

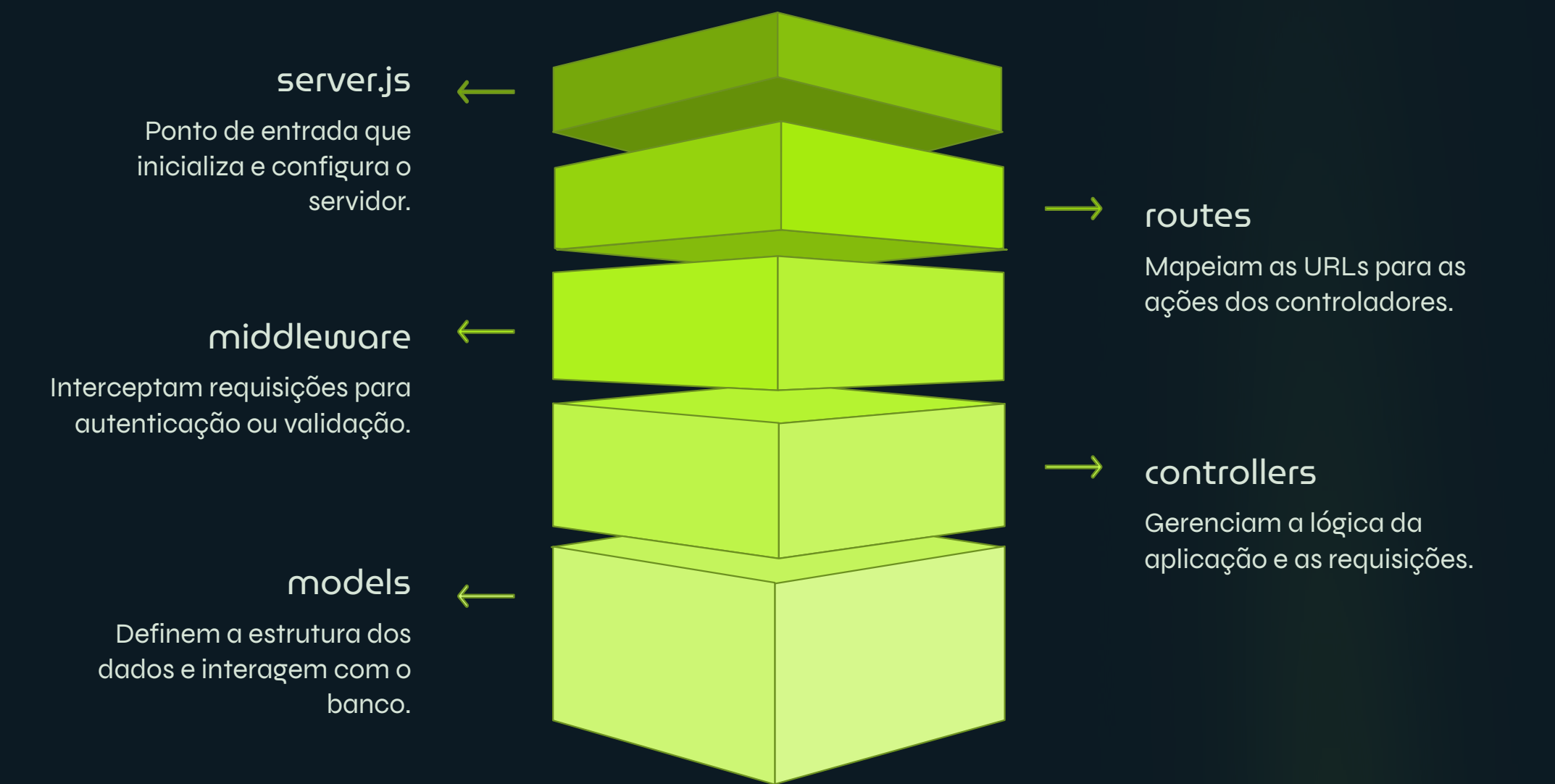
# Fase 0: Preparando a Pistola de Portais

Antes de viajar por dimensões, precisamos montar nosso equipamento.

## 1. Estrutura de Pastos

Uma boa organização evita que seu projeto se transforme em um "Cronenberg". Crie a seguinte estrutura:

```
/
|- controllers/ // A lógica que controla o que acontece em cada rota.
|- models/ // Os "Schemes" ou plantas do nosso banco de dados.
|- routes/ // As URLs da nossa API.
|- middleware/ // Guardas que verificam permissões antes de acessar uma rota.
|- .env // Nossas variáveis secretas.
|- server.js // O ponto de partida do nosso universo.
```



## 2. Instalação das Dependências

Abra o terminal e instale os pacotes necessários:

```
npm install express mongoose bcryptjs jsonwebtoken cors dotenv joi
```

**express**  
Nosso framework para criar o servidor.

**mongoose**  
Para modelar e conversar com nosso banco de dados MongoDB de forma estruturada.

**bcryptjs**  
Para transformar senhas em códigos indecifráveis (hashing).

**jsonwebtoken (JWT)**  
Para criar passes de acesso interdimensionais (tokens).

**cors**  
Para permitir que nosso front-end converse com o backend.

**dotenv**  
Para carregar nossas variáveis secretas do arquivo .env.

**joi**  
Para validar os dados que chegam, garantindo que ninguém envie um Meeseeks onde deveria haver um e-mail.

## 3. Opções de Banco de Dados: Local vs. Atlas (Nuvem)

Você tem duas opções para seu laboratório de dados.

**MongoDB Local**  
É como construir um laboratório na sua própria garagem. Você instala o MongoDB Community Server diretamente na sua máquina. É rápido para desenvolvimento, mas requer que você gerencie tudo.

