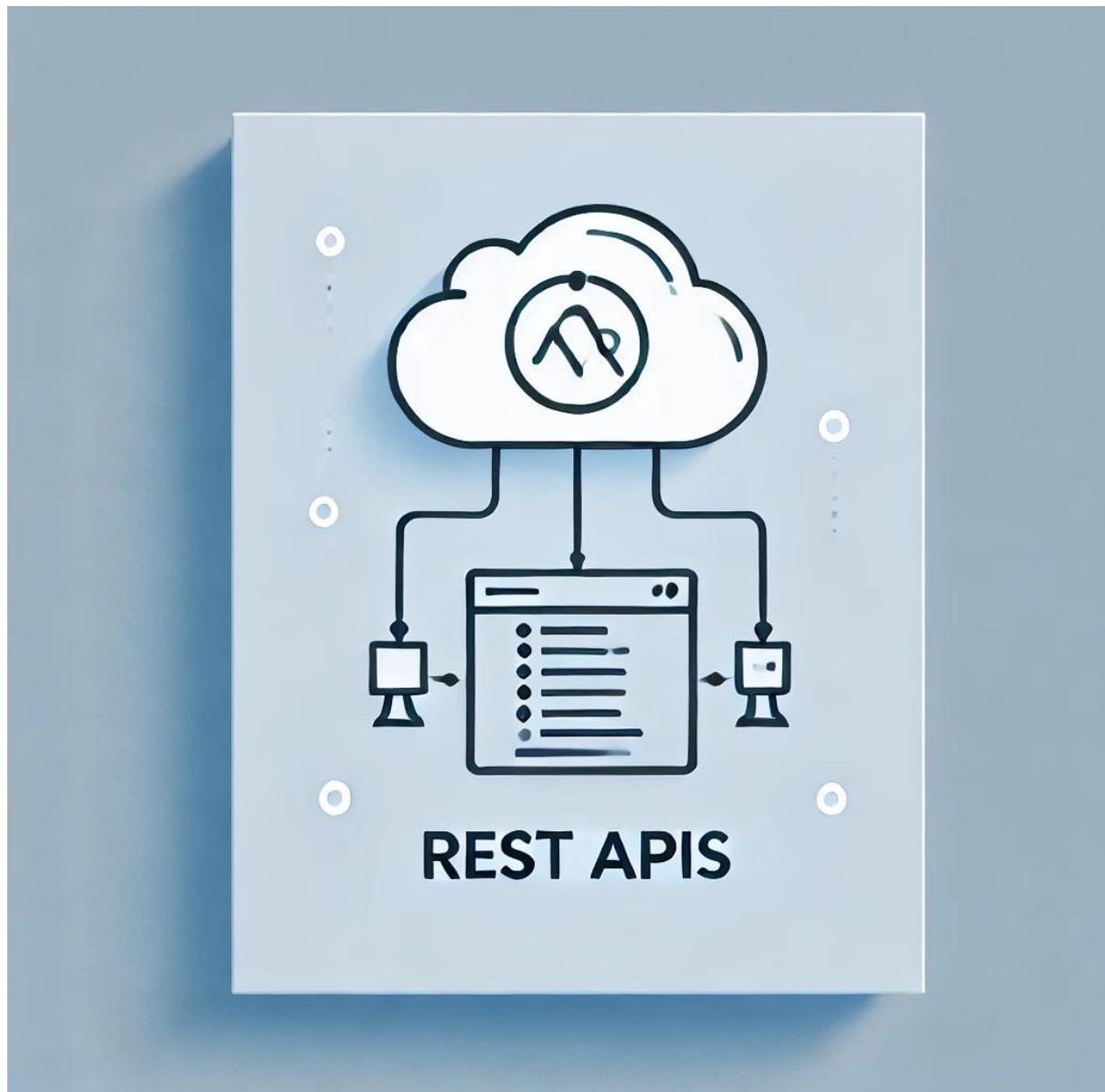


PROJETO API REST EM NODE JS



Projeto API REST

Softi9

Daniel Kucheryavenko

28/01/2025

API de Gerenciamento de Armazenamento de Empresas

Resumo:

Este projeto consiste em uma API RESTfull feita com **Node js, express, SQLite, supertest e jest**. Foi utilizado o **Swagger** para realizar a documentação. Foram usados os seguintes endpoints: **POST, PUT, GET, DELETE**.

