

Se você já tem uma base e quer o **mínimo de esforço** possível, a melhor estratégia em 2026 é fugir da configuração manual e usar **geradores de código** ou **frameworks "opinados"** que já entregam a estrutura pronta.

Aqui estão as três formas mais rápidas, da "quase automática" até a "mais profissional":

1. O Jeito "Relâmpago": Fastify CLI

O **Fastify** é hoje o sucessor espiritual do Express. Ele é mais rápido em performance e possui um gerador oficial que cria uma estrutura completa com pastas para rotas, plugins e testes com um único comando.

- **Por que escolher:** Mínimo de boilerplate e máxima velocidade de execução.

- **Comando:**

```
Bash
npx fastify-cli generate meu-projeto --lang=ts
cd meu-projeto
npm install
npm run dev
```

Isso gera uma API com **JSON Schema** (validação automática de dados) e suporte a **plugins**, pronta para produção.

2. O Jeito "Tudo em Um": NestJS

Se você quer uma API que já venha com "tudo dentro" (banco de dados, validação, documentação Swagger) e não quer perder tempo decidindo qual biblioteca usar para cada coisa.

- **Por que escolher:** O CLI do NestJS cria recursos completos (CRUD) com um comando.
 - **Passo a passo:**
 1. Instale o CLI: `npm i -g @nestjs/cli`
 2. Crie o projeto: `nest new projeto-api`
 3. **A mágica:** Dentro da pasta, rode `nest g resource usuarios`.
 - Isso vai perguntar se você quer REST API. Diga "sim" e ele criará automaticamente o Controller, Service, DTO e Entidade com as rotas GET, POST, PATCH e DELETE prontas.
-

3. O Jeito "Moderno & Minimalista": Hono

Muito popular em 2026, o **Hono** é extremamente leve e funciona em qualquer lugar (Node,

Bun, Cloudflare Workers). A sintaxe é idêntica ao Express, mas muito mais rápida e com suporte nativo a TypeScript.

- **Comando:**

Bash

```
npm create hono@latest meu-app
```

- **Vantagem:** É a menor curva de aprendizado se você já conhece Express, mas com performance de última geração.

Comparativo de Esforço

Ferramenta	Esforço Inicial	Recursos Inclusos	Ideal para...
Fastify CLI	Baixo	Rotas, Plugins, Logger	Performance extrema e simplicidade.
NestJS CLI	Médio	CRUD Automático, Swagger, DI	Projetos que vão crescer (Enterprise).
Express Generator	Muito Baixo	Estrutura básica de pastas	Protótipos muito simples (está ficando legado).

Minha Recomendação Técnica

Se você quer **velocidade de desenvolvimento** sem abrir mão de boas práticas, use o **NestJS**. O comando `nest g resource` economiza horas de escrita de código repetitivo (boilerplates).

Se você quer algo **leve e direto**, vá de **Fastify CLI**.

Qual dessas abordagens parece mais interessante para o seu projeto atual? Posso te ajudar com os comandos de configuração do banco de dados para qualquer uma delas.