

# Do Zero até AôDev: Roteiro de Estudos Back-end - 2 Anos

## Ano 1 Fundamentos e Consolidação

### Mês 1 Retomada com Projeto Prático

- Revisar JS moderno (map, filter, async/await, destructuring, módulos)
- Criar estrutura base do projeto (Node + Express + MVC)
- Criar conexão com PostgreSQL via Prisma
- Implementar: Cadastro/Login com hash da senha (bcrypt)

### Mês 2 CRUD completo e autenticação JWT

- Criar modelos User e DailyReport no Prisma
- Implementar endpoints: POST /reports, GET /reports?from=&to=
- Implementar autenticação com JWT (login, middleware, proteção de rotas)
- Salvar e buscar relatórios por usuário autenticado

### Mês 3 Testes e Validações

- Introdução ao Jest e Supertest
- Escrever testes unitários e de integração para rotas
- Implementar validações com Zod ou Joi nos endpoints
- Aplicar Helmet, CORS e Rate Limiting

### Mês 4 Documentação e PDF

- Criar documentação com Swagger ou Postman Docs
- Criar endpoint GET /reports/summary/weekly
- Gerar PDF com PDFKit ou Puppeteer
- Preparar endpoint: /reports/summary/weekly?format=pdf

### Mês 5 MongoDB e comparação

- Aprender MongoDB básico (Atlas, Compass)
- Criar versão Mongo do DailyReport
- Comparar: SQL x NoSQL (relacional x documentos)
- Criar endpoint opcional para salvar em MongoDB (log ou fallback)

### Mês 6 Deploy e CI/CD

- Aprender Docker básico (containerizar app e DB)
- Deploy em Railway ou Render (backend + banco)
- Adicionar GitHub Actions para rodar testes e deploy
- Criar README bem escrito com instruções de uso

### Mês 7 Arquitetura Profissional

- Reorganizar projeto com pastas: services, repositories, controllers

- Criar regras de negócio em Services
- Aplicar princípios SOLID
- Refatorar endpoints antigos para nova arquitetura

## **Mês 8 Cron Job e Email**

- Aprender agendamento com node-cron
- Criar tarefa: domingo à noite buscar relatórios da semana
- Gerar PDF do resumo e enviar com nodemailer
- Configurar conta de envio (ex: Gmail ou SMTP Test)

## **Mês 9 Clean Code e Git Avançado**

- Aplicar regras de Clean Code (nomes, modularidade, DRY)
- Criar hooks/lint (ex: ESLint, Prettier)
- Aprender comandos Git mais avançados
- Aprender GitFlow ou trunk-based flow

## **Mês 10 Segurança e Boas Práticas**

- Proteger contra ataques: SQL Injection, XSS, CSRF
- Atualizar autenticação com Refresh Token
- Criar RBAC (admin, user)
- Adicionar logs com Winston ou Pino

## **Mês 11 Painel de Produtividade (extra)**

- GET /reports/stats
- Retornar dados como: dias com report, % produtividade, total por tag
- Organizar respostas com DTOs (Data Transfer Objects)

## **Mês 12 Revisão Geral e Portfólio**

- Revisar projeto, refatorar onde for necessário
- Criar vídeo demo explicando o sistema
- Publicar no GitHub com README completo
- Criar portfólio pessoal (GitHub Pages com tuas infos + projetos)

## **Ano 2 Avançado e Preparação para o Mercado**

### **Mês 13 Domínio de Arquitetura**

- Estudar camadas: Service, Repository, Controller
- Aprender Domain-Driven Design (DDD) básico
- Refatorar projeto para arquitetura mais robusta

### **Mês 14 Refresh Token e RBAC**

- Implementar Refresh Token para reautenticação segura

- Criar controle de acesso por função (RBAC)
- Aplicar conceitos de segurança em produção

### **Mês 15 Cache e Performance**

- Estudar Redis para cache
- Aplicar cache em endpoints frequentes (ex: /summary)
- Analisar performance com ferramentas como Postman + ab (Apache Bench)

### **Mês 16 Microserviços**

- Aprender conceitos de microserviços
- Criar microserviço de envio de email
- Integrar com fila (RabbitMQ ou BullMQ)

### **Mês 17 Integrações com APIs externas**

- Integrar com Slack ou Telegram via bot
- Criar alertas automáticos via webhook
- Entender boas práticas de consumo de APIs

### **Mês 18 GraphQL e Websockets**

- Aprender o básico de GraphQL com Apollo Server
- Criar rota alternativa via GraphQL para relatórios
- Usar WebSocket para notificações em tempo real

### **Mês 19 Estrutura de Dados e Algoritmos**

- Estudar listas, pilhas, filas, árvores, grafos e recursão
- Resolver problemas no LeetCode (nível fácil a médio)
- Criar repositório de desafios resolvidos com explicações

### **Mês 20 Preparação para Entrevistas**

- Simular entrevistas técnicas (com tempo cronometrado)
- Treinar explicação de projetos pessoais
- Revisar conceitos chave de sistema distribuído e arquitetura REST

### **Mês 21 Projeto Final Avançado**

- Escolher problema real e modelar uma API completa
- Aplicar mensageria, cache, segurança, testes e CI/CD
- Publicar e documentar como produto completo

### **Mês 22 Revisão Completa e Feedback**

- Refinar todos os projetos
- Pedir revisão de código (code review via GitHub)
- Analisar feedback e aplicar melhorias

## **Mês 23 Divulgação e Portfólio Profissional**

- Criar site pessoal ou Notion portfolio
- Publicar projetos com vídeo, README e link de deploy
- Divulgar em comunidades (LinkedIn, Dev.to, etc.)

## **Mês 24 Entrevistas e Aceleração**

- Aplicar para vagas júnior/pleno
- Participar de eventos, hackathons ou open source
- Refinar pitch pessoal e CV técnico