Objetivo:

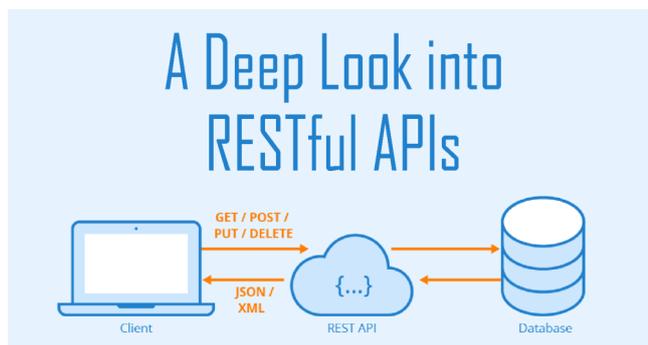
O objetivo deste projeto é projetar uma RESTfull capaz de **gerenciar o sistema de armazenamento de uma empresa**. A API permitirá funcionalidades como adicionar, atualizar, recuperar, e excluir itens do estoque, rastrear inventário e organizar itens por categorias. Será aproveitado para desenvolver o seguinte projeto (front-end) onde iremos desenvolver a parte visual do projeto, que irá fazer pedidos HTTP para a API conseguindo completar e garantir o bom funcionamento.

Ferramentas e Tecnologias Usadas:

- Express
- Node js
- Postman
- HeidiSQL
- SuperTest/ Jest
- SQLite3

O que é uma API (REST)?

Uma **API** (“Application Programming Interface”) **REST** (“Transferência Representacional de Estado”) é conhecida pela ausência de estado. A ausência de estado significa que os servidores não salvam os dados do cliente entre as solicitações. As solicitações do cliente ao servidor são semelhantes aos URLs que ao digitar num navegador para visitar um site. A resposta do servidor corresponde a **dados simples, sem a parte gráfica** típica de uma página Web.



O que é o Postman?

Postman é um software muito usado para testar APIs sem ter de escrever inúmeras linhas de código, possui uma interface moderna e de fácil uso. Conseguimos realizar vários testes dos endpoints de um projeto, guardar e documentar de uma forma simples e eficaz.

Como faço para executar a API localmente?

1. Abrir o projeto no VS Code
2. Abrir o terminal e digitar o seguinte comando:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\DKucheryavenko\Documents\ProjetoFinal> node .\index.js
```

3. Se tudo der certo irá aparecer a seguinte mensagem:

