

Material de Apoio:

- Documentação do Spring Boot

<https://spring.io/quickstart>

- Baeldung

<https://www.baeldung.com/spring-tutorial>

Ferramentas:

- Spring Tool Suite 4
<https://spring.io/tools>
- Postman
- Postgresql
- DBEaver

Aula 1:

- Conceitos de API e Restful;
- Controle de Dependências com Maven
- Versionamento de Código – Git
- Framework Spring

API:

Application Programming Interface

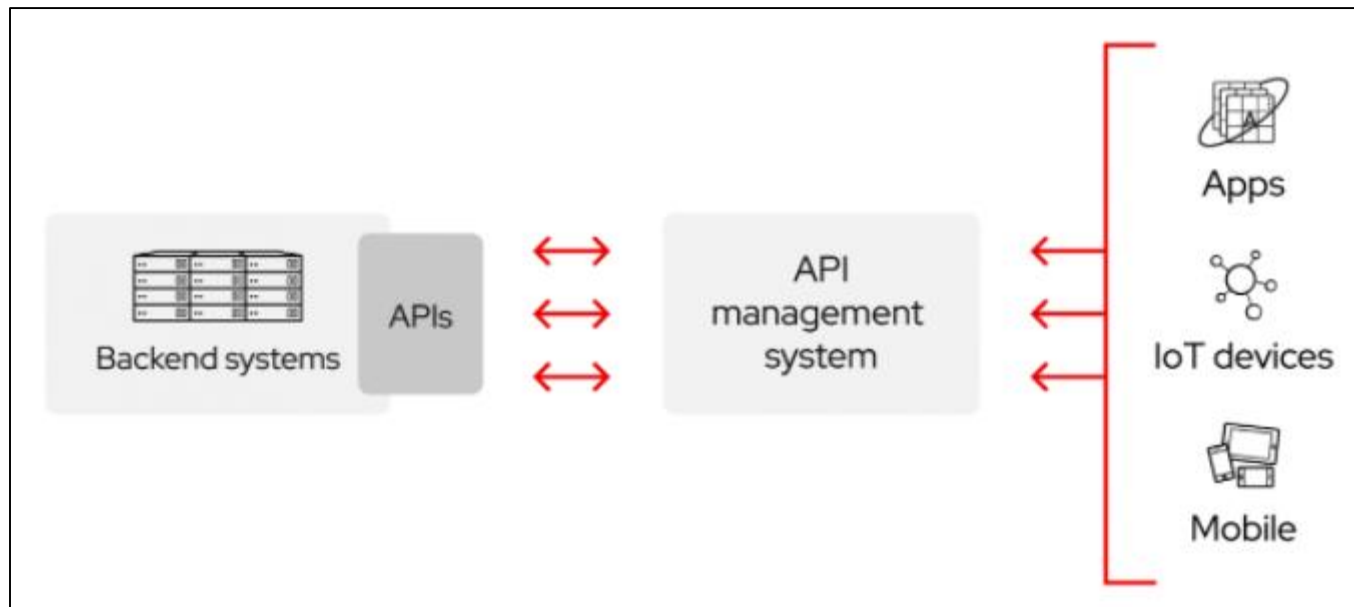
(Interface de Programação de Aplicações)

O que é API?

