

# Spring Boot Essencial

Primeiros Passos  
com Annotations



Marcos Dev



# Índice

Capítulo	Página
0 - O que são Annotations -----	<a href="#">03</a>
1 - Ponto de partida da aplicação -----	<a href="#">06</a>
2 - Gerenciando classes -----	<a href="#">08</a>
3 - Crie APIs com poucas linhas -----	<a href="#">10</a>
4 - Injeção de dependência automática -----	<a href="#">13</a>
5 - Mapeando classes para o banco de dados ---	<a href="#">15</a>
6 - Comunicação com o Banco -----	<a href="#">17</a>
7 - Escrevendo testes com Junit -----	<a href="#">19</a>
8 - Agora as Annotations Estão Nas Suas Mãos -	<a href="#">21</a>
9 - Agradecimentos -----	<a href="#">25</a>

00

0 Que São  
Annotations?

Annotations?  
0 Que São



# O Que São Annotations?

O Spring Boot é uma das ferramentas mais populares para criação de APIs e microserviços em Java. Um dos motivos para sua popularidade são as annotations, que facilitam muito o desenvolvimento.

Neste Ebook, você vai conhecer as principais annotations do Spring Boot com exemplos reais e práticos.

Mas Então, o que são as Annotations?

São comandos colocados diretamente no código (acima de classes, métodos ou atributos) que informam ao Spring como aquele elemento deve se comportar. Por exemplo

```
Spring Boot Essencial - Mar Dev

@RestController
public class ProdutoController {
    // Spring entende que aqui teremos endpoints HTTP
}
```

# Vantagens

## **Menos Código, Mais Clareza**

Você não precisa escrever código repetido. O Spring entende automaticamente o que fazer a partir das annotations.

## **Configuração Automática e Inteligente**

Spring Boot usa annotations para configurar o comportamento da aplicação sem você precisar fazer isso manualmente.

## **Modularidade e Organização**

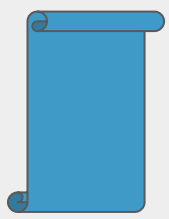
Cada anotação tem uma função clara: controller, serviço, entidade, configuração... Isso ajuda a manter o projeto organizado e de fácil manutenção.

# 01

## Ponto de Partida da Aplicação

da Aplicação

Ponto de Partida



# @SpringBootApplication – Ponto de Partida da Aplicação

Essa annotation marca a classe principal da sua aplicação.

Ela combina outras três: @Configuration, @EnableAutoConfiguration e @ComponentScan.

```
Spring Boot Essencial - Mar Dev

@SpringBootApplication
public class MeuApp {
    public static void main(String[] args) {
        SpringApplication.run(MeuApp.class, args);
    }
}
```

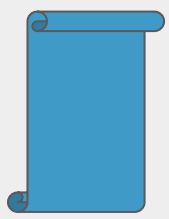
# 02

## Gerenciando Classes

Classes

Gerenciando





# @Component, @Service e @Repository – Gerenciando Classes

Essas annotations dizem ao Spring: "Gerencie essa classe como um bean" (um componente que pode ser injetado em outros lugares).

@Component → usada de forma genérica

@Service → usada para classes de regra de negócio (boa prática)

```
Spring Boot Essencial - Mar Dev

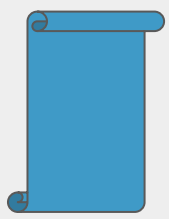
@Service
public class ProdutoService {

    public Produto salvar(Produto produto) {
        // lógica de negócio
        return produto;
    }
}
```

# 03

**Crie APIs com  
poucas linhas**

bomcas Jmpas  
Crie APIs com



## @RestController: Crie APIs com poucas linhas

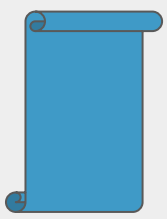
Indica que a classe é um controlador REST, ou seja, responde requisições HTTP com dados (geralmente JSON), e não com páginas HTML

Na prática: é usada em quase toda API RESTful com Spring Boot.

```
Spring Boot Essencial - Mar Dev

@RestController
@RequestMapping("/produtos")
public class ProdutoController {

    @GetMapping
    public List<String> listar() {
        return List.of("Notebook", "Mouse");
    }
}
```



# Crie APIs com poucas linhas

Essas annotations definem os métodos HTTP que a API vai responder.

@GetMapping → requisições GET

@PostMapping → requisições POST

@PutMapping → atualizações (PUT)

@DeleteMapping → exclusões (DELETE)

.

```
Spring Boot Essencial - Mar Dev

@RestController
@RequestMapping("/produtos")
public class ProdutoController {

    @GetMapping
    public List<String> listar() {
        return List.of("Notebook", "Mouse", "Teclado");
    }

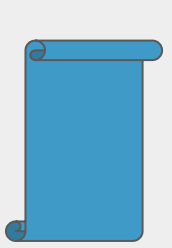
    @PostMapping
    public Produto salvar(@RequestBody Produto produto) {
        return produtoService.salvar(produto);
    }
}
```



# 04

## Injeção de Dependência Automática

Automática  
dependência  
injeção de



# @Autowired: Injeção de dependência automática

No Spring, você não precisa criar objetos "na mão". O próprio framework cuida disso para você através da injeção de dependência.

A annotation @Autowired diz ao Spring: "injeite aqui a instância correta desta classe".

