METRICA

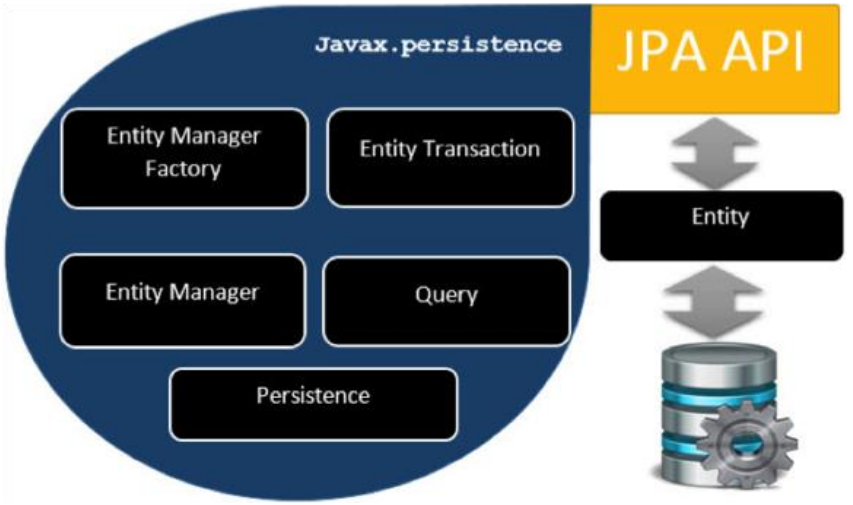
CONSULTING

JPA I

Java Persistence API

JPA es un estándar

- Guardar estado de los objetos
- Código nativo Java
- Mapeo objeto/relacional
- Independiente del gestor de BD
- Basado en etiquetas



