

Como se tornar programador em 2026

Um roteiro de estudos para sair do zero e chegar a uma aplicação publicada. Aprenda construindo com **Node.js e NestJS no backend** e **Next.js no frontend**, usando assistentes de IA para apoiar seu trabalho sem pular os fundamentos.

/ 12 meses de referência / Projetos em todas as etapas / IA como assistente de programação
/ Atualizado em 2026

ANTES DE COMEÇAR

O ritmo importa mais que a pressa

01 / Constância

Estude um pouco, toda semana

Reserve de 8 a 12 horas semanais. Ajuste o prazo à sua rotina; não pule os fundamentos para “chegar logo” ao framework.

02 / Prática

Escreva código desde o início

Para cada conceito, faça algo pequeno sem copiar passo a passo. Use IA para pedir explicações ou testes, mas confira a documentação e entenda o código antes de aceitá-lo.

03 / Foco

Termine antes de trocar de stack

JavaScript, TypeScript, Node.js, NestJS e Next.js já formam uma trilha consistente. Aprenda outras ferramentas quando um projeto pedir.

IA NA TRILHA

Use IA para acelerar, não para pular etapas

A base continua sendo programação. Use assistentes para aprender, investigar problemas e revisar seu trabalho, mantendo você no controle das decisões.

01 / Meses 3 e 4

Tire dúvidas e explore ideias

Peça explicações em outros termos, exemplos pequenos ou caminhos alternativos. Compare as respostas com a documentação e tente resolver por conta própria.

02 / Meses 5 e 6

Investigue erros e crie testes

Compartilhe mensagens de erro e trechos relevantes para levantar hipóteses ou sugerir casos de teste. Reproduza o problema e confirme a solução no seu código.

03 / Meses 9 e 10

Revise sem terceirizar decisões

Use IA para revisar trechos, sugerir refatorações ou melhorar a documentação. Avalie cada sugestão, rode os testes e não envie código que você não entende.

A TRILHA

Do básico ao primeiro produto

Os meses são uma referência, não um prazo de aprovação. Avance quando conseguir explicar e aplicar o que estudou.

01 Meses 1 e 2 Etapa 01

Aprenda a pensar como programador

Comece pela lógica e pelos fundamentos da web. Entenda o que acontece entre o navegador, o servidor e o código que você escreve.

- + Lógica: variáveis, condições, repetições e funções
- + HTML semântico, CSS responsivo e JavaScript no navegador
- + Git e GitHub: commits, branches e README

Entregável: Faça uma página pessoal responsiva e uma lista de tarefas com JavaScript.

02 Meses 3 e 4 Etapa 02

Ganhe fluência em JavaScript e TypeScript

Pratique a linguagem antes de depender de frameworks. Aprenda a organizar o código, lidar com dados e investigar erros com método.

- + Arrays, objetos, módulos, escopo e tratamento de erros
- + Promises, async/await, fetch e consumo de APIs
- + TypeScript: tipos, interfaces, unions e narrowing
- + Use IA como assistente de código para tirar dúvidas e explorar exemplos; revise sugestões, execute os testes e explique cada mudança

Entregável: Crie uma aplicação que consome uma API pública e salva preferências no navegador.

Node.js no backend com NestJS

Node.js é o runtime do backend. Use NestJS para construir APIs com módulos, injeção de dependência e convenções integradas.

- + [Node.js](#): npm, módulos, variáveis de ambiente e event loop
- + [NestJS](#): módulos, controllers, services, injeção de dependência e middlewares
- + HTTP, APIs REST, validação, autenticação e autorização
- + Banco relacional e SQL; modelagem, consultas e migrações. Recomendação: use MySQL ou PostgreSQL
- + Use um assistente IA para investigar erros e sugerir testes; valide cada hipótese no código e na documentação
- + Testes de unidade e integração para regras importantes

Entregável: Publique uma API de catálogo ou tarefas com NestJS, cadastro, login e documentação dos endpoints.

Crie interfaces completas com Next.js

Use React e Next.js para construir a experiência do produto, conectando páginas e componentes aos serviços do seu backend.

- + React: componentes, props, estado, formulários e composição
- + [Next.js](#): App Router, layouts, navegação e renderização
- + Carregamento de dados, estados de erro e acessibilidade
- + Integração com a API NestJS e gerenciamento de sessão

Entregável: Transforme sua API em um produto utilizável, com busca, formulários e área autenticada.

Entregue um projeto full-stack de ponta a ponta

Aproxime seu trabalho do dia a dia de uma equipe: planeje, implemente, teste, revise e coloque uma aplicação no ar.

- + Estrutura de projeto, configuração e segredos fora do código
- + Segurança básica, logs, tratamento consistente de falhas
- + Testes do fluxo principal e revisão de pull requests
- + Use IA como apoio em revisão e documentação, conferindo mudanças, testes e informações antes de publicar
- + Deploy do frontend Next.js, API Node.js com NestJS e banco de dados.
Recomendação: use [Vercel](#) para o frontend e [Railway](#) para o backend e banco de dados.

Entregável: Entregue um app com um problema bem definido, dados persistentes e URL pública, com decisões e código que você consiga explicar.

Pratique a colaboração e consolide seu portfólio

Aproxime seu trabalho do fluxo de uma equipe e organize evidências do que sabe fazer. Aprenda com revisões e feedback para continuar evoluindo.

- + Portfólio com dois ou três projetos finalizados e diferentes
- + README com objetivo, tecnologias, instalação e decisões
- + Issues, branches, pull requests e revisão de código
- + Contribua com um projeto existente e aplique o feedback recebido

Entregável: Peça uma revisão de código e melhore um projeto do portfólio com base no feedback.

Atualize seu LinkedIn e prepare-se para entrevistas

Apresente suas experiências com clareza e pratique entrevistas comportamentais e técnicas com base nos projetos que você construiu.

- + Atualize o LinkedIn: título, resumo, competências e links para projetos
- + Prepare um currículo objetivo e acompanhe suas candidaturas
- + Pratique entrevistas técnicas: resolva exercícios e explique decisões, testes e depuração em voz alta
- + Simule entrevistas comportamentais com exemplos concretos dos seus projetos e peça feedback

Entregável: Atualize seu LinkedIn e currículo, faça uma entrevista técnica simulada e ajuste sua preparação com o feedback.

UMA SEMANA POSSÍVEL

Aprenda, pratique, revise

2 dias Estudo focado de conceitos e documentação

2 dias Implementação do projeto, sem tutorial aberto

1 dia Revisão, testes, checagem das sugestões do assistente IA e próximo passo

PRONTO PARA O PRÓXIMO PASSO?

Seu portfólio deve mostrar o que você resolve.

Uma aplicação publicada, bem explicada e feita por você vale mais que uma lista longa de tecnologias. Comece com o primeiro projeto e evolua a cada etapa.

Mentoria: **acompanhamento com dev de 10+ anos de experiência**, do plano de estudos à primeira vaga. <https://culturadev.com.br/mentoria>