

**UC:** Projetos

**Assunto:**

Projeto Pizzaria – Integração

Preparando para Deploy

65 – Usando Storage para Upload de Imagens

# SujeitoPizza

[Categoria](#) [Produto](#) [→]

## Últimos pedidos



Mesa 35

## Novo produto

**bebidas**

Digite o nome do produto...

Preço do produto...

Digite a descrição do produto...

**Cadastrar produto**

EXPLORER

... ↻

BACKEND

node\_modules

prisma

src

tmp

- 4f039587d393ea86da3b98f3baa23b...
- 8a652b025acaa467024b0a74085c90f9...
- 9b5565412cabf8912fd21c9c26cfaca...
- 040cfab507866ce68a0ce7fdc60e8cc...
- 765f54ffa7f05a9ff902a1a4b85f0e3b...
- b98497f70a15c4d47f7f2209a61bffc1...
- c83f6cc7a8b3dea100c14366e0808e9...
- d3c76275c1e9e682c497a354bd3285...

> tmp

.env

.gitignore

{ package.json

tsconfig.json

≡ yarn-error.log

🔒 yarn.lock

OUTLINE

TIMELINE

8a652b025acaa467024b0a74085c90f9-pizza de calabresa.jpg X

tmp > 8a652b025acaa467024b0a74085c90f9-pizza de calabresa.jpg



pizzaria

Query #1 products [all]

Filter

ENTITIES 217

public

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

information\_schema

pg\_catalog

	id	name	price	description	banner	created_at	update_at	cat
1	0cd97aa3-1583-458e-b0ef-549fd1f40e37	Burger Completo Duplo	45	Hambuger Dupl...	040cfab507...	2024-11-29 ...	2024-11-29 01:57:17.362	1e4ffd
2	1cd5c28d-6392-4d30-9b2d-91278fce7f0	Coca-Cola 2L	10	Coca-Cola gelad...	b98497f70a...	2024-11-25 ...	2024-11-25 23:47:24.095	00d8f
3	37e1bb99-2313-4a82-98c3-769d93409a...	Açaí 500ml	30	pote de açaí de ...	4f039587d3...	2024-12-05 ...	2024-12-05 00:34:31.367	80de4
4	6ad3ded6-115b-468c-89eb-25de1a9459...	Burger Triplo Dev	50	Hamburger Tripl...	9b5565412c...	2024-11-29 ...	2024-11-29 02:14:03.556	1e4ffd
5	7f66d79a-ceb2-4175-b4a6-dc5208a9f50e	Pizza de calabresa	46	Pizza de calabre...	8a652b025a...	2024-11-25 ...	2024-11-25 22:06:31.016	e2142
6	96eb4477-824b-4137-8251-dbce1d72f7...	Dev Burger Salada	40	Burgão Dev de S...	c83f6cc7a8b...	2024-11-29 ...	2024-11-29 01:06:24.05	1e4ffd
7	ca5c64fa-e4dd-4b67-9c33-2c4961d4b0...	Coca lata	5	Coca-Cola lata 3...	765f54ffa7f0...	2024-11-25 ...	2024-11-25 22:17:10.368	00d8f
8	dc850cd8-a417-4719-9494-ddff3f53b8ed	Açaí 300ml	12	Pote de Açaí 30...	d3c76275c1...	2024-11-29 ...	2024-11-29 01:29:50.485	f44fb5

File Edit Selection View Go Run ... ⏪ ⏩ 🔎 backend

EXPLORER

BACKEND

- > node\_modules
- > prisma
- src
  - > @types
  - > config
  - > controllers
  - > middlewares
  - > prisma
  - > services
  - TS routes.ts**
  - TS server.ts**
- > tmp
- > tmp
- .env
- .gitignore
- { package.json
- tsconfig.json
- yarn-error.log
- yarn.lock

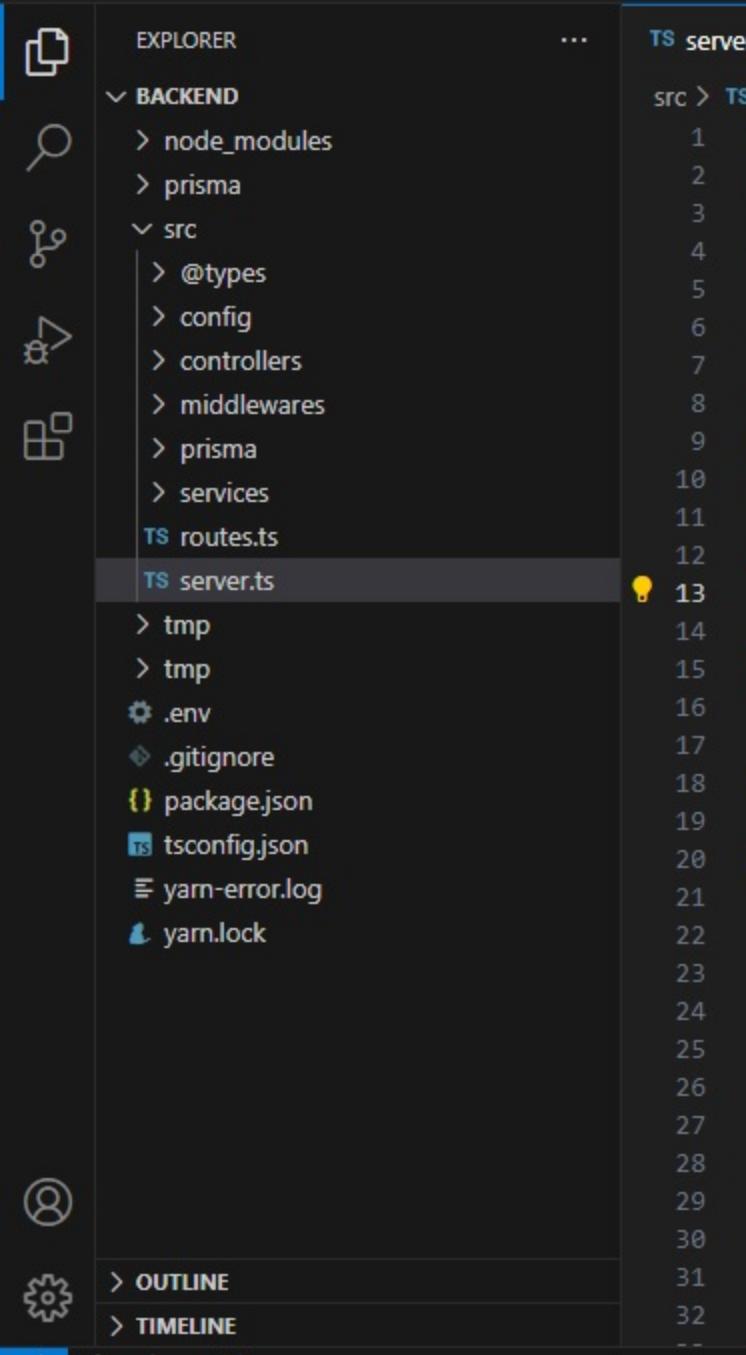
OUTLINE

TIMELINE

TS server.ts

```
src > TS server.ts > ...
1 import express, { Request, Response, NextFunction } from 'express'
2 import 'express-async-errors';
3 import cors from 'cors';
4 import path from 'path';
5
6 import { router } from './routes'
7
8 const app = express();
9 app.use(express.json());
10 app.use(cors());
11
12 app.use(router);
13
14 app.use(
15   '/files',
16   express.static(path.resolve(__dirname, '..', 'tmp')))
17 )
18
19 app.use((err: Error, req: Request, res: Response, next: NextFunction) => {
20   if(err instanceof Error) {
21     // Se for uma instância do tipo error
22     res.status(400).json({
23       error: err.message
24     })
25   }
26
27   res.status(500).json({
28     status: 'error',
29     message: 'Internal server error.'
30   })
31
32 })
```

Ln 16, Col 57 (53 selected) Spaces: 4 CRLF { TypeScript ✓ Prettier



TS server.ts

```
src > TS server.ts > ...
1 import express, { Request, Response, NextFunction } from 'express'
2 import 'express-async-errors';
3 import cors from 'cors';
4 import path from 'path';
5
6 import { router } from './routes'
7
8 const app = express();
9 app.use(express.json());
10 app.use(cors());
11
12 app.use(router);
13
14 // Acessar url da imagem ex: localhost:3333/files/nome-imagem.png
15 app.use(
16   '/files',
17   express.static(path.resolve(__dirname, '..', 'tmp'))
18 )
19
20 app.use((err: Error, req: Request, res: Response, next: NextFunction) => {
21   if(err instanceof Error) {
22     // Se for uma instância do tipo error
23     res.status(400).json({
24       error: err.message
25     })
26   }
27
28   res.status(500).json({
29     status: 'error',
30     message: 'Internal server error.'
31   })
32 }
```

EXPLORER

... ...

BACKEND

> node\_modules

> prisma

src

> @types

> config

> controllers

> middlewares

> prisma

> services

TS routes.ts

TS server.ts

tmp

- 4f039587d393ea86da3b98f3baa23b08-acai.jpg
- 8a652b025acaa467024b0a74085c90...
- 9b5565412cabf8912fd21c9c26cfaca...
- 040cfab507866ce68a0ce7fdc60e8cc...
- 765f54ffa7f05a9ff902a1a4b85f0e3b...
- b98497f70a15c4d47f7f2209a61bffc1...
- c83f6cc7a8b3dea100c14366e0808e9...
- d3c76275c1e9e682c497a354bd3285...

> tmp

.env

.gitignore

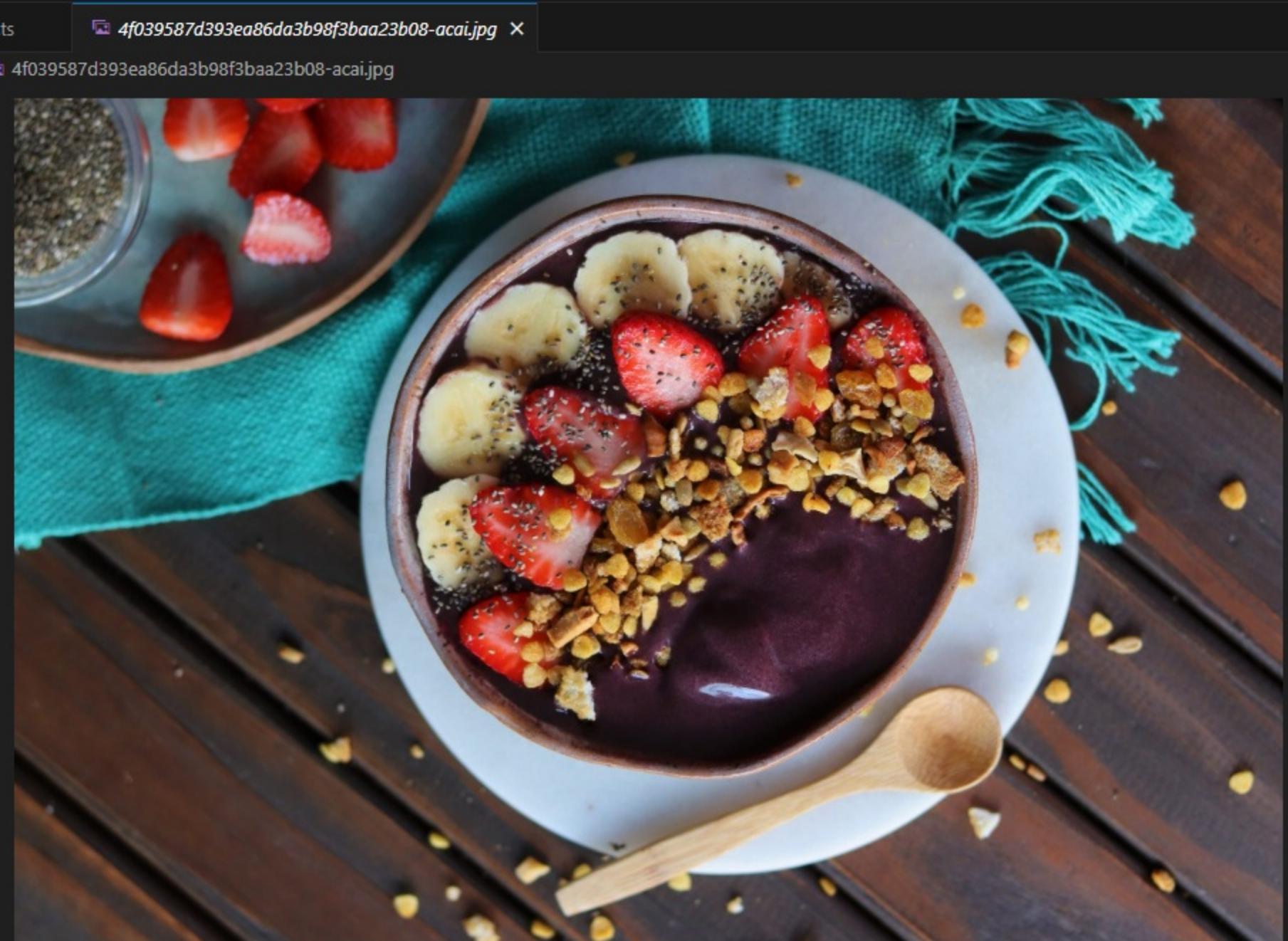
package.json

tsconfig.json

yarn-error.log

OUTLINE

TIMELINE



## Novo produto

**bebidas**

Digite o nome do produto...

Preço do produto...

Digite a descrição do produto...

**Cadastrar produto**

Image and Video Upload, Storage & Processing

cloudinary.com

Cloudinary

PLATFORM SOLUTIONS DEVELOPERS RESOURCES ABOUT US PRICING

SIGN UP FOR FREE

Image and Video API Platform

# Visualize Engaging Experiences

Trusted by 2 million developers and 10,000 enterprise and hyper-growth companies as a critical part of their tech stack.

GET STARTED

LET'S CHAT

o,q\_auto,c\_crop,g\_east,h\_530,w\_630,l\_album\_promo/fl\_layer\_app



Cloudinary Programmable Media offers AI-powered APIs to automate image and video management, transformation and optimized delivery at scale

## Programmable Media

## Digital Asset Management

Yearly  Monthly**Advanced**

Engaging visual experiences with advanced features.

**GET STARTED****\$224** Per month5 Users / 3 Accounts  
600 monthly credits**Plus** ★ MOST POPULAR

Extend your team's digital media capabilities for websites & apps.

**GET STARTED****\$89** Per month3 Users / 2 Accounts  
225 monthly credits**Free**

Start exploring Programmable Media APIs today.

**SIGN UP FREE****\$0** Free forever  
No credit card required3 Users / 1 Account  
25 monthly credits**Enterprise**

For organizations requiring digital media at scale.

**CONTACT US**

Includes enhanced features, security, support, and services.



## Advanced

Engaging visual experiences with advanced features.

[GET STARTED](#)

**\$224** Per month

5 Users / 3 Accounts  
600 monthly credits

## Plus MOST POPULAR

Extend your team's digital media capabilities for websites & apps.

[GET STARTED](#)

**\$89** Per month

3 Users / 2 Accounts  
225 monthly credits

## Free

Start exploring Programmable Media APIs today.

[SIGN UP FREE](#)

**\$0** Free forever  
No credit card required

3 Users / 1 Account  
25 monthly credits

## Enterprise

For organizations requiring digital media at scale.

[CONTACT US](#)

### Features (incl Plus features)

- Custom domain (CNAME) support
- Optional HTTPS SSL Certificate
- Authentication Options

### Features (incl Free features)

- Backup to your own S3 bucket
- Auto-tagging based search
- Access to free and paid add-ons
- Expedited support
- Allowlist/Blocklist asset access
- Video Bandwidth Boost 2:1
- Role-based multi-user

### Features

- Upload widget, API, and search
- Remote fetch, auto-backup, and revision tracking
- Image and video transformations
- Video transcoding and adaptive streaming
- High performance delivery through a CDN

### Features

- Security and compliance review
- Multi-CDN and enterprise grade infrastructure
- Enterprise SLAs and support
- Designated customer success manager
- Custom contracts and invoicing
- User management (SSO, OAuth, SAML)

[GET STARTED](#)**\$224** Per month5 Users / 3 Accounts  
600 monthly credits[GET STARTED](#)**\$89** Per month3 Users / 2 Accounts  
225 monthly credits[SIGN UP FREE](#)**\$0** Free forever  
No credit card required3 Users / 1 Account  
25 monthly credits[CONTACT US](#)Includes enhanced  
features, security, support,  
and services.**Features (incl Plus features)**

- Custom domain (CNAME) support
- Optional HTTPS SSL Certificate
- Authentication Options

**Features (incl Free features)**

- Backup to your own S3 bucket
- Auto-tagging based search
- Access to free and paid add-ons
- Expedited support
- Allowlist/Blocklist asset access
- Video Bandwidth Boost 2:1
- Role-based multi-user administration

**Features**

- Upload widget, API, and search
- Remote fetch, auto-backup, and revision tracking
- Image and video transformations
- Video transcoding and adaptive streaming
- High performance delivery through a CDN
- Support — forums, tickets, and email
- Access to free add-ons

**Features**

- Security and compliance review
- Multi-CDN and enterprise grade infrastructure
- Enterprise SLAs and support
- Designated customer success manager
- Custom contracts and invoicing
- User management (SSO, provisioning API)

## Welcome to Cloudinary

Get started - it's free.

[SIGN UP WITH EMAIL](#)

Or

[SIGN UP WITH GOOGLE](#)

[SIGN UP WITH GITHUB](#)

By signing up, you indicate that you have read, understood and agree to  
Cloudinary's [Terms of Service](#) and [Privacy Policy](#)

Already have an account? [Log in](#)

Trusted by 2M developers and  
10,000+ customers worldwide



Fazer login com o Google

## Escolha uma conta

para prosseguir para [cloudinary.com](https://cloudinary.com)



Usuário E

[usuarioe@gmail.com](mailto:usuarioe@gmail.com)[Usar outra conta](#)

Consulte a [Política de Privacidade](#) e os [Termos de Serviço](#) do app [cloudinary.com](https://cloudinary.com) antes de usá-lo.



Fazer login com o Google

# Fazer login no serviço cloudinary.com

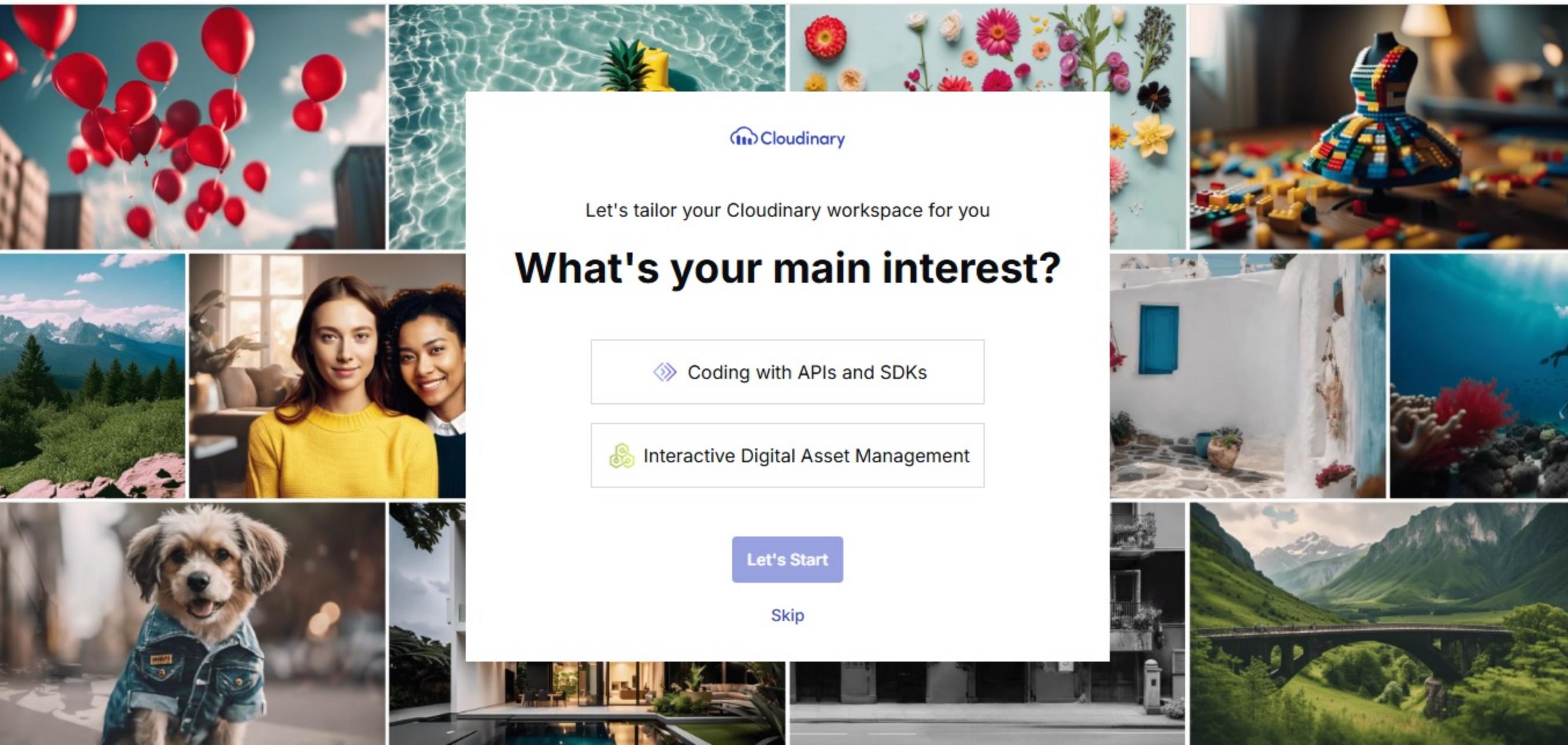


usuarioe@gmail.com

Ao continuar, o Google vai compartilhar seu nome, endereço de e-mail, preferência de idioma e foto do perfil com cloudinary.com. Consulte a [Política de Privacidade](#) e os [Termos de Serviço](#) de cloudinary.com.

Você pode gerenciar o recurso "Fazer login com o Google" na [sua conta](#).

[Cancelar](#)[Continuar](#)



Cloudinary Management Conso

console.cloudinary.com/console/c-b217595c7f0a2d2f171a95fcfa7f99/welcome

Cloudinary

Let's tailor your Cloudinary workspace for you

# What's your main interest?

Coding with APIs and SDKs

Interactive Digital Asset Management

Let's Start

Skip

Cloudinary Management Console

console.cloudinary.com/pm/c-b217595c7f0a2d2f171a95fcfa7f99/getting-started

# Get Started with Programmable Media

Original (JPEG, 1.37 MB) → Optimized (AVIF, 340 KB) → Transformed

## 1 Choose an SDK

All SDKs

**Node.js**

**React**

**Next.js**

**Python**

## 2 Install Cloudinary

Get the Cloudinary Node.js SDK

```
npm  pnpm  yarn  
npm i cloudinary
```

## 3 Upload, Optimize and Transform

Watch tutorial

View API Keys

dgshomtoi

Getting Started

Dashboard

Reports

Manage and Analyze

Media Explorer

Media Library

Transform and Customize

Transformation Center

Manage Transformations

Optimize and Deliver

Overview

Video

Cloudinary Management Console

console.cloudinary.com/pm/c-b217595c7f0a2d2f171a95fcfa7f99/getting-started

dgshomtoi

Getting Started

Dashboard

Reports

Manage and Analyze

Media Explorer

Media Library

Transform and Customize

Transformation Center

Manage Transformations

Optimize and Deliver

Overview

Video

## Install Cloudinary

Get the Cloudinary Node.js SDK

npm pnpm yarn

```
npm i cloudinary
```

## Upload, Optimize and Transform

Watch tutorial

View API Keys

index.js

```
import { v2 as cloudinary } from 'cloudinary';

(async function() {

    // Configuration
    cloudinary.config({
        cloud_name: 'dgshomtoi',
        api_key: '891882163577121',
        api_secret: '<your_api_secret>' // Click 'View API Keys' above to copy your API secret
    });

    // Upload an image
    const uploadResult = await cloudinary.uploader
        .upload(
            'https://res.cloudinary.com/demo/image/upload/getting-started/shoes.jpg', {
                public_id: 'shoes',
            }
        )
        .catch((error) => {
            console.log(error);
        });

    console.log(uploadResult);
});
```

Copy

1

Cloudinary Management Conso x G docs cloudinary - Pesquisa Goo x +

← → C google.com/search?q=docs+cloudinary&sca\_esv=34e05643fe6f2863&sxsrf=ADLYWIKfQM0i6IAUtslyBOK\_fqSz0V8Kw%3A1733435752487&source=hp... ☆

Google docs cloudinary X | ⌨ 🎤 📸 🔎

Todas Imagens Vídeos Shopping Notícias Web Maps Mais Ferramentas

**Patrocinado**



TwicPics

<https://www.twicpics.com> › claudi › alternative

⋮

## Cloud Image Optimization - By Developers, For Developers

Responsive Images Optimization Platform. Best Rated Developer Experience. Easy Integration with Your Tech Stack.

[Clear Documentation](#) · [Smart Cropping](#). AI-based. · [Context-Aware Resizing](#) · [Blog Center](#)

**Patrocinado**



Gumlet

<https://www.gumlet.com>

⋮

## Best Cloudinary alternative

Compare **Cloudinary** to Gumlet — Switch to Gumlet for better image optimization, simplified workflows and 30% less CDN cost. Cut cost by 30%, migrate faster, and deliver stunning images and videos in latest formats.

[Image Format Conversion](#) · [Video Settings](#) · [Video Streaming](#) · [Video Protection](#) · [Video Hosting](#)



Cloudinary

<https://cloudinary.com> › docume...

⋮

## Cloudinary Image & Video Management - Documentation Home

Cloudinary Management Conso x G docs cloudinary - Pesquisa Goo x +

← → C google.com/search?q=docs+cloudinary&lsca\_esv=34e05643fe6f2863&sxsrf=ADLYWIKfQM0i6IAUtslyBOK\_fqSz0V8Kw%3A1733435752487&source=hp... ☆

docs cloudinary

Cloudinary

https://cloudinary.com › docume... · Traduzir esta página ::

## Cloudinary Image & Video Management - Documentation Home

AI-powered image and video APIs, SDKs and widgets that automate the visual asset lifecycle. Getting Started, SDK Quick Starts, Developer Onboarding, Video ...

---

[Upload API Reference](#) >

Cloud-based Upload API with extensive options for uploading ...

---

[Node.js SDK](#) >

Node.js quick start - Image and video upload - Asset Management

---

[Guides](#) >

Cloudinary Logo Cloudinary Docs Docs · Programmable Media ...

---

[Upload](#) >

Cloudinary provides a secure and comprehensive API for easily ...

---

[Image transformations](#) >

For more information on these SDK helper methods, see the ...

[Mais resultados de cloudinary.com »](#)

[https://cloudinary.com/documentation/image\\_upload\\_api\\_reference](https://cloudinary.com/documentation/image_upload_api_reference)

Cloudinary Management Conso x Upload API Reference | Cloudinx +

cloudinary.com/documentation/image\_upload\_api\_reference

Docs Programmable Media Digital Asset Management More Products Resources Pricing Search Documentation SIGN UP FOR FREE

Get Started Guides References SDKs Release Notes

Transformation URL API

Programmable Media > References > Upload API

Upload API

Last updated: Dec-05-2024

The Upload API is a **rate-unlimited** RESTful API that enables you to upload your media assets (resources) and provides a wide range of functionality, including basic and advanced asset management, metadata management, and asset generation.

Cloudinary's backend SDKs wrap these REST APIs, handle authentication, and enable you to perform these methods using your preferred programming language or framework. This reference provides both SDK and REST/cURL syntax and examples for each endpoint method.

**Tip:** MediaFlows, Cloudinary's drag-and-drop workflow builder for image and video, supports all Upload API parameters in a low-code environment. See MediaFlow's documentation on media upload [here](#).

# Upload API reference

## Overview

By default, the API endpoints use the following format:

`https://api.cloudinary.com/v1_1/:cloud_name/:action`

For example, to upload an image asset to the `demo` product environment:

On this page:

- Overview
- Asset management
  - upload
  - explicit
  - rename (public ID)
  - destroy
  - download\_backup
- Metadata management
  - context
  - metadata
  - tags
- Asset generation
  - create\_collage
  - explode

Rate this page:

Cloudinary Management Console x Upload API Reference | Cloudiny Cloudinary URL-Gen (JS) x +

cloudinary.com/documentation/image\_upload\_api\_reference

Docs Programmable Media Digital Asset Management More Products Resources Pricing Search Documentation SIGN UP FOR FREE

Get Started Guides References SDKs Release Notes

Provisioning API

Analyze API (Beta)

Live streaming API (Beta)

Cloudinary CLI

SDK references

- Transformation Builder reference ↗
- React SDK reference ↗
- Vue.js SDK reference ↗
- JavaScript SDK reference ↗
- Angular SDK reference ↗
- PHP SDK reference ↗
- Go SDK reference ↗
- Dart SDK reference ↗
- Flutter SDK reference ↗

Postman collections ↗

Upload Widget API

Product Gallery API

Programmable Media > References > Upload API

The Upload API is a **rate-unlimited** RESTful API that enables you to upload your media assets (resources) and provides a wide range of functionality, including basic and advanced asset management, metadata management, and asset generation.

Cloudinary's backend SDKs wrap these REST APIs, handle authentication, and enable you to perform these methods using your preferred programming language or framework. This reference provides both SDK and REST/cURL syntax and examples for each endpoint method.

**Tip:** MediaFlows, Cloudinary's drag-and-drop workflow builder for image and video, supports all Upload API parameters in a low-code environment. See MediaFlow's documentation on media upload [here](#).

## Overview

By default, the API endpoints use the following format:

`https://api.cloudinary.com/v1_1/:cloud_name/:action`

For example, to upload an image asset to the `demo` product environment:

```
POST https://api.cloudinary.com/v1_1/demo/image/upload
```

On this page:

- Overview
- Asset management
  - upload
  - explicit
  - rename (public ID)
  - destroy
  - download\_backup
- Metadata management
  - context
  - metadata
  - tags
- Asset generation
  - create\_collage
  - explode

Rate this page:

☆ ☆ ☆ ☆ ☆

<https://cloudinary.com/documentation/sdks/js/url-gen/index.html>

# Cloudinary URL-Gen (JS)

@cloudinary/url-gen

## Cloudinary URL-Gen SDK

### About

The Cloudinary URL-Gen SDK allows you to quickly and easily integrate your application with Cloudinary. Effortlessly optimize and transform your cloud's assets.

This SDK can also be used with popular frontend frameworks.

### Additional documentation

This Readme provides basic installation and usage information. For the complete documentation, see the [URL-Gen SDK Guide](#) and the [URL-Gen Reference](#).

NOTE: When using the Cloudinary URL-Gen SDK, it's recommended to use the Cloudinary Java Library instead. This library is more feature-rich and has better support for Java-specific features.

### Cloudinary URL-Gen SDK

About

Additional documentation

Table of Contents

Key Features

Version Support

Note!

