

Send ▾

200 OK

37 ms

237 B

Just Now ▾

Base Environment

Add Cookies

Add Certificates

Filter

order

PUT Finalizar Pedido (Finish Order)

GET Detalhes de um Pedido (Deta...)

GET Listar Pedidos (List Orders)

PUT Enviar Pedido (Send Order)

POST Adicionar um Item ao Pedido ▾

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa

DEL Deletar uma ORDER - Pedido

product

POST Cadastrar Produto

GET Listar Produto por Categoria

master



Beautify JSON

\$.store.books[\*].author



Preferences

Online

Made with ❤ by Kong



Search.. Ctrl + P

Invite

Pizzaria ▾



Sujeito Programador ▾

Run

POST ▾ http://localhost:3333/order

Send ▾

200 OK

66 ms

178 B

22 Minutes Ago ▾

Params

Body

Auth

Headers

Scripts

Docs

Preview

Headers

Cookies

Tests 0 / 0

→ Mock

Console

JSON ▾

```

1 < {
2   "table": 35,
3   "name": "Jorge"
4 }

```

Preview ▾

```

1 < {
2   "id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",
3   "table": 35,
4   "status": false,
5   "draft": true,
6   "name": "Jorge",
7   "created_at": "2024-01-11T10:00:00Z",
8   "update_at": "2024-01-11T10:00:00Z"
9 }

```

Search with Google  
Copy  
Select All

Filter

order

PUT Finalizar Pedido (Finish Order)

GET Detalhes de um Pedido (Deta...)

GET Listar Pedidos (List Orders)

PUT Enviar Pedido (Send Order)

POST Adicionar um Item ao Pedido

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa

DEL Deletar uma ORDER - Pedido

product

POST Cadastrar Produto

GET Listar Produto por Categoria

master



Beautify JSON

\$.store.books[\*].author



Preferences

Online

Made with ❤ by Kong



Search.. Ctrl + P

Invite

Pizzaria



Sujeito Programador ▾

Run

PUT ▾ http://localhost:3333/order/send

Send ▾

200 OK

95.6 ms

175 B

2 Days Ago ▾

Params 1

Body 1

Auth 1

Headers 4

Scripts

Docs

Preview

Headers 8

Cookies

Tests 0 / 0

→ Mock

Console

JSON ▾

```
1 {  
2   "order_id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3"  
3 }
```

Preview ▾

```
1 {  
2   "id": "c9019c83-9a02-4152-a475-21f07e1e39d2",  
3   "table": 9,  
4   "status": false,  
5   "draft": false,  
6   "name": null,  
7   "created_at": "2024-11-26T15:36:31.834Z",  
8   "update_at": "2024-11-26T15:36:31.834Z"  
9 }
```

Filter

order

PUT Finalizar Pedido (Finish Order)

GET Detalhes de um Pedido (Deta...)

GET Listar Pedidos (List Orders)

PUT Enviar Pedido (Send Order) ▾

POST Adicionar um Item ao Pedido

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa

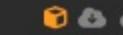
DEL Deletar uma ORDER - Pedido

product

POST Cadastrar Produto

GET Listar Produto por Categoria

master



Beautify JSON

\$.store.books[\*].author



Preferences

Online

Made with ❤ by Kong



Search.. Ctrl + P

Invite

Pizzaria



Sujeito Programador ▾

Run

PUT ▾ http://localhost:3333/order/send

Send ▾

200 OK

121 ms

179 B

Just Now ▾

Base Environment

Add Cookies

Add Certificates

Filter

order

PUT Finalizar Pedido (Finish Order)

GET Detalhes de um Pedido (Deta...)

