

targettrust
treinamento e tecnologia



Riguel Figueiró

Consultor de software na ThoughtWorks
Apaixonado por metodologias ágeis,
integração contínua, qualidade e disseminação
e compartilhamento de conhecimento!

Como falar comigo:

- riguel@dkosoftware.com.br
- www.linkedin.com/in/riguel-figueiro/
- github.com/riguelbf






JAVA DEVELOPER

Imersão Completa



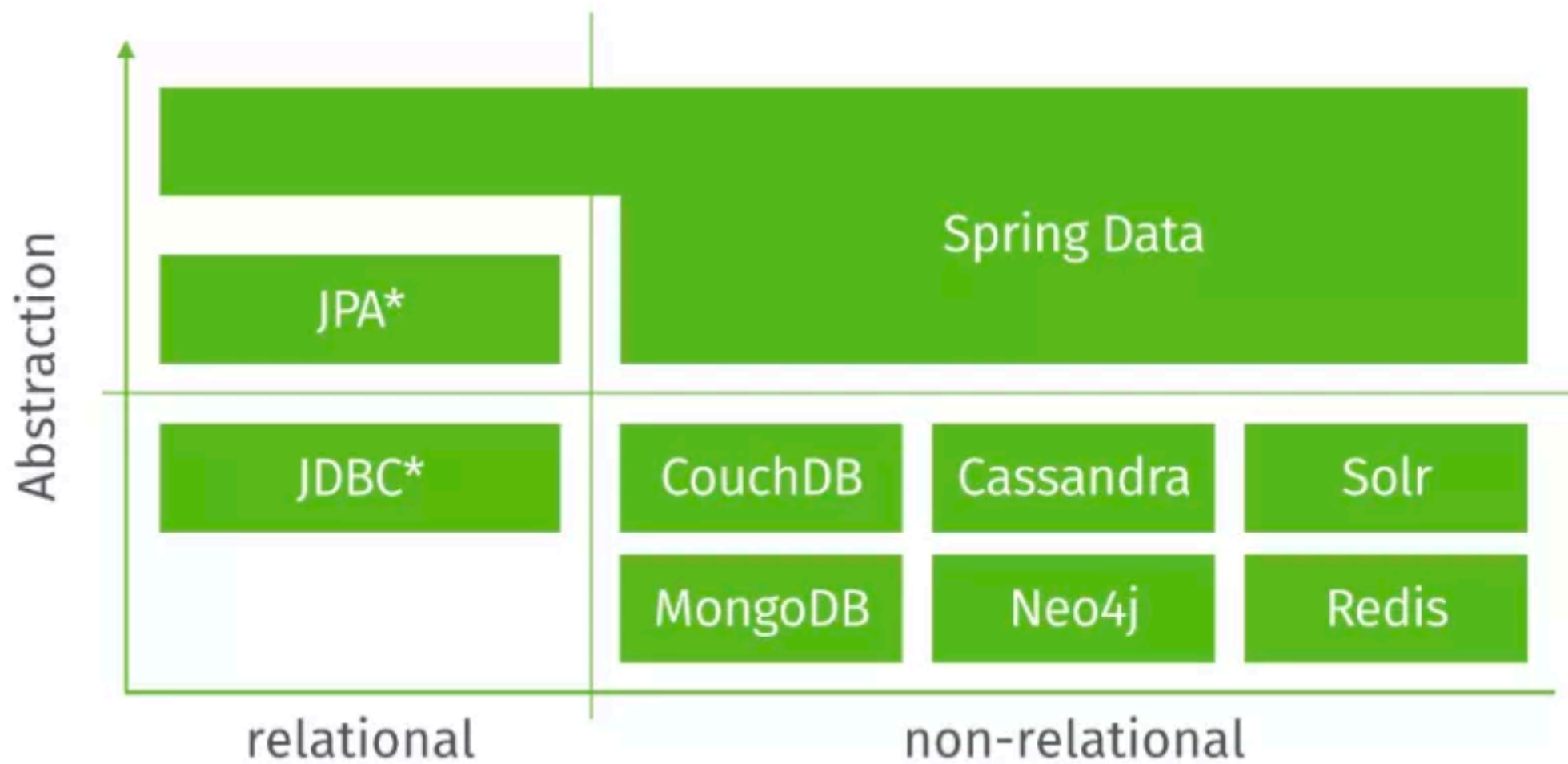


O que é Spring Data



A missão da Spring Data é fornecer um modelo de programação familiar e consistente, baseado em Spring, para o acesso a dados, mantendo as características especiais do armazenamento de dados subjacente.

