

Tempo estimado de leitura:  
5 min

## Módulo #4

Guias de estudo

### Introdução

Nesse módulo estudamos sobre como podemos organizar nossos projetos Java em camadas que facilitam a manutenibilidade deles.

Além disso, vimos sobre alguns padrões de projetos que podem ser usados para evitar repetições de código.

### Camadas

A forma mais básica de organizar um projeto Java é dividir os arquivos em diretórios que especificam a função de cada arquivo dentro do projeto.

No contexto de desenvolvimento de APIs com o Spring, podemos começar dividindo projeto em 3 camadas:

**services, models e controllers:**



Controllers

Dentro da camada **Controllers** ficam todas as classes que implementam os endpoints da API.

Models

Na camada **models**, nós colocamos os modelos da aplicação. Os modelos são classes que especificam as entidades de domínio da API.

Services

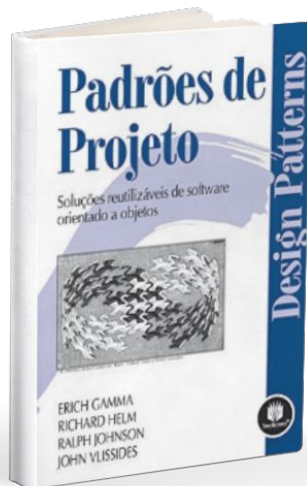
A camada **services** fica responsável por armazenar as classes de serviços, onde as lógicas de negócios são implementadas e executadas.

```
alunos-api ~/dev/java/mentorama/alunos-api
├── .idea
├── .mvn
├── src
│   ├── main
│   │   ├── java
│   │   │   ├── br.com.mentorama.alunosapi
│   │   │   │   ├── controllers
│   │   │   │   │   ├── AlunosController
│   │   │   │   ├── exceptions
│   │   │   │   │   ├── NotFoundException
│   │   │   │   ├── models
│   │   │   │   │   ├── Aluno
│   │   │   │   ├── services
│   │   │   │   │   ├── AlunosService
│   │   │   │   └── AlunosApiApplication
│   │   └── resources
│   └── test
├── target
├── .gitignore
├── HELP.md
├── mvnw
├── mvnw.cmd
└── pom.xml
```

## Padrões de projetos

Padrões de projetos são soluções conhecidas e documentadas para projetos de software. São vitais para se construir sistemas escalonáveis e robustos, evitar repetição de código e permitir que o programador escreva código de forma elegante.

Os padrões de projetos são bem documentados no livro [Padrões de Projetos: Soluções Reutilizáveis de Software Orientados a Objetos](#).



Além desse livro, recomendamos o site **Refactoring Guru** (<https://refactoring.guru/pt-br/design-patterns>) onde também são ensinados padrões de projetos bem conhecidos.