GET Listar Pedidos (List Orders)

PUT Enviar Pedido (Send Order) ▾

POST Adicionar um Item ao Pedido

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa

DEL Deletar uma ORDER - Pedido

product

POST Cadastrar Produto

GET Listar Produto por Categoria

master



Beautify JSON

Preview

Headers ▾

Cookies

Tests 0 / 0

→ Mock

Console

Preview ▾

```
1 {  
2   "order_id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3"  
3 }
```

```
1 {  
2   "id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",  
3   "table": 35,  
4   "status": false,  
5   "draft": false,  
6   "name": "Jorge",  
7   "created_at": "2024-11-29T02:28:31.944Z",  
8   "update_at": "2024-11-29T02:28:31.944Z"  
9 }
```

\$.store.books[\*].author



Preferences

Online

Made with ❤ by Kong

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C +

Query #1 products [all] categories [1e4ffc20-04b2-4c... orders [all] x + 🔍

Filter Y

ENTITIES 217 +

- public
  - \_prisma\_migrations
  - categories
  - items
  - orders**
  - products
  - users
- information\_schema
- pg\_catalog

	id	equals	Enter Value				
1	Od0fef8f-9947-4e0a-94d2-dc2722b9cee0	55	false	false	(NULL)	2024-11-26 ...	2024-11-26 01:30:26.0...
2	c9019c83-9a02-4152-a475-21f07e1e39...	9	true	false	(NULL)	2024-11-26 ...	2024-11-26 15:36:31.8...