```
PS C:\Users\DKucheryavenko\Documents\ProjetoFinal> node .\index.js
Porta: 3000
Conexão estabelecida :)
Tabela "tableCompanies" ON.
Tabela "categories" ON.
Tabela "itens" ON.
```

Dia 11- Conheci o projeto e fiz uma pequena pesquisa de como organizar os ficheiros dentro do projeto, adicionei as bibliotecas **express e node**. **Criei todas as tabelas** que foram pedidas (itens, companies, categories) e adicionando as chaves e as relações. Fiz todos os **endpoints** para a tabela companies e comecei por fazer as dos itens. Conectei a DB ao **HeidiSQL** e consegui fazer um **Error Handling**.

Dia 12- Descobri como **atualizar tabelas sem as apagar**, conseguindo corrigir algumas delas (adicionar e apagar colunas, mudar os tipos de dados). Realizei todos os **requisitos das empresas, dos itens e das categorias**. Implementei **campos obrigatórios** para todas as funções que davam para não ocorrer futuros erros (por exemplo duas empresas com o nome igual).

Dia 13- Troquei o **nome de algumas colunas** por exemplo “companyName” para “Name”. Alterei os **campos dos IDs** nas tabelas e no código para o melhor funcionamento. Corrigi alguns **erros** que tinham vindo a aparecer.

Dia 14- Terminei todos os requisitos funcionais. Explorei a **documentação**, conseguindo realizar uma no Postman e pesquisei sobre a documentação no **Swagger**. Implementei a funcionalidade **Jest e supertest** no projeto. Tirei alguns **printscreens** para a documentação.

Dia 15- Tirei **todos os printscreens** necessários para o relatório, organizei tudo em pastas e com os respetivos nomes de cada imagem. Comecei a **documentação no Swagger**. Implementei o documento “env.”.

Dia 16- Continuei a **documentação Swagger**, acabei por fazer os critérios dos Endpoints do “**Gerenciamento de Itens**”, “**Gerenciamento de Categorias**” e “**Rastreamento de Inventário**”. Usei um exemplo que pedi ao chat GPT para me inspirar. Conheci os critérios deste relatório.

Dia 17- Terminei a documentação **Swagger**. **Corrigi vários erros** nos códigos de status HTTP, conseguindo ganhar mais conhecimento. Comecei a fazer o **relatório**, conseguindo fazer mais de metade.

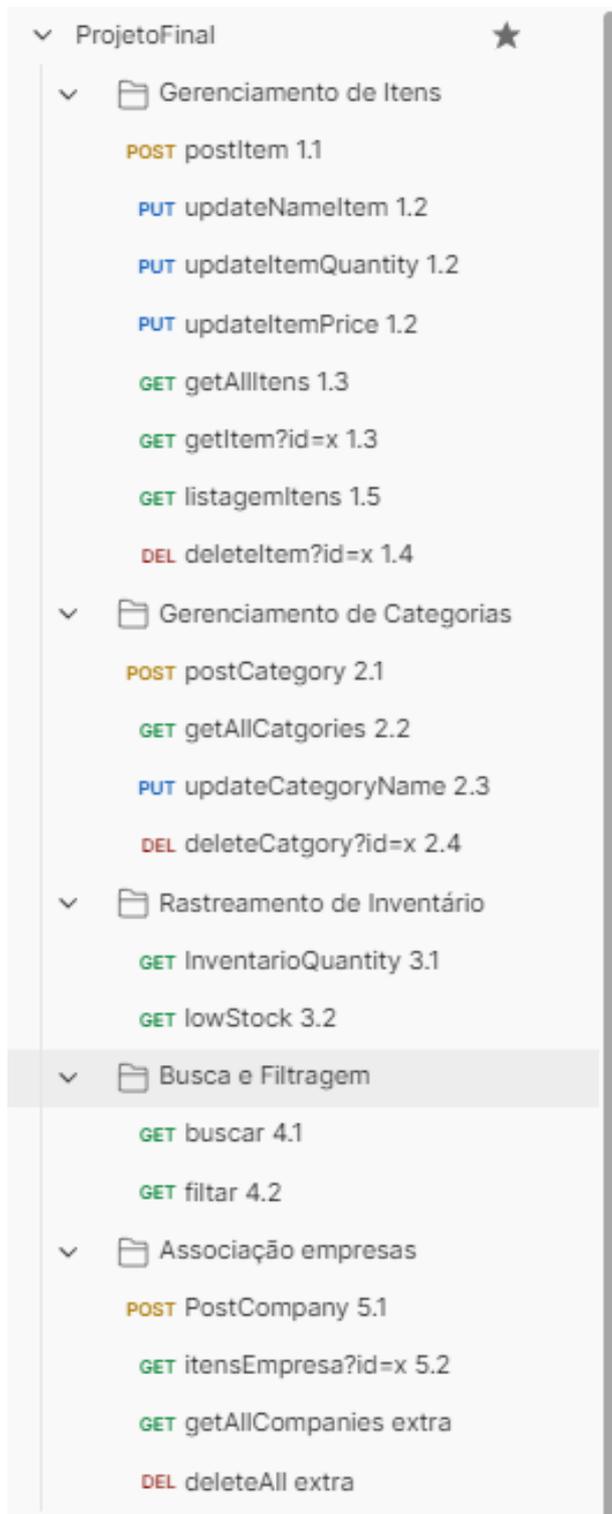
Endpoints Desenvolvidos

Endpoint	Status
POST /itens/postItem	Concluído
PUT /itens/updateNameItem	Concluído
PUT /itens/updateQuantity	Concluído
PUT /itens/updatePrice	Concluído
GET /itens/getItens	Concluído
GET /itens/getItem{ID}	Concluído
GET /itens/listagemItens	Concluído
DELETE /itens/deleteItem{ID}	Concluído
POST /categories/postCategory	Concluído
GET /categories/getAll	Concluído
PUT /categories/updateName	Concluído
DELETE /categories/deleteCategory{ID}	Concluído
GET /inventario/quantity{ID}	Concluído
GET /inventario/lowStock{Query}	Concluído
GET /filter/buscar{Query}	Concluído
GET /filter/filtrar{Query}	Concluído
POST /companies/postCompany	Concluído
GET /companies/itensEmpresa{ID}	Concluído
GET /companies/getCompanies (extra)	Concluído
DELETE /companies/deleteAll (extra)	Concluído

Procedimento de Testes

- **Postman:** Usei para testar todos os endpoints.

Printscreens (Gerenciamento de Itens):



POST http://localhost:3000/itens/postItem

Params Authorization Headers (8) **Body** Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL **JSON** ▾