Installation

Install using your favorite package manager  
(yarn, npm)

Usage

Setup

Transform and Optimize Assets

Generate Image and Video URLs

## Cloudinary URL-Gen (JS) - 1.21.0

[Interfaces](#)[Help & Examples](#)

```
yarn add @cloudinary/url-gen
```

[Copy](#)

## Usage

### Setup

```
// Import the Cloudinary class
import {Cloudinary} from '@cloudinary/url-gen';

// Create your instance
const cld = new Cloudinary({
  cloud: {
    cloudName: 'demo'
  },
  url: {
    secure: true // force https, set to false to force http
  }
});
```

[Copy](#)

### Transform and Optimize Assets

- See full documentation

```
// Create a new instance if you haven't (see above for the details)
const cld = new Cloudinary({/*...*/})

// Let's create a new image
const myImage = cld.image('sample');

// Import the resize transformation and apply it to myImage
import {Resize} from '@cloudinary/url-gen/actions/resize';
```

[Copy](#)

### Cloudinary URL-Gen SDK

[About](#)[Additional documentation](#)[Table of Contents](#)[Key Features](#)[Version Support](#)[Note!](#)[Installation](#)

Install using your favorite package manager (yarn, npm)

### Usage

[Setup](#)[Transform and Optimize Assets](#)[Generate Image and Video URLs](#)[File upload](#)[Transpilation](#)[Testing with Jest](#)[Contributions](#)[Get Help](#)[About Cloudinary](#)[Additional Resources](#)[Back to top](#)



```
        secure: true // force https, set to false to force http
    }
});
```

## Transform and Optimize Assets

- See full documentation

```
// Create a new instance if you haven't (see above for the details)
const cld = new Cloudinary({/*...*/})

// Let's create a new image
const myImage = cld.image('sample');

// Import the resize transformation and apply it to myImage
import {Resize} from '@cloudinary/url-gen/actions/resize';

// Resize the image to 100x100
myImage.resize(Resize.scale().width(100).height(100));

// When we're done, we can apply all our changes and create a URL.
const myURL = myImage.toURL();

// https://res.cloudinary.com/demo/image/upload/c_scale,w_100,h_100/sample
console.log(myURL);
```

[Copy](#)

### Cloudinary URL-Gen SDK

[About](#)[Additional documentation](#)[Table of Contents](#)[Key Features](#)[Version Support](#)[Note!](#)[Installation](#)

Install using your favorite package manager (yarn, npm)

### Usage

[Setup](#)[Transform and Optimize Assets](#)[Generate Image and Video URLs](#)[File upload](#)[Transpilation](#)[Testing with Jest](#)[Contributions](#)[Get Help](#)[About Cloudinary](#)[Additional Resources](#)[Back to top](#)

## Generate Image and Video URLs

The library supports transformations on both images and videos. Please use the appropriate method, as per below:

- Use `cld.image()` to generate image URLs and transformations
- Use `cld.video()` to generate video URLs and transformations

The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** backend
- Toolbar:** Includes icons for file operations like Open, Save, Find, and others.
- Left Sidebar:** Explorer view showing the project structure:
  - BACKEND**: node\_modules, prisma, src (expanded), @types, config, controllers, middlewares, prisma, services, routes.ts, server.ts.
  - tmp**: A folder containing numerous temporary files with random IDs (e.g., 4f039587d393ea86da3b98f3baa23b...).
  - OUTLINE** and **TIMELINE** sections.
- Central Editor Area:** Displays the contents of the **.env** file.

```
1 # Environment variables declared in this file are automatically made available to Prisma.
2 # See the documentation for more detail: https://pris.ly/d/prisma-schema#accessing-environment-variables-from-the-schema
3
4 # Prisma supports the native connection string format for PostgreSQL, MySQL, SQLite, SQL Server, MongoDB and CockroachDB.
5 # See the documentation for all the connection string options: https://pris.ly/d/connection-strings
6
7 DATABASE_URL="postgresql://postgres:root@localhost:5432/pizzaria?schema=public"
8
9 # Secret JWT https://www.md5hashgenerator.com/
10 JWT_SECRET=98388750f63fac47136942aaf8ac79ce
```
- Bottom Status Bar:** Shows file statistics (Ln 10, Col 44, Spaces: 4, UTF-8, LF) and tool buttons for Properties and Prettier.



console.cloudinary.com/pm/c-b217595c7f0a2d2f171a95fcfa7f99/getting-started

**Get Started with Programmable Media**



**1 Choose an SDK**

All SDKs

**Node.js** (selected)

**React**

**Next.js**

**Python**

**2 Install Cloudinary**

Get the Cloudinary Node.js SDK

```
npm  pnpm  yarn  
npm i cloudinary
```

**3 Upload, Optimize and Transform**

Watch tutorial

View API Keys

Billing - Plan Details - Cloudinary x Upload API Reference | Cloudinary x Cloudinary URL-Gen (JS) x +

console.cloudinary.com/settings/c-b217595c7f0a2d2f171a95fcfa7f99/billing/plans

dgshomtoi

Product environment settings

- API Keys
- Upload Settings & Backup
- Upload Presets
- Optimization
- Webhook Notifications
- Security

Account settings

- My Profile
- Account
- Product Environments
- User Management
- Account Security

Billing

Plan Details

Any questions? [Get in touch](#) and let us know how we can help.

Programmable Media

**Free**

\$0

0 / 25 Credits (i)

0% monthly usage

Next Bill Date -

**Change Plan**

Total \$0

<https://console.cloudinary.com/settings/c-b217595c7f0a2d2f171a95fcfa7f99/billing/plans>

dgshomtoi

## Product environment settings

## API Keys

Upload Settings &amp; Backup

Upload Presets

Optimization

Webhook Notifications

Security

## Account settings

My Profile

Account

Product Environments

User Management

Account Security

Billing

## API Keys

Cloud name: dgshomtoi

Manage API key and secret pairs for your product environment. To build the environment variable for each pair, copy the provided format and substitute your actual values for the placeholders. Make sure to store your secrets securely.

+ Generate New API Key

API environment variable

CLOUDINARY\_URL=cloudinary://&lt;your\_api\_key&gt;:&lt;your\_api\_secret&gt;@dgshomtoi



Key Name	Date Created	API Key	API Secret	Status
Untitled	Dec 05, 2024	891882163577121	*****	<input checked="" type="checkbox"/> Active

Page 1 of 1 &lt; &gt;



dgshomtoi

## Product environment settings

## API Keys

Upload Settings &amp; Backup

Upload Presets

Optimization

Webhook Notifications

Security

## Account settings

My Profile

Account

Product Environments

User Management

Account Security

Billing

## API Keys

Cloud name: dgshomtoi

Manage API key and secret pairs for your product environment. To build the environment variable for each pair, copy the provided format and substitute your actual values for the placeholders. Make sure to store your secrets securely.

+ Generate New API Key

API environment variable

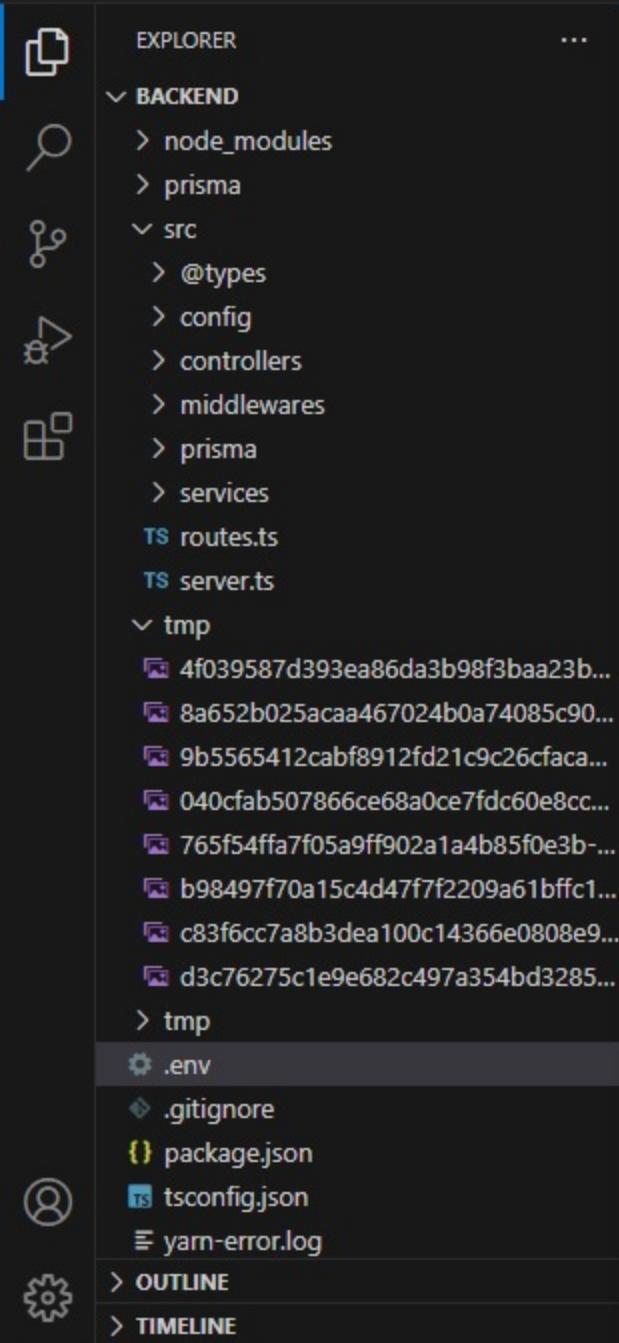
CLOUDINARY\_URL=cloudinary://&lt;your\_api\_key&gt;:&lt;your\_api\_secret&gt;@dgshomtoi

Key Name	Date Created	API Key	Status
Untitled	Dec 05, 2024	8918821635	Active

- Copiar Ctrl+C
- Copiar link para o conteúdo destacado
- Pesquisar "dgshomtoi" no Google
- Imprimir... Ctrl+P
- Abrir no modo de leitura
- Traduzir seleção para o português
- Inspecionar

Page 1 of 1 &lt; &gt;





```
⚙️ .env X  
⚙️ .env  
1  # Environment variables declared in this file are automatically made available to Prisma.  
2  # See the documentation for more detail: https://pris.ly/d/prisma-schema#accessing-environment-variables-from-the-schema  
3  
4  # Prisma supports the native connection string format for PostgreSQL, MySQL, SQLite, SQL Server, MongoDB and CockroachDB.  
5  # See the documentation for all the connection string options: https://pris.ly/d/connection-strings  
6  
7  DATABASE_URL="postgresql://postgres:root@localhost:5432/pizzaria?schema=public"  
8  
9  # Secret JWT https://www.md5hashgenerator.com/  
10 JWT_SECRET=98388750f63fac47136942aaf8ac79ce  
11  
12 # Cloudinary  
13 CLOUDINARY_NAME=dgshomtoi
```

The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** backend
- Explorer Bar (Left):**
  - EXPLORER
  - BACKEND
    - > node\_modules
    - > prisma
    - src
      - > @types
      - > config
      - > controllers
      - > middlewares
      - > prisma
      - > services
    - TS routes.ts
    - TS server.ts
  - tmp
    - 4f039587d393ea86da3b98f3baa23b...
    - 8a652b025acaa467024b0a74085c90...
    - 9b5565412cabf8912fd21c9c26cfaca...
    - 040cfab507866ce68a0ce7fdc60e8cc...
    - 765f54ffa7f05a9ff902a1a4b85f0e3b-...
    - b98497f70a15c4d47f7f2209a61bffc1...
    - c83f6cc7a8b3dea100c14366e0808e9...
    - d3c76275c1e9e682c497a354bd3285...
  - > tmp
  - .env
  - .gitignore
  - { package.json
  - tsconfig.json
  - yarn-error.log
- Code Editor (Center):**
  - File: .env
  - Content:

```
1 # Environment variables declared in this file are automatically made available to Prisma.
2 # See the documentation for more detail: https://pris.ly/d/prisma-schema#accessing-environment-variables-from-the-schema
3
4 # Prisma supports the native connection string format for PostgreSQL, MySQL, SQLite, SQL Server, MongoDB and CockroachDB.
5 # See the documentation for all the connection string options: https://pris.ly/d/connection-strings
6
7 DATABASE_URL="postgresql://postgres:root@localhost:5432/pizzaria?schema=public"
8
9 # Secret JWT https://www.md5hashgenerator.com/
10 JWT_SECRET=98388750f63fac47136942aaf8ac79ce
11
12 # Cloudinary
13 CLOUDINARY_NAME=dgshomtoi
14 CLOUDINARY_KEY=
15 CLOUDINARY_SECRET=
```
- Bottom Status Bar:** Ln 14, Col 1 (34 selected) Spaces: 4 UTF-8 LF Properties ⚡ Prettier

dgshomtoi

## Product environment settings

## API Keys

Upload Settings &amp; Backup

Upload Presets

Optimization

Webhook Notifications

Security

## Account settings

My Profile

Account

Product Environments

User Management

Account Security

Billing

## API Keys

Cloud name: dgshomtoi

Manage API key and secret pairs for your product environment. To build the environment variable for each pair, copy the provided format and substitute your actual values for the placeholders. Make sure to store your secrets securely.

+ Generate New API Key

API environment variable

CLOUDINARY\_URL=cloudinary://&lt;your\_api\_key&gt;:&lt;your\_api\_secret&gt;@dgshomtoi



Key Name	Date Created	API Key	API Secret	Status
Untitled	Dec 05, 2024	891882163577121	*****	<input checked="" type="checkbox"/> Active

Copiar Ctrl+C

Copiar link para o conteúdo destacado

Pesquisar "891882163577121" no Google

Imprimir... Ctrl+P

Abrir no modo de leitura

Traduzir seleção para o português

Inspecionar

Page 1 of 1 &lt; &gt;



A screenshot of the Visual Studio Code (VS Code) interface. The top bar includes standard menu items: File, Edit, Selection, View, Go, Run, and a separator. To the right are icons for back, forward, search, and window controls. The search bar contains the text "backend". The left sidebar (Explorer) shows a file tree with a tree view icon. The "BACKEND" folder is expanded, showing subfolders like "node\_modules", "prisma", and "src" (which contains "@types", "config", "controllers", "middlewares", "prisma", "services", "routes.ts", and "server.ts"). A "tmp" folder is also present with many temporary files. Below the main tree are collapsed sections for ".env", ".gitignore", "package.json", "tsconfig.json", and "yarn-error.log". The main editor area displays a ".env" file with the following content:

```
1 # Environment variables declared in this file are automatically made available to Prisma.
2 # See the documentation for more detail: https://pris.ly/d/prisma-schema#accessing-environment-variables-from-the-schema
3
4 # Prisma supports the native connection string format for PostgreSQL, MySQL, SQLite, SQL Server, MongoDB and CockroachDB.
5 # See the documentation for all the connection string options: https://pris.ly/d/connection-strings
6
7 DATABASE_URL="postgresql://postgres:root@localhost:5432/pizzaria?schema=public"
8
9 # Secret JWT https://www.md5hashgenerator.com/
10 JWT_SECRET=98388750f63fac47136942aaf8ac79ce
11
12 # Cloudinary
13 CLOUDINARY_NAME=dgshomtoi
14 CLOUDINARY_KEY=891882163577121
15 CLOUDINARY_SECRET=
```

The bottom status bar shows the current position as "Ln 14, Col 31 (15 selected)", settings for "Spaces: 4", "UTF-8", "LF", and "Properties", and a "Prettier" button.

console.cloudinary.com/settings/c-b217595c7f0a2d2f171a95fcfa7f99/api-keys

dgshomtoi

Product environment settings

API Keys

Upload Settings & Backup

Upload Presets

Optimization

Webhook Notifications

Security

Account settings

My Profile

Account

Product Environments

User Management

Account Security

Billing

## API Keys

Cloud name: dgshomtoi

Manage API key and secret pairs for your product environment. To build the environment variable for each pair, copy the provided format and substitute your actual values for the placeholders. Make sure to store your secrets securely.

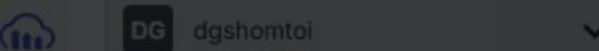
+ Generate New API Key

API environment variable

```
CLOUDINARY_URL=cloudinary://<your_api_key>:<your_api_secret>@dgshomtoi
```

Key Name	Date Created	API Key	API Secret	Status
Untitled	Dec 05, 2024	891882163577121	*****	<input checked="" type="checkbox"/> Active

Page 1 of 1 < >



Product environment settings

## API Keys

Upload Settings &amp; Backup

Upload Presets

Optimization

Webhook Notifications

Security

## Account settings

My Profile

Account

Product Environments

User Management

Account Security

Billing

## API Keys

Cloud name: dgshomtoi

Manage API key and secret pairs for your product environment. To build the environment variable for each pair, copy the provided format and substitute your actual values for the placeholders. Make sure to store your secrets securely.

+ Generate New API Key

API environment variable

CLOUDINARY\_URL=cloudinary://&lt;your\_api\_key&gt;:&lt;your\_api\_secret&gt;@dgshomtoi

## Password Verification



Confirmation code has been sent to your email.

Email Confirmation Code

Cancel

Approve

API Secret

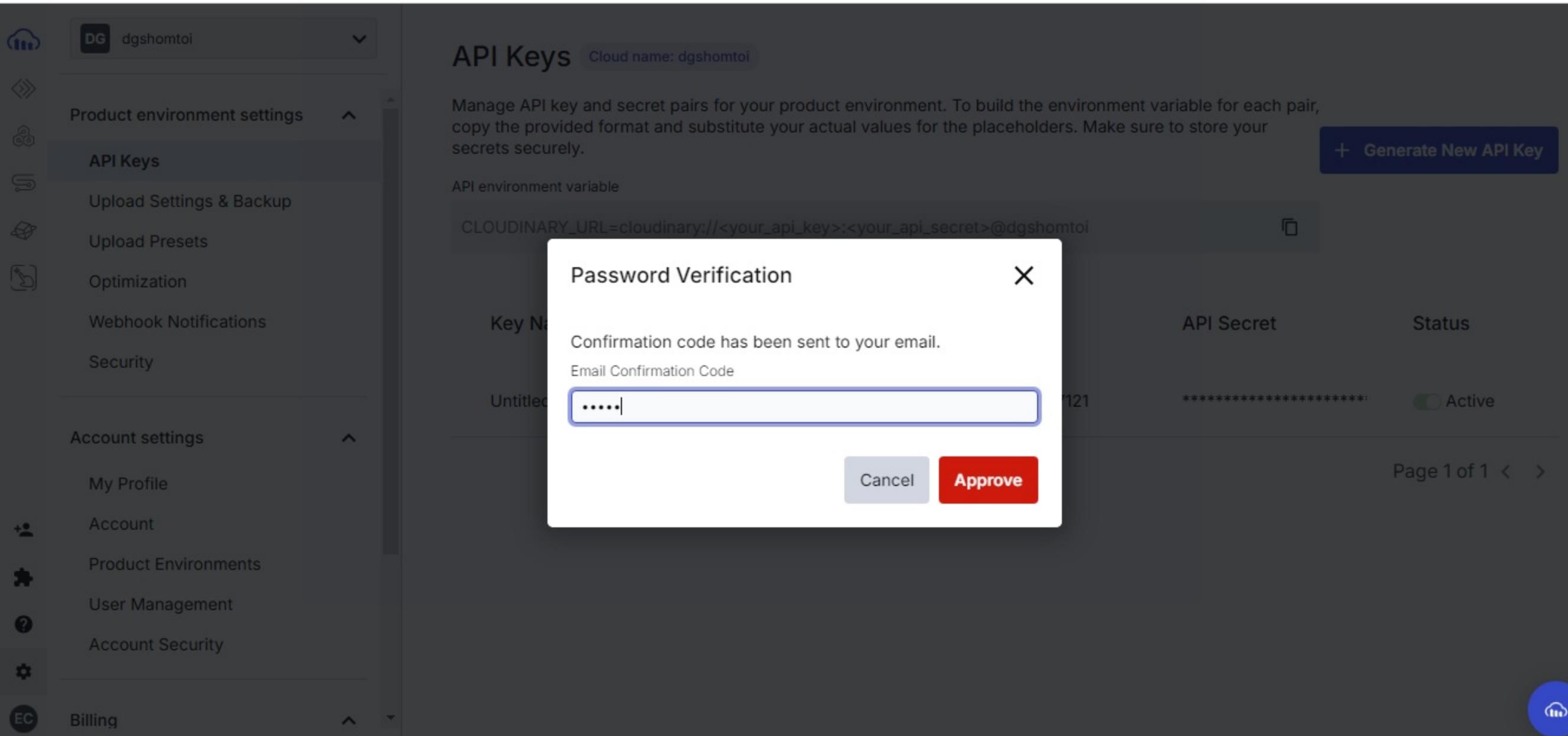
121

\*\*\*\*\*

Active

Page 1 of 1 &lt; &gt;





dgshomtoi

## Product environment settings

## API Keys

Upload Settings &amp; Backup

Upload Presets

Optimization

Webhook Notifications

Security

## Account settings

My Profile

Account

Product Environments

User Management

Account Security

Billing

## API Keys

Cloud name: dgshomtoi

Manage API key and secret pairs for your product environment. To build the environment variable for each pair, copy the provided format and substitute your actual values for the placeholders. Make sure to store your secrets securely.

+ Generate New API Key

API environment variable

CLOUDINARY\_URL=cloudinary://&lt;your\_api\_key&gt;:&lt;your\_api\_secret&gt;@dgshomtoi



Key Name	Date Created	API Key	API Secret	Status
Untitled	Dec 05, 2024	891882163577121	vi9KPAJ3xxG2a...	Active

Page 1 of 1 &lt; &gt;



The screenshot shows the Visual Studio Code (VS Code) interface. The top bar includes standard menu items: File, Edit, Selection, View, Go, Run, etc. The title bar displays the current workspace name: "backend". The left sidebar (Explorer) lists the project structure:

- BACKEND
  - node\_modules
  - prisma
  - src
    - @types
    - config
    - controllers
    - middlewares
    - prisma
    - services
  - routes.ts
  - server.ts- tmp
  - 4f039587d393ea86da3b98f3baa23b...
  - 8a652b025acaa467024b0a74085c90...
  - 9b5565412cabf8912fd21c9c26cfaca...
  - 040cfab507866ce68a0ce7fdc60e8cc...
  - 765f54ffa7f05a9ff902a1a4b85f0e3b-...
  - b98497f70a15c4d47f7f2209a61bffc1...
  - c83f6cc7a8b3dea100c14366e0808e9...
  - d3c76275c1e9e682c497a354bd3285...
- tmp
  - .env
  - .gitignore
  - package.json
  - tsconfig.json
  - yarn-error.log

The main editor area shows the content of the ".env" file:

```
1 # Environment variables declared in this file are automatically made available to Prisma.
2 # See the documentation for more detail: https://pris.ly/d/prisma-schema#accessing-environment-variables-from-the-schema
3
4 # Prisma supports the native connection string format for PostgreSQL, MySQL, SQLite, SQL Server, MongoDB and CockroachDB.
5 # See the documentation for all the connection string options: https://pris.ly/d/connection-strings
6
7 DATABASE_URL="postgresql://postgres:root@localhost:5432/pizzaria?schema=public"
8
9 # Secret JWT https://www.md5hashgenerator.com/
10 JWT_SECRET=98388750f63fac47136942aaf8ac79ce
11
12 # Cloudinary
13 CLOUDINARY_NAME=dgshomtoi
14 CLOUDINARY_KEY=891882163577121
15 CLOUDINARY_SECRET=vI9KPAJ3xxG2a4K3VMHxvJ4zcgA
```

The status bar at the bottom provides information about the current file: Line 15, Column 19 (27 selected), Spaces: 4, UTF-8, LF, Properties, and Prettier.

```
[INFO] 18:18:36 ts-node-dev ver. 2.0.0 (using ts-node ver. 10.9.2, typescript ver. 5.6.3)
Servidor online!!!!
[INFO] 18:42:15 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\server.ts has been modified
Servidor online!!!!
[INFO] 18:42:24 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\server.ts has been modified
Servidor online!!!!
[INFO] 18:42:27 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\server.ts has been modified
Servidor online!!!!
[INFO] 18:42:30 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\server.ts has been modified
Servidor online!!!!
[INFO] 18:42:32 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\server.ts has been modified
Servidor online!!!!
[INFO] 18:42:36 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\server.ts has been modified
Servidor online!!!!
[INFO] 18:42:39 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\server.ts has been modified
Servidor online!!!!
[INFO] 18:42:43 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\server.ts has been modified
Servidor online!!!!
[INFO] 18:42:46 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\server.ts has been modified
Servidor online!!!!
[INFO] 18:42:49 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\server.ts has been modified
Servidor online!!!!
[INFO] 18:42:51 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\server.ts has been modified
Servidor online!!!!
[INFO] 18:42:53 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\server.ts has been modified
Servidor online!!!!
[INFO] 18:42:55 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\server.ts has been modified
Servidor online!!!!
```

Servidor online!!!!  
[INFO] 18:42:32 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\server.ts has been modified  
Servidor online!!!!  
[INFO] 18:42:36 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\server.ts has been modified  
Servidor online!!!!  
[INFO] 18:42:39 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\server.ts has been modified  
Servidor online!!!!  
[INFO] 18:42:43 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\server.ts has been modified  
Servidor online!!!!  
[INFO] 18:42:46 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\server.ts has been modified  
Servidor online!!!!  
[INFO] 18:42:49 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\server.ts has been modified  
Servidor online!!!!  
[INFO] 18:42:51 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\server.ts has been modified  
Servidor online!!!!  
[INFO] 18:42:53 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\server.ts has been modified  
Servidor online!!!!  
[INFO] 18:42:55 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\server.ts has been modified  
Servidor online!!!!

Deseja finalizar o arquivo em lotes (S/N)?

^C

c:\Users\Dev\Desktop\Pizzaria\backend>

c:\Users\Dev\Desktop\Pizzaria\backend>\_

```
next-server (v15.0.3) - X
./src/app/dashboard/components/orders/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/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

./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/product 200 in 1114ms
```

Administrador: Prompt de Comando

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

.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/product 200 in 1114ms

Deseja finalizar o arquivo em lotes (S/N)?

^C

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



```
c:\Users\Dev\Desktop\Pizzaria\backend>npm install cloudinary
```

```
c:\Users\Dev\Desktop\Pizzaria\backend>npm install cloudinary
npm warn deprecated q@1.5.1: You or someone you depend on is using Q, the JavaScript Promise library that gave JavaScript developers strong feelings about promises. They can almost certainly migrate to the native JavaScript promise now. Thank you literally everyone for joining me in this bet against the odds. Be excellent to each other.
npm warn deprecated
npm warn deprecated (For a CapTP with native promises, see @endo/eventual-send and @endo/capt)
added 3 packages, changed 25 packages, and audited 1008 packages in 18s

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

64 vulnerabilities (15 moderate, 10 high, 39 critical)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.

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



express-fileupload



Todas Videos Shopping Imagens Notícias Web Maps Mais

Ferramentas



NPM

<https://www.npmjs.com> › package · Traduzir esta página



## express-fileupload

13 de jul. de 2024 — Simple **express file upload** middleware that wraps around Busboy. Latest version:

1.5.1, last published: 5 months ago.

[Types/express-fileupload](#) · 49 Versions · 852 Dependents



Stack Overflow

<https://stackoverflow.com> › uplo... · Traduzir esta página



## Uploading file with express-fileupload - node.js

23 de ago. de 2017 — I am trying to upload a file with **express-fileupload** and am having no luck getting it to work. I can get the file (in this case an image) to 'upload'.

6 respostas · Melhor resposta: You are pointing the directory where the file would go to, but you are no...

[Node/Express file upload](#) - Stack Overflow

16 de mai. de 2014

[node.js - File upload and move problem with express-fileupload](#)

19 de ago. de 2019

[How do you use express-fileupload correctly](#) - Stack Overflow

1 de jul. de 2020

[req.files is undefined using express-fileupload](#) - Stack Overflow

14 de jul. de 2020

[Mais resultados de stackoverflow.com](#)

[Pro](#) [Teams](#) [Pricing](#) [Documentation](#) Search packages[Search](#)[Sign Up](#)[Sign In](#)

## express-fileupload DT

1.5.1 • Public • Published 5 months ago

[Readme](#)[Code](#)Beta1 Dependency856 Dependents49 Versions

# express-fileupload

Simple express middleware for uploading files.

npm v1.5.1 coverage 95%

## Help us Improve express-fileupload

This package is still very much supported and maintained. But the more help the better. If you're interested any of the following:

### Install

```
> npm i express-fileupload
```



### Repository

[github.com/richardgirges/express-fileupload](https://github.com/richardgirges/express-fileupload)

### Homepage

[github.com/richardgirges/express-fileupload](https://github.com/richardgirges/express-fileupload)

Settings - API Keys - Cloudinary | Upload API Reference | Cloudinary URL-Gen (JS) | express-fileupload - npm

npmjs.com/package/express-fileupload

Search packages

Search Sign Up Sign In

## express-fileupload DT

1.5.1 • Public • Published 5 months ago

Readme Code Beta 1 Dependency 856 Dependents 49 Versions

# express-fileupload

Simple express middleware for uploading files.

npm v1.5.1 PASSED coverage 95%

## Help us Improve express-fileupload

This package is still very much supported and maintained. But the more help the better. If you're interested any of the following:

- Ticket and PR triage
- Feature scoping and implementation
- Maintenance (upgrading packages, fixing security vulnerabilities, etc)

Install

```
> npm i express-fileupload
```

Copiar Ctrl+C

Repository

[github.com](#)

Copiar link para o conteúdo destacado

Pesquisar "npm i express-fileupload" no Google

Imprimir... Ctrl+P

Homepage

[github.com](#)

Abrir no modo de leitura

Traduzir seleção para o português

Inspecionar

Weekly Downloads

435.948

Version License



Administrador: Prompt de Comando

```
c:\Users\Dev\Desktop\Pizzaria\backend>npm i express-fileupload
```

```
c:\Users\Dev\Desktop\Pizzaria\backend>npm i express-fileupload
```

```
added 1 package, and audited 1009 packages in 4s
```

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

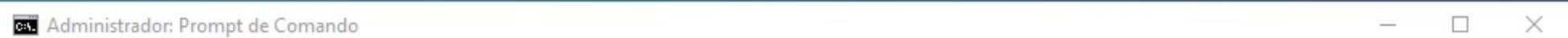
```
64 vulnerabilities (15 moderate, 10 high, 39 critical)
```

```
To address issues that do not require attention, run:  
  npm audit fix
```

```
To address all issues (including breaking changes), run:  
  npm audit fix --force
```

```
Run `npm audit` for details.
```

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



```
c:\Users\Dev\Desktop\Pizzaria\backend>npm install --save-dev @types/express-fileupload
```

Administrador: Prompt de Comando

```
c:\Users\Dev\Desktop\Pizzaria\backend>npm install --save-dev @types/express-fileupload
added 2 packages, and audited 1011 packages in 12s
28 packages are looking for funding
  run `npm fund` for details

64 vulnerabilities (15 moderate, 10 high, 39 critical)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.

