

1. Como Funciona a Internet

Conceitos fundamentais:

O que é a internet e como ela funciona

Funcionamento de navegadores (browsers)

Protocolo HTTP/HTTPS

Sistema de Nomes de Domínio (DNS)

Hospedagem de sites (Hosting)

Dica: Pesquise artigos e vídeos explicativos para ter uma visão visual e prática desses conceitos.

2. Lógica de Programação

Antes de aprender qualquer linguagem, é fundamental entender a lógica de programação para desenvolver o raciocínio lógico na resolução de problemas.

Recomendação: Estude com **Portugol**, uma linguagem estruturada em português, ideal para iniciantes.

Tópicos essenciais:

Variáveis, entrada e saída de dados

Operadores:

Aritméticos

Relacionais

Lógicos

Estruturas:

Condicionais (`if`, `else`)

Repetição (`while`, `for`)

Funções

Vetores (arrays)

Matrizes

3. JavaScript

Fundamentos:

Sintaxe e declaração de variáveis (`var`, `let`, `const`)

Entrada e saída (ex: `console.log`, `prompt-sync`)

Operadores: matemáticos, comparação, lógicos

Estruturas de decisão (`if`, `switch`)

Estruturas de repetição (`for`, `while`, `forEach`)

Funções e escopos

Objetos e arrays

Métodos de arrays (`map`, `filter`, `reduce`, etc.)

Manipulação de datas com `Date`

Promises e assincronismo (`then`, `catch`, `async/await`)

Gerenciadores de Pacotes:

`npm` – [npmjs.com](https://www.npmjs.com)

`yarn` – yarnpkg.com

4. Git e GitHub

Comandos essenciais:

Inicializar repositório: `git init`

Verificar status: `git status`

Adicionar arquivos: `git add`

Comitar mudanças: `git commit`

Ver histórico: `git log`

Ver diferenças: `git diff`

Renomear, mover, remover arquivos: `mv`, `rm`

Restaurar arquivos: `git restore`

Alterar último commit: `git commit --amend`

Navegar para commits antigos: `git checkout <hash>`

Limpar arquivos não rastreados: `git clean`

Reverter commit: `git revert`

Ignorar arquivos: `.gitignore`

Trabalhar com branches: `git branch`, `git checkout -b`

5. SQL (Banco de Dados Relacional)

Operações básicas:

Criar e deletar banco de dados

Tipos de dados

Tabelas:

Criar e manipular tabelas e colunas

Inserir, atualizar e deletar dados

Consultas:

Buscar dados com **SELECT**

Filtros com **WHERE**, operadores lógicos, relacionais e matemáticos

Relacionamentos:

Chaves primárias e estrangeiras

JOINs entre tabelas

Extras:

Constraints (restrições de integridade)

Integração com **Docker**

Interface visual com **pgAdmin**

6. Criação de APIs com Node.js

Configuração e desenvolvimento:

Criar servidor com Node.js

Utilizar **Express.js** para rotas e middlewares

Entender **body**, **header**, **params** e **query**

Ativar CORS

Receber JSON no corpo da requisição

Middlewares personalizados

Ferramentas de desenvolvimento: **nodemon**, **multer**

Testar APIs:

Insomnia

Postman

Swagger – Documentação interativa para APIs REST

ORMs e query builders:

pg – Queries com SQL puro

Sequelize

Mongoose (MongoDB)

Prisma

TypeORM

7. Autenticação

JWT (JSON Web Tokens) – Autenticação baseada em tokens

Passport.js – Estratégia de autenticação para Node.js

Login social – Google, Apple, Facebook, etc.

Serviços externos:

Amazon Cognito

Auth0

8. Cloud Básico e Deploy

Heroku – Deploy rápido para aplicações Node.js

AWS EC2 – Instância de máquina virtual para rodar aplicações

AWS S3 – Hospedagem de arquivos e sites estáticos

PM2 – Gerenciador de processos para apps Node.js

Nginx – Servidor proxy reverso e balanceador de carga

9. Outras Habilidades Recomendadas

Cron – Agendamento de tarefas automatizadas

Redis – Armazenamento em cache e filas

Sentry – Monitoramento e rastreamento de erros

GitHub Actions – Automação de fluxos de trabalho com CI/CD

10. Tópicos Avançados Recomendados

Para seguir se desenvolvendo e se destacar como desenvolvedor:

TypeScript – Superset do JavaScript com tipagem estática

Estrutura de Dados e Algoritmos – Essencial para entrevistas técnicas

ROADMAP PARA PROGRAMADORES BACK END

Programação Orientada a Objetos (POO)

Nest.js – Framework escalável baseado em Node.js + TypeScript

CI/CD – Integração e entrega contínua (CircleCI, BuddyWorks)

Testes:

Jest

Mocha

Supertest

GraphQL – Alternativa ao REST com Apollo Server/Client

Socket.io – Comunicação em tempo real via WebSockets

RabbitMQ – Mensageria assíncrona

Terraform – Infraestrutura como código (IaC)

ElasticSearch – Motor de busca poderoso para grandes volumes de dados