```
1 {
2   "name": "teste",
3   "quantity": 25,
4   "price": 100,
5   "category_id": 2,
6   "company_id": 21
7 }
```

Body Cookies Headers (7) Test Results | ↻ **201 Created**

{ } JSON ▾ ▶ Preview 🌐 Visualize | ▾

```
1 {
2   "status": 201,
3   "produto": 14,
4   "message": "0 produto teste foi adicionado."
5 }
```

POST http://localhost:3000/itens/postItem **Send** ▾

Params Auth Headers (8) **Body** Scripts Settings **Cookies**

raw ▾ **JSON** ▾ **Beautify**

```
1 {
2   "name": "teste",
3   "quantity": 25,
4   "price": 100,
5   "category_id": 2,
6   "company_id": 21
7 }
```

Body ▾ ↻ **401 Unauthorized** - 8 ms - 296 B - 🌐 📧 ⋮

{ } JSON ▾ ▶ Preview 🌐 Visualize | ▾ 📄 📄 🔍 🔗

```
1 {
2   "status": 401,
3   "message": "0 item teste já existe."
4 }
```

POST http://localhost:3000/itens/postItem Send

Params Auth Headers (8) Body Scripts Settings Cookies

raw JSON Beautify

```
1 {
2   "name": "teste2",
3   "quantity": 25,
4   "price": 100,
5   "category_id": 223,
6   "company_id": 21
7 }
```

Body 404 Not Found - 6 ms - 306 B

{ } JSON Preview Visualize

```
1 {
2   "status": 404,
3   "message": "A categoria com id '223' não existe."
4 }
```

POST http://localhost:3000/itens/postItem Send

Params Auth Headers (8) Body Scripts Settings Cookies

raw JSON Beautify

```
1 {
2   "name": "teste2",
3   "quantity": 25,
4   "price": 100,
5   "category_id": 2,
6   "company_id": 21123
7 }
```

Body 404 Not Found - 6 ms - 306 B

{ } JSON Preview Visualize

```
1 {
2   "status": 404,
3   "message": "A empresa com id '21123' não existe."
4 }
```

POST http://localhost:3000/itens/postItem Send

Params Auth Headers (8) **Body** Scripts Settings Cookies

raw JSON Beautify

```
1 {
2   "name": "",
3   "quantity": 25,
4   "price": 100,
5   "category_id": 2,
6   "company_id": 21
7 }
```

Body 400 Bad Request - 12 ms - 305 B

{ } JSON Preview Visualize