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

EXPLORER

... TS server.ts X

BACKEND

- > node\_modules
- > prisma
- src
  - > @types
  - > config
  - > controllers
  - > middlewares
  - > prisma
  - > services
- TS routes.ts
- TS server.ts
- > tmp
- > tmp
- .env
- .gitignore
- { package-lock.json
- { package.json
- tsconfig.json
- = yarn-error.log
-   yarn.lock

OUTLINE

TIMELINE

```
src > TS server.ts > ...
1 import express, { Request, Response, NextFunction } from 'express'
2 import 'express-async-errors';
3 import cors from 'cors';
4 import path from 'path';
5
6 import { router } from './routes'
7 import fileUpload from 'express-fileupload';
8
9 const app = express();
10 app.use(express.json());
11 app.use(cors());
12 app.use(fileUpload({
13   limits: { fileSize: 50 * 1024 * 1024 } // Enviar imagem de no máximo 50mb
14 }))
15 app.use(router);
16
17 // Acessar url da imagem ex: localhost:3333/files/nome-imagem.png
18 app.use(
19   '/files',
20   express.static(path.resolve(__dirname, '..', 'tmp'))
21 )
22
23 app.use((err: Error, req: Request, res: Response, next: NextFunction) => {
24   if(err instanceof Error) {
25     // Se for uma instância do tipo error
26     res.status(400).json({
27       error: err.message
28     })
29   }
30
31   res.status(500).json({
32     status: 'error',
33   })
34 })
```

A screenshot of a code editor interface, likely Visual Studio Code, displaying a file named `TS routes.ts`. The file contains TypeScript code for a REST API endpoint handling. The code includes imports for `multer` and various controller classes like `CreateUserController`, `AuthUserController`, etc. It defines routes for users, categories, products, orders, and sessions, each with specific methods like `post` and `get`. A yellow warning icon is present near the product-related code. The editor's sidebar shows a project structure with files like `node_modules`, `prisma`, and `src`. The bottom status bar indicates the file has 207 selected characters.

```
const upload = multer(uploadConfig.upload("./tmp"))

// -- ROTAS USER --
router.post('/users', new CreateUserController().handle)
router.post('/session', new AuthUserController().handle)

router.get('/me', isAuthenticated, new DetailUserController().handle)

// -- ROTAS CATEGORY --
router.post('/category', isAuthenticated, new CreateCategoryController().handle)
router.get('/category', isAuthenticated, new ListCategoryController().handle)

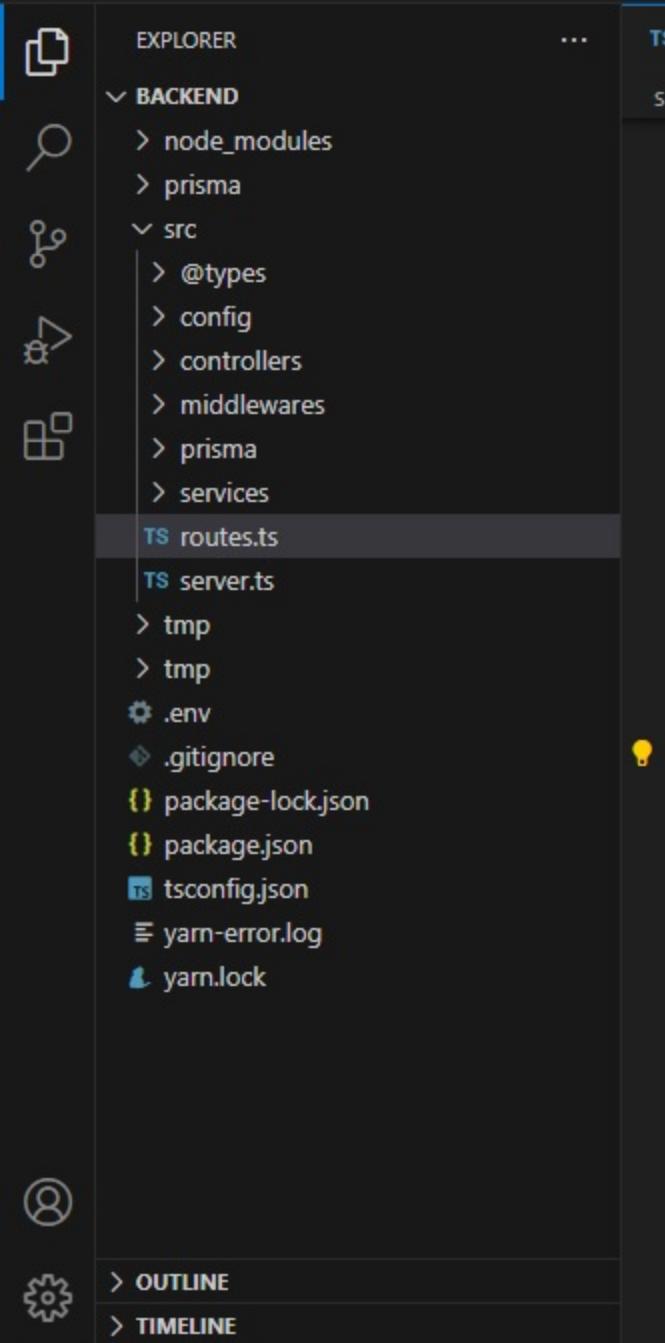
// -- ROTAS PRODUCT --
// router.post('/product', isAuthenticated, upload.single('file'), new CreateProductController().handle)
router.post('/product', isAuthenticated, upload.single('file'), new CreateProductController().handle)
router.get('/category/product', isAuthenticated, new ListByCategoryController().handle)

// -- ROTAS ORDER --
router.post('/order', isAuthenticated, new CreateOrderController().handle)
router.delete('/order', isAuthenticated, new RemoveOrderController().handle)

router.post('/order/add', isAuthenticated, new AddItemController().handle)
router.delete('/order/remove', isAuthenticated, new RemoveItemController().handle)
router.put('/order/send', isAuthenticated, new SendOrderController().handle)

router.get('/orders', isAuthenticated, new ListOrdersController().handle)
router.get('/order/detail', isAuthenticated, new DetailOrderController().handle)

router.put('/order/finish', isAuthenticated, new FinishOrderController().handle)
```



## TS routes.ts

```
src > TS routes.ts > ...
31  const upload = multer(uploadConfig.upload("./tmp"))
32
33 // -- ROTAS USER --
34 router.post('/users', new CreateUserController().handle)
35
36 router.post('/session', new AuthUserController().handle)
37
38 router.get('/me', isAuthenticated, new DetailUserController().handle)
39
40 // -- ROTAS CATEGORY --
41 router.post('/category', isAuthenticated, new CreateCategoryController().handle)
42
43 router.get('/category', isAuthenticated, new ListCategoryController().handle)
44
45 // -- ROTAS PRODUCT --
46 // router.post('/product', isAuthenticated, upload.single('file'), new CreateProductController().handle)
47 router.post('/product', isAuthenticated, upload.single('file'), new CreateProductController().handle)
48 router.get('/category/product', isAuthenticated, new ListByCategoryController().handle)
49
50 // -- ROTAS ORDER --
51
52 router.post('/order', isAuthenticated, new CreateOrderController().handle)
53 router.delete('/order', isAuthenticated, new RemoveOrderController().handle)
54
55 router.post('/order/add', isAuthenticated, new AddItemController().handle)
56 router.delete('/order/remove', isAuthenticated, new RemoveItemController().handle)
57 router.put('/order/send', isAuthenticated, new SendOrderController().handle)
58
59 router.get('/orders', isAuthenticated, new ListOrdersController().handle)
60 router.get('/order/detail', isAuthenticated, new DetailOrderController().handle)
61
62 router.put('/order/finish', isAuthenticated, new FinishOrderController().handle)
```

EXPLORER

... TS routes.ts X

src > TS routes.ts > ...

- > node\_modules
- > prisma
- ✓ src
  - > @types
  - > config
  - > controllers
  - > middlewares
  - > prisma
  - > services
- TS routes.ts
- TS server.ts
- > tmp
- > tmp
- ⚙ .env
- diamond .gitignore
- { package-lock.json
- { package.json
- tsconfig.json
- ≡ yarn-error.log
- blue lock yarn.lock

OUTLINE

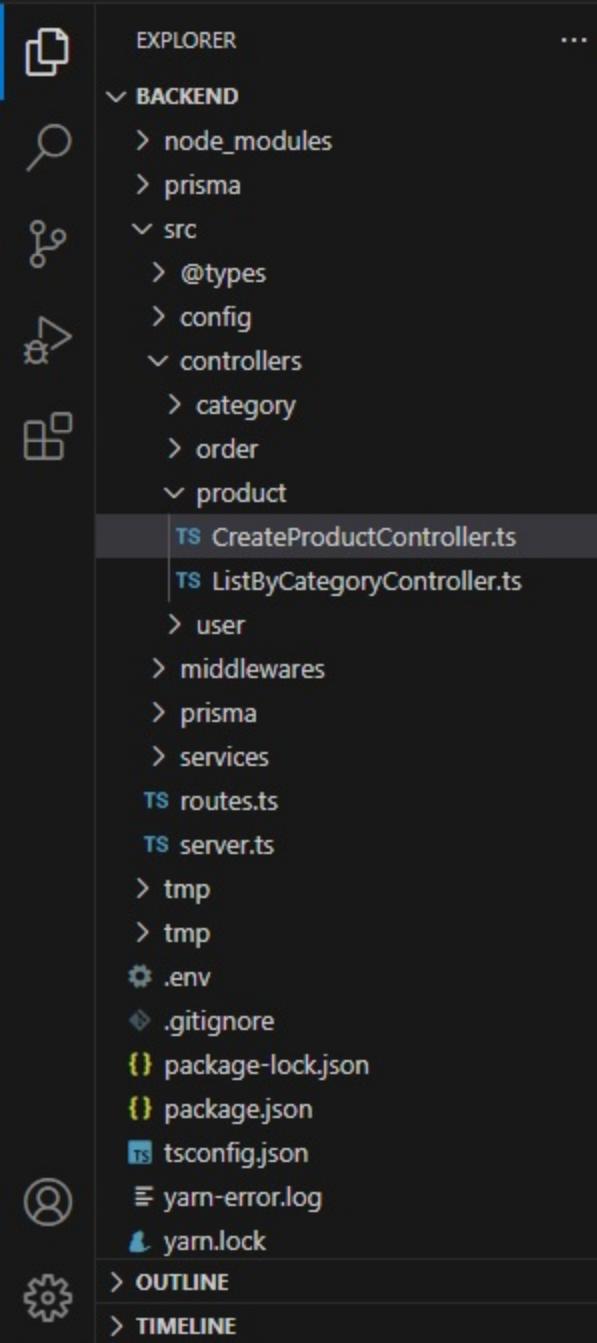
TIMELINE

```
31  const upload = multer(uploadConfig.upload("./tmp"))
32
33 // -- ROTAS USER --
34 router.post('/users', new CreateUserController().handle)
35
36 router.post('/session', new AuthUserController().handle)
37
38 router.get('/me', isAuthenticated, new DetailUserController().handle)
39
40 // -- ROTAS CATEGORY --
41 router.post('/category', isAuthenticated, new CreateCategoryController().handle)
42
43 router.get('/category', isAuthenticated, new ListCategoryController().handle)
44
45 // -- ROTAS PRODUCT --
46 // router.post('/product', isAuthenticated, upload.single('file'), new CreateProductController().handle)
47 router.post('/product', isAuthenticated, new CreateProductController().handle)
48 router.get('/category/product', isAuthenticated, new ListByCategoryController().handle)
49
50 // -- ROTAS ORDER --
51
52 router.post('/order', isAuthenticated, new CreateOrderController().handle)
53 router.delete('/order', isAuthenticated, new RemoveOrderController().handle)
54
55 router.post('/order/add', isAuthenticated, new AddItemController().handle)
56 router.delete('/order/remove', isAuthenticated, new RemoveItemController().handle)
57 router.put('/order/send', isAuthenticated, new SendOrderController().handle)
58
59 router.get('/orders', isAuthenticated, new ListOrdersController().handle)
60 router.get('/order/detail', isAuthenticated, new DetailOrderController().handle)
61
62 router.put('/order/finish', isAuthenticated, new FinishOrderController().handle)
```

The screenshot shows a code editor interface with a dark theme. The top navigation bar includes File, Edit, Selection, View, Go, Run, and a search bar containing "backend". The left sidebar has icons for Explorer, Search, Problems, and others. The Explorer panel shows a project structure under "BACKEND": node\_modules, prisma, src (containing @types, config, controllers, category, order, product), routes.ts, server.ts, tmp, .env, .gitignore, package-lock.json, package.json, tsconfig.json, yarn-error.log, and yarn.lock. The main editor area displays a TypeScript file named "CreateProductController.ts". The code implements a controller for creating products, using Express and a CreateProductService. It handles file uploads and returns a JSON response. A yellow warning icon is visible near the beginning of the file.

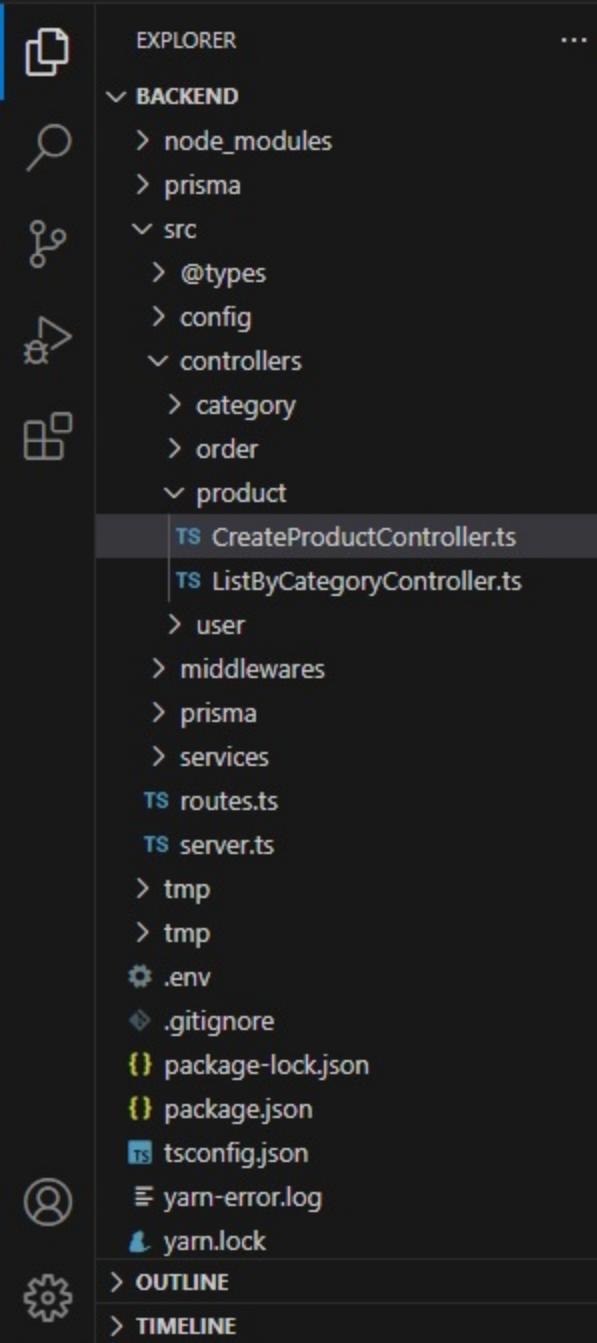
```
TS routes.ts TS CreateProductController.ts X
src > controllers > product > TS CreateProductController.ts > CreateProductController > handle
1 import { Request, Response } from "express";
2 import { CreateProductService } from "../../services/product/CreateProductService";
3
4 class CreateProductController{
5     async handle(req: Request, res: Response){
6         const { name, price, description, category_id } = req.body;
7
8         const createProductService = new CreateProductService();
9
10        if(!req.file){
11            throw new Error("error upload file")
12        } else {
13
14            const { originalname, filename: banner} = req.file;
15
16            const product = await createProductService.execute({
17                name,
18                price,
19                description,
20                banner,
21                category_id
22            });
23
24            res.json(product)
25        }
26    }
27}
28
29 export { CreateProductController }
```

Ln 10, Col 13 (8 selected) Spaces: 4 UTF-8 CRLF { TypeScript ✓ Prettier



TS routes.ts    TS CreateProductController.ts X

```
src > controllers > product > TS CreateProductController.ts > CreateProductController > handle
1 import { Request, Response } from "express";
2 import { CreateProductService } from "../../services/product/CreateProductService";
3
4 class CreateProductController{
5     async handle(req: Request, res: Response){
6         const { name, price, description, category_id } = req.body;
7
8         const createProductService = new CreateProductService();
9
10        if(!req.files || Object.keys(req.files).length === 0){
11            throw new Error("error upload file")
12        } else {
13
14            const { originalname, filename: banner} = req.file;
15
16            const product = await createProductService.execute({
17                name,
18                price,
19                description,
20                banner,
21                category_id
22            });
23
24            res.json(product)
25        }
26    }
27}
28
29 export { CreateProductController }
```



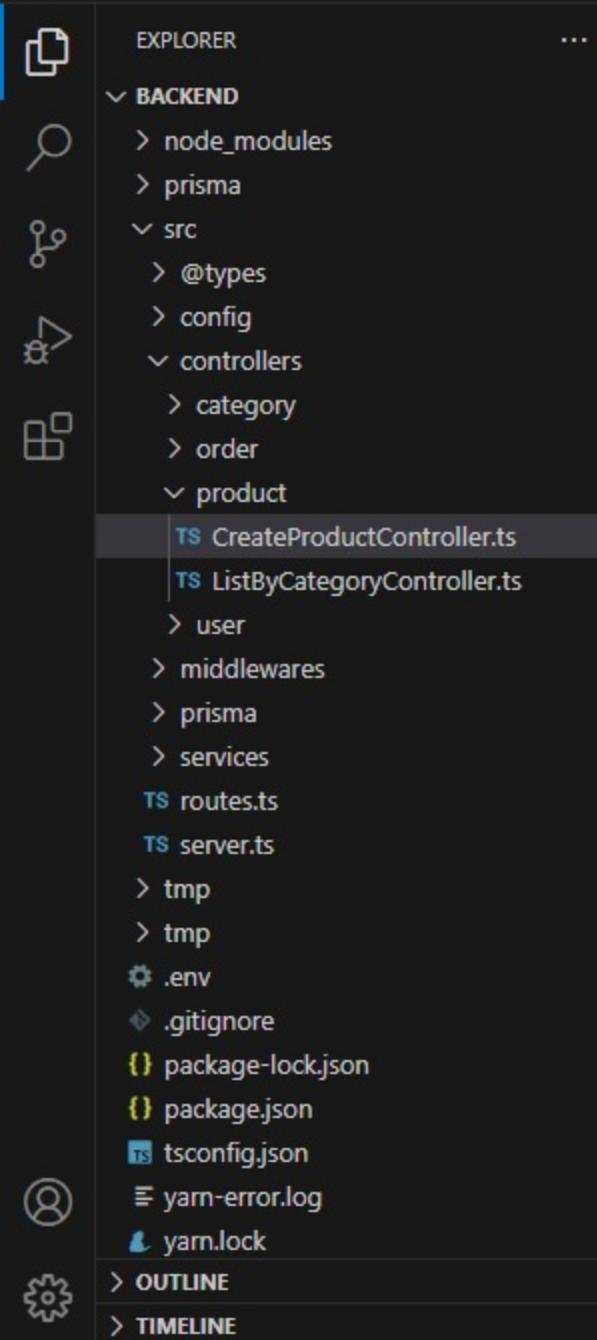
TS routes.ts TS CreateProductController.ts X

```
src > controllers > product > TS CreateProductController.ts > CreateProductController > handle
1 import { Request, Response } from "express";
2 import { CreateProductService } from "../../services/product/CreateProductService";
3
4 class CreateProductController{
5     async handle(req: Request, res: Response){
6         const { name, price, description, category_id } = req.body;
7
8         const createProductService = new CreateProductService();
9
10        if(!req.files || Object.keys(req.files).length === 0){
11            throw new Error("error upload file")
12        } else {
13
14            const { originalname, filename: banner} = req.file;
15
16            const product = await createProductService.execute({
17                name,
18                price,
19                description,
20                banner,
21                category_id
22            });
23
24            res.json(product)
25        }
26    }
27}
28
29 export { CreateProductController }
```

The screenshot shows a code editor interface with a dark theme. The top navigation bar includes File, Edit, Selection, View, Go, Run, and a search bar containing "backend". The left sidebar has icons for Explorer, Search, Problems, and others. The Explorer panel shows a project structure under "BACKEND": node\_modules, prisma, src (containing @types, config, controllers, category, order, product), routes.ts, server.ts, tmp, .env, .gitignore, package-lock.json, package.json, tsconfig.json, yarn-error.log, and yarn.lock. The main editor area displays a TypeScript file named "CreateProductController.ts". The code implements a controller for creating products, using Express and a CreateProductService. It handles file uploads and returns a JSON response. The file is 29 lines long and exports the CreateProductController.

```
TS routes.ts TS CreateProductController.ts X
src > controllers > product > TS CreateProductController.ts > CreateProductController > handle
1 import { Request, Response } from "express";
2 import { CreateProductService } from "../../services/product/CreateProductService";
3
4 class CreateProductController{
5     async handle(req: Request, res: Response){
6         const { name, price, description, category_id } = req.body;
7
8         const createProductService = new CreateProductService();
9
10        if(!req.files || Object.keys(req.files).length === 0){
11            throw new Error("error upload file image")
12        } else {
13
14            const { originalname, filename: banner} = req.file;
15
16            const product = await createProductService.execute({
17                name,
18                price,
19                description,
20                banner,
21                category_id
22            });
23
24            res.json(product)
25        }
26    }
27}
28
29 export { CreateProductController }
```

Ln 11, Col 53 (5 selected) Spaces: 4 UTF-8 CRLF { TypeScript ✓ Prettier



TS routes.ts    TS CreateProductController.ts X

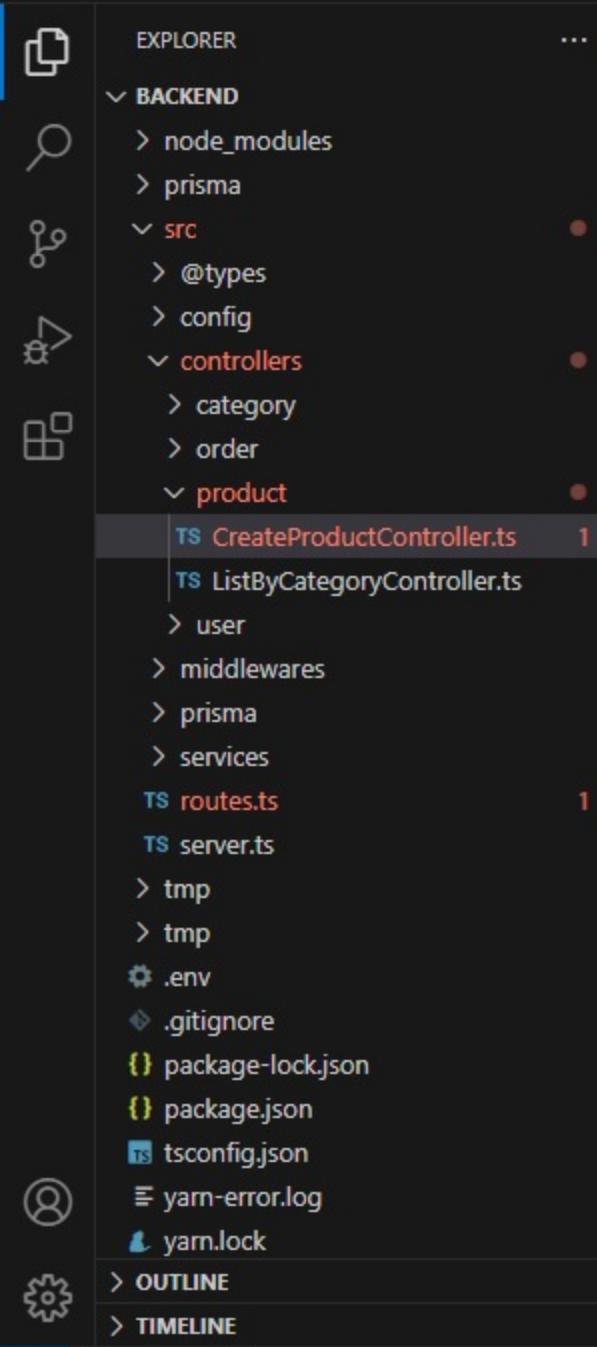
```
src > controllers > product > TS CreateProductController.ts > CreateProductController > handle
1 import { Request, Response } from "express";
2 import { CreateProductService } from "../../services/product/CreateProductService";
3
4 class CreateProductController{
5     async handle(req: Request, res: Response){
6         const { name, price, description, category_id } = req.body;
7
8         const createProductService = new CreateProductService();
9
10        if(!req.files || Object.keys(req.files).length === 0){
11            throw new Error("error upload file image")
12        } else {
13
14            const { originalname, filename: banner} = req.file;
15
16            const product = await createProductService.execute({
17                name,
18                price,
19                description,
20                banner,
21                category_id
22            );
23
24            res.json(product)
25        }
26    }
27}
28
29 export { CreateProductController }
```

The screenshot shows a dark-themed code editor interface with the following details:

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** backend
- Left Sidebar (Explorer):**
  - EXPLORER
  - BACKEND
    - node\_modules
    - prisma
    - src
      - @types
      - config
      - controllers
        - category
        - order
        - product
          - CreateProductController.ts (selected)
          - ListByCategoryController.ts
    - middlewares
    - prisma
    - services
    - routes.ts
    - server.ts
    - tmp
    - .env
    - .gitignore
    - package-lock.json
    - package.json
    - tsconfig.json
    - yarn-error.log
    - yarn.lock
  - OUTLINE
  - TIMELINE- Central Area:** TS CreateProductController.ts 1 X
- Code Content:**

```
src > controllers > product > TS CreateProductController.ts > CreateProductController > handle
1 import { Request, response, Response } from "express";
2 import { CreateProductService } from "../../services/product/CreateProductService";
3 import { UploadedFile } from "express-fileupload";

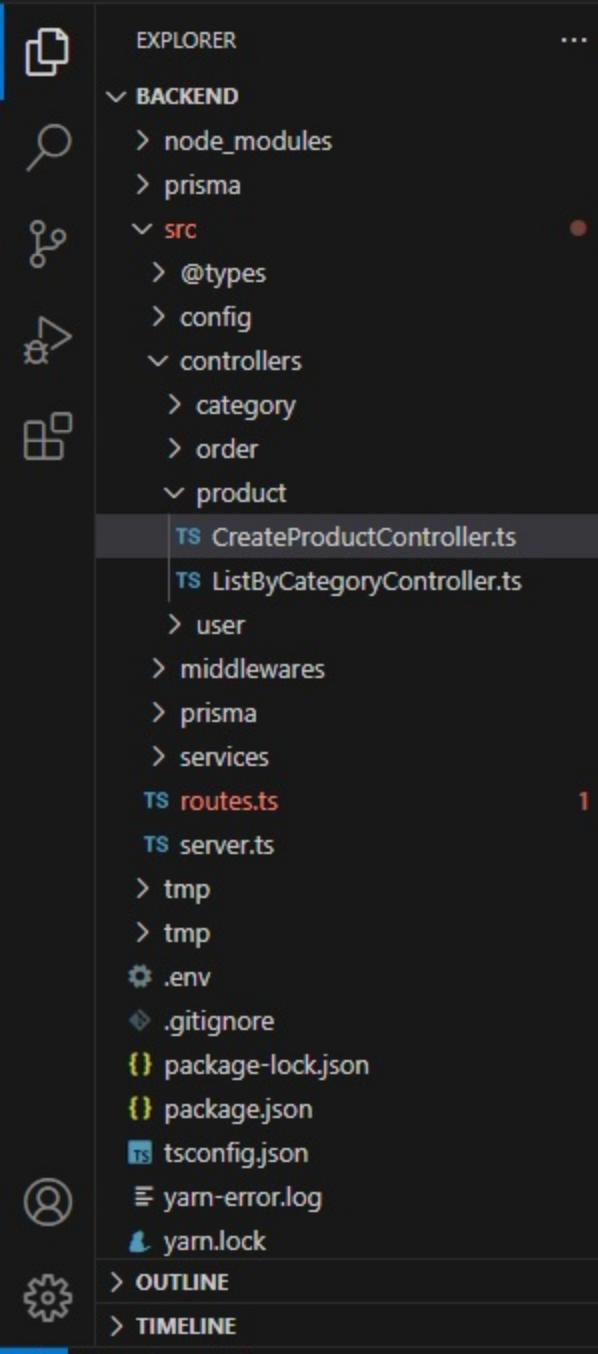
4
5 class CreateProductController{
6     async handle(req: Request, res: Response){
7         const { name, price, description, category_id } = req.body;
8
9         const createProductService = new CreateProductService();
10
11        if(!req.files || Object.keys(req.files).length === 0){
12            throw new Error("error upload file image")
13        } else {
14
15            const file:UploadedFile = req.files['file']
16
17            console.log(file);
18
19            return response.json({})
20
21            const product = await createProductService.execute({
22                name,
23                price,
24                description,
25                banner,
26                category_id
27            });
28
29            res.json(product)
30        }
31    }
32 }
```
- Bottom Status Bar:** 4 selections (135 characters selected) Spaces: 4 CRLF { TypeScript ✓ Prettier



TS routes.ts 1 TS CreateProductController.ts 1

```
src > controllers > product > TS CreateProductController.ts > CreateProductController > handle
1 import { Request, response, Response } from "express";
2 import { CreateProductService } from "../../services/product/CreateProductService";
3 import { UploadedFile } from "express-fileupload";

4
5 class CreateProductController{
6     async handle(req: Request, res: Response){
7         const { name, price, description, category_id } = req.body;
8
9         const createProductService = new CreateProductService();
10
11        if(!req.files || Object.keys(req.files).length === 0){
12            throw new Error("error upload file image")
13        } else {
14
15            const file:UploadedFile = req.files['file']
16
17            console.log(file);
18
19            return response.json({})
20
21        const product = await createProductService.execute({
22            name,
23            price,
24            description,
25            banner,
26            category_id
27        });
28
29        res.json(product)
30    }
31}
32}
```



## TS routes.ts 1 TS CreateProductController.ts X

```
src > controllers > product > TS CreateProductController.ts > CreateProductController > handle
1 import { Request, response, Response } from "express";
2 import { CreateProductService } from "../../services/product/CreateProductService";
3 import { UploadedFile } from "express-fileupload";
4
5 class CreateProductController{
6     async handle(req: Request, res: Response){
7         const { name, price, description, category_id } = req.body;
8
9         const create ProductService = new Create ProductService();
10
11        if(!req.files || Object.keys(req.files).length === 0){
12            throw new Error("error upload file image")
13        } else {
14
15            const file:UploadedFile = req.files['file']
16
17            console.log(file);
18
19            return response.json({})
20
21        //.....const product = await create ProductService.execute({
22        //.....name,
23        //.....price,
24        //.....description,
25        //.....banner,
26        //.....category_id
27        //....});
28
29        //....res.json(product)
30    }
31}
32}
```

Administrator: C:\Windows\system32\cmd.exe

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

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

[INFO] 19:40:45 ts-node-dev ver. 2.0.0 (using ts-node ver. 10.9.2, typescript ver. 5.6.3)
Servidor online!!!!
```

