

RELOJOARIA



ETAPAS
BACKEND



ETAPA 1

ESTRUTURAS DE PASTAS

CRIE A SEGUINTE ESTRUTURA NO VS CODE:

```
✓ RELOJOARIA
  ✓ backend
    ✓ routes
      .env
    package-lock.json
    package.json
  server.js
```

OBJETIVO: ORGANIZAR O PROJETO DE FORMA MODULAR PARA
FACILITAR A MANUTENÇÃO E O CRESCIMENTO DA API DE RELOJOARIA.



ETAPA 2

INICIALIZAR BACKEND

1. Abra o terminal na pasta /backend

No VS Code: Terminal → New Terminal

2. Execute o comando:

```
npm init -y
```

3. Instale o Express:

```
npm install express
```

OBJETIVO: PREPARAR O AMBIENTE NODE.JS E INSTALAR O EXPRESS PARA DAR VIDA À API DA RELOJOARIA.

ESTA ETAPA ESTABELECE A FUNDACÃO TECNOLÓGICA DA SUA API.





ETAPA 3

TABELA DE PRODUTOS

Crie uma tabela de produtos no Supabase

```
CREATE TABLE public.relojoaria_produtos (
    -- Chave primária: UUID gerado automaticamente
    id UUID DEFAULT uuid_generate_v4() PRIMARY KEY,
    -- Detalhes do produto
    name VARCHAR(100) NOT NULL,
    description TEXT,
    -- '⌚' para relógio de pulso, '⌚' para relógio de mesa, etc.
    emoji VARCHAR(10) NOT NULL DEFAULT '⌚',
    -- Preço deve ser positivo
    price DECIMAL(10, 2) NOT NULL CHECK (price >= 0),
    -- Estoque (pode ser negativo se permitir encomendas)
    stock INTEGER NOT NULL DEFAULT 0,
    -- Auditoria (tracking de tempo)
    created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),
    updated_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()
);
```





ETAPA 4

ROTAS DO SUPABASE

1. Instale as dependências necessárias:

```
npm install cors dotenv @supabase/supabase-js
```

2. Crie arquivo .env na pasta /backend:

```
SUPABASE_URL=https://seu-projeto.supabase.co  
SUPABASE_KEY=sua-chave-publica-aqui  
PORT=3000
```

Objetivo: Substituir a simulação de dados em memória por uma persistência real e integrar a API ao banco de dados Supabase para rotas completas e funcionais.

Esta etapa transforma sua API de um protótipo em uma aplicação robusta, onde os dados dos relógios sobrevivem ao desligamento do servidor.

ETAPA 5

TESTAR O SERVIDOR

1. No terminal da pasta /backend, execute:

```
node server.js
```

Se vir "Servidor rodando em http://localhost:3000" no terminal,
parabéns! ✓

2. Abra seu navegador e acesse:

```
http://localhost:3000
```

Objetivo: Verificar se o backend está ativo e respondendo
corretamente às requisições básicas.
Antes de construir as rotas complexas, você precisa ter certeza
de que o servidor está de pé.