```
1 {
2   "status": 400,
3   "message": "Todos os campos são obrigatórios"
4 }
```

POST http://localhost:3000/itens/postItem Send

Params Auth Headers (8) **Body** Scripts Settings Cookies

raw JSON Beautify

```
1 {
2   "name": "arroz"
3   "quantity": 25,
4   "price": 2,
5   "category_id": 2,
6   "company_id": 21
7 }
```

Body 409 Conflict - 14 ms - 300 B

{ } JSON Preview Visualize

```
1 {
2   "status": 409,
3   "message": "Erro de sintaxe: JSON inválido."
4 }
```

PUT http://localhost:3000/itens/updateNameItem Send

Params Auth Headers (8) **Body** Scripts Settings Cookies

raw JSON Beautify

```
1 {
2   "id": "14",
3   "name": "teste2"
4 }
5 }
```

Body 200 OK • 19 ms • 278 B

{ } JSON Preview Visualize

```
1 {
2   "message": "Update realizado",
3   "status": 200
4 }
```

PUT http://localhost:3000/itens/updateNameItem Send

Params Auth Headers (8) **Body** Scripts Settings Cookies

raw JSON Beautify

```
1 {
2   "id": "1412",
3   "name": "teste2"
4 }
5 }
```

Body 404 Not Found • 6 ms • 291 B

{ } JSON Preview Visualize

```
1 {
2   "status": 404,
3   "message": "O id 1412 não existe."
4 }
```

PUT ▼ | http://localhost:3000/itens/updateNameItem Send ▼

Params Auth Headers (8) **Body** ● Scripts Settings Cookies

raw ▼ JSON ▼ Beautify

```
1 {
2   "id": "14",
3   "name": "caderno"
4 }
5 }
```

Body ▼ ↻ 401 Unauthorized • 7 ms • 308 B • 🌐 📄 ⋮

{} JSON ▼ ▶ Preview 🔄 Visualize ▼ 🔄 📄 🔍 🔗

```
1 {
2   "status": 401,
3   "message": "O nome caderno já esta a ser usado."
4 }
```

PUT ▼ | http://localhost:3000/itens/updateQuantity Send ▼

Params Auth Headers (8) **Body** ● Scripts Settings Cookies

raw ▼ JSON ▼ Beautify

```
1 {
2   "id": "14",
3   "quantity": 30
4 }
```

Body ▼ ↻ 200 OK • 15 ms • 278 B • 🌐 📄 ⋮

{} JSON ▼ ▶ Preview 🔄 Visualize ▼ 🔄 📄 🔍 🔗

```
1 {
2   "message": "Update realizado",
3   "status": 200
4 }
```

PUT ▼ | http://localhost:3000/itens/updateQuantity Send ▼

Params Auth Headers (8) **Body** ● Scripts Settings Cookies

raw ▼ JSON ▼ Beautify

```
1 {
2   "id": "142",
3   "quantity": 30
4 }
```

Body ▼ ↺ 404 Not Found • 5 ms • 290 B • 🌐 📄 ⋮

{ } JSON ▼ ▶ Preview 🔄 Visualize ▼ 🔗 📄 🔍 🔗

```
1 {
2   "status": 404,
3   "message": "O id 142 não existe."
4 }
```

PUT ▼ | http://localhost:3000/itens/updatePrice Send ▼

Params Auth Headers (8) **Body** ● Scripts Settings Cookies

raw ▼ JSON ▼ Beautify

```
1 {
2   "id": "14",
3   "price": 12
4 }
```

Body ▼ ↺ 200 OK • 15 ms • 278 B • 🌐 📄 ⋮

{ } JSON ▼ ▶ Preview 🔄 Visualize ▼ 🔗 📄 🔍 🔗

```
1 {
2   "message": "Update realizado",
3   "status": 200
4 }
```

PUT ▼ | http://localhost:3000/itens/updatePrice Send ▼

Params Auth Headers (8) **Body** ● Scripts Settings Cookies

raw ▼ JSON ▼ Beautify