```
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 3.2s
```

**Novo produto**

bebidas



Digite o nome do produto...

Preço do produto...

Digite a descrição do produto...

**Cadastrar produto**

Google Chrome Abrir

Área de Trabalho > arquivos

Pesquisar em arquivos

Organizar Nova pasta

hamburger 1

açaí

açaí

hamburger duplo

hamburger triplo

hamburger

sorvete 1

Categoria Produto [→]

Nome: hamburger 1 Arquivos personalizados

Abrir Cancelar

Digite o nome do produto...

Preço do produto...

Digite a descrição do produto...

Cadastrar produto

Search Star Camera Download Print More

The image shows a dual-screen setup. The left screen displays a file explorer window titled 'Área de Trabalho > arquivos' (Desktop > files) with a search bar 'Pesquisar em arquivos'. It lists several files: 'açaí' (two images), 'hamburger 1' (selected and highlighted with a blue border), 'hamburger duplo', 'hamburger triplo', 'hamburger', and 'sorvete 1'. The right screen shows a dark-themed application window for product management. At the top, there are buttons for 'Categoria' and 'Produto' with a right arrow, and standard file operations like search, star, camera, download, print, and more. Below these are three input fields: 'Nome' (Name) set to 'hamburger 1', 'Arquivos personalizados' (Personalized files), 'Abrir' (Open) button, and 'Cancelar' (Cancel) button. Below the input fields are three text input boxes: 'Digite o nome do produto...' (Type product name...), 'Preço do produto...' (Product price...), and 'Digite a descrição do produto...' (Type product description...). A large green button at the bottom is labeled 'Cadastrar produto' (Register product).

## Novo produto



bebidas

Digite o nome do produto...

Preço do produto...

Digite a descrição do produto...

Cadastrar produto

## Novo produto



hamburger

Hamburger Simples

25

Hamburger Simples nº 1

**Cadastrar produto**

## Novo produto



hamburger

Hamburger Simples

25

Hamburger Simples nº 1

Carregando...

Produto Registrado com Sucesso

**Últimos pedidos** 

Mesa 3

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

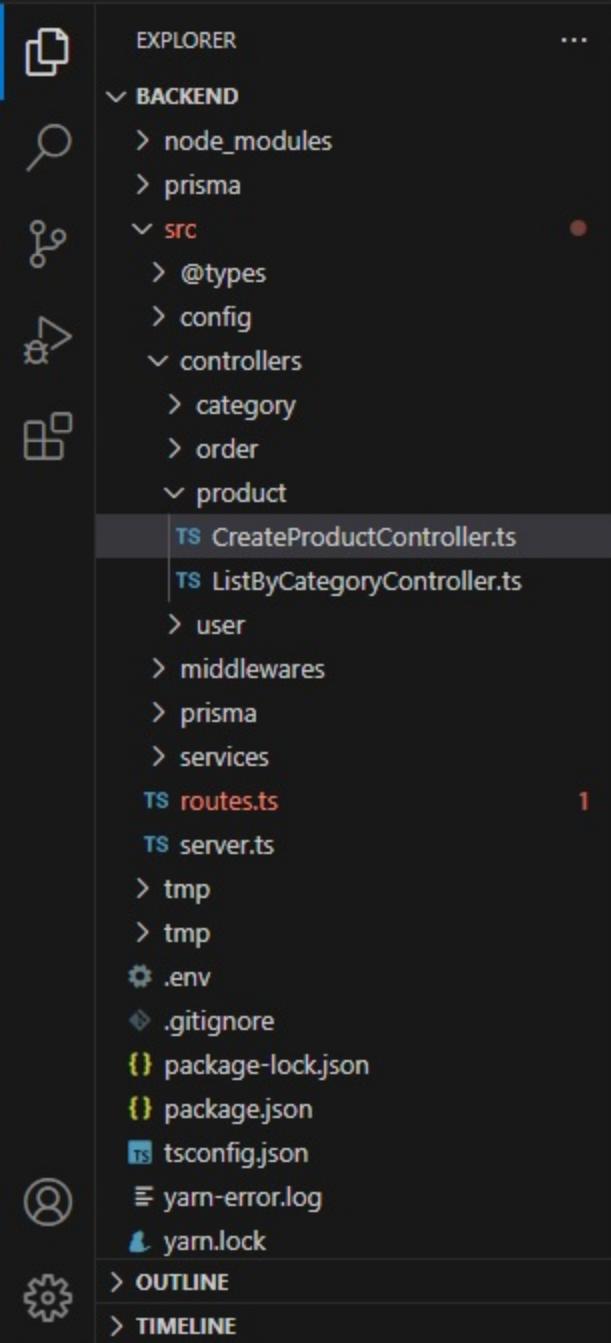
```
[INFO] 19:40:45 ts-node-dev ver. 2.0.0 (using ts-node ver. 10.9.2, typescript ver. 5.6.3)
```

```
Servidor online!!!!
```

```
{
  name: 'hamburger 1.jpg',
  data: <Buffer ff d8 ff e0 00 10 4a 46 49 46 00 01 01 00 00 01 00 01 00 00 ff db 00 84 00 09 06 07 13 12 12 13 12 13 12
13 15 15 17 17 1a 1a 17 17 17 17 18 17 1a 18 ... 5798 more bytes>,
  size: 5848,
  encoding: '7bit',
  tempFilePath: '',
  truncated: false,
  mimetype: 'image/jpeg',
  md5: '0eb2d8983b5458fef486fc78bde87cda',
  mv: [Function: mv]
}
```

```
Error: Cannot set headers after they are sent to the client
```

```
  at ServerResponse.setHeader (node:_http_outgoing:703:11)
  at ServerResponse.header (c:\Users\Dev\Desktop\Pizzaria\backend\node_modules\express\lib\response.js:794:10)
  at ServerResponse.send (c:\Users\Dev\Desktop\Pizzaria\backend\node_modules\express\lib\response.js:174:12)
  at ServerResponse.json (c:\Users\Dev\Desktop\Pizzaria\backend\node_modules\express\lib\response.js:278:15)
  at c:\Users\Dev\Desktop\Pizzaria\backend\src\server.ts:31:21
  at newFn (c:\Users\Dev\Desktop\Pizzaria\backend\node_modules\express-async-errors\index.js:16:20)
  at Layer.handle_error (c:\Users\Dev\Desktop\Pizzaria\backend\node_modules\express\lib\router\layer.js:71:5)
  at trim_prefix (c:\Users\Dev\Desktop\Pizzaria\backend\node_modules\express\lib\router\index.js:326:13)
  at c:\Users\Dev\Desktop\Pizzaria\backend\node_modules\express\lib\router\index.js:286:9
  at Function.process_params (c:\Users\Dev\Desktop\Pizzaria\backend\node_modules\express\lib\router\index.js:346:12)
```



TS routes.ts 1 TS CreateProductController.ts X

```
src > controllers > product > TS CreateProductController.ts > CreateProductController > handle
1 import { Request, response, Response } from "express";
2 import { CreateProductService } from "../../services/product/CreateProductService";
3 import { UploadedFile } from "express-fileupload";
4
5 class CreateProductController{
6     async handle(req: Request, res: Response){
7         const { name, price, description, category_id } = req.body;
8
9         const createProductService = new CreateProductService();
10
11        if(!req.files || Object.keys(req.files).length === 0){
12            throw new Error("error upload file image")
13        } else {
14
15            const file:UploadedFile = req.files['file']
16
17            console.log(file);
18
19            return response.json({})
20
21        //     const product = await createProductService.execute(
22        //         name,
23        //         price,
24        //         description,
25        //         banner,
26        //         category_id
27        // );
28
29        // res.json(product)
30    }
31}
32}
```

dgshomtoi ▼Product environment settings ^

API Keys



Upload Settings &amp; Backup



Upload Presets



Optimization



Webhook Notifications



Security

Account settings ^

My Profile



Account



Product Environments



User Management



Account Security

Billing ^

## API Keys

Cloud name: dgshomtoi

Manage API key and secret pairs for your product environment. To build the environment variable for each pair, copy the provided format and substitute your actual values for the placeholders. Make sure to store your secrets securely.

+ Generate New API Key

API environment variable

CLOUDINARY\_URL=cloudinary://<your\_api\_key>:<your\_api\_secret>@dgshomtoi copy

Key Name	Date Created	API Key	API Secret	Status
Untitled	Dec 05, 2024	891882163577121	vI9KPAJ3xxG2a...	<span>Active</span>

Page 1 of 1 < >



dgshomtoi

Assets  
Digital asset management

Upload Settings &amp; Backup



Upload Presets



Optimization



Webhook Notifications



Security



Account settings



My Profile



Account



Product Environments



User Management



Account Security



Billing



## API Keys

Cloud name: dgshomtoi

Manage API key and secret pairs for your product environment. To build the environment variable for each pair, copy the provided format and substitute your actual values for the placeholders. Make sure to store your secrets securely.

[+ Generate New API Key](#)

API environment variable

CLOUDINARY\_URL=cloudinary://&lt;your\_api\_key&gt;:&lt;your\_api\_secret&gt;@dgshomtoi



Key Name	Date Created	API Key	API Secret	Status
Untitled	Dec 05, 2024	891882163577121	vI9KPAJ3xxG2a...	<input checked="" type="checkbox"/> Active

Page 1 of 1 &lt; &gt;

console.cloudinary.com/console/c-b217595c7f0a2d2f171a95fcfa7f99/media\_library/homepage

**Media Library** Home Assets Folders Collections Moderation **Upload** ...

Welcome! 20

**Media Library**

Studio

App Marketplace

Structured Metadata

Dashboard

Preferences

Let's Start Discovering

Search Media Library

What do you want to do today?

 **Adapt your images to different devices**  
[Resize & Crop](#)

 **Prepare your products for publishing**  
[Background replace](#)

 **Brand your images**

 **Remove irrelevant items in the image**  
1

← → C https://console.cloudinary.com/console/c-b217595c7f0a2d2f171a95fcfa7f99/media\_library/search?q=&view\_mode=mosaic

dgshomtoi

Welcome! 20

Media Library

Studio

App Marketplace

Structured Metadata

Dashboard

Preferences

Media Library

Home Assets Folders Collections Moderation

Search Media Library

Upload

Advanced Search

Type to filter

Display name Folders Creation date Tags Formats Asset types Orientations

+ Add more

Refresh Sort by: Relevance View Settings Preview X

Showing 57 assets



Select an item in the grid to preview it

1

console.cloudinary.com/console/c-b217595c7f0a2d2f171a95fcfa7f99/media\_library/search?q=&view\_mode=mosaic

dgshomtoi

Welcome!

Media Library

Studio

App Marketplace

Structured Metadata

Dashboard

Preferences

Media Library

Assets

Folders

Collections

Moderation

Search Media Library

Upload

Advanced Search

Type to filter

Display name

Folders

Creation date

Tags

Formats

Asset types

Orientations

Add more

Refresh

Sort by: Relevance

View

Settings

Mosaic

List

Card

Showing 57 assets

Preview

Select an item in the grid to preview it

1

dgshomtoi

Welcome!

Media Library

Studio

App Marketplace

Structured Metadata

Dashboard

Preferences

# Media Library

Home Assets Folders Collections Moderation

Search Media Library

Upload

Advanced Search

Type to filter

Display name Folders Creation date Tags Formats Asset types Orientations Add more

Refresh View Settings Preview X

Assets (57)

Display name	Containing folder	Asset type	Format	Size
cld-sample-5	Home	Image	JPG	370
cld-sample-4	Home	Image	JPG	799
cld-sample-3	Home	Image	JPG	884
cld-sample-2	Home	Image	JPG	578
cld-sample	Home	Image	JPG	465
Logo	Home	Image	PNG	10.6

Select an item in the grid to preview it

1

EXPLORER

BACKEND

- > node\_modules
- > prisma
- src
  - > @types
  - > config
  - controllers
    - > category
    - > order
    - product
      - TS CreateProductController.ts
      - TS ListByCategoryController.ts
- > user
- > middlewares
- > prisma
- > services
- TS routes.ts
- TS server.ts
- > tmp
- > tmp
- .env
- .gitignore
- { package-lock.json
- { package.json
- tsconfig.json
- yarn-error.log
- yarn.lock

OUTLINE

TIMELINE

TS routes.ts 1 TS CreateProductController.ts X

```
src > controllers > product > TS CreateProductController.ts > ...
1 import { Request, response, Response } from "express";
2 import { CreateProductService } from "../../services/product/CreateProductService";
3 import { UploadedFile } from "express-fileupload";
4
5 import { v2 as cloudinary, UploadApiResponse } from 'cloudinary'
6
7 cloudinary.config({
8   cloud_name: process.env.CLOUDINARY_NAME,
9   api_key: process.env.CLOUDINARY_KEY,
10  api_secret: process.env.CLOUDINARY_SECRET,
11})
12
13 class CreateProductController{
14   async handle(req: Request, res: Response){
15     const { name, price, description, category_id } = req.body;
16
17     const createProductService = new CreateProductService();
18
19     if(!req.files || Object.keys(req.files).length === 0){
20       throw new Error("error upload file image")
21     } else {
22
23       const file:UploadedFile = req.files['file']
24
25       console.log(file);
26
27       return response.json({})
28
29      // const product = await createProductService.execute({
30      //   name,
31      //   price,
32      //   description,
```

2 selections (223 characters selected) Spaces: 4 UTF-8 CRLF ⓘ TypeScript ✅ Prettier

The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** backend
- Toolbars:** Standard window controls (minimize, maximize, close) are visible in the top right.
- Left Sidebar (EXPLORER):** Shows the project structure:
  - BACKEND
  - node\_modules
  - prisma
  - src
    - @types
    - config
    - controllers
      - category
      - order
    - product
      - CreateProductController.ts (selected)
      - ListByCategoryController.ts
    - user
    - middlewares
    - prisma
    - services
  - routes.ts
  - server.ts
  - tmp
  - .env
  - .gitignore
  - package-lock.json
  - package.json
  - tsconfig.json
  - yarn-error.log
  - yarn.lock
- Right Panel:** The main editor area displays the content of `CreateProductController.ts`:

```
api_key: process.env.CLOUDINARY_KEY,
api_secret: process.env.CLOUDINARY_SECRET,
}

class CreateProductController{
    async handle(req: Request, res: Response){
        const { name, price, description, category_id } = req.body;

        const create ProductService = new Create ProductService();

        if(!req.files || Object.keys(req.files).length === 0){
            throw new Error("error upload file image")
        } else {

            const file: UploadedFile = req.files['file']

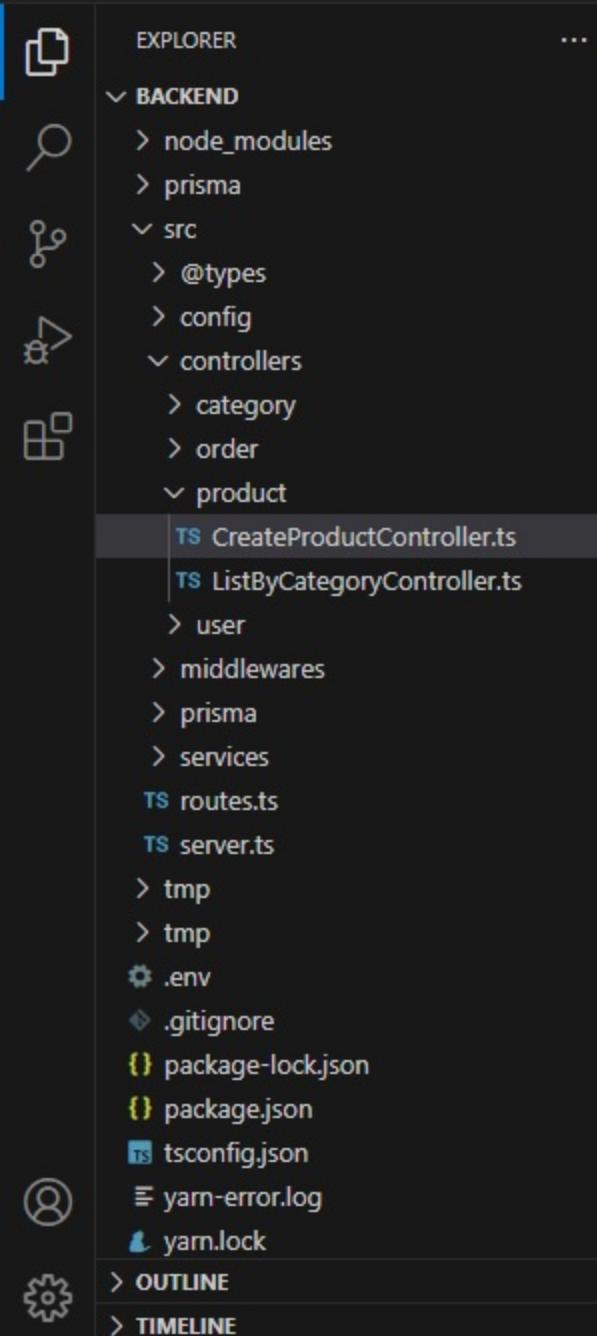
            console.log(file);

            return response.json({})
        }
    }
}
```
- Bottom Status Bar:** Ln 25, Col 13 (18 selected) | Spaces: 4 | UTF-8 | CRLF | {TypeScript} | Prettier

The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** backend
- Toolbars:** Standard window controls (minimize, maximize, close) are visible in the top right.
- Left Sidebar (EXPLORER):** Shows the project structure:
  - BACKEND
  - node\_modules
  - prisma
  - src
    - @types
    - config
    - controllers
      - category
      - order
      - product
        - CreateProductController.ts (selected)
        - ListByCategoryController.ts
    - middlewares
    - prisma
    - services
    - routes.ts
    - server.ts
    - tmp
    - .env
    - .gitignore
    - package-lock.json
    - package.json
    - tsconfig.json
    - yarn-error.log
    - yarn.lock
  - OUTLINE
  - TIMELINE
- Active Editor:** TS CreateProductController.ts (Line 1, Column 1)
- Code Content:**

```
src > controllers > product > TS CreateProductController.ts > CreateProductController > handle
    9     api_key: process.env.CLOUDINARY_KEY,
    10    api_secret: process.env.CLOUDINARY_SECRET,
    11  }
    12
    13 class CreateProductController{
    14     async handle(req: Request, res: Response){
    15         const { name, price, description, category_id } = req.body;
    16
    17         const create ProductService = new Create ProductService();
    18
    19         if(!req.files || Object.keys(req.files).length === 0){
    20             throw new Error("error upload file image")
    21         } else {
    22
    23             const file: UploadedFile = req.files['file']
    24
    25             console.log(file);
    26
    27             return response.json({})
    28
    29             // const product = await create ProductService.execute(
    30             //     name,
    31             //     price,
    32             //     description,
    33             //     banner,
    34             //     category_id
    35             // );
    36
    37             // res.json(product)
    38         }
    39     }
    40 }
```
- Bottom Status Bar:** 2 selections (42 characters selected), Spaces: 4, UTF-8, CRLF, {TypeScript}, Prettier



TS routes.ts    TS CreateProductController.ts X

```
src > controllers > product > TS CreateProductController.ts > CreateProductController > handle
```

```
12
13 class CreateProductController{
14     async handle(req: Request, res: Response){
15         const { name, price, description, category_id } = req.body;
16
17         const create ProductService = new Create ProductService();
18
19         if(!req.files || Object.keys(req.files).length === 0){
20             throw new Error("error upload file image")
21         } else {
22
23             const file: UploadedFile = req.files['file']
24
25             const resultFile = await new Promise((resolve, reject) => {
26                 cloudinary.uploader.upload_stream({}, function (error, result) {
27                     if(error){
28                         reject(error);
29                     }
30                 })
31
32                     resolve(result)
33                     .end(file.data)
34             })
35
36             console.log(resultFile);
37
38             response.json({})
39
40             // const product = await create ProductService.execute({
41             //     name,
42             //     price,
43             //     description,
```

Administrator: C:\Windows\system32\cmd.exe

been modified  
Servidor online!!!!  
[INFO] 20:05:50 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\controllers\product\CreateProductController.ts has been modified  
Servidor online!!!!  
[INFO] 20:05:51 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\controllers\product\CreateProductController.ts has been modified  
Servidor online!!!!  
[INFO] 20:05:52 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\controllers\product\CreateProductController.ts has been modified  
Compilation error in c:\Users\Dev\Desktop\Pizzaria\backend\src\controllers\product\CreateProductController.ts  
[ERROR] 20:05:53 Unable to compile TypeScript:  
src/controllers/product/CreateProductController.ts(49,9): error TS1003: Identifier expected.  
  
[INFO] 20:05:54 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\controllers\product\CreateProductController.ts has been modified  
Servidor online!!!!  
[INFO] 20:05:55 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\controllers\product\CreateProductController.ts has been modified  
Servidor online!!!!  
[INFO] 20:05:58 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\controllers\product\CreateProductController.ts has been modified  
Servidor online!!!!  
[INFO] 20:07:01 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\controllers\product\CreateProductController.ts has been modified  
Servidor online!!!!  
[INFO] 20:07:14 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\controllers\product\CreateProductController.ts has been modified  
Servidor online!!!!

```
[ERROR] 20:05:53 Unable to compile TypeScript:  
src/controllers/product/CreateProductController.ts(49,9): error TS1003: Identifier expected.  
  
[INFO] 20:05:54 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\controllers\product\CreateProductController.ts has  
been modified  
Servidor online!!!!  
[INFO] 20:05:55 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\controllers\product\CreateProductController.ts has  
been modified  
Servidor online!!!!  
[INFO] 20:05:58 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\controllers\product\CreateProductController.ts has  
been modified  
Servidor online!!!!  
[INFO] 20:07:01 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\controllers\product\CreateProductController.ts has  
been modified  
Servidor online!!!!  
[INFO] 20:07:14 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\controllers\product\CreateProductController.ts has  
been modified  
Servidor online!!!!  
Deseja finalizar o arquivo em lotes (S/N)?  
^C  
c:\Users\Dev\Desktop\Pizzaria\backend>  
c:\Users\Dev\Desktop\Pizzaria\backend>  
c:\Users\Dev\Desktop\Pizzaria\backend>  
c:\Users\Dev\Desktop\Pizzaria\backend>
```

Administrator: C:\Windows\system32\cmd.exe

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

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

[INFO] 20:09:25 ts-node-dev ver. 2.0.0 (using ts-node ver. 10.9.2, typescript ver. 5.6.3)  
Servidor online!!!!



# SujeitoPizza

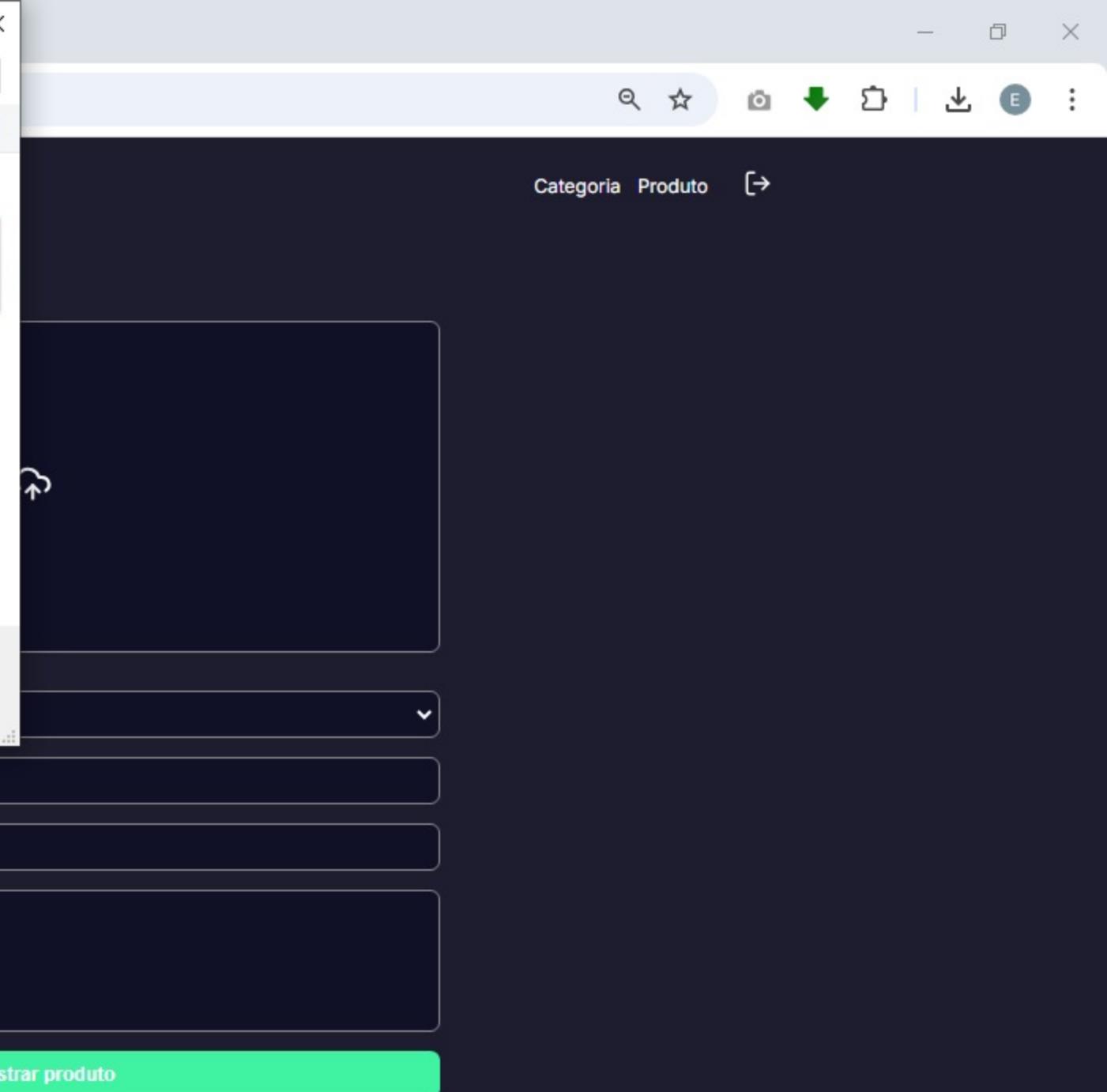
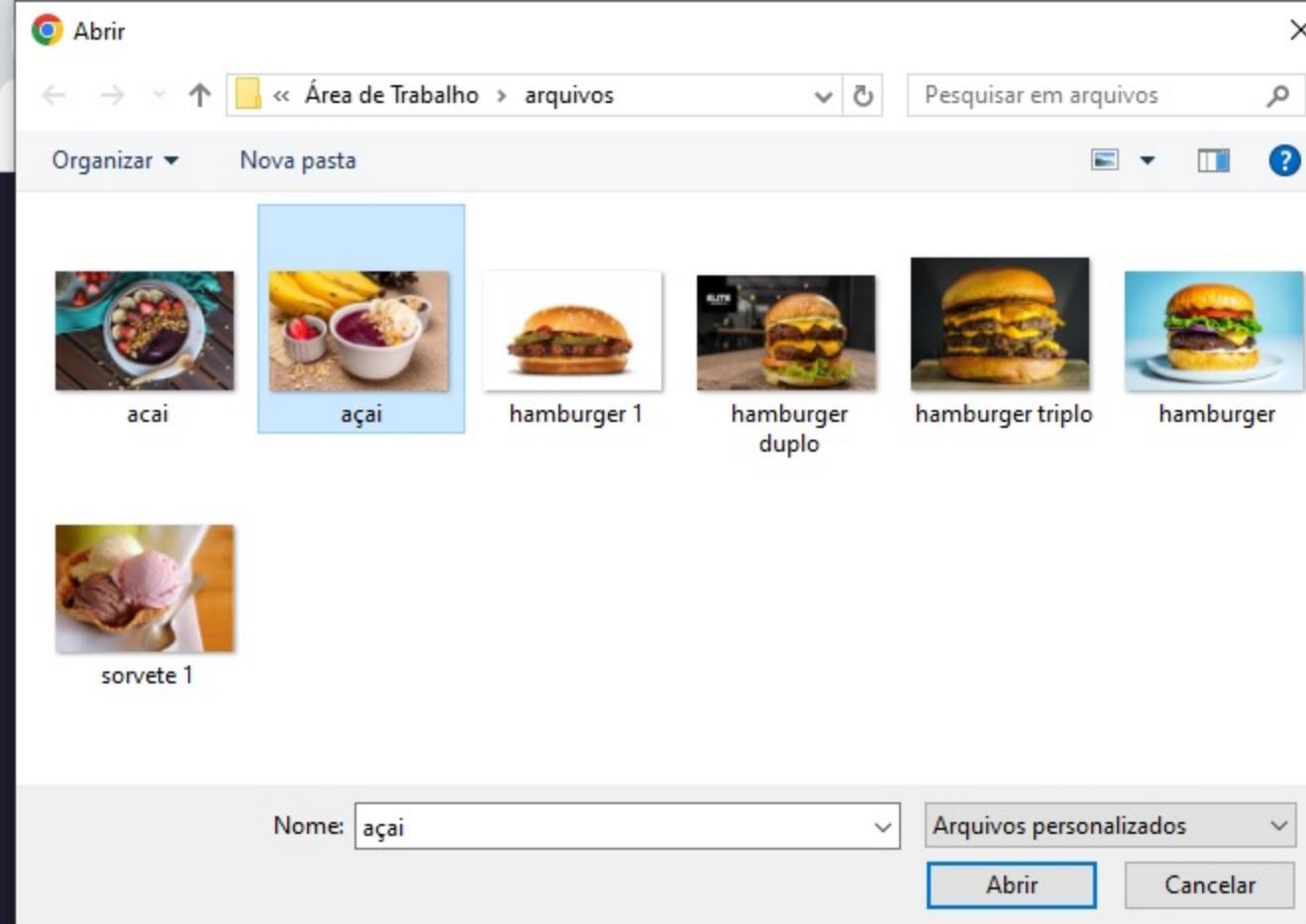
Categoria Produto [→]

## Últimos pedidos

Mesa 35

## Novo produto

[Cadastrar produto](#)



## Novo produto



sobremesas



Açaí

35

Açaí Pequeno

**Cadastrar produto**

# SujeitoPizza

[Categoria](#) [Produto](#) [→]

## Últimos pedidos



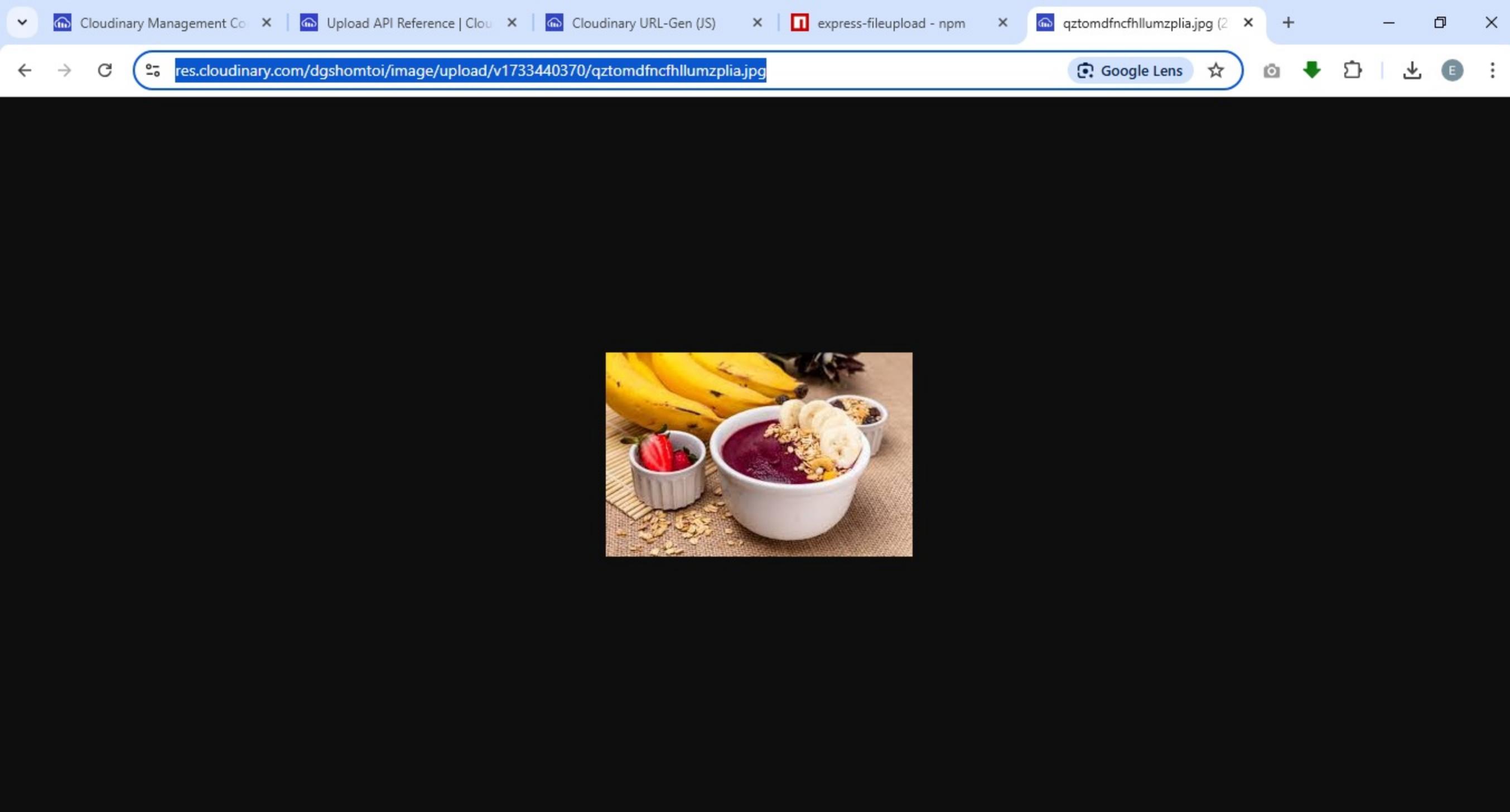
Mesa 35

✓ Produto Registrado com Sucesso