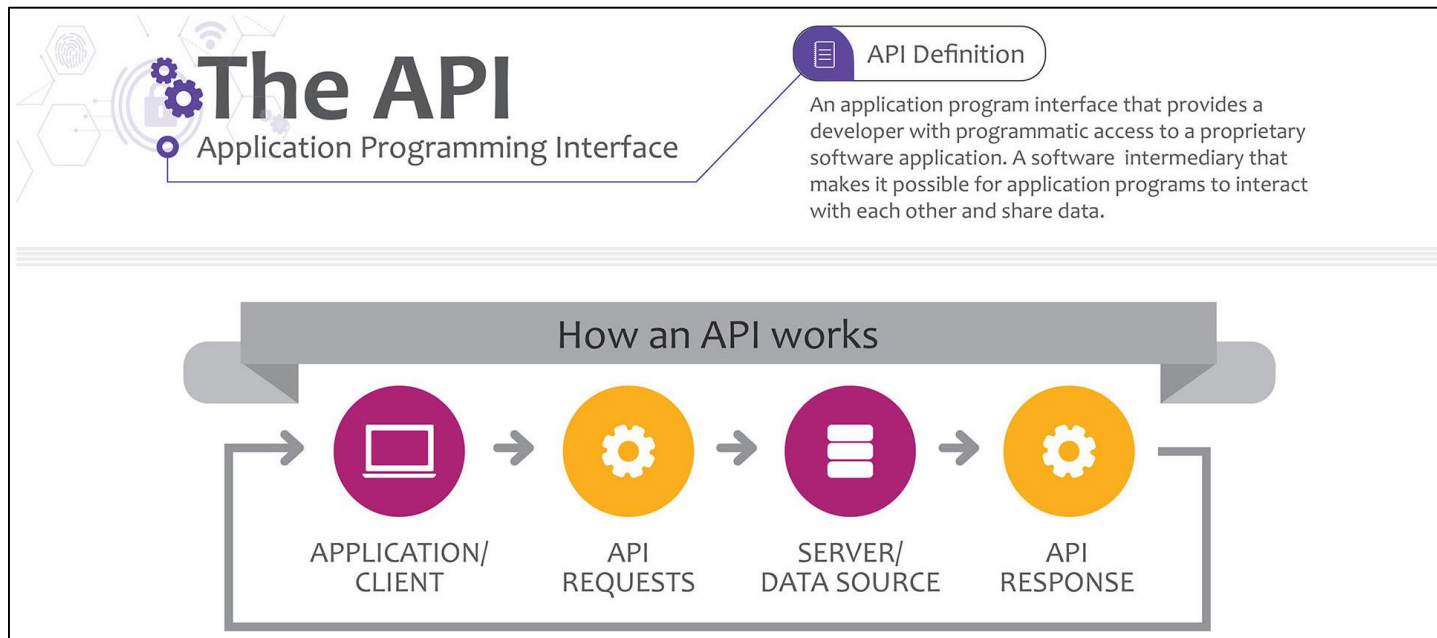
API é um conjunto de definições e protocolos usado no desenvolvimento e na [integração](#) de software de aplicações. API é um acrônimo em inglês que significa interface de programação de aplicações.

Uma API permite que sua solução ou serviço se comunique com outros produtos e serviços sem precisar saber como eles foram implementados. Isso simplifica o desenvolvimento de aplicações, gerando economia de tempo e dinheiro. Ao desenvolver novas ferramentas e soluções (ou ao gerenciar aquelas já existentes), as APIs oferecem a flexibilidade necessária para simplificar o design, a administração e o uso, além de fornecer oportunidades de inovação.

As APIs costumam ser vistas como contratos, com documentações que representam um acordo entre as partes interessadas. Se uma dessas partes enviar uma solicitação remota estruturada de uma forma específica, isso determinará como o software da outra parte responderá.



Fonte: redhat.com



Fonte: JM Baxi Group

REST e RESTful:

Representational State Transfer

(Transferência de Estado Representacional)

REST e RESTful:

Princípios, regras e restrições para a criação de projetos com interfaces bem definidas.

REST:

Princípios, regras e restrições de arquitetura.

RESTful:

Capacidade de aplicar, por um sistema, os princípios REST.

RESTful:

GET <http://domínio.com/alunos>

DELETE <http://domínio.com/alunos/fulano>

POST <http://domínio.com/alunos>
(data {nome: ciclano})