```
1 {
2   "id": "142",
3   "price": 12
4 }
```

Body ▼ ↺ **404 Not Found** - 4 ms - 290 B - 🌐 📄 ⋮

{ } JSON ▼ ▶ Preview 🔄 Visualize ▼ 🔗 📄 🔍 🔗

```
1 {
2   "status": 404,
3   "message": "O id 142 não existe."
4 }
```

GET ▼ | http://localhost:3000/itens/getItem?id=14 Send ▼

Params ● Auth Headers (6) **Body** Scripts Settings Cookies

Body ▼ ↺ **200 OK** - 4 ms - 328 B - 🌐 📄 ⋮

{ } JSON ▼ ▶ Preview 🔄 Visualize ▼ 🔗 📄 🔍 🔗

```
1 {
2   "Item": [
3     {
4       "id": 14,
5       "name": "teste2",
6       "quantity": 30,
7       "price": 12,
8       "category_id": 2,
9       "company_id": 21
10    }
11  ]
12 }
```

GET http://localhost:3000/itens/getItem?id=142 Send

Params Auth Headers (6) Body Scripts Settings Cookies

Body 404 Not Found - 4 ms - 303 B

JSON Preview Visualize

```
1 {
2   "status": 404,
3   "message": "Não foi possível encontrar o Item"
4 }
```

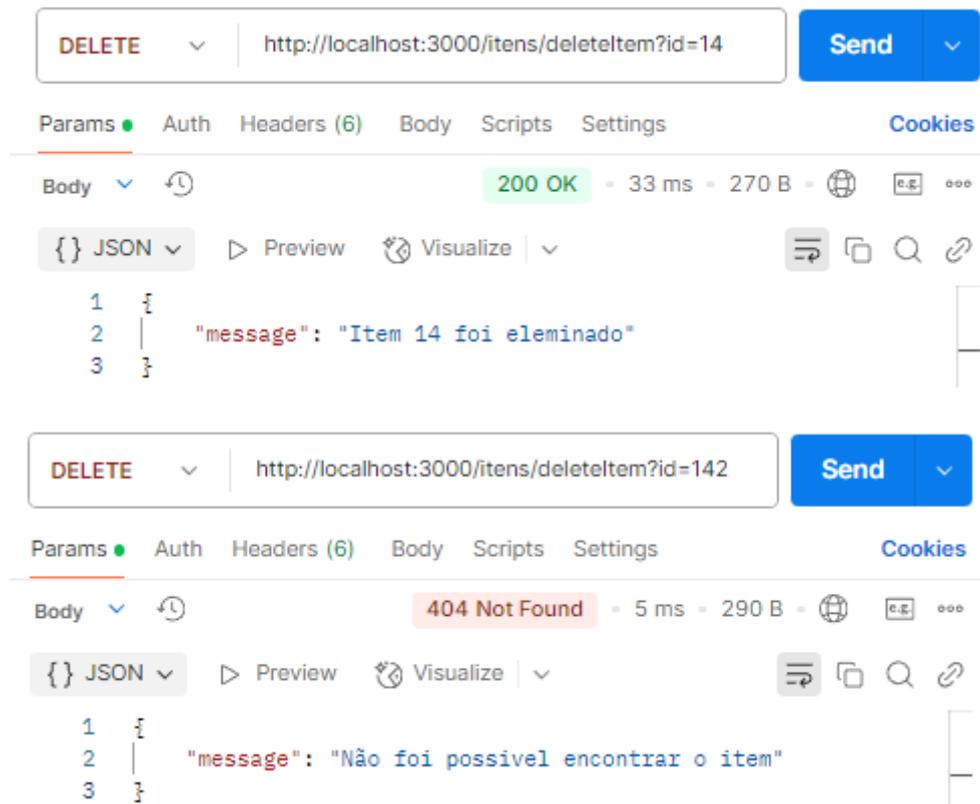
GET http://localhost:3000/itens/getItens Send

Params Auth Headers (6) Body Scripts Settings Cookies

Body 200 OK - 6 ms - 579 B

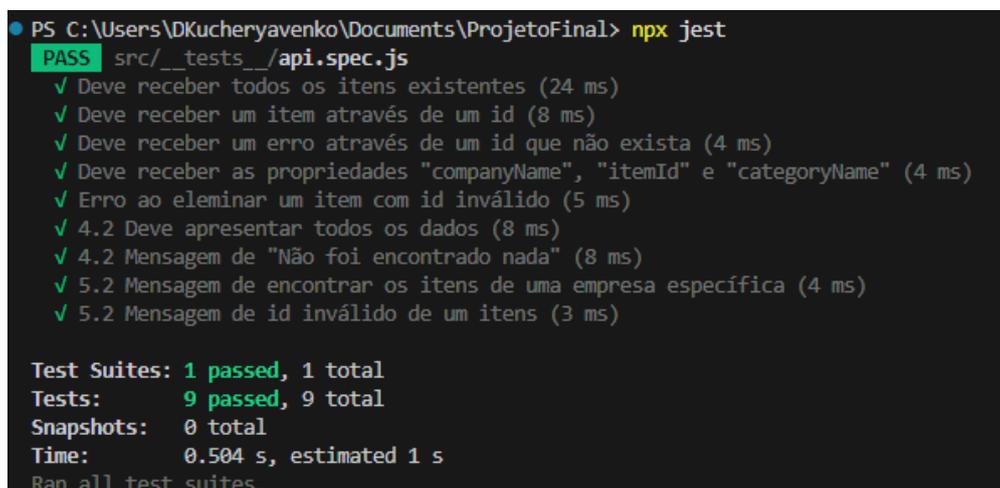
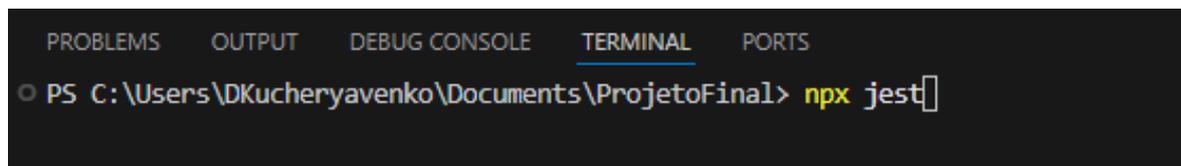
JSON Preview Visualize