```
Seleccionar Administrador: C:\Windows\system32\cmd.exe
— □ X
Servidor online!!!!
{
  asset_id: 'e276c9ae01f9936370ea52f4c8e7d21d',
  public_id: 'qztomdfncfhllumzplia',
  version: 1733440370,
  version_id: '60da456ada30d7f8f10b52a808a492bb',
  signature: '3d786fbe3015ce9ff2e766c31c10df2137782d01',
  width: 275,
  height: 183,
  format: 'jpg',
  resource_type: 'image',
  created_at: '2024-12-05T23:12:50Z',
  tags: [],
  bytes: 11695,
  type: 'upload',
  etag: '173fe1e00ce22c63f059002b3c70e841',
  placeholder: false,
  url: 'http://res.cloudinary.com/dgshomtoi/image/upload/v1733440370/qztomdfncfhllumzplia.jpg',
  secure_url: 'https://res.cloudinary.com/dgshomtoi/image/upload/v1733440370/qztomdfncfhllumzplia.jpg',
  asset_folder: '',
  display_name: 'qztomdfncfhllumzplia',
  original_filename: 'file',
  api_key: '891882163577121'
}
Error: Cannot set headers after they are sent to the client
  at ServerResponse.setHeader (node:_http_outgoing:703:11)
  at ServerResponse.header (c:\Users\Dev\Desktop\Pizzaria\backend\node_modules\express\lib\response.js:794:10)
  at ServerResponse.send (c:\Users\Dev\Desktop\Pizzaria\backend\node_modules\express\lib\response.js:174:12)
  at ServerResponse.json (c:\Users\Dev\Desktop\Pizzaria\backend\node_modules\express\lib\response.js:278:15)
  at c:\Users\Dev\Desktop\Pizzaria\backend\src\server.ts:31:21
  at newFn (c:\Users\Dev\Desktop\Pizzaria\backend\node_modules\express-async-errors\index.js:16:20)
  at Layer.handle_error (c:\Users\Dev\Desktop\Pizzaria\backend\node_modules\express\lib\router\layer.js:71:5)
  at trim_prefix (c:\Users\Dev\Desktop\Pizzaria\backend\node_modules\express\lib\router\index.js:326:13)
  at c:\Users\Dev\Desktop\Pizzaria\backend\node_modules\express\lib\router\index.js:286:9
  at Function.process_params (c:\Users\Dev\Desktop\Pizzaria\backend\node_modules\express\lib\router\index.js:346:12)
```

```
Seleccionar Administrador: C:\Windows\system32\cmd.exe
Servidor online!!!!
{
  asset_id: 'e276c9ae01f9936370ea52f4c8e7d21d',
  public_id: 'qztomdfncfhllumzplia',
  version: 1733440370,
  version_id: '60da456ada30d7f8f10b52a808a492bb',
  signature: '3d786fbe3015ce9ff2e766c31c10df2137782d01',
  width: 275,
  height: 183,
  format: 'jpg',
  resource_type: 'image',
  created_at: '2024-12-05T23:12:50Z',
  tags: [],
  bytes: 11695,
  type: 'upload',
  etag: '173fe1e00ce22c63f059002b3c70e841',
  placeholder: false,
  url: 'http://res.cloudinary.com/dgshomtoi/image/upload/v1733440370/qztomdfncfhllumzplia.jpg',
  secure_url: 'https://res.cloudinary.com/dgshomtoi/image/upload/v1733440370/qztomdfncfhllumzplia.jpg',
  asset_folder: '',
  display_name: 'qztomdfncfhllumzplia',
  original_filename: 'file',
  api_key: '891882163577121'
}
Error: Cannot set headers after they are sent to the client
  at ServerResponse.setHeader (node:_http_outgoing:703:11)
  at ServerResponse.header (c:\Users\Dev\Desktop\Pizzaria\backend\node_modules\express\lib\response.js:794:10)
  at ServerResponse.send (c:\Users\Dev\Desktop\Pizzaria\backend\node_modules\express\lib\response.js:174:12)
  at ServerResponse.json (c:\Users\Dev\Desktop\Pizzaria\backend\node_modules\express\lib\response.js:278:15)
  at c:\Users\Dev\Desktop\Pizzaria\backend\src\server.ts:31:21
  at newFn (c:\Users\Dev\Desktop\Pizzaria\backend\node_modules\express-async-errors\index.js:16:20)
  at Layer.handle_error (c:\Users\Dev\Desktop\Pizzaria\backend\node_modules\express\lib\router\layer.js:71:5)
  at trim_prefix (c:\Users\Dev\Desktop\Pizzaria\backend\node_modules\express\lib\router\index.js:326:13)
  at c:\Users\Dev\Desktop\Pizzaria\backend\node_modules\express\lib\router\index.js:286:9
  at Function.process_params (c:\Users\Dev\Desktop\Pizzaria\backend\node_modules\express\lib\router\index.js:346:12)
```

```
Seleccionar Administrador: C:\Windows\system32\cmd.exe
Servidor online!!!!
{
  asset_id: 'e276c9ae01f9936370ea52f4c8e7d21d',
  public_id: 'qztomdfncfhllumzplia',
  version: 1733440370,
  version_id: '60da456ada30d7f8f10b52a808a492bb',
  signature: '3d786fbe3015ce9ff2e766c31c10df2137782d01',
  width: 275,
  height: 183,
  format: 'jpg',
  resource_type: 'image',
  created_at: '2024-12-05T23:12:50Z',
  tags: [],
  bytes: 11695,
  type: 'upload',
  etag: '173fe1e00ce22c63f059002b3c70e841',
  placeholder: false,
  url: 'http://res.cloudinary.com/dgshomtoi/image/upload/v1733440370/qztomdfncfhllumzplia.jpg',
  secure_url: 'https://res.cloudinary.com/dgshomtoi/image/upload/v1733440370/qztomdfncfhllumzplia.jpg',
  asset_folder: '',
  display_name: 'qztomdfncfhllumzplia',
  original_filename: 'file',
  api_key: '891882163577121'
}
Error: Cannot set headers after they are sent to the client
  at ServerResponse.setHeader (node:_http_outgoing:703:11)
  at ServerResponse.header (c:\Users\Dev\Desktop\Pizzaria\backend\node_modules\express\lib\response.js:794:10)
  at ServerResponse.send (c:\Users\Dev\Desktop\Pizzaria\backend\node_modules\express\lib\response.js:174:12)
  at ServerResponse.json (c:\Users\Dev\Desktop\Pizzaria\backend\node_modules\express\lib\response.js:278:15)
  at c:\Users\Dev\Desktop\Pizzaria\backend\src\server.ts:31:21
  at newFn (c:\Users\Dev\Desktop\Pizzaria\backend\node_modules\express-async-errors\index.js:16:20)
  at Layer.handle_error (c:\Users\Dev\Desktop\Pizzaria\backend\node_modules\express\lib\router\layer.js:71:5)
  at trim_prefix (c:\Users\Dev\Desktop\Pizzaria\backend\node_modules\express\lib\router\index.js:326:13)
  at c:\Users\Dev\Desktop\Pizzaria\backend\node_modules\express\lib\router\index.js:286:9
  at Function.process_params (c:\Users\Dev\Desktop\Pizzaria\backend\node_modules\express\lib\router\index.js:346:12)
```



console.cloudinary.com/console/c-b217595c7f0a2d2f171a95fcfa7f99/media\_library/search?q=&view\_mode=list

dgshomtoi

Welcome!

Media Library

Studio

App Marketplace

Structured Metadata

Dashboard

Preferences

# Media Library

Home Assets Folders Collections Moderation

Search Media Library

Upload

Advanced Search

Type to filter

Display name Folders Creation date Tags Formats Asset types Orientations + Add more

Refresh View Settings Preview X

Assets (57)

Display name	Containing folder	Asset type	Format	Size
cld-sample-5	Home	Image	JPG	370
cld-sample-4	Home	Image	JPG	795
cld-sample-3	Home	Image	JPG	884
cld-sample-2	Home	Image	JPG	578
cld-sample	Home	Image	JPG	465
Icon	Home	Image	PNG	10.6

Select an item in the grid to preview it

1

console.cloudinary.com/console/c-b217595c7f0a2d2f171a95fcfa7f99/media\_library/search?q=&view\_mode=list

dgshomtoi

## Media Library

Assets

Home Folders Collections Moderation

Search Media Library

Upload

Welcome!

Media Library

Studio

App Marketplace

Structured Metadata

Dashboard

Preferences

Advanced Search

'qztomdfncfhllumzplia' is selected.

Refresh View Settings

Assets (58)

Display name	Containing folder	Asset type	Format	Size
qztomdfncfhllumzplia	Home	Image	JPG	11.42 KB
cld-sample-5	Home	Image	JPG	370
cld-sample-4	Home	Image	JPG	799
cld-sample-3	Home	Image	JPG	884
cld-sample-2	Home	Image	JPG	578
cld-sample	Home	Image	JPG	465



Location Home

Format JPG

File size 11.42 KB

Dimensions 275 x 183

Last replaced via API

1

The screenshot shows the Media Library interface with a context menu open over an asset named "qztomdfncfhllumzplia". The menu items include Open, Edit (Early Access), Open Containing Folder, Download, Share, Copy URL, Rename, Replace, Add to Collection, Relate to..., Access Control, and Delete. The "Edit" item is highlighted with a blue background and white text. The asset details panel on the right shows the following information for the selected asset:

Asset type	Format	Size
Image	JPG	370
Location  Home		
Format  JPG		
File size 11.42 KB		
Dimensions 275 x 183		
Last replaced  via API		

Below the asset details, there is a circular notification badge with the number "1".



**View Actual Size** 

 Download

Summary

Metadata

Analysis

3

1

## Download Options

Download for Social



## Templates

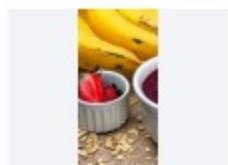
Sort by: Template name ↑

Create New Template



## Original

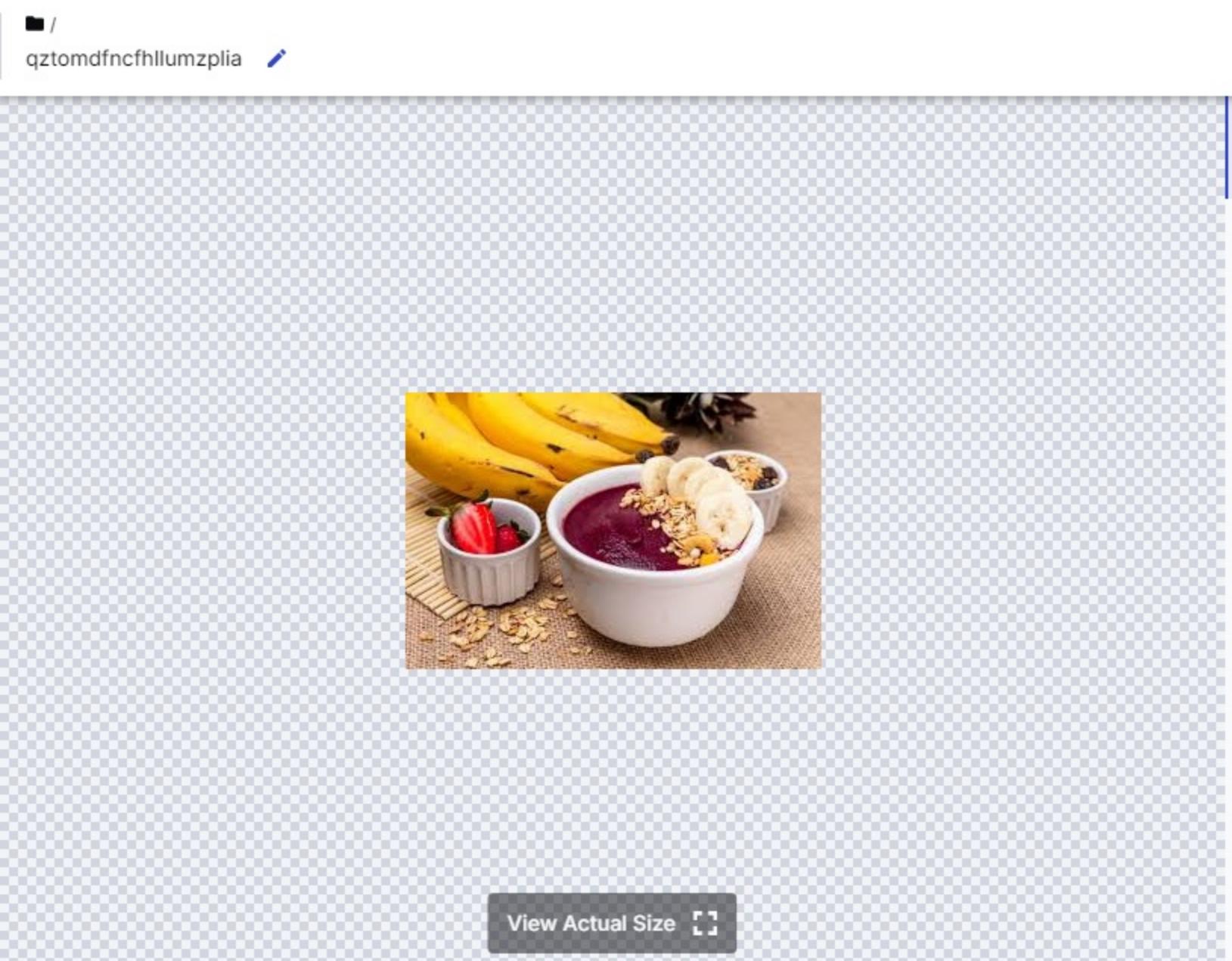
275 x 183 11.42 KB JPG



### Banner (tall)



### Banner (wide)



 Download

i  
Summary

Metadata

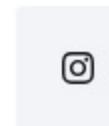
Analysis

Related Asse

Comments

## Download Options

Download for Social



## Templates

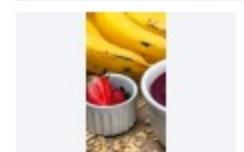
Sort by: Template name ↑

Create New Template



Original 

275 × 183 11.42 KB JPG



### Banner (tall)



## Banner (wide)

console.cloudinary.com/console/c-b217595c7f0a2d2f171a95fcfa7f99/media\_library/search?q=&view\_mode=list

dgshomtoi

## Media Library

Assets Folders Collections Moderation

Search Media Library

Upload

Welcome!

Media Library

Studio

App Marketplace

Structured Metadata

Dashboard

Preferences

Advanced Search

'qztomdfncfhllumzplia' is selected.

Refresh View Settings

Assets (58)

Display name	Containing folder	Asset type	Format	Size
<input checked="" type="checkbox"/>  qztomdfncfhllumzplia	Home	Image	JPG	11.4
<input type="checkbox"/>  cld-sample-5	Home	Image	JPG	370
<input type="checkbox"/>  cld-sample-4	Home	Image	JPG	799
<input type="checkbox"/>  cld-sample-3	Home	Image	JPG	884
<input type="checkbox"/>  cld-sample-2	Home	Image	JPG	578
<input type="checkbox"/>  cld-sample	Home	Image	JPG	465



i

Location Home

Format JPG

File size 11.42 KB

Dimensions 275 x 183

Last via API

1

# **UC: Projetos**

## **Assunto:**

Projeto Pizzaria – Integração  
Preparando para Deploy  
66 – Ajustando Envio de Imagens



EXPLORER

... TS routes.ts

BACKEND

- > node\_modules
- > prisma
- src
  - > @types
  - > config
  - controllers
    - > category
    - > order
  - product
    - TS CreateProductController.ts
    - TS ListByCategoryController.ts
- > user
- > middlewares
- > prisma
- > services
- TS routes.ts
- TS server.ts
- > tmp
- > tmp
- .env
- .gitignore
- { package-lock.json
- { package.json
- tsconfig.json
- yarn-error.log
- yarn.lock

> OUTLINE

> TIMELINE

✖ 0 ▲ 0 ⌂ 0

Ln 38, Col 13 (17 selected) Spaces: 4 UTF-8 CRLF { TypeScript ✓ Prettier

TS CreateProductController.ts X

src > controllers > product > TS CreateProductController.ts > CreateProductController > handle

```
12
13 class CreateProductController{
14     async handle(req: Request, res: Response){
15         const { name, price, description, category_id } = req.body;
16
17         const create ProductService = new Create ProductService();
18
19         if(!req.files || Object.keys(req.files).length === 0){
20             throw new Error("error upload file image")
21         } else {
22
23             const file: UploadedFile = req.files['file']
24
25             const resultFile = await new Promise((resolve, reject) => {
26                 cloudinary.uploader.upload_stream({}, function (error, result) {
27                     if(error){
28                         reject(error);
29                     }
30
31                     resolve(result)
32                 }).end(file.data)
33             })
34
35             console.log(resultFile);
36
37             response.json({})
38
39             // const product = await create ProductService.execute({
40             //     name,
41             //     price,
42             //     description,
43             // }
```

The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** backend
- Toolbars:** Standard Windows-style toolbar with icons for close, minimize, maximize, and others.
- Left Sidebar (EXPLORER):**
  - BACKEND folder
    - node\_modules
    - prisma
  - src folder
    - @types
    - config
    - controllers
      - category
      - order
    - product
      - CreateProductController.ts (selected)
      - ListByCategoryController.ts
      - user
    - middlewares
    - prisma
    - services
  - routes.ts
  - server.ts
  - tmp
  - .env
  - .gitignore
  - package-lock.json
  - package.json
  - tsconfig.json
  - yarn-error.log
  - yarn.lock
- Central Area:** The main code editor window displays `CreateProductController.ts`. The code handles file uploads and interacts with a service to create a product.

```
TS routes.ts          TS CreateProductController.ts X
src > controllers > product > TS CreateProductController.ts > CreateProductController > handle
19      if(!req.files || Object.keys(req.files).length === 0) {
20          throw new Error("error upload file image")
21      } else {
22
23          const file:UploadedFile = req.files['file']
24
25          const resultFile = await new Promise((resolve, reject) => {
26              cloudinary.uploader.upload_stream({}, function (error, result) {
27                  if(error){
28                      reject(error);
29                      return;
30                  }
31
32                  resolve(result)
33              }).end(file.data)
34          })
35
36          console.log(resultFile);
37
38          //... const product = await createProductService.execute({
39          //...     // name,
40          //...     // price,
41          //...     // description,
42          //...     // banner,
43          //...     // category_id
44          //... });
45
46          // res.json(product)
47      }
48  }
49}
50
51 export { CreateProductController }
```
- Bottom Status Bar:** Ln 38, Col 1 (241 selected), Spaces: 4, UTF-8, CRLF, {TypeScript}, {Prettier}

A screenshot of a code editor interface, likely Visual Studio Code, displaying a TypeScript file named `CreateProductController.ts`. The code implements a controller for handling product creation requests.

```
TS routes.ts          TS CreateProductController.ts 1 X
src > controllers > product > TS CreateProductController.ts > CreateProductController > handle
19      if(!req.files || Object.keys(req.files).length === 0) {
20          throw new Error("error upload file image")
21      } else {
22
23          const file:UploadedFile = req.files['file']
24
25          const resultFile = await new Promise((resolve, reject) => {
26              cloudinary.uploader.upload_stream({}, function (error, result) {
27                  if(error){
28                      reject(error);
29                      return;
30                  }
31
32                  resolve(result)
33              }).end(file.data)
34          })
35
36          console.log(resultFile);
37
38          const product = await productService.execute({
39              name,
40              price,
41              description,
42              banner,
43              category_id
44          });
45
46          res.json(product)
47      }
48  }
49}
50
51 export { CreateProductController }
```

The code includes logic to handle file uploads, validate the request body, and interact with a `productService` to execute the creation process. A tooltip is visible over the `category_id` field in the object being passed to the `execute` method.

The screenshot shows a dark-themed code editor interface with the following details:

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** backend
- Left Sidebar (EXPLORER):** BACKEND (node\_modules, prisma), src (@types, config, controllers: category, order, product), routes.ts, server.ts, tmp, .env, .gitignore, package-lock.json, package.json, tsconfig.json, yarn-error.log, yarn.lock.
- Current File:** TS CreateProductController.ts (1)
- Code Content:** The file contains TypeScript code for a CreateProductController. It uses promises to handle file uploads via Cloudinary and interacts with a ProductService to create a new product. A yellow squiggly underline is under the word "banner".

```
const resultFile = await new Promise((resolve, reject) => {
    cloudinary.uploader.upload_stream({}, function (error, result) {
        if(error){
            reject(error);
            return;
        }
        resolve(result)
    }).end(file.data)
})

console.log(resultFile);

const product = await createProductService.execute({
    name,
    price,
    description,
    banner,
    category_id
});

res.json(product)
}

export { CreateProductController }
```
- Bottom Status Bar:** Ln 42, Col 17 (7 selected) Spaces: 4 UTF-8 CRLF {TypeScript} Prettier

EXPLORER

... TS routes.ts TS CreateProductController.ts X

BACKEND

- > node\_modules
- > prisma
- src
  - > @types
  - > config
  - controllers
    - > category
    - > order
    - product
      - TS CreateProductController.ts
      - TS ListByCategoryController.ts

> user

> middlewares

> prisma

> services

TS routes.ts

TS server.ts

> tmp

> tmp

.env

.gitignore

{ package-lock.json

{ package.json

tsconfig.json

yarn-error.log

yarn.lock

> OUTLINE

> TIMELINE

```
src > controllers > product > TS CreateProductController.ts > CreateProductController > handle > product > banner
22
23     const file:UploadedFile = req.files['file']
24
25     const resultFile = await new Promise((resolve, reject) => {
26         cloudinary.uploader.upload_stream({}, function (error, result) {
27             if(error){
28                 reject(error);
29             }
30         })
31
32         resolve(result)
33     }).end(file.data)
34 }
35
36     console.log(resultFile);
37
38     const product = await createProductService.execute(
39         name,
40         price,
41         description,
42         banner: '',
43         category_id
44     );
45
46     res.json(product)
47 }
48 }
49 }
50
51 export { CreateProductController }
```

EXPLORER

... TS routes.ts TS CreateProductController.ts X

BACKEND

- > node\_modules
- > prisma
- src
  - > @types
  - > config
  - controllers
    - > category
    - > order
  - product
    - TS CreateProductController.ts
    - TS ListByCategoryController.ts
- > user
- > middlewares
- > prisma
- > services
- TS routes.ts
- TS server.ts
- > tmp
- > tmp
- .env
- .gitignore
- { package-lock.json
- { package.json
- tsconfig.json
- yarn-error.log
- yarn.lock

> OUTLINE

> TIMELINE

```
src > controllers > product > TS CreateProductController.ts > CreateProductController > handle
12
13 class CreateProductController{
14     async handle(req: Request, res: Response){
15         const { name, price, description, category_id } = req.body;
16
17         const createProductService = new CreateProductService();
18
19         if(!req.files || Object.keys(req.files).length === 0){
20             throw new Error("error upload file image")
21         } else {
22
23             const file:UploadedFile = req.files['file']
24
25             const resultFile = await new Promise((resolve, reject) => {
26                 cloudinary.uploader.upload_stream({}, function (error, result) {
27                     if(error){
28                         reject(error);
29                         return;
30                     }
31
32                     resolve(result)
33                 }).end(file.data)
34             })
35
36             console.log(resultFile);
37
38             const product = await createProductService.execute({
39                 name,
40                 price,
41                 description,
42                 banner: '',
43                 category_id
44             })
45
46             res.status(201).json(product)
47         }
48     }
49 }
```

EXPLORER

BACKEND

- > node\_modules
- > prisma
- src
  - > @types
  - > config
  - controllers
    - > category
    - > order
  - product
    - TS CreateProductController.ts
    - TS ListByCategoryController.ts
- > user
- > middlewares
- > prisma
- > services
- TS routes.ts
- TS server.ts
- > tmp
- > tmp
- .env
- .gitignore
- { package-lock.json
- { package.json
- tsconfig.json
- yarn-error.log
- yarn.lock

OUTLINE

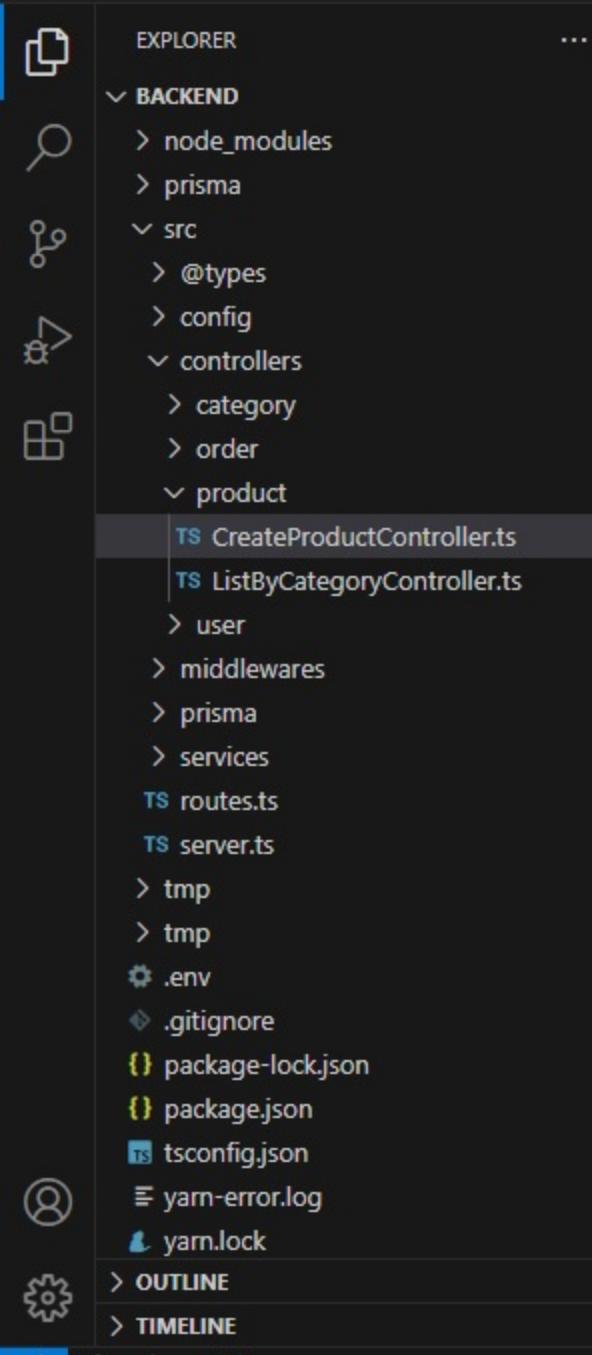
TIMELINE

TS routes.ts TS CreateProductController.ts X

src > controllers > product > TS CreateProductController.ts > TS CreateProductController > handle > resultFile

```
12
13 class CreateProductController{
14     async handle(req: Request, res: Response){
15         const { name, price, description, category_id } = req.body;
16
17         const createProductService = new CreateProductService();
18
19         if(!req.files || Object.keys(req.files).length === 0){
20             throw new Error("error upload file image")
21         } else {
22
23             const file:UploadedFile = req.files['file']
24
25             const resultFile: UploadApiResponse = await new Promise((resolve, reject) => {
26                 cloudinary.uploader.upload_stream({}, function (error, result) {
27                     if(error){
28                         reject(error);
29                         return;
30                     }
31
32                     resolve(result)
33                 }).end(file.data)
34             })
35
36             console.log(resultFile);
37
38             const product = await createProductService.execute({
39                 name,
40                 price,
41                 description,
42                 banner: "",
43                 category_id
44             })
45
46             res.status(201).json(product)
47         }
48     }
49 }
```

Ln 25, Col 19 (29 selected) Spaces: 4 UTF-8 CRLF { TypeScript ✓ Prettier



TS routes.ts    TS CreateProductController.ts X

```
src > controllers > product > TS CreateProductController.ts > ...
1 import { Request, response, Response } from "express";
2 import { CreateProductService } from "../../services/product/CreateProductService";
3 import { UploadedFile } from "express-fileupload";
4 
5 import { v2 as cloudinary, UploadApiResponse } from 'cloudinary'
6 
7 cloudinary.config({
8   cloud_name: process.env.CLOUDINARY_NAME,
9   api_key: process.env.CLOUDINARY_KEY,
10  api_secret: process.env.CLOUDINARY_SECRET,
11 })
12 
13 class CreateProductController{
14   async handle(req: Request, res: Response){
15     const { name, price, description, category_id } = req.body;
16 
17     const createProductService = new CreateProductService();
18 
19     if(!req.files || Object.keys(req.files).length === 0){
20       throw new Error("error upload file image")
21     } else {
22 
23       const file:UploadedFile = req.files['file']
24 
25       const resultFile: UploadApiResponse = await new Promise((resolve, reject) => {
26         cloudinary.uploader.upload_stream({}, function (error, result) {
27           if(error){
28             reject(error);
29             return;
30           }
31 
32           resolve(result)
33         })
34       })
35     }
36   }
37 }
```

A screenshot of the Microsoft Visual Studio Code (VS Code) interface. The top navigation bar includes File, Edit, Selection, View, Go, Run, and other standard options. A search bar at the top right contains the text "backend". The left sidebar features the Explorer, Search, Problems, and Outline/Timeline sections. The main workspace shows two tabs open: "routes.ts" and "CreateProductController.ts". The "CreateProductController.ts" tab is active, displaying the following TypeScript code:

```
src > controllers > product > CreateProductController.ts > CreateProductController > handle
14     async handle(req: Request, res: Response){
15         const { name, price, description, category_id } = req.body;
16
17         const create ProductService = new Create ProductService();
18
19         if(!req.files || Object.keys(req.files).length === 0){
20             throw new Error("error upload file image")
21         } else {
22
23             const file: UploadedFile = req.files['file']
24
25             const resultFile: UploadApiResponse = await new Promise((resolve, reject) => {
26                 cloudinary.uploader.upload_stream({}, function (error, result) {
27                     if(error){
28                         reject(error);
29                         return;
30                     }
31
32                     resolve(result)
33                 }).end(file.data)
34             })
35
36             console.log(resultFile);
37
38             const product = await create ProductService.execute({
39                 name,
40                 price,
41                 description,
42                 banner: '',
43                 category_id
44             });
45
46             res.json(product)
47         }
48     }
49
50     res.json(products)
51 }
```

The code handles file uploads and creates products using a ProductService. It logs the uploaded file details and returns the created product. The status bar at the bottom indicates the current line (Ln 36), column (Col 13), and selected count (24 selected), along with settings for spaces (4), CRLF, and file type (TypeScript). The status bar also shows icons for Prettier and a bell.

The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** backend
- Explorer:** Shows the project structure:
  - BACKEND
    - node\_modules
    - prisma
  - src
    - @types
    - config
    - controllers
      - category
      - order
      - product
        - CreateProductController.ts (selected)
        - ListByCategoryController.ts
  - middlewares
  - prisma
  - services
  - routes.ts
  - server.ts
  - tmp
  - .env
  - .gitignore
  - package-lock.json
  - package.json
  - tsconfig.json
  - yarn-error.log
  - yarn.lock
- Code Editor:** The active file is `CreateProductController.ts`. The code implements a controller for handling product creation. It checks if there is a file upload, handles it using Cloudinary, and then creates a product in the database using a service. Finally, it returns the created product as JSON.

