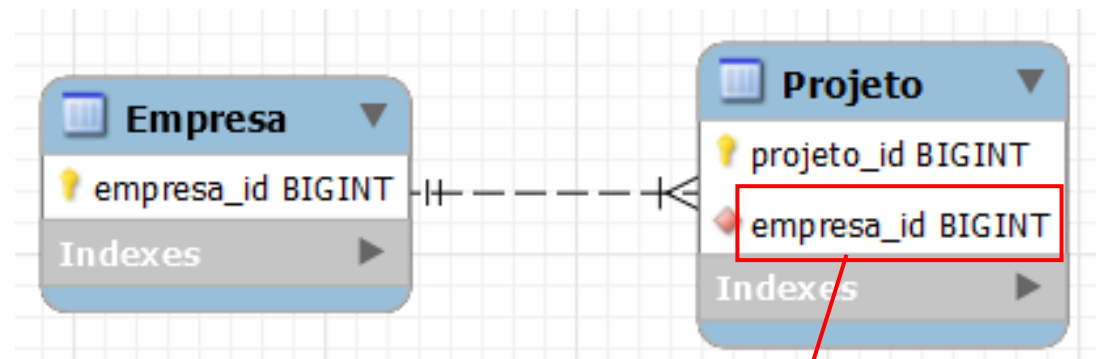


Springboot

Relacionamentos entre classes de modelo

Relacionamento 1.. *



- As configurações relacionadas ao mapeamento serão feitas utilizando anotações JPA nas classes do modelo:

@Entity

@Table(name="Empresa")

public class Empresa {

//...

@OneToMany(mappedBy="empresa")

private Set<Projeto> projetos;

// getters and setters

}

@Entity

@Table(name="Projeto")

public class Projeto {

//...

@ManyToOne

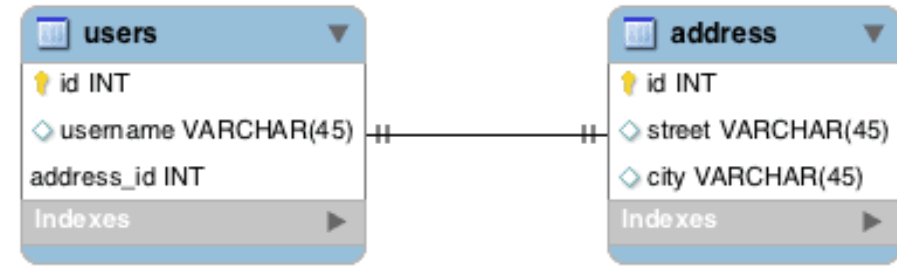
@JoinColumn(name="empresa_id", nullable=false)

private Empresa empresa;

// getters and setters

}

Relacionamento 1..1



```
@Entity
@Table(name = "users")
public class User {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name = "id")
    private Long id;
    //...
    @OneToOne(cascade = CascadeType.ALL)
    @JoinColumn(name = "address_id",
referencedColumnName = "id")
    private Address address;

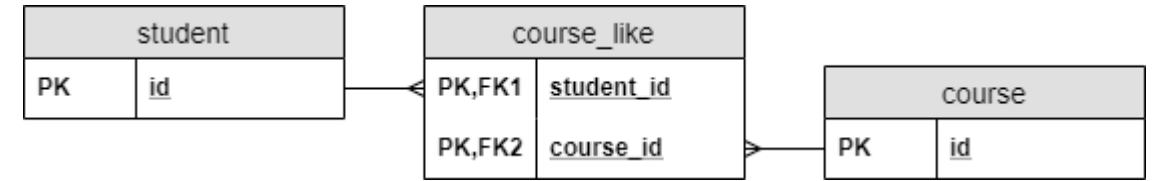
    // ... getters and setters
}
```

```
@Entity
@Table(name = "address")
public class Address {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name = "id")
    private Long id;
    //...

    @OneToOne(mappedBy = "address")
    private User user;

    //... getters and setters
}
```

Relacionamento **



@Entity

```
class Student {  
    @Id  
    Long id;  
    @ManyToMany  
    @JoinTable(  
        name = "course_like",  
        joinColumns = @JoinColumn(name = "student_id"),  
        inverseJoinColumns = @JoinColumn(name = "course_id"))  
    Set<Course> likedCourses;  
    // additional properties  
    // standard constructors, getters, and setters  
}
```

@Entity

```
class Course {  
    @Id  
    Long id;  
    @ManyToMany  
    Set<Student> likes;  
    // additional properties  
    // standard constructors, getters, and setters  
}
```

Tipos de atributos das classes

```
@Column(name = "encontro_hora_fim")
```

```
private LocalDateTime horaFim;
```

```
@Column(name = "objeto_aprendizagem_descricao", columnDefinition = "longtext")
```

```
private String descricao;
```

```
@Lob
```

```
@Column(name = "objeto_aprendizagem_arquivo", columnDefinition = "longblob")
```

```
private byte[] arquivo;
```

Consultas JPA

```
public interface AtividadeRepository extends JpaRepository<Atividade, Long> {  
    @Query("SELECT a from Atividade a WHERE a.status != -1")  
    List<Atividade> getActiveAtividades();  
}
```

```
public interface AcompanhamentoRepository extends JpaRepository<Acompanhamento, Long> {  
    List<Acompanhamento> findByAtividade_IdAndParticipante_Estudante_Id(Long atividadeId, Long estudantId);  
}
```

@Repository

```
public interface ArquivoRepository extends JpaRepository<Arquivo, String>{  
    @Query(value = "SELECT * FROM arquivo WHERE arquivo_id = :id AND arquivo_status >= 0", nativeQuery =  
true)  
    <Arquivo> buscarArquivoAtivosPorId(int id);
```