Spring Data





O que é JPA

Java Persistence API é uma API padrão da linguagem Java que descreve uma interface comum para frameworks de persistência de dados. A JPA define um meio de mapeamento objeto-relacional para objetos Java simples e comuns, denominados beans de entidade.

Conceitos fundamentais

A interface central na abstração do repositório do Spring Data é `Repository`. Leva a classe de domínio para gerenciar, bem como o tipo de ID da classe de domínio como argumentos de tipo. Essa interface age principalmente como uma interface de marcação para capturar os tipos com os quais trabalhar e para ajudá-lo a descobrir as interfaces que a estendem. O `CrudRepository` fornece funcionalidade CRUD sofisticada para a classe de entidade que está sendo gerenciada. Fácil de entender e desenvolver aplicativos com spring framework

CrudRepository Interface

1. Salva a entidade em questão.
2. Retorna a entidade identificada pelo ID fornecido.
3. Retorna todas as entidades.
4. Retorna o número de entidades.
5. Eliminar a entidade especificada.
6. Indica se existe uma entidade com o ID fornecido.

```
public interface CrudRepository<T, ID extends Serializable>
    extends Repository<T, ID> {

    <S extends T> S save(S entity);           ❶

    Optional<T> findById(ID primaryKey);      ❷

    Iterable<T> findAll();                     ❸

    long count();                              ❹

    void delete(T entity);                     ❺

    boolean existsById(ID primaryKey);         ❻

    // ... more functionality omitted.
}
```

Query Methods

Repositórios de funcionalidade CRUD padrão geralmente têm consultas sobre o armazenamento de dados subjacente. Com o Spring Data, declarar essas consultas torna-se um processo de quatro etapas.



1 - Declare uma interface estendendo o Repository ou uma de suas subinterfaces e digite-a na classe de domínio e no tipo de ID que ela deve manipular, conforme mostrado no exemplo a seguir:

```
interface PersonRepository extends Repository<Person, Long> { ... }
```

2 - Declarar métodos de consulta na interface:

```
interface PersonRepository extends Repository<Person, Long> {  
    List<Person> findByLastname(String lastname);  
}
```

3 - Configure o Spring para criar instâncias de proxy para essas interfaces, com o JavaConfig ou com a configuração XML.

Para usar a configuração Java, crie uma classe semelhante à seguinte:

A

```
import org.springframework.data.jpa.repository.config.EnableJpaRepositories;

@EnableJpaRepositories
class Config {}
```

B

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:jpa="http://www.springframework.org/schema/data/jpa"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd
    http://www.springframework.org/schema/data/jpa
    http://www.springframework.org/schema/data/jpa/spring-jpa.xsd">

  <jpa:repositories base-package="com.acme.repositories"/>

</beans>
```

4 -Injete a instância do repositório e use-a, conforme mostrado no exemplo a seguir:

```
class SomeClient {  
  
    private final PersonRepository repository;  
  