```
1 {
2   "itens": [
3     {
4       "id": 2,
5       "name": "caderno",
6       "quantity": 20,
7       "price": 12,
8       "category_id": 2,
9       "company_id": 21
10    },
11    {
12      "id": 4,
13      "name": "jeans",
14      "quantity": 5,
15      "price": 35,
16      "category_id": 15,
17      "company_id": 27
18    },
19    {
20      "id": 5,
21      "name": "caneta",
22      "quantity": 100,
23      "price": 1.5,
24      "category_id": 2,
25      "company_id": 21
26    },
27    {
28      "id": 14,
29      "name": "teste2",
30      "quantity": 30,
31      "price": 12,
32      "category_id": 2,
33      "company_id": 21
34    }
35  ]
36 }
```



- **Jest/ Supertest:** Usei para fazer testes de código, é muito usado em projetos coletivos.

Printscreens (Jest/ Supertest):



Conclusão

Demorei uma semana e alguns dias a terminar o projeto. Este projeto fez-me adquirir novas aprendizagens, pois nunca tinha feito uma API antes e muito menos trabalhar com JavaScript e SQLite. Tive vários momentos em que não sabia o que tinha de fazer, o que me fazia ir procurar, consegui obter vários conhecimentos não só em programação como em aprendizagens para o futuro. Fiz vários apontamentos no meu caderno, conseguindo estudar aquilo que se precisava. Com isto tudo consegui desenvolver o meu projeto!