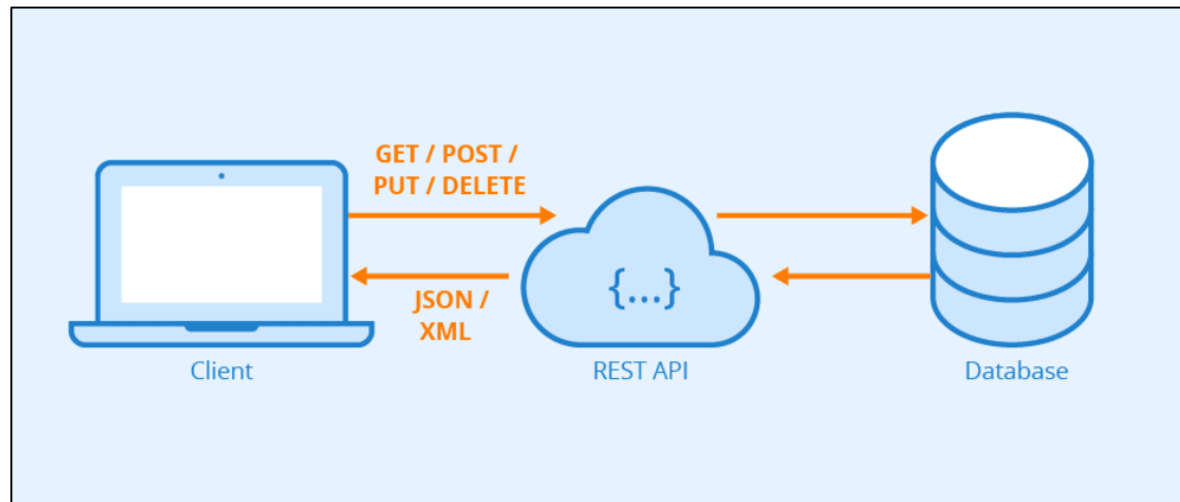
RESTful:

Métodos HTTP:

- GET
- POST
- PUT
- DELETE

Veja mais em: <https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Methods>

RESTFUL:



Fonte: Dev Community

Spring Framework:



Microservices

Quickly deliver production-grade features with independently evolvable microservices.



Reactive

Spring's asynchronous, nonblocking architecture means you can get more from your computing resources.



Cloud

Your code, any cloud—we've got you covered. Connect and scale your services, whatever your platform.



Web apps

Frameworks for fast, secure, and responsive web applications connected to any data store.



Serverless

The ultimate flexibility. Scale up on demand and scale to zero when there's no demand.



Event Driven

Integrate with your enterprise. React to business events. Act on your streaming data in realtime.



Batch

Automated tasks. Offline processing of data at a time to suit you.

(Maven : Controle de Dependências):

- Gerencia a utilização de bibliotecas (.jar) no projeto
- Automatiza a tarefa, substituindo o download e inserção manual de bibliotecas;
- Configuração de dependências a partir do pom.xml

(Maven : Controle de Dependências):

```
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-security</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
  . . .
```


