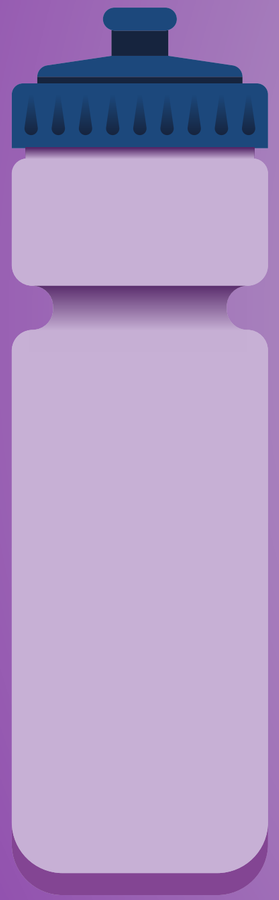
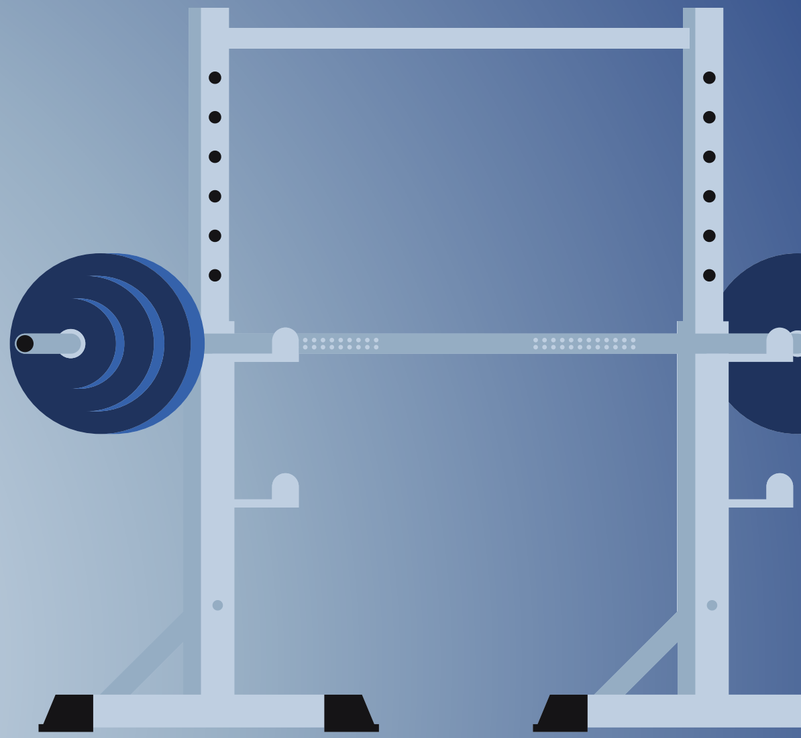
SUPLE-FIT
LOJA DE SUPLEMENTOS

Suplementos alimentares para potencializar seus treinos e alcançar seus objetivos



1º PASSO

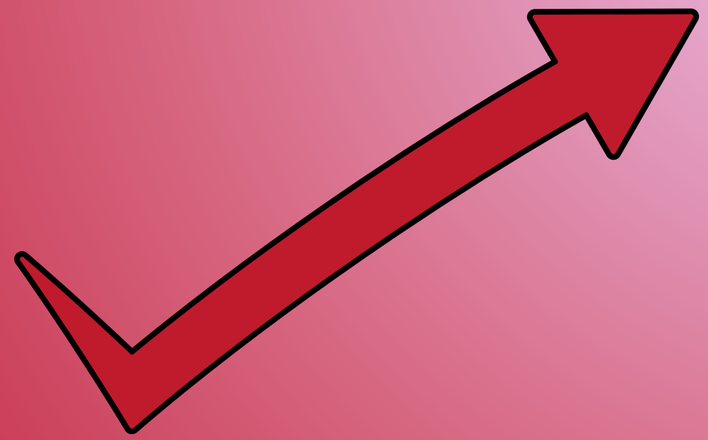
Criar a pasta ‘LOJASUPLEMENTOS’, jogar no VS Code e em seguida criar as seguintes pastas e os seguintes arquivos.



```
projeto-tutorial/
├── /backend/
│   ├── server.js
│   ├── package.json
│   └── /routes/
│       └── api.js
└── /frontend/
    └── index.html
```

2º PASSO

Abrir o terminal e colocar os seguintes códigos: cd backend, npm init -y e npm install express.