    SomeClient(PersonRepository repository) {  
        this.repository = repository;  
    }  
  
    void doSomething() {  
        List<Person> persons = repository.findByLastname("Matthews");  
    }  
}
```





Configurando o projeto para Spring Data







HEROKU

Adicionando o Postgres

 Personal  >  targetdb

 Open app More 

Overview

Resources

Deploy

Metrics

Activity

Access


Settings

Dynos

This app has no process types yet

Add a Procfile to your app in order to define its process types. [Learn more](#)

Add-ons

 Heroku Postgres

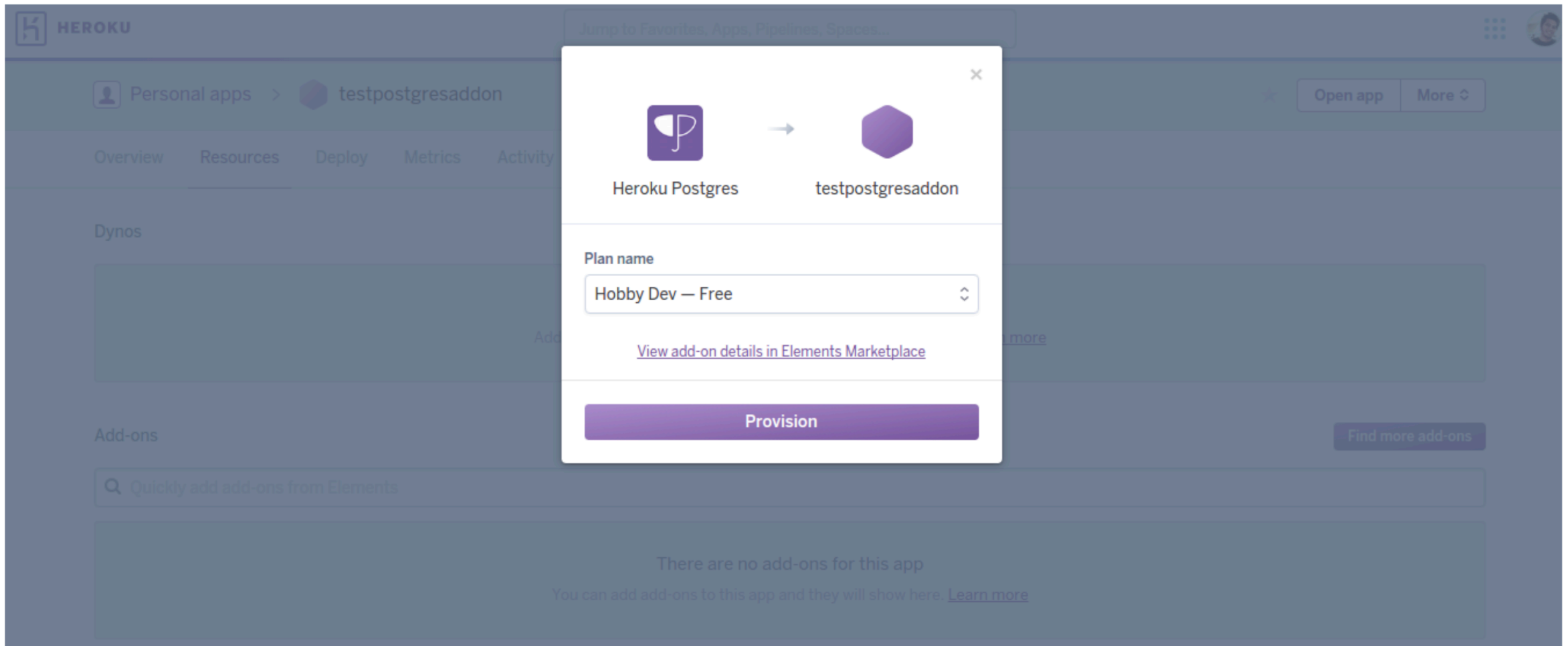
There are no add-ons for this app

You can add add-ons to this app and they will show here. [Learn more](#)

Estimated Monthly Cost

\$0.00

No problem, it is FREE!





Arquivo de configuração de repositório
disponível para download

Adicionando as dependências