**MongoDB Atlas (Recomendado)**  
É como usar um dos laboratórios secretos do Rick espalhados pelo multiverso. O Atlas é a versão em nuvem do MongoDB. É a opção recomendada para este projeto.  
**Vantagens:** Não precisa instalar nada na sua máquina; Possui um nível gratuito perfeito para projetos de desenvolvimento; Seus dados ficam seguros na nuvem, acessíveis de qualquer dimensão (ou computador).

### Passos Resumidos para Configurar o Atlas:

- Crie uma conta gratuita no site do MongoDB Atlas.
- Crie um novo "Cluster" no plano gratuito (M0). Escolha um provedor de nuvem e uma região próximos a você.
- Segurança de Rede: Vá para Network Access e adicione seu endereço IP atual. Para simplificar durante o desenvolvimento, você pode clicar em "Allow Access from Anywhere" (0.0.0.0/0), mas lembre-se que isso não é seguro para produção!
- Usuário do Banco: Vá para Database Access e crie um novo usuário e senha. Anote-os bem!
- Obtenha a Connection String: Volte para a visão geral do seu Cluster, clique em "Connect", escolha "Drivers", e copie a string de conexão fornecida. Ela será parecida com mongodb+srv://:@cluster....

## 4. Conexão com o MongoDB (server.js)

Crie o arquivo server.js e adicione o código para se conectar ao seu banco de dados.

### server.js

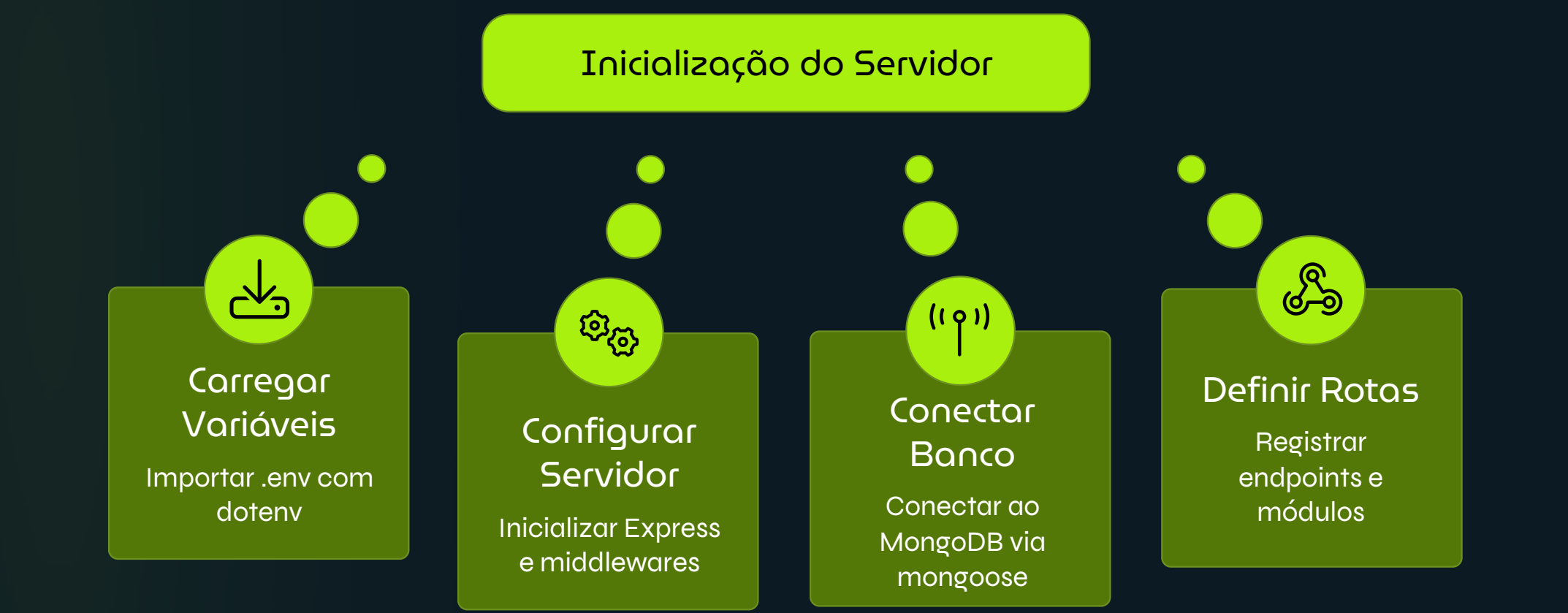
```
require('dotenv').config();
const express = require('express');
const mongoose = require('mongoose');
const cors = require('cors');

const app = express();
app.use(cors());
app.use(express.json());

// Conexão com o MongoDB
const MONGO_URI = process.env.MONGO_URI || 'mongodb://localhost:27017/ricknmorty-shop';

mongoose.connect(MONGO_URI)
.then(() => console.log('Conectado ao MongoDB... Wubba Lubba Dub Dub!'))
.catch(err => console.error('Não foi possível conectar ao MongoDB...', err));

// Rotas (vamos adicionar depois)
// app.use('/api/users', require('./routes/userRoutes'));
// app.use('/api/auth', require('./routes/authRoutes'));
const PORT = process.env.PORT || 5000;
app.listen(PORT, () => console.log(`Servidor rodando na dimensão ${PORT}`));
```



Não se esqueça de criar um arquivo .env na raiz e adicionar sua string de conexão que você copiou do MongoDB Atlas. Lembre-se de substituir pela senha que você criou.

```
MONGO_URI=mongodb+srv://seu-usuario:sua-senha@cluster...
```