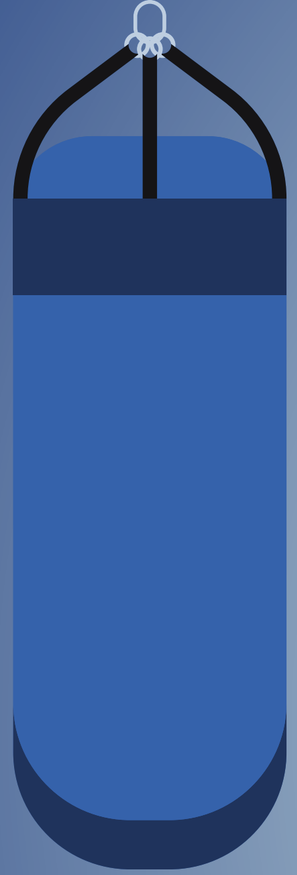
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\Aluno\Desktop\lojasuplementos> cd .\backend\
PS C:\Users\Aluno\Desktop\lojasuplementos\backend> npm init -y
Wrote to C:\Users\Aluno\Desktop\lojasuplementos\backend\package.json:

{
  "name": "backend",
  "version": "1.0.0",
  "main": "server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "node server.js"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "@supabase/supabase-js": "^2.84.0",
    "cors": "^2.8.5",
    "dotenv": "^17.2.3",
    "express": "^5.1.0"
  },
  "devDependencies": {},
  "description": ""
}



PS C:\Users\Aluno\Desktop\lojasuplementos\backend> npm install express
```

3º PASSO

Colocar o código no arquivo ‘server.js’

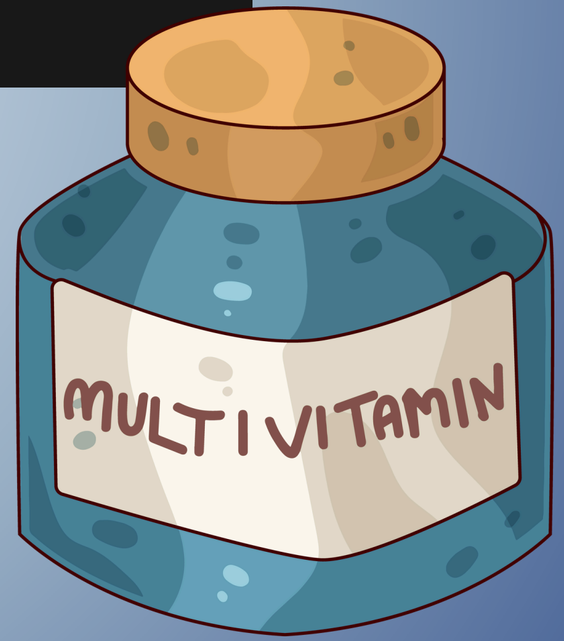
```
const express = require('express');
const cors = require('cors');
require('dotenv').config();
const { createClient } = require('@supabase/supabase-js');

const app = express();
const PORT = process.env.PORT || 3000;

// Middleware
app.use(cors());
app.use(express.json());

// Inicializar Supabase
const supabase = createClient(
  process.env.SUPABASE_URL,
  process.env.SUPABASE_KEY
);

// Rota teste
app.get('/', (req, res) => {
  res.json({ mensagem: 'API rodando!' });
});

// GET - Buscar todos os produtos
app.get('/produtos', async (req, res) => {
  try {
    const { data, error } = await supabase
      .from('produtos')
      .select('*');
    if (error) throw error;
    res.json(data);
  } catch (error) {
    res.status(400).json({ erro: error.message });
  }
});

// GET - Buscar produto por ID
app.get('/produtos/:id', async (req, res) => {
  try {
    const { data, error } = await supabase
      .from('produtos')
      .select('*')
      .eq('id', req.params.id)
      .single();
    if (error) throw error;
    res.json(data);
  } catch (error) {
    res.status(400).json({ erro: error.message });
  }
});

// POST - Criar novo produto
app.post('/produtos', async (req, res) => {
  try {
    const { nome, categoria, preco, estoque, descricao } = req.body;
    const { data, error } = await supabase
      .from('produtos')
      .insert([{ nome, categoria, preco, estoque, descricao }])
      .select();
    if (error) throw error;
    res.json(data);
  } catch (error) {
    res.status(400).json({ erro: error.message });
  }
});

// PUT - Atualizar produto
app.put('/produtos/:id', async (req, res) => {
  try {
    const { nome, preco } = req.body;
    const { data, error } = await supabase
      .from('produtos')
      .update({ nome, preco })
      .eq('id', req.params.id)
      .select();
    if (error) throw error;
    res.json(data);
  } catch (error) {
    res.status(400).json({ erro: error.message });
  }
});