```
34
35 // In this section you declare the dependencies for your production and test code
36 dependencies {
37
38     compile("org.springframework.boot:spring-boot-starter-web")
39     compile("org.springframework.boot:spring-boot-starter-data-jpa")
40     compile("org.postgresql:postgresql:9.4-1200-jdbc41")
41 }
42
```

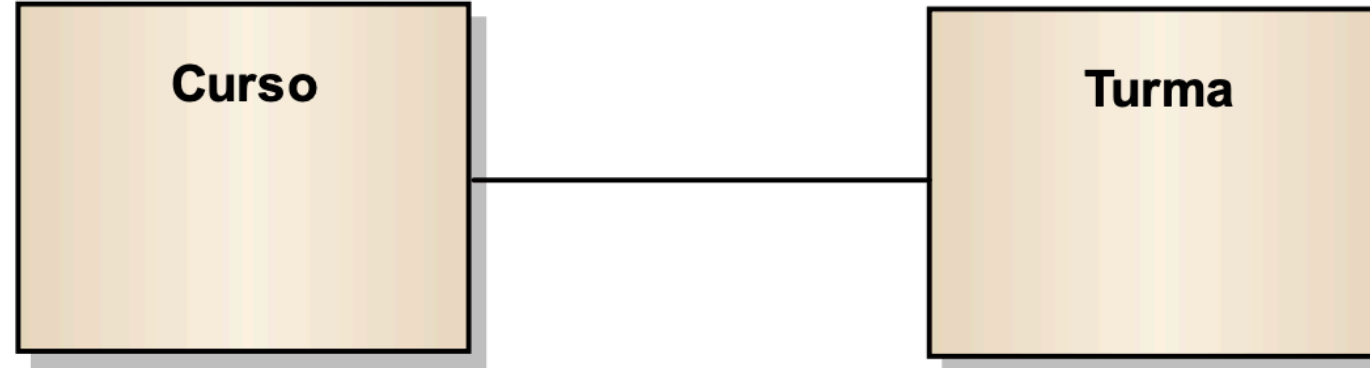
Direcionamento de relacionamentos

Bidirecional: – Cada uma das entidades tem um campo de relacionamento que consulta a outra. Exemplo: Curso sabe que instâncias de Turma ele possui e Turma sabe a qual Curso ela pertence.

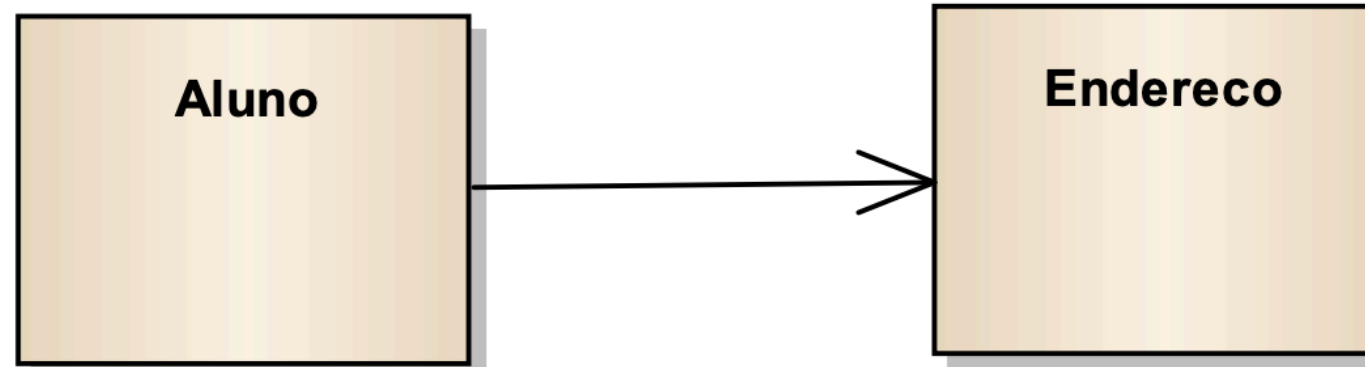
Unidirecional: – Somente uma das entidades tem um campo de relacionamento que consulta a outra. Exemplo: Aluno tem acesso ao seu Endereço, mas Endereço não tem acesso ao Aluno a que pertence.