```
src > controllers > product > CreateProductController.ts > CreateProductController > handle > product > banner
19     if(!req.files || Object.keys(req.files).length === 0){
20         throw new Error("error upload file image")
21     } else {
22
23         const file:UploadedFile = req.files['file']
24
25         const resultFile: UploadApiResponse = await new Promise((resolve, reject) => {
26             cloudinary.uploader.upload_stream({}, function (error, result) {
27                 if(error){
28                     reject(error);
29                     return;
30                 }
31
32                     resolve(result)
33                 }).end(file.data)
34             })
35
36             const product = await createProductService.execute({
37                 name,
38                 price,
39                 description,
40                 banner: banner,
41                 category_id
42             });
43
44             res.json(product)
45         }
46     }
47 }
48
49 export { CreateProductController }
```
- Status Bar:** Ln 40, Col 27 (2 selected), Spaces: 4, UTF-8, CRLF, {TypeScript}, Prettier

EXPLORER

... TS routes.ts

BACKEND

- > node\_modules
- > prisma
- src
  - > @types
  - > config
  - controllers
    - > category
    - > order
  - product
    - TS CreateProductController.ts
    - TS ListByCategoryController.ts
- > user
- > middlewares
- > prisma
- > services
- TS routes.ts
- TS server.ts
- > tmp
- > tmp
- .env
- .gitignore
- { package-lock.json
- { package.json
- tsconfig.json
- yarn-error.log
- yarn.lock

> OUTLINE

> TIMELINE

TS CreateProductController.ts X

```
src > controllers > product > TS CreateProductController.ts > TS CreateProductController > handle > TS product > TS banner
19     if(!req.files || Object.keys(req.files).length === 0) {
20         throw new Error("error upload file image")
21     } else {
22
23         const file:UploadedFile = req.files['file']
24
25         const resultFile: UploadApiResponse = await new Promise((resolve, reject) => {
26             cloudinary.uploader.upload_stream({}, function (error, result) {
27                 if(error){
28                     reject(error);
29                     return;
30                 }
31
32                     resolve(result)
33                 }).end(file.data)
34             })
35
36             const product = await createProductService.execute({
37                 name,
38                 price,
39                 description,
40                 banner: resultFile.url,
41                 category_id
42             });
43
44             res.json(product)
45         }
46     }
47 }
48
49 export { CreateProductController }
```

# SujeitoPizza

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

## Últimos pedidos

Mesa 35

Administrator: C:\Windows\system32\cmd.exe

been modified  
Servidor online!!!!  
[INFO] 20:30:06 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\controllers\product\CreateProductController.ts has been modified  
Servidor online!!!!  
[INFO] 20:30:09 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\controllers\product\CreateProductController.ts has been modified  
Servidor online!!!!  
[INFO] 20:30:41 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\controllers\product\CreateProductController.ts has been modified  
Servidor online!!!!  
[INFO] 20:32:44 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\controllers\product\CreateProductController.ts has been modified  
Compilation error in c:\Users\Dev\Desktop\Pizzaria\backend\src\controllers\product\CreateProductController.ts  
[ERROR] 20:32:45 Unable to compile TypeScript:  
src/controllers/product/CreateProductController.ts(25,31): error TS1110: Type expected.  
  
[INFO] 20:32:47 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\controllers\product\CreateProductController.ts has been modified  
Servidor online!!!!  
[INFO] 20:32:49 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\controllers\product\CreateProductController.ts has been modified  
Servidor online!!!!  
[INFO] 20:34:46 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\controllers\product\CreateProductController.ts has been modified  
Servidor online!!!!  
[INFO] 20:34:48 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\controllers\product\CreateProductController.ts has been modified  
Servidor online!!!!  
[INFO] 20:36:03 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\controllers\product\CreateProductController.ts has been modified  
Servidor online!!!!  
[INFO] 20:36:08 Restarting: c:\Users\Dev\Desktop\Pizzaria\backend\src\controllers\product\CreateProductController.ts has been modified  
Servidor online!!!!

```
./src/app/dashboard/components/modal/styles.module.scss  
./src/app/dashboard/components/modal/index.tsx  
./src/app/dashboard/components/orders/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/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 586ms
```

```
true
```

```
GET /dashboard/product 200 in 169ms
```

```
true
```

```
GET /dashboard 200 in 64ms
```

```
true
```

```
GET /dashboard 200 in 187ms
```

```
GET /favicon.ico 200 in 19ms
```

Sujeito Pizza

Categoria Produto [→]

## **Novo produto**



**bebidas** ▾

Google Chrome Abrir

Área de Trabalho > arquivos

Pesquisar em arquivos

Organizar Nova pasta

acai

açai

hamburger 1

hamburger duplo

hamburger triplo

hamburger

pizza frango

sorvete 1

Categoria Produto [→]

Nome: pizza frango Arquivos personalizados

Abrir Cancelar

Digite o nome do produto...

Preço do produto...

Digite a descrição do produto...

Cadastrar produto

Search Star Camera Download Print More

The image shows a dual-screen setup. The left screen displays a file explorer window titled 'Área de Trabalho > arquivos' (Desktop > files) with various food items like açaí, hamburgers, and pizza listed. The right screen shows a product management interface with a dark theme. It has fields for 'Nome' (Name), 'Arquivos personalizados' (Custom files), and buttons for 'Abrir' (Open) and 'Cancelar' (Cancel). Below these are input fields for product name, price, and description, each with a placeholder text. A large green button at the bottom is labeled 'Cadastrar produto' (Register product).

## Novo produto



pizzas

Pizza Frango com Catupiry

60

Pizza Top

**Cadastrar produto**

**Últimos pedidos**

Mesa 3

 Produto Registrado com Sucesso

console.cloudinary.com/console/c-b217595c7f0a2d2f171a95fcfa7f99/media\_library/search?q=&view\_mode=list

dgshomtoi

## Media Library

Assets

Home Folders Collections Moderation

Search Media Library

Upload

Welcome!

Media Library

Studio

App Marketplace

Structured Metadata

Dashboard

Preferences

Advanced Search

'qztomdfncfhllumzplia' is selected.

Refresh View Settings

Assets (58)

Display name	Containing folder	Asset type	Format	Size
qztomdfncfhllumzplia	Home	Image	JPG	11.4
cld-sample-5	Home	Image	JPG	370
cld-sample-4	Home	Image	JPG	799
cld-sample-3	Home	Image	JPG	884
cld-sample-2	Home	Image	JPG	578
cld-sample	Home	Image	JPG	465

Location Home

Format JPG

File size 11.42 KB

Dimensions 275 x 183

Last replaced via API

1

console.cloudinary.com/console/c-b217595c7f0a2d2f171a95fcfa7f99/media\_library/search?q=&view\_mode=list

dgshomtoi

Welcome!

Media Library

Studio

App Marketplace

Structured Metadata

Dashboard

Preferences

## Media Library

Assets Folders Collections Moderation

Search Media Library

Upload

Advanced Search

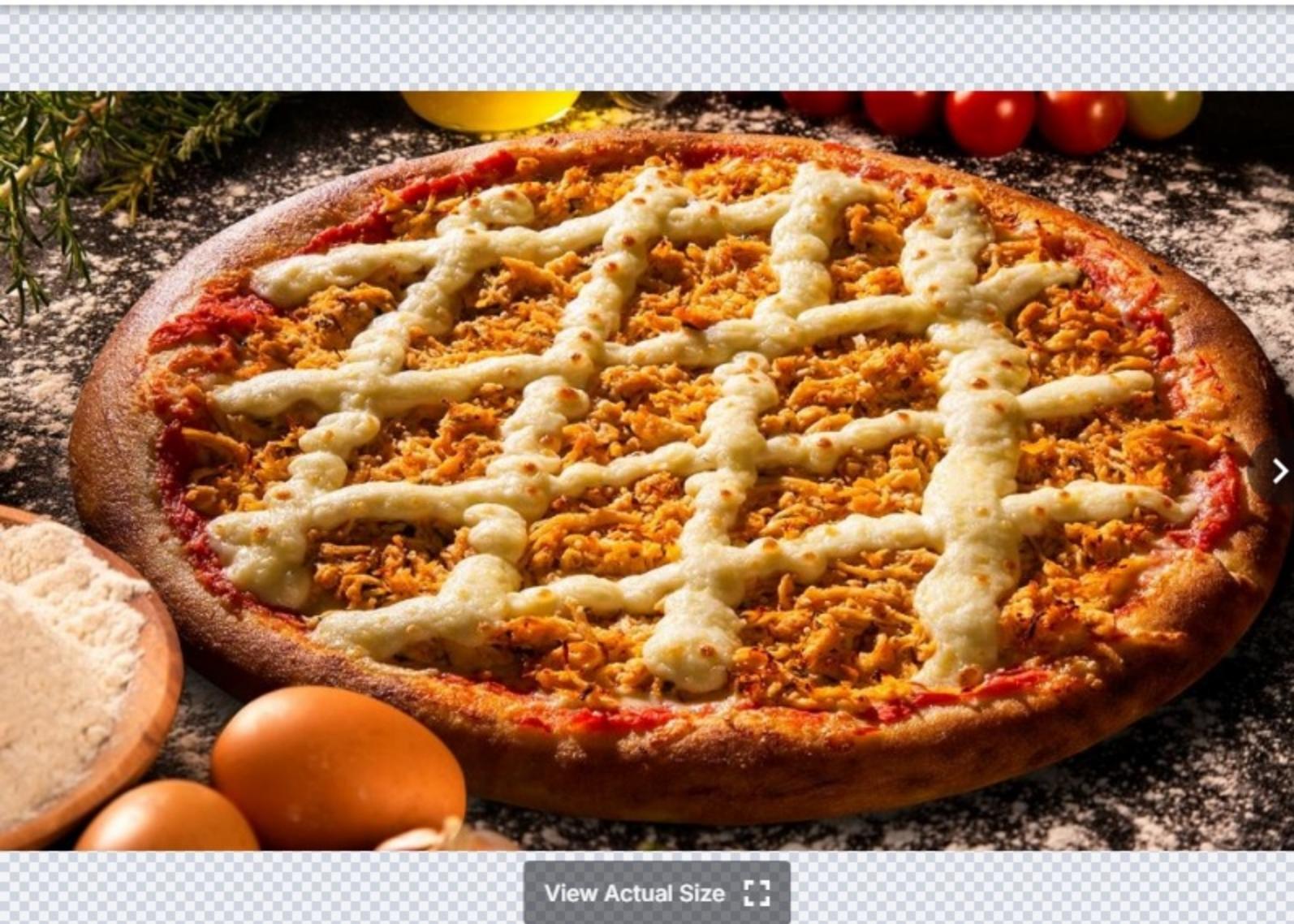
'v4ptcwdg998mrk1kn8qz' is selected.

Refresh

View Settings Preview

Assets (59)

Display name	Containing folder	Asset type	Format	Size	Dimensions	Duration	Delivery ...	Access ...
v4ptcwdg998m...	Home	Image	JPG	144.73 KB	1000 × 577			
qztomdfncfhllumzplia	Home	Image	JPG	11.42 KB	275 × 183		Upload	Public
cld-sample-5	Home	Image	JPG	370.25 KB	1870 × 12...		Upload	Public
cld-sample-4	Home	Image	JPG	799.41 KB	1870 × 12...		Upload	Public
cld-sample-3	Home	Image	JPG	884.6 KB	1870 × 12...		Upload	Public
cld-sample-2	Home	Image	JPG	578.35 KB	1870 × 12...		Upload	Public



**View Actual Size** 

 Download

Summary

Metadata

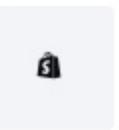
Analysis

Related Assets

 Comments

## Download Options

Download for Social



## Templates

Sort by: Template name

Create New Template



Original

1000 × 577 144.73 KB JPG



### Banner (tall)



Banner (wide)

console.cloudinary.com/console/c-b217595c7f0a2d2f171a95fcfa7f99/media\_library/search?q=&view\_mode=list

dgshomtoi

Welcome!

Media Library

Studio

App Marketplace

Structured Metadata

Dashboard

Preferences

1

## Media Library

Home Assets Folders Collections Moderation

Search Media Library

Upload

Advanced Search

'v4ptcwdg998mrk1kn8qz' is selected.

Refresh View Settings Preview

Assets (59)

Display name	Containing folder	Asset type	Format	Size	Dimensions	Duration	Delivery ...	Access ...
v4ptcwdg998mrk1kn8...	Home	Image	JPG	144.73 KB	1000 × 577		Upload	Public
qztomdfncfhllumzplia	Home	Image	JPG	11.42 KB	275 × 183		Upload	Public
cld-sample-5	Home	Image	JPG	370.25 KB	1870 × 12...		Upload	Public
cld-sample-4	Home	Image	JPG	799.41 KB	1870 × 12...		Upload	Public
cld-sample-3	Home	Image	JPG	884.6 KB	1870 × 12...		Upload	Public
cld-sample-2	Home	Image	JPG	578.35 KB	1870 × 12...		Upload	Public

pizzaria

Query #1 products [all]

Filter

ENTITIES 217

public

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

information\_schema

pg\_catalog

	id	name	price	description	banner	created_at	update_at	cat
1	0cd97aa3-1583-458e-b0ef-549fd1f40e37	Burger Completo Duplo	45	Hambuger Dupl...	040cfab507...	2024-11-29 ...	2024-11-29 01:57:17.362	1e4ffd
2	1cd5c28d-6392-4d30-9b2d-91278fce7f0	Coca-Cola 2L	10	Coca-Cola gelad...	b98497f70a...	2024-11-25 ...	2024-11-25 23:47:24.095	00d8f
3	37e1bb99-2313-4a82-98c3-769d93409a...	Açaí 500ml	30	pote de açaí de ...	4f039587d3...	2024-12-05 ...	2024-12-05 00:34:31.367	80de4
4	6ad3ded6-115b-468c-89eb-25de1a9459...	Burger Triplo Dev	50	Hamburger Tripl...	9b5565412c...	2024-11-29 ...	2024-11-29 02:14:03.556	1e4ffd
5	7f66d79a-ceb2-4175-b4a6-dc5208a9f50e	Pizza de calabresa	46	Pizza de calabre...	8a652b025a...	2024-11-25 ...	2024-11-25 22:06:31.016	e2142
6	96eb4477-824b-4137-8251-dbce1d72f7...	Dev Burger Salada	40	Burgão Dev de S...	c83f6cc7a8b...	2024-11-29 ...	2024-11-29 01:06:24.05	1e4ffd
7	ca5c64fa-e4dd-4b67-9c33-2c4961d4b0...	Coca lata	5	Coca-Cola lata 3...	765f54ffa7f0...	2024-11-25 ...	2024-11-25 22:17:10.368	00d8f
8	dc850cd8-a417-4719-9494-ddff3f53b8ed	Açaí 300ml	12	Pote de Açaí 30...	d3c76275c1...	2024-11-29 ...	2024-11-29 01:29:50.485	f44fb5



## pizzaria



<> Query #1



	<input type="text"/> id	<input type="text"/> equals	<input type="text"/> Enter Value					<input type="button"/>	<input type="button"/>						
	<input checked="" type="checkbox"/> id	<input type="checkbox"/> text	<input type="checkbox"/> name	<input type="checkbox"/> text	<input type="checkbox"/> price	<input type="checkbox"/> text	<input type="checkbox"/> description	<input type="checkbox"/> text	<input type="checkbox"/> banner	<input type="checkbox"/> text	<input type="checkbox"/> created_at	<input type="checkbox"/> timestamp(3)	<input type="checkbox"/> update_at	<input type="checkbox"/> timestamp(3)	<input checked="" type="checkbox"/> key
1	0cd97aa3-1583-458e-b0ef-549fd1f40e37	Burger Completo Duplo	45	Hambuger Dupl...	040cfab507...	2024-11-29 ...	2024-11-29 01:57:17.362	1e...							
2	1cd5c28d-6392-4d30-9b2d-91278fce7f0	Coca-Cola 2L	10	Coca-Cola gelad...	b98497f70a...	2024-11-25 ...	2024-11-25 23:47:24.095	00...							
3	37e1bb99-2313-4a82-98c3-769d93409a...	Açaí 500ml	30	pote de açaí de ...	4f039587d3...	2024-12-05 ...	2024-12-05 00:34:31.367	80...							
4	6ad3ded6-115b-468c-89eb-25de1a9459...	Burger Triplo Dev	50	Hamburger Tripl...	9b5565412c...	2024-11-29 ...	2024-11-29 02:14:03.556	1e...							
5	7f66d79a-ceb2-4175-b4a6-dc5208a9f50e	Pizza de calabresa	46	Pizza de calabre...	8a652b025a...	2024-11-25 ...	2024-11-25 22:06:31.016	e2...							
6	96eb4477-824b-4137-8251-dbce1d72f7...	Dev Burger Salada	40	Burgão Dev de S...	c83f6cc7a8b...	2024-11-29 ...	2024-11-29 01:06:24.05	1e...							
7	ca5c64fa-e4dd-4b67-9c33-2c4961d4b0...	Coca lata	5	Coca-Cola lata 3...	765f54ffa7f0...	2024-11-25 ...	2024-11-25 22:17:10.368	00...							
8	dc850cd8-a417-4719-9494-ddff3f53b8ed	Açaí 300ml	12	Pote de Açaí 30...	d3c76275c1...	2024-11-29 ...	2024-11-29 01:29:50.485	f44...							
9	fd6c8de7-da0e-4373-9a75-10528f325179	Pizza Frango com Catupiry	60	Pizza Top	http://res.clo...	2024-12-05 ...	2024-12-05 23:42:01.785	e2...							

pizzaria

Query #1 products [all]

id equals Enter Value

ENTITIES 217

public

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

information\_schema

pg\_catalog

	name	price	description	banner
1	0e37 Burger Completo Duplo	45	Hambuger Dupl...	040cfab507866ce68a0ce7fdc60e8cc8-hamburger duplo.jpg
2	ef7f0 Coca-Cola 2L	10	Coca-Cola gelad...	b98497f70a15c4d47f7f2209a61bffc1-coca-cola 2l.jpg
3	09a... Açaí 500ml	30	pote de açaí de ...	4f039587d393ea86da3b98f3baa23b08-acai.jpg
4	459... Burger Triplo Dev	50	Hamburger Tripl...	9b5565412cabf8912fd21c9c26cfaca6-hamburger triplo.jpg
5	9f50e Pizza de calabresa	46	Pizza de calabre...	8a652b025acaa467024b0a74085c90f9-pizza de calabresa.jpg
6	2f7... Dev Burger Salada	40	Burgão Dev de S...	c83f6cc7a8b3dea100c14366e0808e99-hamburger.jpg
7	4b0... Coca lata	5	Coca-Cola lata 3...	765f54ffa7f05a9ff902a1a4b85f0e3b-coca-cola lata 350ml.jpg
8	b8ed Açaí 300ml	12	Pote de Açaí 30...	d3c76275c1e9e682c497a354bd32855a-aÃ§ai.jpg
9	5179 Pizza Frango com Catupiry	60	Pizza Top	http://res.cloudinary.com/dgshc...121/v4ptcwg998mrk1kn8qz.jpg

Edit in modal Shift+Enter

Set as NULL

Copy Ctrl+C

Copy as TSV for Excel

Copy as JSON

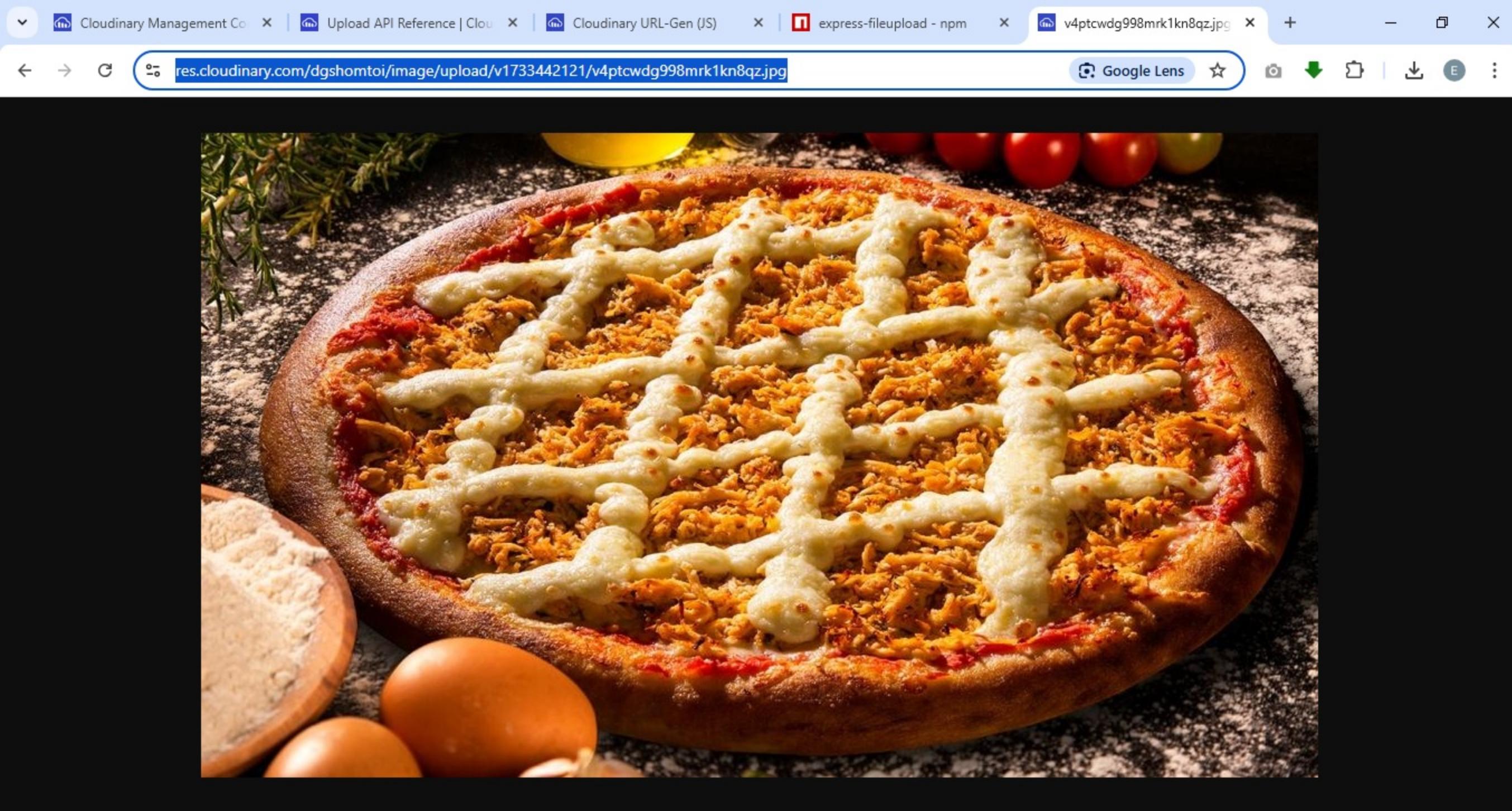
Copy as Markdown

Copy as SQL

Paste Ctrl+V

Clone row Ctrl+D

Delete row Delete





Pedidos feitos

Últimos pedidos

Novo



### Detalhes do pedido

**Mesa 35****Qtd: 2 - Burger Triplo Dev - R\$ 100**

Hamburger Triplo com Cheddar

**Qtd: 1 - Dev Burger Salada - R\$ 40**

Burgão Dev de Salada com 1 hambúrguer de 160g

**Valor Total: R\$ 140**[Concluir pedido](#)

The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** frontend
- Toolbar:** Includes icons for file operations like Open, Save, Find, and Refresh.
- Explorer:** Shows the project structure under the **FRONTEND** folder:
  - .next
  - cache
  - server
  - static
  - types
  - app-build-manifest.json
  - build-manifest.json
  - package.json
  - react-loadable-manifest.json
  - trace
  - node\_modules
  - public
  - src
    - app
      - dashboard
        - category
      - components
        - button
        - header
      - modal
        - index.tsx (highlighted)
        - styles.module.scss
      - orders
      - product
      - layout.tsx
      - page.tsx
- OUTLINE
- TIMELINE

- Code Editor:** Displays the content of the **index.tsx** file.

```
src > app > dashboard > components > modal > index.tsx > Modalorder
21     <X size={40} color="#FF3f4b" />
22
23
24     <article className={styles.container}>
25         <h2>Detalhes do pedido</h2>
26
27         <span className={styles.table}>
28             Mesa <b>{order[0].order.table}</b>
29         </span>
30
31         {order[0].order?.name && (
32             <span className={styles.name}>
33                 <b>{order[0].order.name}</b>
34             </span>
35         )}
36
37         {order.map( item => (
38             <section className={styles.item} key={item.id}>
39                 <span>
40                     Qtd: {item.amount} - <b>{item.product.name}</b> - R$ {parseFloat(item.product.price) * item.amount}
41                 </span>
42                 <span className={styles.description}>
43                     {item.product.description}</span>
44             </section>
45         ))
46
47         <h3 className={styles.total}>Valor Total: R$ {calculateTotalOrder(order)} </h3>
48
49         <button className={styles.buttonOrder} onClick={handleFinishOrder}>
50             Concluir pedido
51         </button>
52     ...
```
- Status Bar:** Ln 47, Col 84 (28 selected) Spaces: 2 UTF-8 CRLF {TypeScript JSX} ✓ Prettier

File Edit Selection View Go Run ... ← → 🔍 frontend

EXPLORER

FRONTEND

- .next
  - cache
  - server
  - static
  - types
- { app-build-manifest.json
- { build-manifest.json
- { package.json
- { react-loadable-manifest.json
- trace
- > node\_modules
- > public
- src
  - app
    - dashboard
      - category
    - components
      - button
      - header
    - modal
      - index.tsx
      - styles.module.scss
    - orders
    - product
    - layout.tsx
    - page.tsx

index.tsx

```
src > app > dashboard > components > modal > index.tsx > Modalorder > order.map() callback
21   <X size={40} color="#FF3f4b" />
22   </button>
23
24   <article className={styles.container}>
25     <h2>Detalhes do pedido</h2>
26
27     <span className={styles.table}>
28       Mesa <b>{order[0].order.table}</b>
29     </span>
30
31     {order[0].order?.name && (
32       <span className={styles.name}>
33         <b>{order[0].order.name}</b>
34       </span>
35     )}
36
37     {order.map( item => (
38       <section className={styles.item} key={item.id}>
39         <img
40           src={item.product.banner}
41           width={120}
42           height={120}
43         />
44         <span>
45           Qtd: {item.amount} - <b>{item.product.name}</b> - R$ {parseFloat(item.product.price) * item.amount}
46         </span>
47         <span className={styles.description}>
48           {item.product.description}</span>
49         </section>
50       ))
51
52       <h3 className={styles.total}>Valor Total: R$ {calculateTotalOrder(order)} </h3>
```

Ln 39, Col 1 (138 selected) Spaces: 2 UTF-8 CRLF ⓘ TypeScript JSX ✅ Prettier



Search.. Ctrl + P

Invite

Pizzaria



Sujeito Programador ▾

Run

POST ▾ http://localhost:3333/session

Send ▾

200 OK

133 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": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJwIjoiQWxtaXigQW50ZSIiImVtYWI
6 >

```

- 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



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

133 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": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJwY1IjoiQWxtaXgQW50ZSIiImVtYWI
6 }>

```

Search with Google

Copy

Select All

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



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

16.4 ms

178 B

7 Days Ago ▾

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



Preview

Headers ▾

Cookies

Tests 0 / 0

→ Mock

Console

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 }
```

\$.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

117 ms

177 B

Just Now ▾

Params

Body

Auth

Headers

Scripts

Docs

Preview

Headers

Cookies

Tests

Mock

Console

JSON ▾

```
1 {  
2   "table": 22,  
3   "name": "John"  
4 }
```

Preview ▾

```
1 {  
2   "id": "d2f50263-8ecd-48ae-89ab-8a7eaed1fb38",  
3   "table": 22,  
4   "status": false,  
5   "draft": true,  
6   "name": "John",  
7   "created_at": "2024-12-05T23:56:06.592Z",  
8   "update_at": "2024-12-05T23:56:06.592Z"  
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 ▾

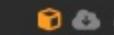
DEL Deletar uma ORDER - Pedido

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

POST ▾ http://localhost:3333/order

Send ▾

200 OK

117 ms

177 B

Just Now ▾

Params

Body

Auth

Headers

Scripts

Docs

Preview

Headers

Cookies

Tests

Mock

Console

JSON ▾

```
1 {  
2   "table": 22,  
3   "name": "John"  
4 }
```

Preview ▾

```
1 {  
2   "id": "d2f50263-8ecd-48ae-89ab-5e7e0a4d42e0",  
3   "table": 22,  
4   "status": false,  
5   "draft": true,  
6   "name": "John",  
7   "created_at": "2024-12-05T23:56:12.000Z",  
8   "update_at": "2024-12-05T23:56:12.000Z"  
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

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

135 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": "d2f50263-8ecd-48ae-89ab-8a7eaed1fb38",
3   "product_id": "96eb4477-824b-4137-8251-dbce1d72f755",
4   "amount": 1
5 }
```

```

1 {
2   "id": "471a49ae-7db2-4dd8-8f6d-476040a6e166",
3   "amount": 1,
4   "created_at": "2024-12-03T23:07:45.492Z",
5   "update_at": "2024-12-03T23:07:45.492Z",
6   "order_id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",
7   "product_id": "96eb4477-824b-4137-8251-dbce1d72f755"
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

**pizzaria** C +

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

	id	equals	Enter Value		
1	0cd97aa3-1583-458e-b0ef-549fd1f40e37	Burger Completo Duplo	45	Hambuger Dupl...	040cfab507866ce68a0ce7fdc60e8cc8-hamburger duplo.jpg
2	1cd5c28d-6392-4d30-9b2d-91278fccef7f0	Coca-Cola 2L	10	Coca-Cola gelad...	b98497f70a15c4d47f7f2209a61bffc1-coca-cola 2l.jpg
3	37e1bb99-2313-4a82-98c3-769d93409a...	Açaí 500ml	30	pote de açaí de ...	4f039587d393ea86da3b98f3baa23b08-acai.jpg
4	6ad3ded6-115b-468c-89eb-25de1a9459...	Burger Triplo Dev	50	Hamburger Tripl...	9b5565412cabf8912fd21c9c26cfaca6-hamburger triplo.jpg
5	7f66d79a-ceb2-4175-b4a6-dc5208a9f50e	Pizza de calabresa	46	Pizza de calabre...	8a652b025acaa467024b0a74085c90f9-pizza de calabresa.jpg
6	96eb4477-824b-4137-8251-dbce1d72f7...	Dev Burger Salada	40	Burgão Dev de S...	c83f6cc7a8b3dea100c14366e0808e99-hamburger.jpg
7	ca5c64fa-e4dd-4b67-9c33-2c4961d4b0...	Coca lata	5	Coca-Cola lata 3...	765f54ffa7f05a9ff902a1a4b85f0e3b-coca-cola lata 350ml.jpg
8	dc850cd8-a417-4719-9494-ddff3f53b8ed	Açaí 300ml	12	Pote de Açaí 30...	d3c76275c1e9e682c497a354bd32855a-aÃ§ai.jpg
9	fd6c8de7-da0e-4373-9a75-10528f325179	Pizza Frango com Catupiry	60	Pizza Top	<a href="http://res.cloudinary.com/dgshomtoi/image/upload/v1733442121/v4p">http://res.cloudinary.com/dgshomtoi/image/upload/v1733442121/v4p</a>

**fd6c8de7-da0e-4373-9a75-10528f325179**

- 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



Search.. Ctrl + P

Invite

Pizzaria ▾



Sujeito Programador ▾

Run

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

Send ▾

200 OK

135 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": "d2f50263-8ecd-48ae-89ab-8a7eaed1fb38",  
3   "product_id": "fd6c8de7-da0e-4373-9a75-10528f325179",  
4   "amount": 1  
5 }
```

Preview ▾

```
1 ▶ {  
2   "id": "471a49ae-7db2-4dd8-8f6d-476040a6e166",  
3   "amount": 1,  
4   "created_at": "2024-12-03T23:07:45.492Z",  
5   "update_at": "2024-12-03T23:07:45.492Z",  
6   "order_id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",  
7   "product_id": "96eb4477-824b-4137-8251-dbce1d72f755"  
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

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

135 ms

237 B

2 Days Ago ▾



Base Environment

Add Cookies

Add Certificates

Filter



Params

Body

Auth

Headers

Scripts

Docs

Preview

Headers

Cookies

Tests 0 / 0

→ Mock

Console

JSON ▾

```
1 {  
2   "order_id": "d2f50263-8ecd-48ae-89ab-8a7eaed1fb38",  
3   "product_id": "fd6c8de7-da0e-4373-9a75-10528f325179",  
4   "amount": 2  
5 }
```

```
1 {  
2   "id": "471a49ae-7db2-4dd8-8f6d-476040a6e166",  
3   "amount": 1,  
4   "created_at": "2024-12-03T23:07:45.492Z",  
5   "update_at": "2024-12-03T23:07:45.492Z",  
6   "order_id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",  
7   "product_id": "96eb4477-824b-4137-8251-dbce1d72f755"  
8 }
```

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

38 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

Preview

Headers ▾ 8

Cookies

Tests 0 / 0

→ Mock

Console

Preview ▾

```
1 {  
2   "order_id": "d2f50263-8ecd-48ae-89ab-8a7eaed1fb38",  
3   "product_id": "fd6c8de7-da0e-4373-9a75-10528f325179",  
4   "amount": 2  
5 }
```

```
1 {  
2   "id": "ff19d9f5-a3f1-4018-84ab-b5113b034713",  
3   "amount": 2,  
4   "created_at": "2024-12-06T00:00:25.237Z",  
5   "update_at": "2024-12-06T00:00:25.237Z",  
6   "order_id": "d2f50263-8ecd-48ae-89ab-8a7eaed1fb38",  
7   "product_id": "fd6c8de7-da0e-4373-9a75-10528f325179"  
8 }
```

\$.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

38 ms

237 B

1 Minute Ago ▾

Params

Body

Auth

Headers

Scripts

Docs

Preview

Headers

Cookies

Tests 0 / 0

→ Mock

Console

JSON ▾

```

1 {
2   "order_id": "d2f50263-8ecd-48ae-89ab-8a7eaed1fb38"
3   "product_id": "fd6c8de7-da0e-4373-9a75-1"
4   "amount": 2
5 }
```

Cut

Copy

Paste

Base64 &gt;

Cookie

Faker &gt;

File

Hash &gt;

JSONPath

OS &gt;

Prompt

Request &gt;

Response &gt;

Timestamp &gt;

UUID &gt;

```

1 {
2   "id": "ff19d9f5-a3f1-4018-84ab-b5113b034713",
3   "amount": 2,
4   "created_at": "2024-12-06T00:00:25.237Z",
5   "update_at": "2024-12-06T00:00:25.237Z",
6   "order_id": "d2f50263-8ecd-48ae-89ab-8a7eaed1fb38",
7   "product_id": "fd6c8de7-da0e-4373-9a75-10528f325179"
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

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

99.8 ms

179 B

7 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": "d2f50263-8ecd-48ae-89ab-8a7eaed1fb38"  
3 }
```

Preview ▾

```
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 }
```

- Base Environment
- Add Cookies
- Add Certificates

 ▼ ▲ ↶ ↷ +

- 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

cloud master 📦 cloud cloud

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

133 ms

351 B

10 Minutes Ago ▾

Base Environment

Params

Body



Auth

Headers

4

Scripts

Docs

Preview

Headers

8

Cookies

Tests 0 / 0

→ Mock

Console

Inherit from parent ▾

Preview ▾

```

1  {
2    "id": "6467d8c9-295d-48ab-8ec0-e036875247d9",
3    "name": "Almir Ante",
4    "email": "almir@teste.com",
5    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyYWI1IjoiQWxtaXigQW50ZSIiImVtYwI
6  }

```

Search with Google

Copy

Select All



Select an auth type from above

\$store.books[\*].author

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

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

Send ▾

200 OK

99.8 ms

179 B

7 Days Ago ▾

Base Environment

Add Cookies

Add Certificates

Filter

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

category

users

master



Preferences

Params 1

Body 1

Auth 1

Headers 4

Scripts

Docs

Preview

Headers 8

Cookies

Tests 0 / 0

→ Mock

Console

Bearer Token ▾

ENABLED



TOKEN

.....