O Spring vê o @Autowired, encontra uma instância de ProdutoService e injeta no ProdutoController. Você não precisa se preocupar com `new ProdutoService()`.



Spring Boot Essencial - Mar Dev

```
@RestController
public class ProdutoController {

    @Autowired
    private ProdutoService produtoService;
}
```

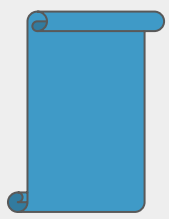
# 05

**Mapeando classes  
para o banco de  
dados**

qqqos

para o banco de

mapeando classes



## @Entity e @Id: Mapeando classes para o banco de dados

Quando usamos Spring Boot com JPA, transformamos classes Java em tabelas do banco de dados. Isso é feito com a annotation @Entity.

Já a annotation @Id marca qual campo será a chave primária dessa tabela.

```
Spring Boot Essencial - Mar Dev

@RestController
public class ProdutoController {

    @Autowired
    private ProdutoService produtoService;
}
```

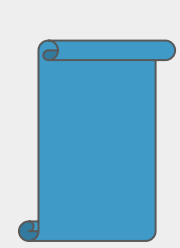


# 06

**Comunicação com o  
banco**

pruço

comunicação com o



# @Repository: Comunicação com o banco

Usada para classes que acessam o banco de dados.

Com JpaRepository, você evita escrever código repetitivo.

Na prática: um CRUD pronto em segundos.

```
Spring Boot Essencial - Mar Dev

@Repository
public interface ProdutoRepository
    extends JpaRepository<Produto, Long> {
}
```

# 07

## Escrevendo testes com JUnit

com JUnit

Escrevendo testes

# @Test: Escrevendo testes com JUnit

Em projetos Spring Boot, usamos JUnit como framework principal de testes.

E a annotation @Test marca os métodos que devem ser executados como testes unitários.

```
Spring Boot Essencial - Mar Dev

@SpringBootTest
class ProdutoServiceTest {

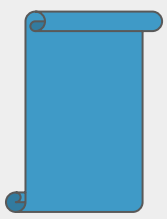
    @Test
    void deveSalvarProduto() {
        Produto produto = new Produto("Fone de PC");
        assertNotNull(produto);
    }
}
```



# 08

**Agora as Annotations  
Estão Nas Suas Mãos**

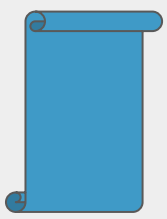
Estão Nas Suas Mãos  
Agora as Annotations



# **Conclusão - Agora as Annotations Estão Nas Suas Mãos**

Você chegou até aqui

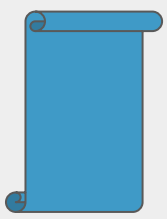
É isso significa que agora você já entende como as annotations funcionam, para que servem e como elas tornam o desenvolvimento com Spring Boot muito mais simples e eficiente.



# Conclusão - Agora as Annotations Estão Nas Suas Mãos

## 0 Que Você Aprendeu

- Que uma annotation é uma forma inteligente de comunicar intenções ao Spring;
- Como usar as principais annotations:  
`@RestController`, `@Service`, `@Autowired`,  
`@Entity`, `@Transactional`, entre outras;
- Como aplicar cada uma em exemplos reais e práticos;
- E como essas pequenas "etiquetas" no código resolvem grandes problemas no dia a dia.



# Conclusão - Agora as Annotations Estão Nas Suas Mãos



## Próximos Passos

Agora é hora de praticar !

Aqui vão algumas ideias simples para colocar o conhecimento em ação:

- Crie uma API de produtos com `@RestController` e `@Service`;
- Use `@Entity` e `@Repository` para salvar dados em um banco H2 ou PostgreSQL;
- Faça testes simples com `@SpringBootTest`.

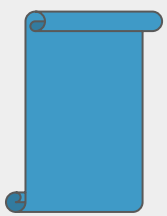
Cada vez que você usa uma annotation com confiança, está um passo mais perto de dominar o ecossistema Spring.



# 09

✨ **Agradecimentos** ✨

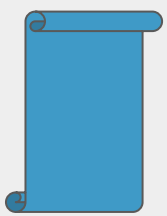
✨ Agradecimentos ✨



# Agradecimentos

Gostaria de expressar minha sincera gratidão a todos que contribuíram para a criação deste eBook, especialmente à incrível comunidade de desenvolvedores que, com seu conhecimento e entusiasmo, tornam o universo da programação mais rico e acessível a todos.

Este eBook, focado nas annotations em Java para iniciantes e com uma temática divertida de jogos, só foi possível graças à troca constante de ideias, dicas e experiências de todos vocês, que sempre buscam aprender, compartilhar e crescer juntos.



# Agradecimentos

Cada comentário, sugestão e feedback contribuíram para que este conteúdo fosse mais do que apenas um guia técnico – tornou-se uma jornada colaborativa e de aprendizado mútuo. Agradeço a cada desenvolvedor, de iniciante a experiente, que faz parte dessa comunidade vibrante e unida.

Nosso caminho na programação é constante, e que possamos continuar a aprender e ensinar uns aos outros. Que este eBook seja mais uma ferramenta para o seu crescimento e para o seu prazer em programar.

Muito obrigado!

# Spring Boot Essencial

Primeiros Passos  
com Annotations



Marcos Dev