// DELETE - Deletar produto
app.delete('/produtos/:id', async (req, res) => {
  try {
    const { error } = await supabase
      .from('produtos')
      .delete()
      .eq('id', req.params.id);
    if (error) throw error;
    res.json({ mensagem: 'Produto deletado' });
  } catch (error) {
    res.status(400).json({ erro: error.message });
  }
});

app.listen(PORT, () => {
  console.log(`Servidor rodando em http://localhost:${PORT}`);
});
```



4º PASSO

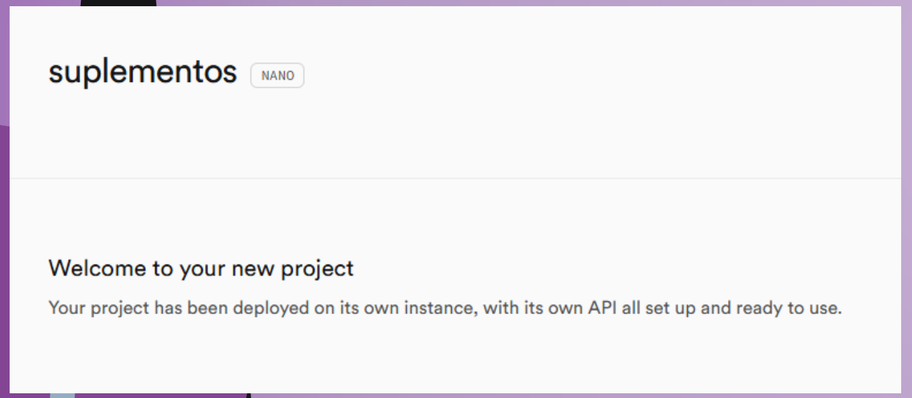
Executar o ‘node server.js’

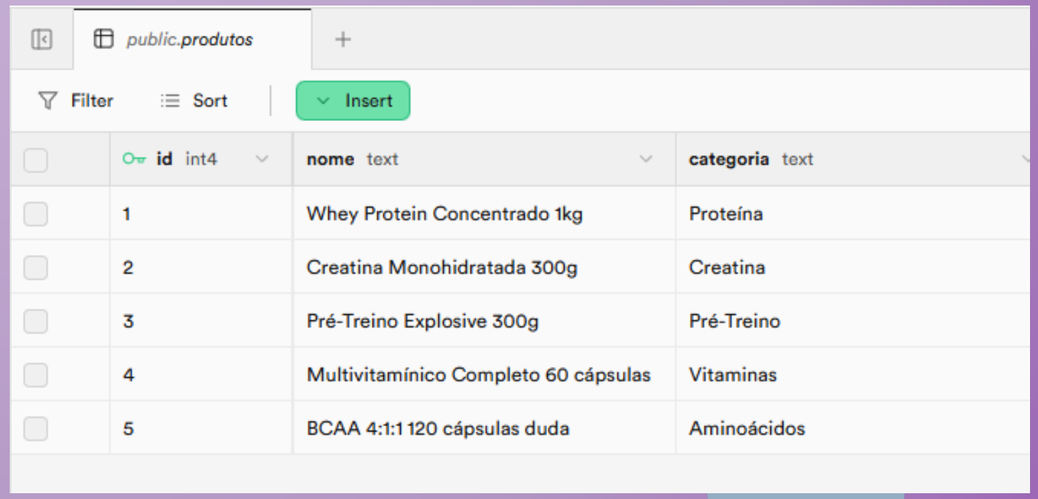
```
PS C:\Users\Aluno\Desktop\lojasuplementos> cd .\backend\
PS C:\Users\Aluno\Desktop\lojasuplementos\backend> node server.js
[dotenv@17.2.3] injecting env (3) from .env -- tip: ⚙ suppress all logs with { quiet: true }
Servidor rodando em http://localhost:3000
```



5º PASSO

Criar novo projeto no supabase, e criar a tabela ‘produtos’.





6º PASSO

Abrir o arquivo ‘.env’ e adicionar o URL e a KEY do seu supabase

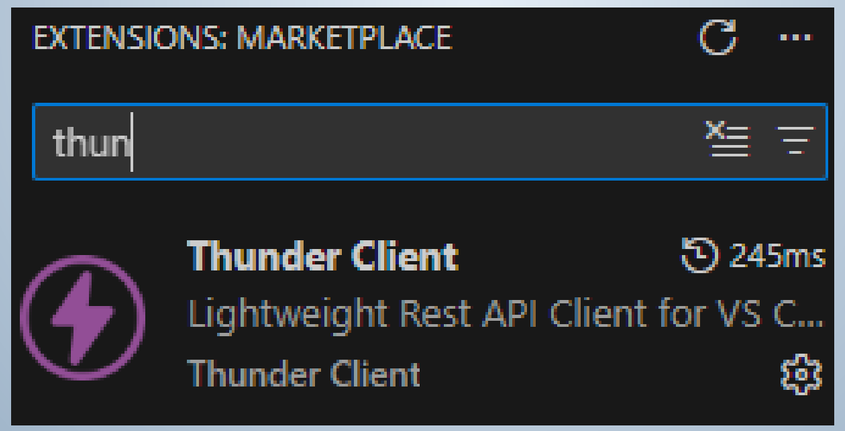
server.js | New Request | .env | api.js | index.

backend > .env

```
1  SUPABASE_URL=https://rrgtxwofyasgezugqqpx.supabase.co
2  SUPABASE_KEY=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdX
3  PORT=3000
```

7º PASSO

Baixar o Thunder Client para poder testar os comandos.



8º PASSO

Testar os comandos:
GET, GET, POST, PUT e DELETE