Spring Boot :: Arquitetura do Projeto RESTful

Entities

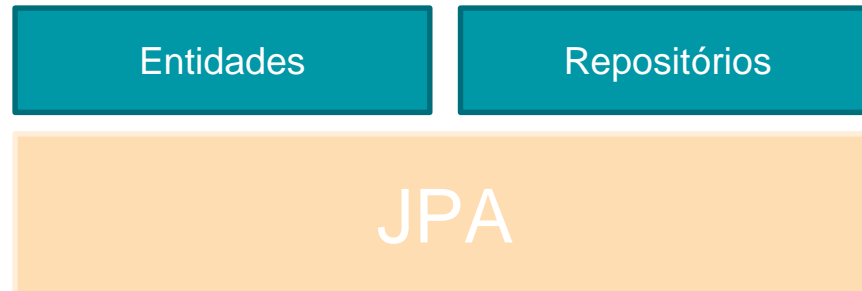
Repositories

Services

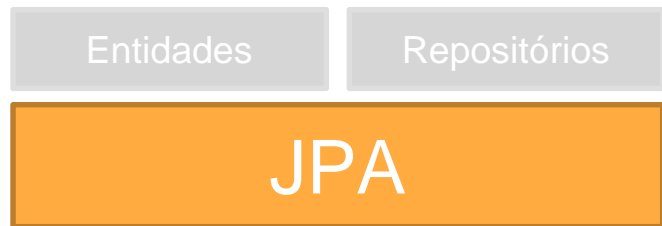
Controllers

Spring Boot

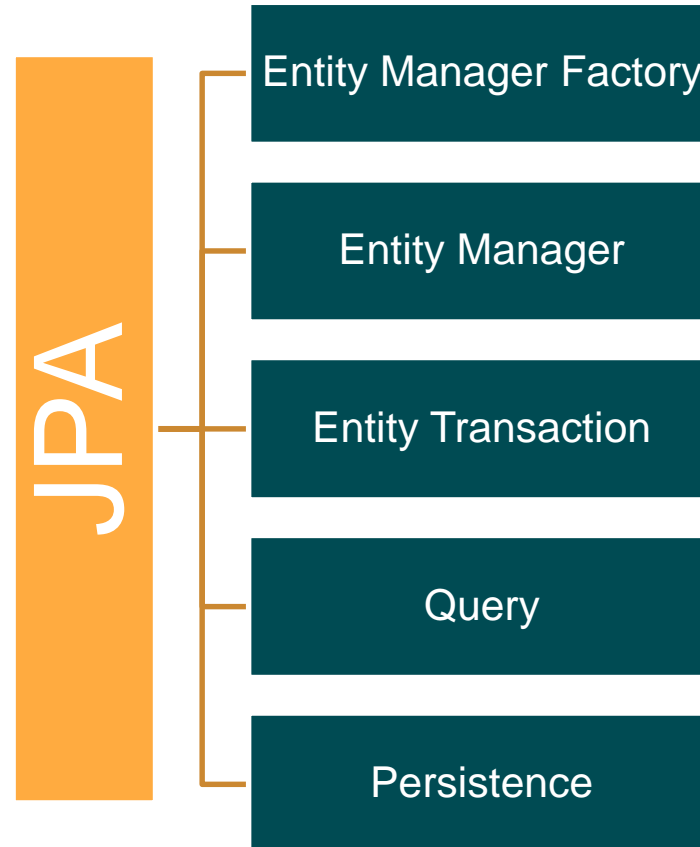
Spring Boot :: Acesso a Dados



Spring Boot :: Acesso a Dados



- Java Persistence API
- Framework baseado em POJOS (objetos Java)
- Mapeamento Objeto-Relacional (ORM)
- Outras funcionalidades

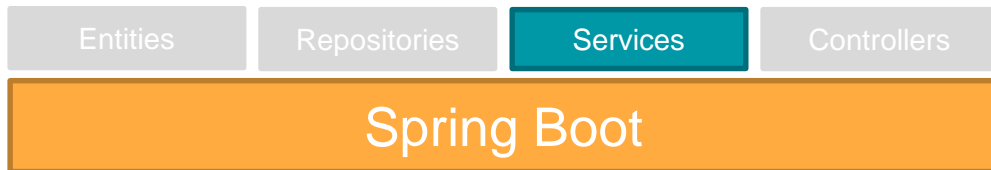


Spring Boot :: Acesso a Dados



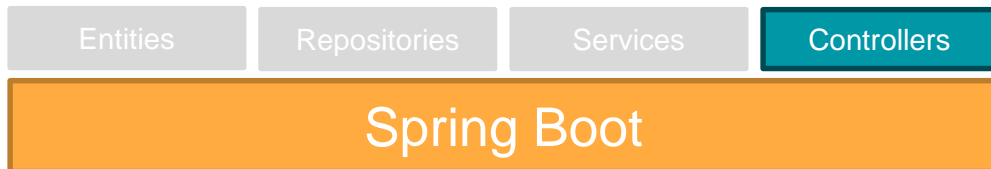
- Entidades: @Entity
- Repositórios: Repository Interface -
JpaRepository | PagingAndSortingRepository | CrudRepository

Spring Boot :: Regras/Lógica de Negócio



- Serviços: @Service
- Fazem a interface entre o endpoint (controller) e o repositório (repository)

Spring Boot :: Controller/Endpoint



- Controller: `@RestController`
- Faz a interface com a aplicação externa
- Contem os endereços (URI) dos recursos disponíveis