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C +

Query #1 products [all] categories [1e4ffc20-04b2-4c] orders [all] x +

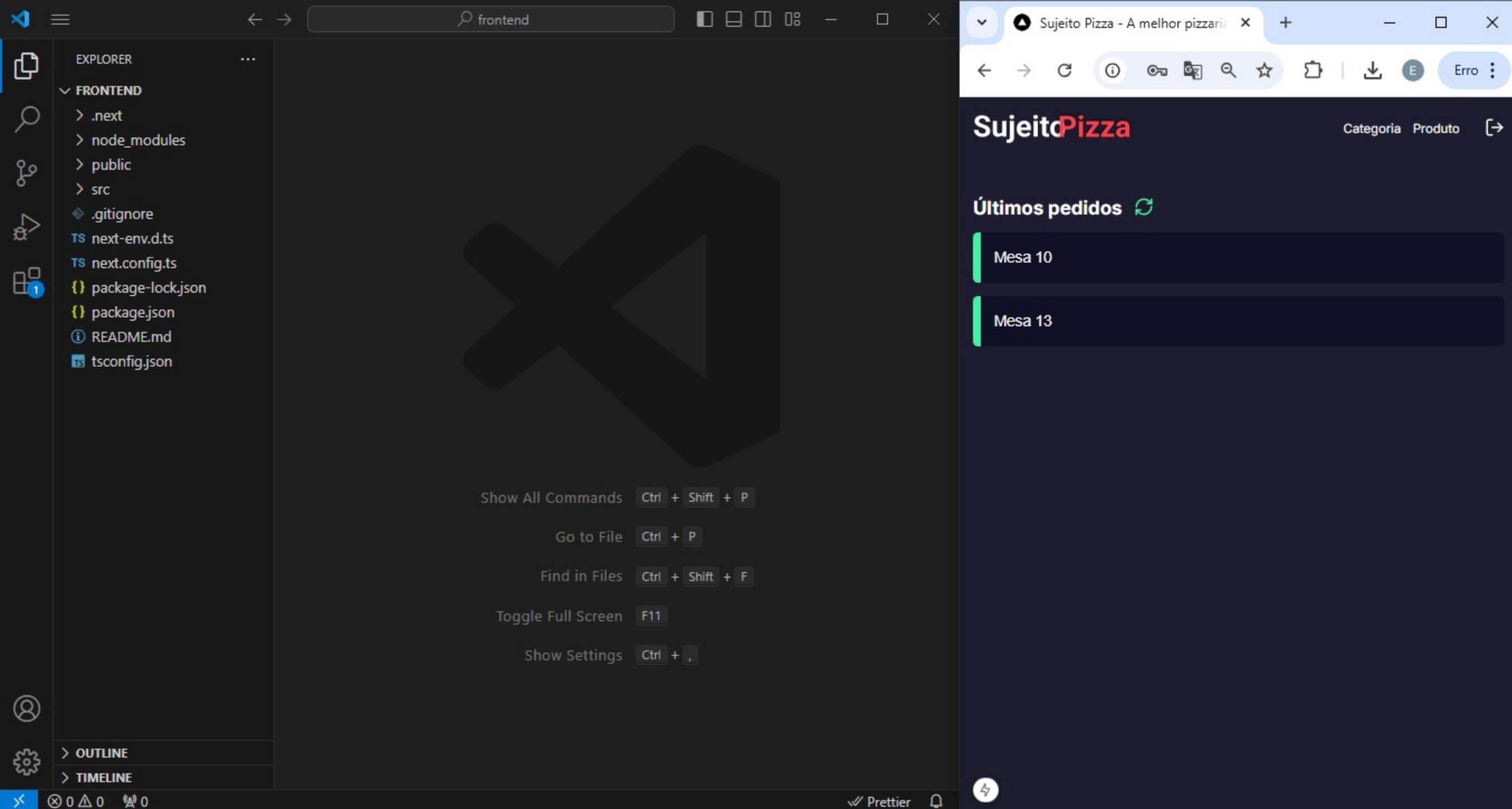
Filter Y +

ENTITIES 217 +

- public
  - \_prisma\_migrations
  - categories
  - items
  - orders
  - products
  - users
- information\_schema
- pg\_catalog

	id	equals	Enter Value	table	int4(32,0)	status	bool	draft	bool	name	text	created_at...	update_at	timestamp(3)
1	0d0fef8f-9947-4e0a-94d2-dc2722b9cee0	55			55	false	false	false	(NULL)			2024-11-26 ...	2024-11-26 01:30:26.0...	
2	c9019c83-9a02-4152-a475-21f07e1e39...	9			9	true	false	false	(NULL)			2024-11-26 ...	2024-11-26 15:36:31.8...	
3	ee42b24f-cdc5-48fd-87d3-1758a06afae3	35			35	false	false	false	Jorge			2024-11-29 ...	2024-11-29 02:28:31.9...	

Refresh Table (ctrl+r or F5)



The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "page.tsx" is the active file.
- Content:** The code is a TypeScript component named "Dashboard". It imports "Orders" and "api" from local modules, and "getCookieServer" from a service. It defines an asynchronous function "getOrders" that uses "await" to get orders from the API, setting a header with a Bearer token. It returns an empty array if there's an error. The component then renders an "Orders" component.
- Status Bar:** Shows line 1, column 1 (540 selected), spaces: 2, and encoding: UTF-8.

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** Includes links for "Categoria", "Produto", and a search bar.
- Section:** "Últimos pedidos" (Last orders) displays two items:
  - Mesa 10
  - Mesa 13
- Status Bar:** Shows a lightning bolt icon, indicating the page is loading or has been updated.



Search.. Ctrl + P

Invite

Pizzaria



Sujeito Programador ▾

Run

POST ▾ http://localhost:3333/session

Send ▾

200 OK

216 ms

351 B

41 Minutes Ago ▾

Base Environment

Add Cookies

Add Certificates

Filter

GET Listar Pedidos (List Orders)

PUT Enviar Pedido (Send Order)