PREFIX

Bearer

Preview ▾

```
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

Online

Made with ❤ by Kong



Search.. Ctrl + P

Invite

Pizzaria



Sujeito Programador ▾

Run

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

Send ▾

200 OK

156 ms

178 B

Just Now ▾

Base Environment

Add Cookies

Add Certificates

Filter

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

category

users

master



Beautify JSON

Preview

Headers ▾

Cookies

Tests 0 / 0

→ Mock

Console

JSON ▾

```
1 {  
2   "order_id": "d2f50263-8ecd-48ae-89ab-8a7eaed1fb38"  
3 }
```

Preview ▾

```
1 {  
2   "id": "d2f50263-8ecd-48ae-89ab-8a7eaed1fb38",  
3   "table": 22,  
4   "status": false,  
5   "draft": false,  
6   "name": "John",  
7   "created_at": "2024-12-05T23:56:06.592Z",  
8   "update_at": "2024-12-05T23:56:06.592Z"  
9 }
```

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



Preferences

Online

Made with ❤ by Kong

# SujeitoPizza

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

## Últimos pedidos

Mesa 35

Sujeito Pizza

Categoria Produto →

**Últimos pedidos** 

Mesa 2

Mesa 3



Pedidos Pendentes



### Detalhes do pedido

Mesa 22 John



Qtd: 2 - Pizza Frango com Catupiry - R\$ 120

Pizza Top

**Valor Total: R\$ 120****Concluir pedido**



Pedidos | Itens

Últimos pedidos

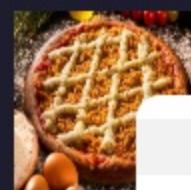
Mesa 17

-



## Detalhes do pedido

Mesa 22 John



Qtd: 2 - Piz

Pizza Top

Valor Tot

Concluir pe

Abrir imagem em uma nova guia

Salvar imagem como...

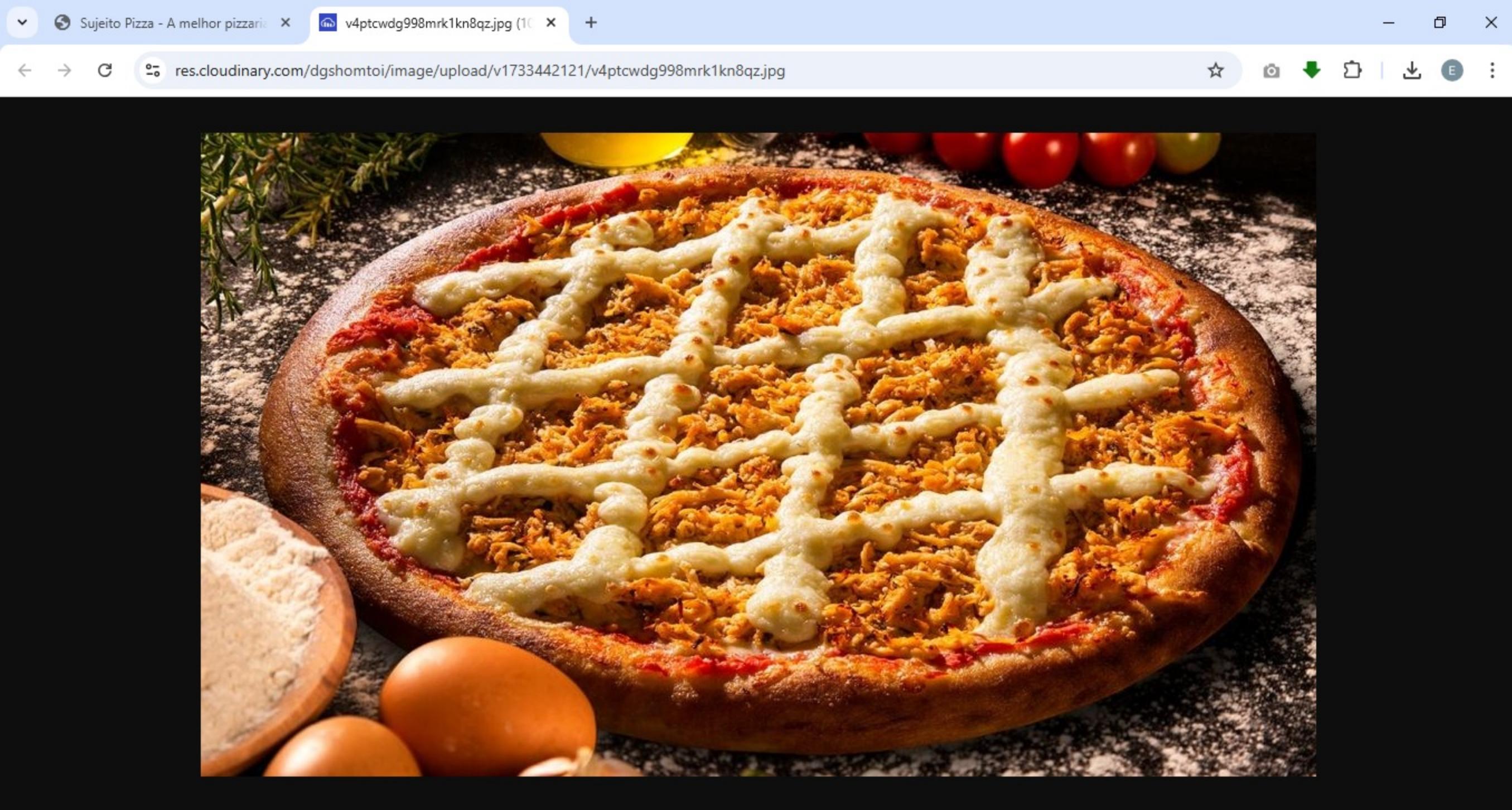
Copiar imagem

Copiar endereço da imagem

Criar QR code para esta imagem

Pesquisar com o Lens

Inspecionar



EXPLORER

... 

BACKEND    

> node\_modules

> prisma

src    

> @types

> config

controllers    

> category

> order

> product

> user

> middlewares

> prisma

> services

routes.ts

server.ts

tmp    

- 4f039587d393ea86da3b98f3baa23b...
- 8a652b025acaa467024b0a74085c90...
- 9b5565412cabf8912fd21c9c26cfaca...
- 040cfab507866ce68a0ce7fdc60e8cc...
- 765f54ffa7f05a9ff902a1a4b85f0e3b...
- b98497f70a15c4d47f7f2209a61bffc1...
- c83f6cc7a8b3dea100c14366e0808e9...
- d3c76275c1e9e682c497a354bd3285...

> tmp

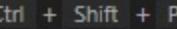
.env

OUTLINE

TIMELINE

✖ 0 △ 0 ⌂ 0



Show All Commands 

Go to File 

Find in Files 

Toggle Full Screen 

Show Settings 

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

pizzaria C +

Query #1 products [all] items [all] x +

Filter Y

ENTITIES 217 +

public

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

information\_schema

pg\_catalog

	id	amount	created_at...	update_at	order_id	product_id
1	0d94f740-43ba-4b43-9663-0014655235...	3	2024-11-26 ...	2024-11-26 17:09:39.369	0d0fef8f-9947-4e0a-9...	7f66d79a-ceb2-4175-b...
2	1900a07a-65b4-409d-812b-b8aa9a5c7a...	2	2024-11-29 ...	2024-11-29 02:41:38.002	ee42b24f-cdc5-48fd-...	6ad3ded6-115b-468c-8...
3	1fc66991-1e7e-42a7-a75e-217ccb149c...	2	2024-11-26 ...	2024-11-26 14:14:15.76	0d0fef8f-9947-4e0a-9...	ca5c64fa-e4dd-4b67-9...
4	29e7be3a-8cc6-4914-9835-1e10a3f092...	1	2024-11-26 ...	2024-11-26 15:42:40.002	c9019c83-9a02-4152...	1cd5c28d-6392-4d30-9...
5	471a49ae-7db2-4dd8-8f6d-476040a6e1...	1	2024-12-03 ...	2024-12-03 23:07:45.492	ee42b24f-cdc5-48fd-...	96eb4477-824b-4137-8...
6	ff19d9f5-a3f1-4018-84ab-b5113b034713	2	2024-12-06 ...	2024-12-06 00:00:25.237	d2f50263-8ecd-48ae-...	fd6c8de7-da0e-4373-9...

Set as NULL

Copy Ctrl+C

Copy as TSV for Excel

Copy as JSON

Copy as Markdown

Copy as SQL

Clone row Ctrl+D

Delete row Delete

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

pizzaria C +

Query #1 products [all] items [all] x +

Filter Y

ENTITIES 217 +

public

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

information\_schema

pg\_catalog

	id	amount	created_at...	update_at	order_id	product_id
1	0d94f740-43ba-4b43-9663-0014655235...	3	2024-11-26 ...	2024-11-26 17:09:39.369	0d0fef8f-9947-4e0a-9...	7f66d79a-ceb2-4175-b...
2	1900a07a-65b4-409d-812b-b8aa9a5c7a...	2	2024-11-29 ...	2024-11-29 02:41:38.002	ee42b24f-cdc5-48fd-...	6ad3ded6-115b-468c-8...
3	1fc66991-1e7e-42a7-a75e-217ccb149c...	2	2024-11-26 ...	2024-11-26 14:14:15.76	0d0fef8f-9947-4e0a-9...	ca5c64fa-e4dd-4b67-9...
4	29e7be3a-8cc6-4914-9835-1e10a3f092...	1	2024-11-26 ...	2024-11-26 15:42:40.002	c9019c83-9a02-4152	1cd5c28d-6392-4d30-9...
5	471a49ae-7db2-4dd8-8f6d-476040a6e1...	1	2024-12-03 ...	2024-12-03 23:07:45.492	ee42b24f-cdc5-48fd-...	96eb4477-824b-4137-8...
6	ff19d9f5-a3f1-4018-84ab-b5113b034713	2	2024-12-06 ...	2024-12-06 00:00:25.237	d2f50263-8ecd-48ae-	fd6c8de7-da0e-4373-9...

Set as NULL

Copy Ctrl+C

Copy as TSV for Excel

Copy as JSON

Copy as Markdown

Copy as SQL

Clone rows 1 - 6 Ctrl+D

Delete rows 1 - 6 Delete

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

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

Filter +

ENTITIES 217 +

public

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

information\_schema

pg\_catalog

id text amount int4(32,0) created\_at... update\_at timestamp(3) order\_id text product\_id text

	id	amount	created_at...	update_at	order_id	product_id
1	0d94f740-43ba-4b43-9663-0014655235...	3	2024-11-26 ...	2024-11-26 17:09:39.369	0d0fef8f-9947-4e0a-1...	7f66d79a-ceb2-4175-b...
2	1900a07a-65b4-409d-812b-b8aa9a5c7a...	2	2024-11-29 ...	2024-11-29 02:41:38.002	ee42b24f-cdc5-48fd-...	6ad3ded6-115b-468c-8...
3	1fc66991-1e7e-42a7-a75e-217ccb149c...	2	2024-11-26 ...	2024-11-26 14:14:15.76	0d0fef8f-9947-4e0a-1...	ca5c64fa-e4dd-4b67-9...
4	29e7be3a-8cc6-4914-9835-1e10a3f092...	1	2024-11-26 ...	2024-11-26 15:42:40.002	c9019c83-9a02-4152...	1cd5c28d-6392-4d30-9...
5	471a49ae-7db2-4dd8-8f6d-476040a6e1...	1	2024-12-03 ...	2024-12-03 23:07:45.492	ee42b24f-cdc5-48fd-...	96eb4477-824b-4137-8...
6	ff19d9f5-a3f1-4018-84ab-b5113b034713	2	2024-12-06 ...	2024-12-06 00:00:25.237	d2f50263-8ecd-48ae...	fd6c8de7-da0e-4373-9...

The screenshot shows the Prisma Studio interface for a PostgreSQL database named 'pizzaria'. The top navigation bar includes tabs for 'products [all]' and 'items [all]'. A search bar at the top right contains the text 'Enter Value'. On the left, a sidebar lists database entities: '\_prisma\_migrations', 'categories', 'items', 'orders', 'products', 'users', 'information\_schema', and 'pg\_catalog'. The 'public' schema is expanded. The main area displays a table for the 'items' entity with columns: id (text), amount (int4(32,0)), created\_at..., update\_at (timestamp(3)), order\_id (text), and product\_id (text). A filter bar at the top allows setting conditions like 'id equals Enter Value'. A message at the bottom right indicates '6 changes successfully applied'.

pizzaria

Filter

ENTITIES 217

public

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

Query #1

products [all]

items [all]

id equals Enter Value

No Data

6 changes successfully applied

localhost:5432/pizzaria

postgresql

Structure

Unknown just now

1 >

C + ⚙

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

pizzaria C +

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

Filter Y

ENTITIES 217 +

public

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

information\_schema

pg\_catalog

id table status draft name created\_at... update\_at

	id	table	status	draft	name	created_at...	update_at
1	0d0fef8f-9947-4e0a-94d2-dc2722b9cee0	55	true	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...
3	d2f50263-8ecd-48ae-89ab-8a7eaed1fb38	22	false	false	John	2024-12-05 ...	2024-12-05 23:56:06.5...
4	ee42b24f-cdc5-48fd-87d3-1758a06afae3	35	false	false	(EMPTY)	2024-11-29 ...	2024-11-29 02:28:31.9...

Set as NULL

- Copy Ctrl+C
- Copy as TSV for Excel
- Copy as JSON
- Copy as Markdown
- Copy as SQL
- Clone rows 1 - 4 Ctrl+D
- Delete rows 1 - 4 Delete

localhost:5432/pizzaria postgresql Structure Unknown just now 1 > C +

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

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

Filter

ENTITIES 217 +

public

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

information\_schema

pg\_catalog

id table status draft name created\_at... update\_at

	id	table	status	draft	name	created_at...	update_at
1	0d0fef8f-9947-4e0a-94d2-dc2722b9cee0	55	true	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...
3	d2f50263-8ecd-48ae-89ab-8a7eaed1fb38	22	false	false	John	2024-12-05 ...	2024-12-05 23:56:06.5...
4	ee42b24f-cdc5-48fd-87d3-1758a06afae3	35	false	false	(EMPTY)	2024-11-29 ...	2024-11-29 02:28:31.9...

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

pizzaria C +

Query #1 products [all] orders [all] x +

Filter Y

ENTITIES 217 +

public

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

information\_schema

pg\_catalog

No Data

4 changes successfully applied

pizzaria

Filter

ENTITIES 217

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

Query #1 products [all]

id equals Enter Value

	id	name	price	description	banner
1	fd6c8de7-da0e-4373-9a75-10528f325179	Pizza Frango com Catupiry	60	Pizza Top	http://res.cloudinary.com/dgshomtoi/image/upload/v1733442121/v4p
2	37e1bb99-2313-4a82-98c3-769d93409a...	Açaí 500ml	30	pote de açaí de ...	4f039587d393ea86da3b98f3baa23b08-acai.jpg
3	6ad3ded6-115b-468c-89eb-25de1a9459...	Burger Triplo Dev	50	Hamburger Tripl...	9b5565412cabf8912fd21c9c26cfaca6-hamburger triplo.jpg
4	0cd97aa3-1583-458e-b0ef-549fd1f40e37	Burger Completo Duplo	45	Hambuger Dupl...	040cfab507866ce68a0ce7fdc60e8cc8-hamburger duplo.jpg
5	dc850cd8-a417-4719-9494-ddff3f53b8ed	Açaí 300ml	12	Pote de Açaí 30...	d3c76275c1e9e682c497a354bd32855a-aÃ§ai.jpg
6	96eb4477-824b-4137-8251-dbce1d72f7...	Dev Burger Salada	40	Burgão Dev de S...	c83f6cc7a8b3dea100c14366e0808e99-hamburger.jpg
7	1cd5c28d-6392-4d30-9b2d-91278fce7f0	Coca-Cola 2L	10	Coca-Cola gelad...	b98497f70a15c4d47f7f2209a61bffc1-coca-cola 2l.jpg
8	ca5c64fa-e4dd-4b67-9c33-2c4961d4b0...	Coca lata	5	Coca-Cola lata 3...	765f54ffa7f05a9ff902a1a4b85f0e3b-coca-cola lata 350ml.jpg
9	7f66d79a-ceb2-4175-b4a6-dc5208a9f50e	Pizza de calabresa	46	Pizza de calabre...	8a652b025acaa467024b0a74085c90f9-pizza de calabresa.jpg

Set as NULL

- Copy Ctrl+C
- Copy as TSV for Excel
- Copy as JSON
- Copy as Markdown
- Copy as SQL
- Clone rows 2 - 9 Ctrl+D
- Delete rows 2 - 9 Delete

pizzaria

Query #1 products [all]

id equals Enter Value

ENTITIES 217

public

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

information\_schema

pg\_catalog

	id	name	price	description	banner
1	fd6c8de7-da0e-4373-9a75-10528f325179	Pizza Frango com Catupiry	60	Pizza Top	http://res.cloudinary.com/dgshomtoi/image/upload/v1733442121/v4p...
2	37e1bb99-2313-4a82-98c3-769d93409a...	Açaí 500ml	30	pote de açaí de ...	4f039587d393ea86da3b98f3baa23b08-acai.jpg
3	6ad3ded6-115b-468c-89eb-25de1a9459...	Burger Triplo Dev	50	Hamburger Tripl...	9b5565412cabf8912fd21c9c26cfaca6-hamburger triplo.jpg
4	0cd97aa3-1583-458e-b0ef-549fd1f40e37	Burger Completo Duplo	45	Hambuger Dupl...	040cfab507866ce68a0ce7fdc60e8cc8-hamburger duplo.jpg
5	dc850cd8-a417-4719-9494-ddff3f53b8ed	Açaí 300ml	12	Pote de Açaí 30...	d3c76275c1e9e682c497a354bd32855a-aÃ§ai.jpg
6	96eb4477-824b-4137-8251-dbce1d72f7...	Dev Burger Salada	40	Burgão Dev de S...	c83f6cc7a8b3dea100c14366e0808e99-hamburger.jpg
7	1cd5c28d-6392-4d30-9b2d-91278fce7f0	Coca-Cola 2L	10	Coca-Cola gelad...	b98497f70a15c4d47f7f2209a61bffc1-coca-cola 2l.jpg
8	ca5c64fa-e4dd-4b67-9c33-2c4961d4b0...	Coca lata	5	Coca-Cola lata 3...	765f54ffa7f05a9ff902a1a4b85f0e3b-coca-cola lata 350ml.jpg
9	7f66d79a-ceb2-4175-b4a6-dc5208a9f50e	Pizza de calabresa	46	Pizza de calabre...	8a652b025acaa467024b0a74085c90f9-pizza de calabresa.jpg

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

pizzaria C +

Query #1 products [all] +

Filter Y

ENTITIES 217 +

public

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

information\_schema

pg\_catalog

id equals Enter Value

	id	name	price	description	banner
1	fd6c8de7-da0e-4373-9a75-10528f325179	Pizza Frango com Catupiry	60	Pizza Top	http://res.cloudinary.com/dgshomtoi/image/upload/v1733442121/v4p

8 changes successfully applied X

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

pizzaria C +

Query #1 products [all] orders [all] x +

Filter Y

ENTITIES 217 +

public

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

information\_schema

pg\_catalog

id equals Enter Value

	table	status	draft	name	created_at...	update_at
1	54387899-64b2-4231-bd3e-da05d9fc88...	22	false	John	2024-12-06 ...	2024-12-06 00:17:09.4...

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

pizzaria C + Query #1 products [all] orders [all] items [all] +

Filter ENTITIES 217 public \_prisma\_migrations categories items orders products users information\_schema pg\_catalog

id equals Enter Value

	id	amount	created_at...	update_at	order_id	product_id
1	098b188b-5183-4f17-a4b4-f250344ffecd	2	2024-12-06 ...	2024-12-06 00:18:03.831	54387899-64b2-4231	fd6c8de7-da0e-4373-9...

+

**Últimos pedidos** 

Mesa 22

Sujeito Pizza

Categoría Producto [→]

## **Novo produto**



**bebidas** ▾

## Cadastrar produto

Abrir

Área de Trabalho > arquivos

Pesquisar em arquivos

Organizar Nova pasta

acai açaí hamburger 1 hamburger duplo hamburger triplo hamburger

pizza frango sorvete 1 pizza calabresa

Categoria Produto [→]

Nome: pizza calabresa Arquivos personalizados

Abrir Cancelar

Digite o nome do produto...

Preço do produto...

Digite a descrição do produto...

Cadastrar produto

Search Star Camera Download Print E

This image shows a screenshot of a Windows file explorer window and a product management application interface. The file explorer window is titled 'Área de Trabalho > arquivos' and contains several food-related images and their names: 'acai', 'açaí', 'hamburger 1', 'hamburger duplo', 'hamburger triplo', 'hamburger', 'pizza frango', 'sorvete 1', and 'pizza calabresa'. The 'pizza calabresa' image is highlighted with a blue selection box. Below the file explorer is a modal dialog box for adding a new product. The dialog has fields for 'Nome' (Name) set to 'pizza calabresa', 'Arquivos personalizados' (Custom files), and two buttons: 'Abrir' (Open) and 'Cancelar' (Cancel). The main application interface has sections for 'Categoria' (Category) and 'Produto' (Product) with a right-pointing arrow. It also features input fields for product name, price, and description, and a large green 'Cadastrar produto' (Add product) button at the bottom.

# SujeitoPizza

Categoria Produto [→]

## Novo produto



pizzas

Pizza Calabresa

65

Pizza Saborosa

Cadastrar produto



# SujeitoPizza

Categoria Produto [→]

## Últimos pedidos

Mesa 22

 Produto Registrado com Sucesso

dgshomtoi

Welcome!

Media Library

Studio

App Marketplace

Structured Metadata

Dashboard

Preferences

# Media Library

Assets

Folders

Collections

Moderation

Search Media Library

Upload

Advanced Search

'v4ptcwdg998mrk1kn8qz' is selected.

Assets (59)

Display name	Containing folder	Asset type	Format	Size	Dimensions	Duration	Delivery...	Access...
v4ptcwdg998mrk1kn8...	Home	Image	JPG	144.73 KB	1000 × 577		Upload	Public
qztomdfncfhllumzplia	Home	Image	JPG	11.42 KB	275 × 183		Upload	Public
cld-sample-5	Home	Image	JPG	370.25 KB	1870 × 12...		Upload	Public
cld-sample-4	Home	Image	JPG	799.41 KB	1870 × 12...		Upload	Public
cld-sample-3	Home	Image	JPG	884.6 KB	1870 × 12...		Upload	Public
cld-sample-2	Home	Image	JPG	578.35 KB	1870 × 12...		Upload	Public

console.cloudinary.com/console/c-b217595c7f0a2d2f171a95fcfa7f99/media\_library/search?q=&view\_mode=list

dgshomtoi

## Media Library

Assets Folders Collections Moderation

Search Media Library

Upload

Welcome! 20

Media Library Studio App Marketplace Structured Metadata Dashboard Preferences

Advanced Search

'dijhowuimzzunakmifkg' is selected.

Refresh View Settings Preview

Assets (60)

Display name	Containing folder	Asset type	Format	Size	Dimensions	Duration	Delivery ...	Access ...
dijhowuimzzunakmifkg	Home	Image	JPG	51.66 KB	600 × 388		<a href="#">Upload</a>	<a href="#">Public</a>
v4ptcwdg998mrk1kn8...	Home	Image	JPG	144.73 KB	1000 × 577		<a href="#">Upload</a>	<a href="#">Public</a>
qztomdfncfhllumzplia	Home	Image	JPG	11.42 KB	275 × 183		<a href="#">Upload</a>	<a href="#">Public</a>
cld-sample-5	Home	Image	JPG	370.25 KB	1870 × 12...		<a href="#">Upload</a>	<a href="#">Public</a>
cld-sample-4	Home	Image	JPG	799.41 KB	1870 × 12...		<a href="#">Upload</a>	<a href="#">Public</a>
cld-sample-3	Home	Image	JPG	884.6 KB	1870 × 12...		<a href="#">Upload</a>	<a href="#">Public</a>

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

pizzaria C + Query #1 products [all] +

Filter Enter Value

ENTITIES 217 +

public

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

information\_schema

pg\_catalog

	id	name	price	description	banner
1	fd6c8de7-da0e-4373-9a75-10528f325179	Pizza Frango com Catupiry	60	Pizza Top	http://res.cloudinary.com/dgshomtoi/image/upload/v1733442121/v4p

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

pizzaria C + Query #1 products [all] +

Filter +

ENTITIES 217 +

public

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

information\_schema

pg\_catalog

id equals Enter Value

	id	name	price	description	banner
1	49b330fb-5693-4520-9774-c984fcdb921a	Pizza Calabresa	65	Pizza Saborosa	http://res.cloudinary.com/dgshomtoi/image/upload/v173344770/dijh
2	fd6c8de7-da0e-4373-9a75-10528f325179	Pizza Frango com Catupiry	60	Pizza Top	http://res.cloudinary.com/dgshomtoi/image/upload/v1733442121/v4p

pizzaria

Query #1 products [all]

id equals Enter Value

ENTITIES 217

public

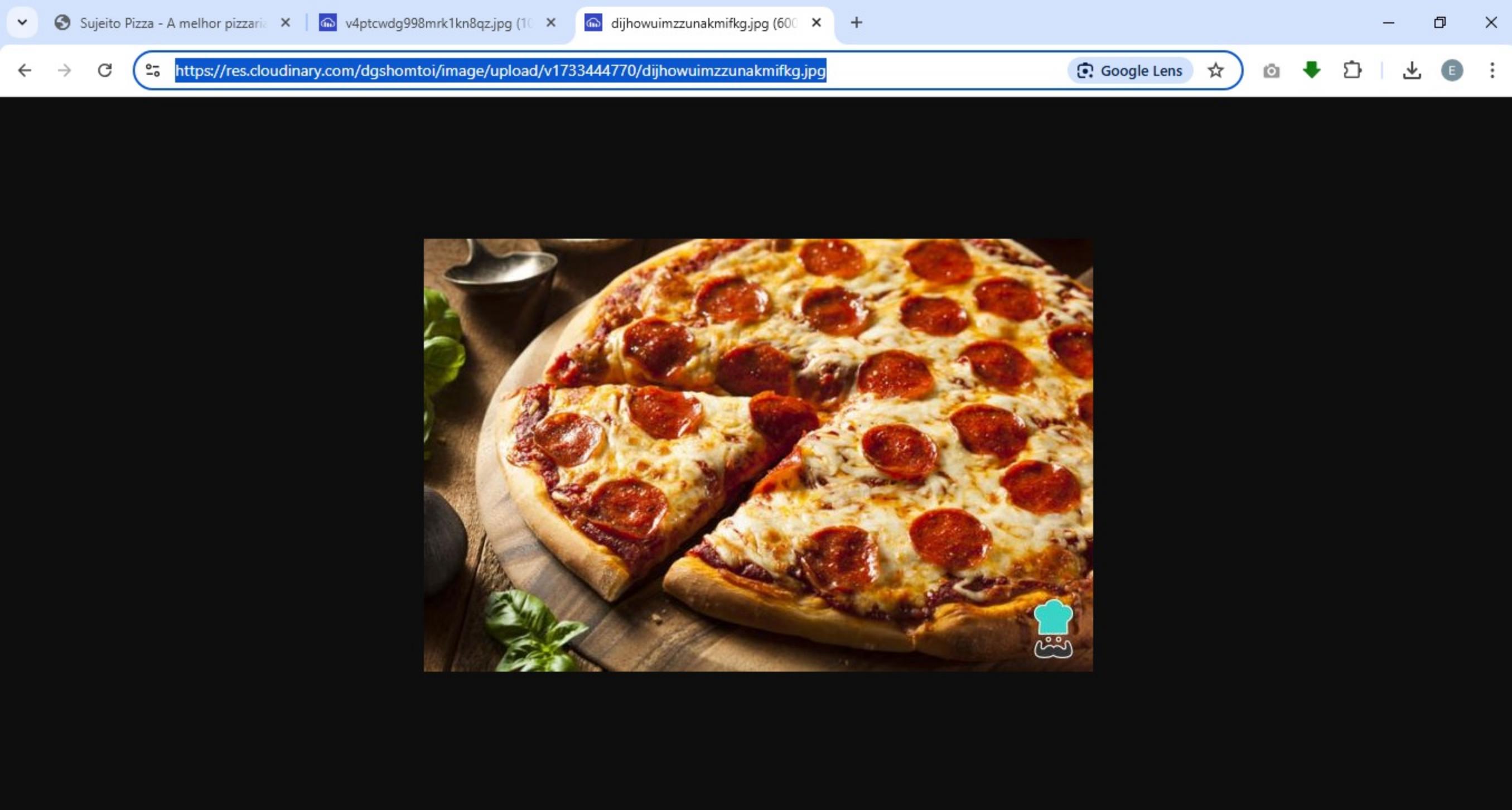
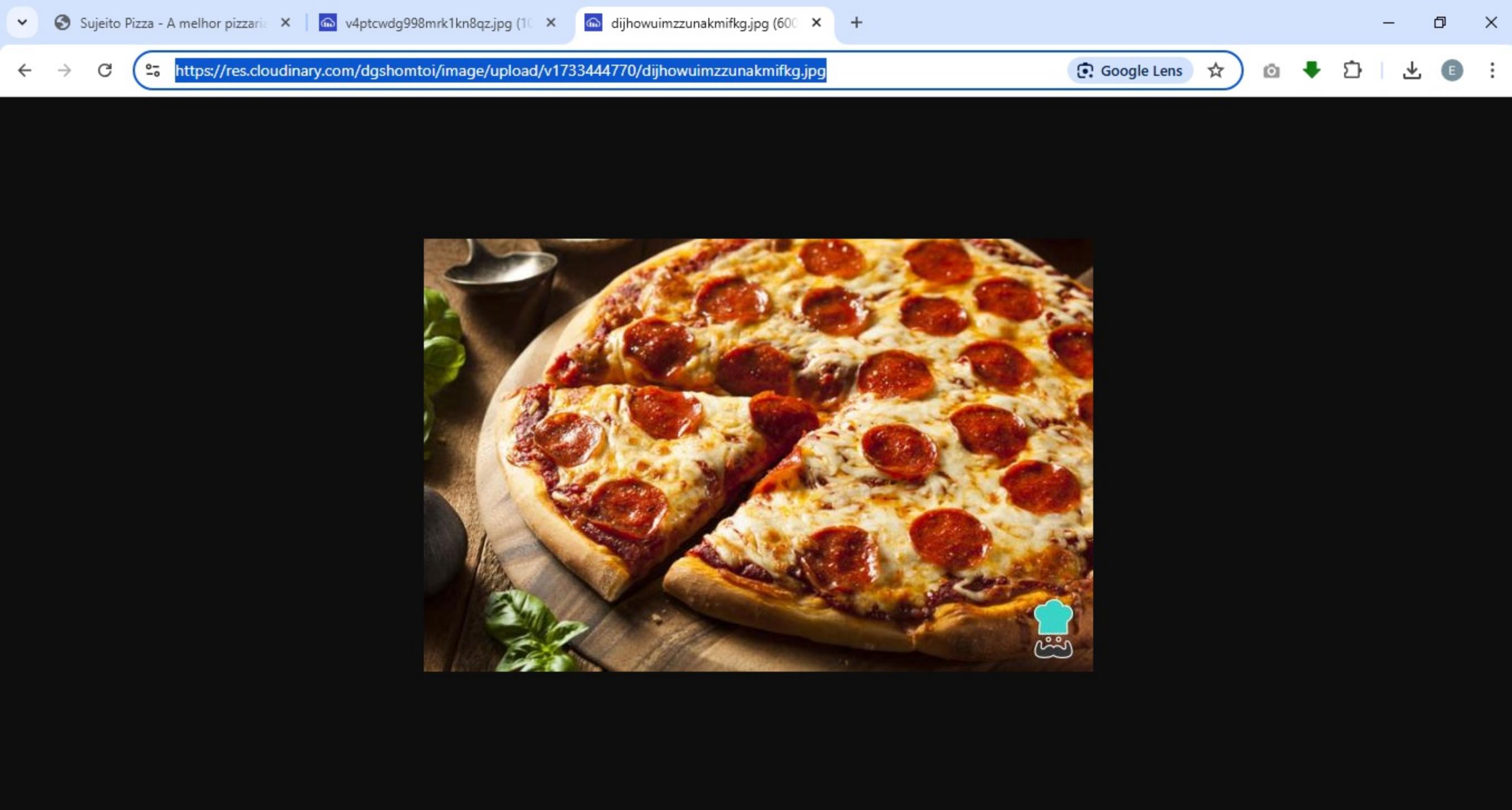
- \_prisma\_migrations
- categories
- items
- orders
- products
- users

information\_schema

pg\_catalog

	nome	price	description	banner
1	Pizza Calabresa	65	Pizza Saborosa	http://res.cloudinary.com/dgshomtoi/image/upload/v1733444770/dijhowuimzzunakmifkg.jpg
2	Pizza Frango com Catupiry	60	Pizza Top	http://res.cloudinary.com/dgshomtoi/image/upload/v1733442121/v4ptcwdg998mrk1kn8qz.jpg

Edit in modal Shift+Enter  
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





# SujeitoPizza

Categoria Produto [→]

## Últimos pedidos

Mesa 22

pizzaria

Filter

ENTITIES 217

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

Structure ↗ # Unknown just now

Query #1 products [all] orders [all]

id equals Enter Value

id text table int4(32,0) status bool draft bool name text created\_at... update\_at timestamp(3)

+ 🔎

No Data

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

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

Filter ENTITIES 217 +

public

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

information\_schema

pg\_catalog

	id	equals	Enter Value																	
1	49b330fb-5693-4520-9774-c984fcdb921a	Pizza Calabresa	65	Pizza Saborosa	http://res.clo...	2024-12-06 ...	2024-12-06 00:26:10.858	e21												
2	fd6c8de7-da0e-4373-9a75-10528f325179	Pizza Frango com Catupiry	60	Pizza Top	http://res.clo...	2024-12-05 ...	2024-12-05 23:42:01.785	e21												

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

pizzaria C +

Query #1 products [all] orders [all] items [all] x +

Filter Y

ENTITIES 217 +

public

- \_prisma\_migrations
- categories
- items (selected)
- orders
- products
- users

information\_schema

pg\_catalog

id equals Enter Value

id text amount int4(32,0) created\_at... update\_at timestamp(3) order\_id text product\_id text

No Data

No Data

localhost:5432/pizzaria postgresql Structure Unknown just now 1 > C +

Sujeito Pizza

Categoria Produto [→]

→

Últimos pedidos

Mesa 2

Sujeito Pizza

Categoria Produto [→]

Últimos pedidos 



Nenhum pedido aberto no momento..



Search.. Ctrl + P

Invite

Pizzaria



Sujeito Programador ▾

Run

POST ▾ http://localhost:3333/order

Send ▾

200 OK

35 ms

178 B

Just Now ▾

Base Environment

Add Cookies

Add Certificates

Filter

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

category

users

master



Beautify JSON

Preview

Headers 8

Cookies

Tests 0 / 0

→ Mock

Console

JSON ▾