Exemplo

Bidirecional



Unidirecional



Multiplicidades

Um para Um: – Cada instância de entidade relacionada com uma única outra instância de entidade

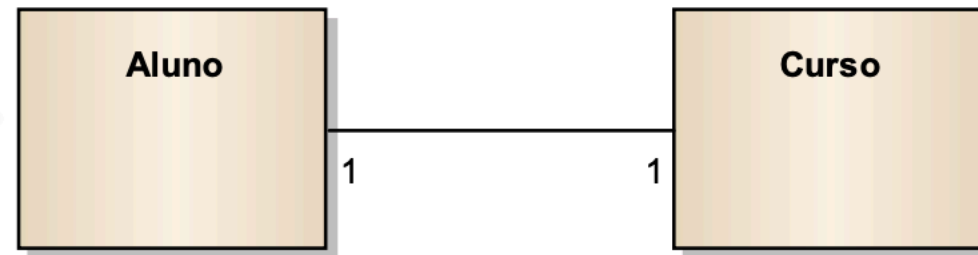
Um para Muitos: – Uma instância de entidade relacionada com outras múltiplas instâncias de entidade

Muitos para Um: – Contrário de um para muitos

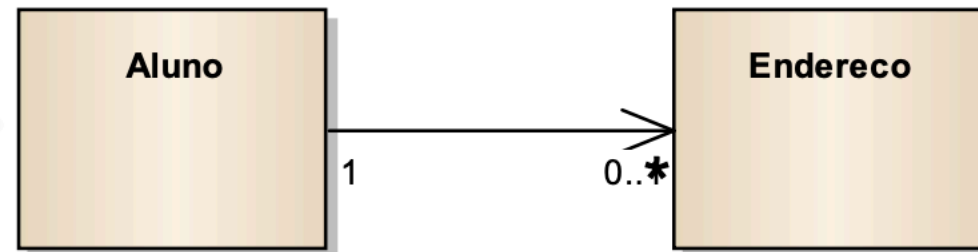
Muitos para Muitos: – Múltiplas instâncias de entidade relacionadas com outras múltiplas instâncias de entidade

Revisão

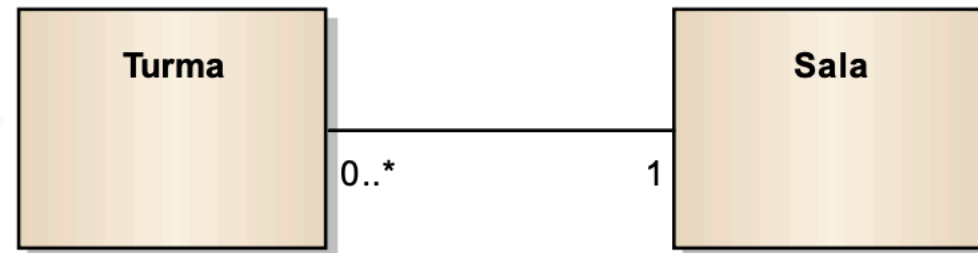
Um para Um
`@OneToOne`



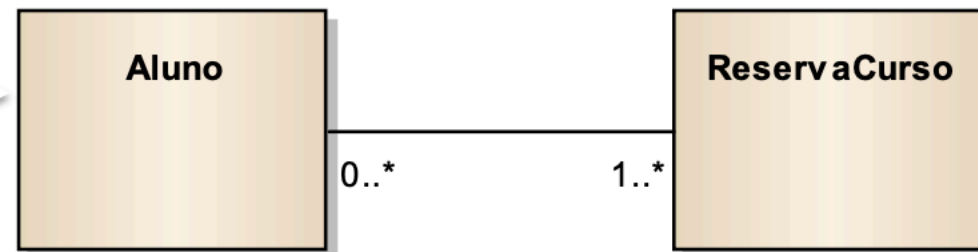
Um para Muitos
`@OneToMany`



Muitos para Um
`@ManyToOne`



Muitos para Muitos
`@ManyToMany`





Pergunta

Façam o máximo de perguntas que desejarem!!



Exercícios

Conteúdo se encontra do portal