POST Adicionar um Item ao Pedido

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa

DEL Deletar uma ORDER - Pedido

product

category

users

POST Criar usuário

POST Logar usuário

GET Detalhes do usuário

master



Preferences

Params

Body ●

Auth

Headers 4

Scripts

Docs

Preview

Headers 8

Cookies

Tests 0 / 0

→ Mock

Console

URL PREVIEW

http://localhost:3333/session



QUERY PARAMETERS

Import from URL ● Bulk Edit

+ Add

Delete all

Description

name

value

PATH PARAMETERS

Path parameters are url path segments that start with a colon ':' e.g. ':id'



```

1 {
2   "id": "6467d8c9-295d-4ab-8ec0-e036875247d9",
3   "name": "Almir Ante",
4   "email": "almir@teste.com",
5   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1IjoiQWxtaXgQW50ZSIiMvTyWI
6   sIjoiYWxtaXJAdGVzdGUuY29tIiwiZWFOIjoxNzMyODQ3MDU2LCJleHAiOjE3MzU0MzkwNTYs
7   InN1YiI6IjY0Njk0GM5LT15NWQtNDhhYi04ZNMwLWUuMzY4NzUyNDdkOSJ9.hlyfiui7-
8   BeqQxHsWx21S71WEmVZ5x4a1RmwEkpw4xA"
9 }
```

\$.store.books[\*].author



Online

Made with ❤ by Kong



Search.. Ctrl + P

Invite

Pizzaria

Sujeito Programador ▾

Run

POST ▾ http://localhost:3333/session

Send ▾

200 OK

216 ms

351 B

41 Minutes Ago ▾

Base Environment

Add Cookies

Add Certificates

Filter

GET Listar Pedidos (List Orders)

PUT Enviar Pedido (Send Order)

POST Adicionar um Item ao Pedido

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa

DEL Deletar uma ORDER - Pedido

product

category

users

POST Criar usuário

POST Logar usuário

GET Detalhes do usuário

master



Preview

Headers

Cookies

Tests 0 / 0

→ Mock

Console

Preview ▾

```

1 {
2   "id": "6467d8c9-295d-4ab-8ec0-e036875247d9",
3   "name": "Almir Ante",
4   "email": "almir@teste.com",
5   "token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJW1lIjoiQwxtAXIgQW50ZSIiMvTyW1
SIjoiYwxtaxJAdGVzdGUuY29tIiwiZWFOIjoxNzMyODQ3MDU2LCJleHAIoje3MzU0MzkwNTYs
InN1YiI6IjY0Njd0KGM5LT15NWQtNDhhYi04ZWmwlWUuMzY4NzUyNDdk0SJ9.hlyfiu7-
3eqQxHsWx21571WEmVZ5x4a1RmWE"
6 }
```

Search with Google

Copy

Select All

\$.store.books[\*].author



Preferences

Online

Made with ❤ by Kong



Search.. Ctrl + P

Invite

Pizzaria ▾



Sujeito Programador ▾

Run

GET ▾ http://localhost:3333/orders

Send ▾

200 OK

158 ms

178 B

2 Days Ago ▾

Params 1

Body

Auth 

Headers 3

Scripts

Docs

Preview

Headers 8

Cookies

Tests 0 / 0

→ Mock

Console

Bearer Token ▾

ENABLED 

TOKEN

nN1Yil6ljY0NjdlOGM5LTi5NWQtNDhhYi04ZWMyLWUwMzY4! 🔍

PREFIX

Bearer

Preview ▾

```
1 [  
2 {  
3   "id": "0d0fef8f-9947-4e0a-94d2-dc2722b9cee0",  
4   "table": 55,  
5   "status": false,  
6   "draft": false,  
7   "name": null,  
8   "created_at": "2024-11-26T01:30:26.021Z",  
9   "update_at": "2024-11-26T01:30:26.021Z"  
10 }  
11 ]
```

Base Environment

Add Cookies

Add Certificates

Filter

order

PUT Finalizar Pedido (Finish Order)

GET Detalhes de um Pedido (Deta...)

