

Trilha de aprendizado

1. Fundamentos de Java (Revisado)

Vamos manter o foco em temas que realmente ajudam você a entender a base da linguagem e seus conceitos fundamentais antes de avançar para tópicos mais complexos.

- **Configuração do ambiente de desenvolvimento:** Instalar o JDK e configurar um editor de código (IntelliJ IDEA ou Eclipse).
 - **Sintaxe básica:** Tipos de dados, operadores, controle de fluxo (if, else, loops).
 - **Arrays:** Como declarar, inicializar e usar arrays.
 - **Entrada de dados:** Usando a classe **Scanner** para capturar entradas do usuário.
 - Lista de Exercícios para praticar lógica ao fim do módulo
-

2. Programação Orientada a Objetos (POO)

Este módulo é essencial para aprender os conceitos de **POO** e é fundamental para avançar no uso de Java.

- **Conceitos básicos de OOP:** O que é orientação a objetos e como ela funciona (classes, objetos, métodos e atributos).
 - **Herança:** Como classes herdam características de outras classes.
 - **Polimorfismo:** Como os objetos podem ter diferentes formas de se comportar dependendo do contexto.
 - **Encapsulamento:** Como proteger os dados e definir o acesso a eles (usando **private**, **public**, etc).
 - **Abstração:** Como esconder a complexidade interna e expor apenas o necessário.
 - **Interfaces:** Como usar interfaces para definir contratos e permitir a flexibilidade de implementação.
 - **Injeção de Dependência (DI):** Essencial para o Spring, como objetos podem ser passados para outros objetos sem precisar instanciá-los diretamente.
-

3. Avançando para Tópicos Intermediários

Agora, com a base sólida de OOP, podemos avançar para tópicos mais intermediários, como coleções, exceções e recursos do Java 8.

- **Coleções (Collections):** Listas, sets, mapas e como usá-los (após entender OOP e como trabalhar com objetos).

- **Exceções (Exceptions):** Trabalhar com erros e exceções, como capturá-los com `try-catch` e usar exceções personalizadas.
 - **Java 8+ Features:** Lambda, Streams, e Optional, para trabalhar de forma mais eficiente com coleções e manipulação de dados.
-

4. Trabalhando com Banco de Dados e JDBC

Agora podemos integrar Java com bancos de dados.

- **Conceitos básicos de SQL:** Como criar, ler, atualizar e deletar dados.
 - **JDBC:** Como conectar Java a um banco de dados.
 - **Hibernate (ou JPA):** Framework ORM para mapeamento objeto-relacional.
-

5. Spring Framework

Aqui, o foco é trabalhar com o Spring para desenvolver aplicações modernas.

- **Spring Core:** Injeção de dependência, gerenciamento de beans, IoC (Inversion of Control).
 - **Spring MVC:** Para desenvolvimento de aplicações web.
 - **Spring Boot:** Simplificação de configuração e desenvolvimento de APIs.
 - **Spring Security:** Implementação de autenticação e autorização.
-

6. Projetos Práticos

Começar a aplicar tudo o que aprendeu em projetos reais.

- **API RESTful com Spring Boot:** Crie uma API simples de CRUD.
 - **Aplicação web com Spring MVC:** Sistema de login básico ou blog.
 - **Integração com Banco de Dados:** Utilizando JPA e Spring Data.
 - **Microserviços com Spring Boot:** Para desenvolvimento de soluções distribuídas.
-

7. Avançando para Microserviços e Arquitetura

Por fim, tópicos mais avançados para construir sistemas robustos.

- **Spring Cloud:** Para construção de aplicações distribuídas.
- **Spring WebFlux:** Para programação reativa no Spring.

- **Spring Boot + Docker + Kubernetes:** Para empacotamento e orquestração de microserviços.
-