**Cronograma de Estudos para se Tornar um Programador Mês 1: Fundamentos de Programação**

**Semana 1: Sintaxe Básica**

● **Conteúdo**: Introdução à sintaxe da linguagem (Java, PHP, JavaScript, Python ou outra).

○ Estrutura básica de um programa, comentários, funções simples. ● **Prática**: Pequenos scripts como calcular soma, subtração, etc.

**Semana 2: Tipos, Variáveis e Constantes**

● **Conteúdo**: Tipos de dados, variáveis e constantes.

○ Diferentes tipos de dados (números, strings, booleanos). ○ Escopo de variáveis (global e local).

● **Prática**: Criar scripts que manipulam variáveis.

**Semana 3: Expressões e Operadores**

● **Conteúdo**: Operadores aritméticos, lógicos e relacionais. ○ Expressões matemáticas e lógicas.

● **Prática**: Desenvolver pequenos problemas matemáticos e usar operadores de comparação.

**Semana 4: Estruturas de Controle**

● **Conteúdo**: Controle de fluxo com if, else, switch e laços de repetição (for, while).

● **Prática**: Implementar um menu interativo com opções baseadas em switch-case e loops para repetição de tarefas.

**Mês 2: Programação Orientada a Objetos (POO) Semana 5-6: Conceitos de POO**

● **Conteúdo**: Conceitos de classes, objetos, herança, polimorfismo e encapsulamento.

○ Linguagem: Java PHP Python TypeScript ou outra.

● **Prática**: Desenvolver pequenos projetos POO, como um sistema de gerenciamento de biblioteca ou controle de estoque.

**Semana 7: Controle de Versionamento**

● **Conteúdo**: Introdução ao Git e GitHub.

○ Controle de versão colaborativo.

● **Prática**: Criar repositórios no GitHub e versionar os projetos desenvolvidos.

**Semana 8: Revisão e Projeto**

● **Projeto**: Desenvolver uma aplicação que usa POO e versioná-la com Git.

○ **Exemplo**: Criar um gerenciador de tarefas com funcionalidades de adicionar, remover e listar tarefas.

**Mês 3: Desenvolvimento Web – Frontend**

**Semana 9-10: HTML, CSS e JavaScript**

● **Conteúdo**: Estrutura básica de páginas web com HTML e CSS. ○ Introdução ao JavaScript para interatividade.

○ Design responsivo. (Neste momento não precisa ir muito a fundo)

● **Prática**: Desenvolver uma página web simples e interativa (ex.: formulário ou galeria de imagens).

**Semana 11-12: Frameworks de Frontend**

● **Conteúdo**: Introdução a frameworks como React ou Vue.js. ○ Componentes, estado, e props.

● **Prática**: Desenvolver uma aplicação de lista de tarefas (To-Do App) usando um framework frontend.

**Mês 4: Desenvolvimento Web – Backend**

**Semana 13-14: Introdução ao Backend e APIs**

● **Conteúdo**: HTTP, RESTful APIs, introdução ao desenvolvimento backend com Spring, Laravel, Node.js (ou Flask/Django).

● **Prática**: Criar uma API simples para gerenciar usuários ou controle de estoque (ex.: CRUD de usuários).

**Semana 15-16: Bancos de Dados**

● **Conteúdo**: Bancos de dados relacionais (SQL).

○ Conceitos de CRUD (Create, Read, Update, Delete).

○ Relacionamentos 1:1; 1:n e n:n

● **Prática**: Desenvolver um sistema completo com Spring, Laravel ou Node.js e um banco de dados (ex.: MySQL).

**Mês 6: Boas Práticas e Projeto Final**

**Semana 21-22: Testes e Qualidade de Código**

● **Conteúdo**: Testes unitários, TDD (Test Driven Development), refatoração de código.

○ Ferramentas: Jest (JavaScript), JUnit (Java).

● **Prática**: Escrever testes para os projetos desenvolvidos nas semanas anteriores.

**Semana 23-24: Projeto Final**

● **Projeto**: Desenvolver uma aplicação completa com Frontend e Backend integrados.

○ **Exemplo de Projeto**: Sistema de gerenciamento de usuários com autenticação, CRUD, banco de dados e design responsivo.

**Opcional: Estruturas de Dados e Algoritmos**

**Estruturas de Dados Avançadas**

● **Conteúdo**: Árvores, grafos, tabelas hash.

**Algoritmos de Ordenação e Busca**

● **Conteúdo**: Algoritmos de ordenação como QuickSort e MergeSort. ○ Busca binária.

● **Prática**: Resolver problemas de ordenação e busca em diferentes cenários.