GET Listar Pedidos (List Orders) ▾

PUT Enviar Pedido (Send Order)

POST Adicionar um Item ao Pedido

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa

DEL Deletar uma ORDER - Pedido

product

category

users

master



\$.store.books[\*].author

Preferences

Online

Made with ❤ by Kong



Search.. Ctrl + P

Invite

Pizzaria ▾



Sujeito Programador ▾

Run

GET ▾ http://localhost:3333/orders

Send ▾

200 OK

118 ms

358 B

Just Now ▾

Base Environment

Add Cookies

Add Certificates

Filter

order

PUT Finalizar Pedido (Finish Order)

GET Detalhes de um Pedido (Deta...)

GET Listar Pedidos (List Orders) ▾

PUT Enviar Pedido (Send Order)

POST Adicionar um Item ao Pedido

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa

DEL Deletar uma ORDER - Pedido

product

category

users

master



Params 1

Body

Auth 1

Headers 3

Scripts

Docs

Preview

Headers 8

Cookies

Tests 0 / 0

→ Mock

Console

Bearer Token ▾

ENABLED



TOKEN

.....

PREFIX

Bearer

Preview ▾

```
1 [  
2 {  
3   "id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",  
4   "table": 35,  
5   "status": false,  
6   "draft": false,  
7   "name": "Jorge",  
8   "created_at": "2024-11-29T02:28:31.944Z",  
9   "update_at": "2024-11-29T02:28:31.944Z"  
10 },  
11 {  
12   "id": "0d0fef8f-9947-4e0a-94d2-dc2722b9ceea",  
13   "table": 55,  
14   "status": false,  
15   "draft": false,  
16   "name": null,  
17   "created_at": "2024-11-26T01:30:26.021Z",  
18   "update_at": "2024-11-26T01:30:26.021Z"  
19 }  
20 ]
```

\$.store.books[\*].author



Preferences

Online

Made with ❤ by Kong

The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
  - .next
  - node\_modules
  - public
  - src
    - app
    - lib
      - cookieClient.ts
      - cookieServer.ts
- File List:** Shows files and folders in the current directory:
  - page.tsx
  - TS order.type.ts
  - src > lib > TS order.type.ts
  - 1
  - services
  - TS middleware.ts
  - ↳ .gitignore
  - TS next-env.d.ts
  - TS next.config.ts
  - { package-lock.json
  - { package.json
  - ⓘ README.md
  - tsconfig.json

Sujeito Pizza - A melhor pizzaria

SujeitoPizza

Categoria Produto [→]

Últimos pedidos ⟳

- Mesa 10
- Mesa 13

The image shows a developer's workspace with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure with files like `page.tsx`, `TS order.type.ts`, `cookieClients.ts`, `cookieServer.ts`, and `order.type.ts`.
- File `TS order.type.ts`:** Contains the following TypeScript code:

```
1 export interface OrderProps{  
2   id: string;  
3   table: number;  
4   name: string;  
5   draft: boolean;  
6   status: boolean;  
7 }
```

**Browser Preview:**

The browser window displays the application "Sujeito Pizza - A melhor pizzaria".

- Header:** Shows "SujeitoPizza" with navigation links for "Categoria" and "Produto".
- Section "Últimos pedidos":** Displays two recent orders:
  - Mesa 10
  - Mesa 13

At the bottom of the browser window, there are standard browser controls (back, forward, search, etc.) and a "Erro" button.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

**Code Editor (VS Code):**

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "page.tsx" is the active file.
- Content:** The code implements an asynchronous function `getOrders` that fetches data from an API using `await`. It handles errors and returns an empty array if no data is found. It also exports a `Dashboard` component.

```
src > app > dashboard > page.tsx > getOrders
1 import { Orders } from "./components/orders"
2 import { api } from "@/services/api"
3 import { getCookieServer } from "@/lib/cookieServer";
4 import { OrderProps } from "@/lib/order.type";
5
6 async function getOrders(): Promise<OrderProps[] | []> {
7   try{
8     const token = await getCookieServer();
9
10    const response = await api.get("/orders", {
11      headers: {
12        Authorization: `Bearer ${token}`
13      }
14    })
15
16    return response.data || [];
17  }catch(err){
18    console.log(err);
19    return [];
20  }
21
22 }
23
24 export default function Dashboard(){
25   return(
26     <>
27       <Orders/>
28     </>
29   )
30 }
```