```
1 {  
2   "table": 90,  
3   "name": "Norma"  
4 }
```

Preview ▾

```
1 {  
2   "id": "aa6116fc-cf76-4628-a12d-c6b031d1a256",  
3   "table": 90,  
4   "status": false,  
5   "draft": true,  
6   "name": "Norma",  
7   "created_at": "2024-12-06T00:40:46.453Z",  
8   "update_at": "2024-12-06T00:40:46.453Z"  
9 }
```

Preferences

Online

Made with ❤ by Kong



Search.. Ctrl + P

Invite

Pizzaria ▾

Sujeito Programador ▾

Run

POST ▾ http://localhost:3333/order

Send ▾

200 OK

35 ms

178 B

Just Now ▾



Base Environment

Add Cookies

Add Certificates

Filter



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

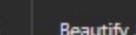
POST Cadastrar Produto

GET Listar Produto por Categoria

category

users

master



Beautify JSON

Preview

Headers ▾

Cookies

Tests 0 / 0

→ Mock

Console

JSON ▾

```
1 {  
2   "table": 90,  
3   "name": "Norma"  
4 }
```

Preview ▾

```
1 {  
2   "id": "aa6116fc-cf76-4628-a12d-c6b031d1a256",  
3   "table": 90,  
4   "status": false,  
5   "draft": true,  
6   "name": "Norma",  
7   "created_at": "2024-12-06T00:40:46.453Z",  
8   "update_at": "2024-12-06T00:40:46.453Z"  
9 }
```

Search with Google

Copy

Select All

\$.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

72 ms

237 B

24 Minutes Ago ▾

Base Environment

Add Cookies

Add Certificates

Filter

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

category

users

master



Beautify JSON

Preview

Headers ▾ 8

Cookies

Tests 0 / 0

→ Mock

Console

Preview ▾

```
1 {  
2   "order_id": "aa6116fc-cf76-4628-a12d-c6b031d1a256",  
3   "product_id": "fd6c8de7-da0e-4373-9a75-10528f325179",  
4   "amount": 2  
5 }
```

```
1 {  
2   "id": "098b188b-5183-4f17-a4b4-f250344ffecd",  
3   "amount": 2,  
4   "created_at": "2024-12-06T00:18:03.831Z",  
5   "update_at": "2024-12-06T00:18:03.831Z",  
6   "order_id": "54387899-64b2-4231-bd3e-da05d9fc8875",  
7   "product_id": "fd6c8de7-da0e-4373-9a75-10528f325179"  
8 }
```

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



Preferences

Online

Made with ❤ by Kong

pizzaria

Query #1 products [all] orders [all] items [all]

id equals Enter Value

	id	name	price	description	banner	created_at	update_at	...
1	49b330fb-5693-4520-9774-c984fcdb921a	Pizza Calabresa	65	Pizza Saborosa	http://res.clo...	2024-12-06 ...	2024-12-06 00:26:10.858	e21
2	fd6c8de7-da0e-4373-9a75-10528f325179	Pizza Frango com Catupiry	60	Pizza Top	http://res.clo...	2024-12-05 ...	2024-12-05 23:42:01.785	e21

ENTITIES 217

public

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

information\_schema

pg\_catalog

fd6c8de7-da0e-4373-9a75-10528f325179

Edit in modal Shift+Enter  
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



Search.. Ctrl + P

Invite

Pizzaria



Sujeito Programador ▾

Run

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

Send ▾

200 OK

35 ms

237 B

Just Now ▾

Params

Body

Auth

Headers

4

Scripts

Docs

Preview

Headers

8

Cookies

Tests

0 / 0

→ Mock

Console

JSON ▾

```
1 {  
2   "order_id": "aa6116fc-cf76-4628-a12d-c6b031d1a256",  
3   "product_id": "fd6c8de7-da0e-4373-9a75-10528f325179",  
4   "amount": 2  
5 }
```

Preview ▾

```
1 {  
2   "id": "43d47c1a-f7c5-4b8d-b961-a6bc8690770f",  
3   "amount": 2,  
4   "created_at": "2024-12-06T00:43:30.584Z",  
5   "update_at": "2024-12-06T00:43:30.584Z",  
6   "order_id": "aa6116fc-cf76-4628-a12d-c6b031d1a256",  
7   "product_id": "fd6c8de7-da0e-4373-9a75-10528f325179"  
8 }
```

- Base Environment
- Add Cookies
- Add Certificates
- Filter
- 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

### category

### users

### master



Beautify JSON

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



Preferences

Online

Made with ❤ by Kong

pizzaria C +

Query #1 products [all] orders [all] items [all] +

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	update_at	key
1	49b330fb-5693-4520-9774-c984fcdb921a	Pizza Calabresa	65	Pizza Saborosa	http://res.clo...	2024-12-06 ...	2024-12-06 00:26:10.858	e21
2	fd6c8de7-da0e-4373-9a75-10	Edit in modal Shift+Enter Set as NULL	60	Pizza Top	http://res.clo...	2024-12-05 ...	2024-12-05 23:42:01.785	e21

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



Search.. Ctrl + P

Invite

Pizzaria ▾



Sujeito Programador ▾

Run

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

Send ▾

200 OK

138 ms

237 B

Just Now ▾

Base Environment

Add Cookies

Add Certificates

Filter

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

category

users

master



Beautify JSON

Preview

Headers ▾ 8

Cookies

Tests 0 / 0

→ Mock

Console

Preview ▾

```
1 {  
2   "order_id": "aa6116fc-cf76-4628-a12d-c6b031d1a256",  
3   "product_id": "49b330fb-5693-4520-9774-c984fcdb921a",  
4   "amount": 1  
5 }
```

```
1 {  
2   "id": "4c3d7bb0-613f-4a04-8271-d84c11ef1d0f",  
3   "amount": 1,  
4   "created_at": "2024-12-06T00:45:09.240Z",  
5   "update_at": "2024-12-06T00:45:09.240Z",  
6   "order_id": "aa6116fc-cf76-4628-a12d-c6b031d1a256",  
7   "product_id": "49b330fb-5693-4520-9774-c984fcdb921a"  
8 }
```

\$.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

83 ms

179 B

Just Now ▾

Base Environment

Add Cookies

Add Certificates

Filter

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

category

users

master



Beautify JSON

Preview

Headers ▾ 8

Cookies

Tests 0 / 0

→ Mock

Console

JSON ▾

```
1 {  
2   "order_id": "aa6116fc-cf76-4628-a12d-c6b031d1a256"  
3 }
```

Preview ▾

```
1 {  
2   "id": "aa6116fc-cf76-4628-a12d-c6b031d1a256",  
3   "table": 90,  
4   "status": false,  
5   "draft": false,  
6   "name": "Norma",  
7   "created_at": "2024-12-06T00:40:46.453Z",  
8   "update_at": "2024-12-06T00:40:46.453Z"  
9 }
```

Preferences

Online

Made with ❤ by Kong

Sujeito Pizza

Categoría Producto [→]

**Últimos pedidos** 



Nenhum pedido aberto no momento.

Sujeito Pizza

Categoria Produto [→]

**Últimos pedidos** 

Mesa 90



 Pedidos atualizados com sucesso!



Pedidos | Mesas

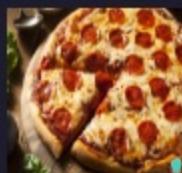


### Detalhes do pedido

Mesa 90 Norma



Qtd: 2 - **Pizza Frango com Catupiry** - R\$ 120  
Pizza Top



Qtd: 1 - **Pizza Calabresa** - R\$ 65  
Pizza Saborosa

**Valor Total: R\$ 185**

**Concluir pedido**



Pedidos Pendentes

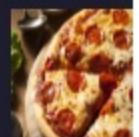
### Detalhes do pedido

Mesa 90 Norma



Qtd: 2 - Piz

Pizza Top



Qtd: 1 - Piz

Pizza Sabo

Valor Tot

Concluir pedido

Abrir imagem em uma nova guia

Salvar imagem como...

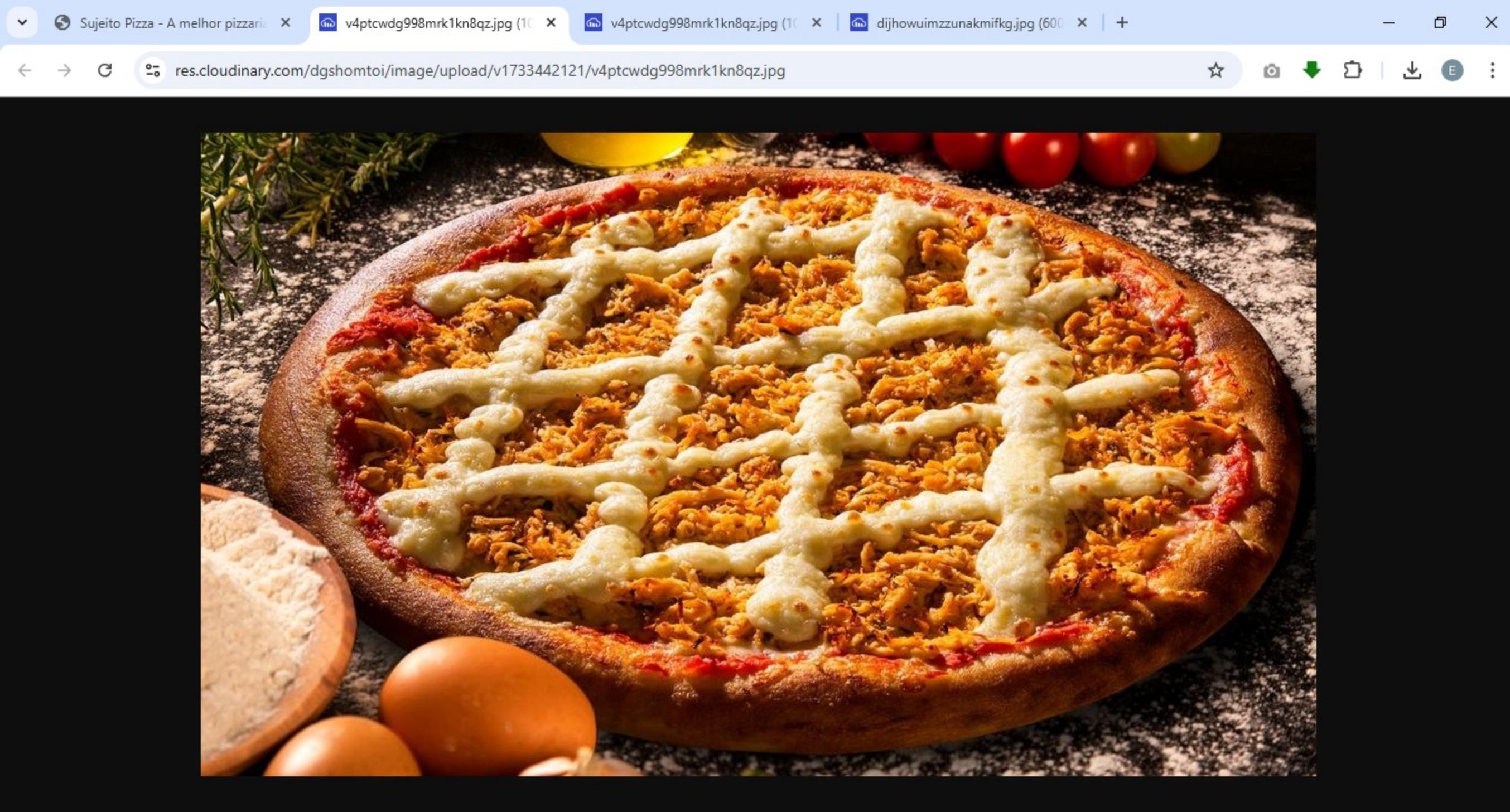
Copiar imagem

Copiar endereço da imagem

Criar QR code para esta imagem

Pesquisar com o Lens

Inspecionar



File Edit Selection View Go Run ... ⟲ ⟳ 🔎 frontend

EXPLORER

FRONTEND

- .next
  - cache
  - server
  - static
  - types
- { app-build-manifest.json
- { build-manifest.json
- { package.json
- { react-loadable-manifest.json
- trace
- > node\_modules
- > public
- src
  - app
    - dashboard
      - category
    - components
      - button
      - header
    - modal
      - index.tsx
      - styles.module.scss
    - orders
    - product
    - layout.tsx
    - page.tsx

index.tsx

```
src > app > dashboard > components > modal > index.tsx > ...
1  "use client"
2
3  import styles from './styles.module.scss'
4  import { Import, X } from 'lucide-react'
5  import { use } from 'react'
6  import { OrderContext } from '@/providers/order'
7  import { calculateTotalOrder } from '@/lib/helper'
8  import Image from 'next/image'
9
10 export function Modalorder(){
11   const { onRequestClose, order, finishOrder } = use(OrderContext);
12
13   async function handleFinishOrder() {
14     await finishOrder(order[0].order.id)
15   }
16
17   return(
18     <dialog className={styles.dialogContainer}>
19
20       <section className={styles.dialogContent}>
21         <button className={styles.dialogBack} onClick={onRequestClose}>
22           <X size={40} color="#FF3f4b" />
23         </button>
24
25         <article className={styles.container}>
26           <h2>Detalhes do pedido</h2>
27
28           <span className={styles.table}>
29             Mesa <b>{order[0].order.table}</b>
30           </span>
31
32           {order[0].order?.name && (
33             <div>{order[0].order.name}</div>
34           )
35         }
36       </article>
37     </section>
38   </dialog>
39 }
```

Ln 8, Col 1 (30 selected) Spaces: 2 UTF-8 CRLF { TypeScript JSX ✓ Prettier

File Edit Selection View Go Run ... ⟲ ⟳ 🔎 frontend

EXPLORER

FRONTEND

- .next
  - cache
  - server
  - static
  - types
- { app-build-manifest.json
- { build-manifest.json
- { package.json
- { react-loadable-manifest.json
- trace
- > node\_modules
- > public
- src
  - app
    - dashboard
      - category
    - components
      - button
      - header
    - modal
      - index.tsx
      - styles.module.scss
    - orders
    - product
    - layout.tsx
    - page.tsx

index.tsx

src > app > dashboard > components > modal > index.tsx > Modalorder > order.map() callback

```
30     </span>
31
32     {order[0].order?.name && (
33         <span className={styles.name}>
34             <b>{order[0].order.name}</b>
35         </span>
36     )}
37
38     {order.map( item => (
39         <section className={styles.item} key={item.id}>
40             <img
41                 src={item.product.banner}
42                 width={120}
43                 height={120}
44             />
45             <span>
46                 Qtd: {item.amount} - <b>{item.product.name}</b> - R$ {parseFloat(item.product.price) * item.amount}
47             </span>
48             <span className={styles.description}>
49                 {item.product.description}</span>
50         </section>
51     )));
52
53     <h3 className={styles.total}>Valor Total: R$ {calculateTotalOrder(order)} </h3>
54
55     <button className={styles.buttonOrder} onClick={handleFinishOrder}>
56         Concluir pedido
57     </button>
58
59     </article>
60
61     </section>
```

Ln 40, Col 16 (3 selected) Spaces: 2 UTF-8 CRLF ⓘ TypeScript JSX ✅ Prettier

File Edit Selection View Go Run ... ⏪ ⏩ 🔎 frontend

EXPLORER

FRONTEND

- .next
  - cache
  - server
  - static
  - types
- { app-build-manifest.json
- { build-manifest.json
- { package.json
- { react-loadable-manifest.json
- trace
- node\_modules
- public

src

- app
  - dashboard
    - category
  - components
    - button
    - header
  - modal
    - index.tsx
    - styles.module.scss
  - orders
  - product
  - layout.tsx
  - page.tsx

OUTLINE

TIMELINE

index.tsx

src > app > dashboard > components > modal > index.tsx > Modalorder > order.map() callback

```
30     </span>
31
32     {order[0].order?.name && (
33         <span className={styles.name}>
34             <b>{order[0].order.name}</b>
35         </span>
36     )}
37
38     {order.map( item => (
39         <section className={styles.item} key={item.id}>
40             <Image
41                 src={item.product.banner}
42                 alt="Foto do Produto"
43                 width={120}
44                 height={120}
45             />
46             <span>
47                 Qtd: {item.amount} - <b>{item.product.name}</b> - R$ {parseFloat(item.product.price) * item.amount}
48             </span>
49             <span className={styles.description}>
50                 {item.product.description}</span>
51         </section>
52     )));
53
54     <h3 className={styles.total}>Valor Total: R$ {calculateTotalOrder(order)} </h3>
55
56     <button className={styles.buttonOrder} onClick={handleFinishOrder}>
57         Concluir pedido
58     </button>
59
60     </article>
61
```

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

```
./src/app/dashboard/components/modal/index.tsx
./src/app/dashboard/components/orders/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/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
```

Deseja finalizar o arquivo em lotes (S/N)?

^C

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

A screenshot of a code editor interface, likely Visual Studio Code, displaying a file named `next.config.ts`. The file contains TypeScript code for configuring a Next.js application. The code imports a type from the "next" package and defines a constant `nextConfig` which includes an `images` object with a `remotePatterns` array containing a single entry for Cloudinary.

```
import type { NextConfig } from "next";
const nextConfig: NextConfig = {
  /* config options here */
  images: {
    remotePatterns: [
      {
        hostname: 'res.cloudinary.com'
      }
    ]
  }
};
export default nextConfig;
```

The code editor's sidebar shows a project structure under the `FRONTEND` folder, including `src`, `app`, `dashboard`, `components`, `fonts`, `signup`, `favicon.ico`, `globals.scss`, `layout.tsx`, `page.module.scss`, `page.tsx`, `lib`, `providers`, `services`, `middleware.ts`, `.gitignore`, `next-env.d.ts`, and `next.config.ts`.

At the bottom of the screen, there are status indicators for file changes, a search bar, and a navigation bar with icons for file operations like close, minimize, maximize, and exit.

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 3.2s
```

# SujeitoPizza

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

## Últimos pedidos

Mesa 90



Pedidos Pendentes



### Detalhes do pedido

Mesa 90 Norma



Qtd: 2 - **Pizza Frango com Catupiry** - R\$ 120  
Pizza Top



Qtd: 1 - **Pizza Calabresa** - R\$ 65  
Pizza Saborosa

**Valor Total: R\$ 185**

**Concluir pedido**

File Edit Selection View Go Run ... ← → 🔍 frontend

EXPLORER

FRONTEND

- .next
  - trace
- node\_modules
- public
- src
  - app
    - dashboard
      - category
    - components
      - button
      - header
    - modal
      - 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
- OUTLINE
- TIMELINE

index.tsx TS next.config.ts

src > app > dashboard > components > modal > index.tsx > Modalorder > order.map() callback

```
30     </span>
31
32     {order[0].order?.name && (
33         <span className={styles.name}>
34             <b>{order[0].order.name}</b>
35         </span>
36     )}
37
38     {order.map( item => (
39         <section className={styles.item} key={item.id}>
40             <Image
41                 src={item.product.banner}
42                 alt="Foto do Produto"
43                 width={120}
44                 height={120}
45             />
46             <span>
47                 Qtd: {item.amount} - <b>{item.product.name}</b> - R$ {parseFloat(item.product.price) * item.amount}
48             </span>
49             <span className={styles.description}>
50                 {item.product.description}</span>
51             </section>
52         ))
53
54     <h3 className={styles.total}>Valor Total: R$ {calculateTotalOrder(order)} </h3>
55
56     <button className={styles.buttonOrder} onClick={handleFinishOrder}>
57         Concluir pedido
58     </button>
59
60     </article>
61
```

Ln 40, Col 1 (179 selected) Spaces: 2 UTF-8 CRLF ⓘ TypeScript JSX ✅ Prettier

A screenshot of a code editor interface, likely Visual Studio Code, displaying a TypeScript file named `index.tsx` within a project structure for a Next.js application.

The project structure in the Explorer sidebar shows:

- `FRONTEND`
- `.next`
- `node_modules`
- `public`
- `src`
  - `app`
    - `dashboard`
      - `category`
    - `components`
      - `button`
      - `header`
    - `modal`
      - `index.tsx` (selected)
      - `styles.module.scss`
    - `orders`
    - `product`
    - `layout.tsx`
    - `page.tsx`
    - `fonts`
    - `signup`
  - `favicon.ico`
  - `globals.scss`
  - `layout.tsx`
  - `page.module.scss`
  - `page.tsx`
  - `lib`

The `index.tsx` file content is as follows:

```
30     </span>
31
32     {order[0].order?.name && (
33         <span className={styles.name}>
34             <b>{order[0].order.name}</b>
35         </span>
36     )}
37
38     {order.map( item => (
39         <section className={styles.item} key={item.id}>
40             <span>
41                 Qtd: {item.amount} - <b>{item.product.name}</b> - R$ {parseFloat(item.product.price) * item.amount}
42             </span>
43             <span className={styles.description}>
44                 {item.product.description}</span>
45             </section>
46     ))}
47
48     <h3 className={styles.total}>Valor Total: R$ {calculateTotalOrder(order)} </h3>
49
50     <button className={styles.buttonOrder} onClick={handleFinishOrder}>
51         Concluir pedido
52     </button>
53
54     </article>
55
56     </section>
57
58     </dialog>
59 )
60 }
```

The code editor includes standard UI elements like a search bar at the top, a status bar at the bottom, and a set of icons on the far left.



Pedidos Pendentes

Últimos pedidos

Novo



### Detalhes do pedido

Mesa 90 Norma

Qtd: 2 - **Pizza Frango com Catupiry** - R\$ 120

Pizza Top

Qtd: 1 - **Pizza Calabresa** - R\$ 65

Pizza Saborosa

**Valor Total: R\$ 185****Concluir pedido**

File Edit Selection View Go Run ... ⟲ ⟳ 🔎 frontend

EXPLORER

FRONTEND

- .next
  - trace
- node\_modules
- public
- src
  - app
    - category
    - components
      - button
      - header
    - modal
      - 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
- OUTLINE
- TIMELINE

index.tsx TS next.config.ts

```
src > app > dashboard > components > modal > index.tsx > ...
1 "use client"
2
3 import styles from './styles.module.scss'
4 import { Import, X } from 'lucide-react'
5 import { use } from 'react'
6 import { OrderContext } from '@/providers/order'
7 import { calculateTotalOrder } from '@/lib/helper'
8 import Image from 'next/image'
9
10 export function Modalorder(){
11   const { onRequestClose, order, finishOrder } = use(OrderContext);
12
13   async function handleFinishOrder() {
14     await finishOrder(order[0].order.id)
15   }
16
17   return(
18     <dialog className={styles.dialogContainer}>
19
20       <section className={styles.dialogContent}>
21         <button className={styles.dialogBack} onClick={onRequestClose}>
22           <X size={40} color="#FF3f4b" />
23         </button>
24
25         <article className={styles.container}>
26           <h2>Detalhes do pedido</h2>
27
28           <span className={styles.table}>
29             Mesa <b>{order[0].order.table}</b>
30           </span>
31
32           {order[0].order?.name && (
33             <div>
34               <h3>Nome:</h3>
35               <p>{order[0].order.name}</p>
36             </div>
37           )}
38         </article>
39       </section>
40     </dialog>
41   )
42 }
```

Ln 8, Col 1 (30 selected) Spaces: 2 UTF-8 CRLF { TypeScript JSX ✓ Prettier

A screenshot of a code editor interface, likely Visual Studio Code, displaying a TypeScript file named `index.tsx` from a project structure named `FRONTEND`.

The project tree on the left shows the following directory structure:

- `FRONTEND`
  - `.next`
    - `trace`
    - `node_modules`
    - `public`
  - `src`
    - `app`
      - `dashboard`
        - `category`
        - `components`
          - `button`
          - `header`
        - `modal`
          - `index.tsx` (selected)
          - `styles.module.scss`
      - `orders`
      - `product`
      - `layout.tsx`
      - `page.tsx`
      - `fonts`
      - `signup`
      - `favicon.ico`
      - `globals.scss`
      - `layout.tsx`
      - `page.module.scss`
      - `page.tsx`
    - `lib`
    - `OUTLINE`
    - `TIMELINE`

The code editor tab bar shows `index.tsx` and `next.config.ts`. The current file, `index.tsx`, contains the following code:

```
use client

import styles from './styles.module.scss'
import { Import, X } from 'lucide-react'
import { use } from 'react'
import { OrderContext } from '@/providers/order'
import { calculateTotalOrder } from '@/lib/helper'

export function Modalorder(){
  const { onRequestClose, order, finishOrder } = use(OrderContext);

  async function handleFinishOrder() {
    await finishOrder(order[0].order.id)
  }

  return(
    <dialog className={styles.dialogContainer}>
      <section className={styles.dialogContent}>
        <button className={styles.dialogBack} onClick={onRequestClose}>
          <X size={40} color="#FF3f4b" />
        </button>

        <article className={styles.container}>
          <h2>Detalhes do pedido</h2>

          <span className={styles.table}>
            Mesa <b>{order[0].order.table}</b>
          </span>

          {order[0].order?.name && (
            <span className={styles.name}>

```

The code uses React hooks like `useClient`, `use`, and `useEffect` along with custom hooks `useOrderContext` and `useCalculateTotalOrder` from the `lib` folder.