**Browser Preview:**

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** Shows "Categoria" and "Produto" buttons.
- Section:** "Últimos pedidos" displays two recent orders: "Mesa 10" and "Mesa 13".

frontend

Sujeito Pizza - A melhor pizzaria

EXPLORER

FRONTEND

src > app > dashboard > page.tsx > Dashboard

```
  6  async function getOrders(): Promise<OrderProps[] | []> {
  7    try{
  8      const token = await getCookieServer();
  9
 10     const response = await api.get("/orders", {
 11       headers: {
 12         Authorization: `Bearer ${token}`
 13       }
 14     })
 15
 16     return response.data || [];
 17   }catch(err){
 18     console.log(err);
 19     return [];
 20   }
 21 }
 22
 23
 24 export default async function Dashboard(){
 25
 26   const orders = await getOrders();
 27
 28   console.log(orders);
 29
 30   return(
 31     <>
 32       <Orders/>
 33     </>
 34   )
 35 }
```

Categoria Produto [→]

Últimos pedidos

Mesa 10

Mesa 13

Editar

Últimos pedidos

Mesa 10

Mesa 13

Novo produto

1440.jpg

Nome: suave

Preço: R\$ 300,00

Descrição: Pote de Açaí 380ML

Cadastrar pedido

Pesquisar

0 0 0 0

2 selections (66 characters selected) Spaces: 2 UTF-8 CRLF TypeScript JSX Prettier

URGENTE 00:15 29/11/2024

00:15 29/11/2024

next-server (v15.0.3) - More info: <https://sass-lang.com/d/legacy-js-api>

```
Import trace for requested module:  
./src/app/dashboard/components/orders/styles.module.scss  
./src/app/dashboard/components/orders/index.tsx  
./src/app/dashboard/page.tsx  
[  
 {  
   id: 'ee42b24f-cdc5-48fd-87d3-1758a06afae3',  
   table: 35,  
   status: false,  
   draft: false,  
   name: 'Jorge',  
   created_at: '2024-11-29T02:28:31.944Z',  
   update_at: '2024-11-29T02:28:31.944Z'  
 },  
 {  
   id: '0d0fef8f-9947-4e0a-94d2-dc2722b9cee0',  
   table: 55,  
   status: false,  
   draft: false,  
   name: null,  
   created_at: '2024-11-26T01:30:26.021Z',  
   update_at: '2024-11-26T01:30:26.021Z'  
 }]  
GET /dashboard 200 in 9465ms  
<w> [webpack.cache.PackFileCacheStrategy/webpack.FileSystemInfo] Resolving '.....typescript/lib/typescript' in c:\Users\Dev\Desktop\Pizzaria\frontend\node_modules\next\dist\build for build dependencies doesn't lead to expected result 'C:\
```

```
next-server (v15.0.3)

./src/app/dashboard/components/orders/index.tsx
./src/app/dashboard/page.tsx
GET /favicon.ico 200 in 7082ms
true
[
  {
    id: 'ee42b24f-cdc5-48fd-87d3-1758a06afae3',
    table: 35,
    status: false,
    draft: false,
    name: 'Jorge',
    created_at: '2024-11-29T02:28:31.944Z',
    update_at: '2024-11-29T02:28:31.944Z'
  },
  {
    id: '0d0fef8f-9947-4e0a-94d2-dc2722b9cee0',
    table: 55,
    status: false,
    draft: false,
    name: null,
    created_at: '2024-11-26T01:30:26.021Z',
    update_at: '2024-11-26T01:30:26.021Z'
  }
]
GET /dashboard 200 in 350ms
GET /favicon.ico 200 in 16ms
```