Mapeo O/R

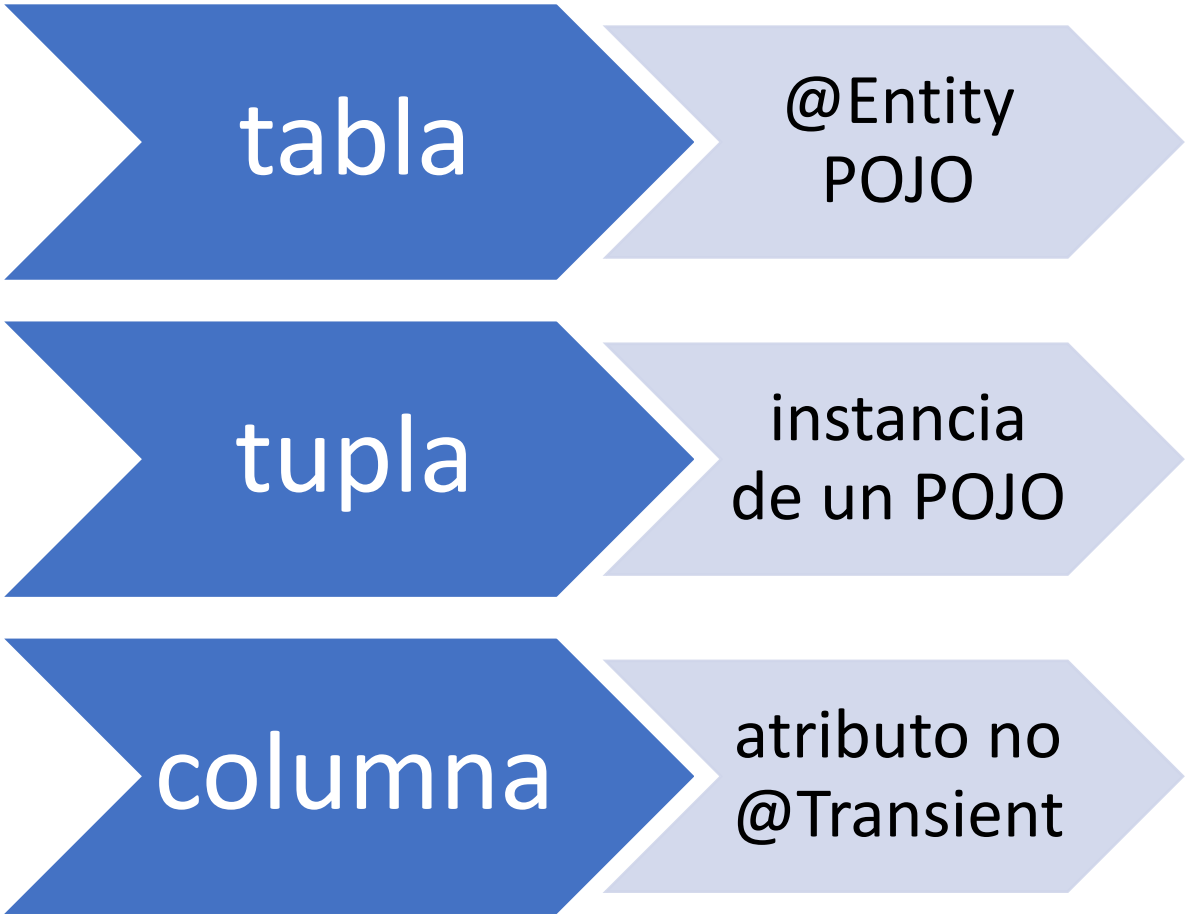


tabla
@Entity
POJO
tupla
instancia
de un POJO
columna
atributo no
@Transient

Entidades @Entity

Una instancia es una tupla

- nombres de los atributos
- @Table para definir otros nombres o índices

Deben ser de tipo JavaBean

- constructor vacío
- getter y setter

Todos los atributos no @Transcient son persistentes

Cada instancia debe ser única, identificada por uno o mas de sus atributos

Ejemplo @Entity

```
@Entity
@Table(
    name = "EMPLOYEES" ,
    schema = "jpatutorial",
    indexes = {@Index(name = "name_index", columnList = "name",unique = true)}
)
public class Employee {
    private Long id;
    private String name;

    /**
     * GETs and SETs
     */
}
```

Tipo de atributos permitidos

no *static* o *final*

tipos primitivos y sus *wrapper*

tipos básicos derivados

- String
- BigInteger

fechas

- Date
- Calendar
- LocalDate
- LocalTime

otras entidades

clases @Embedable

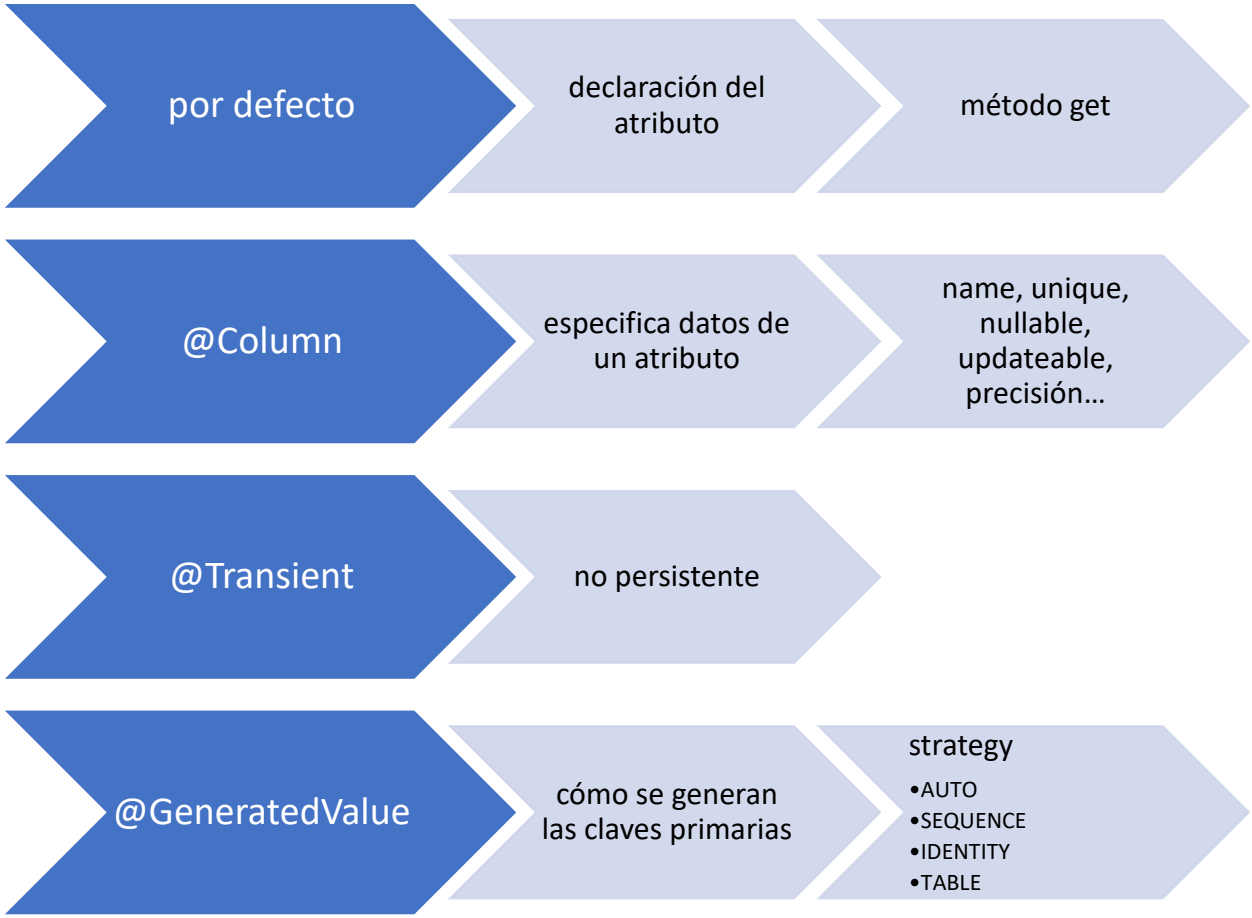
collection de @Entity

clases @Serializable

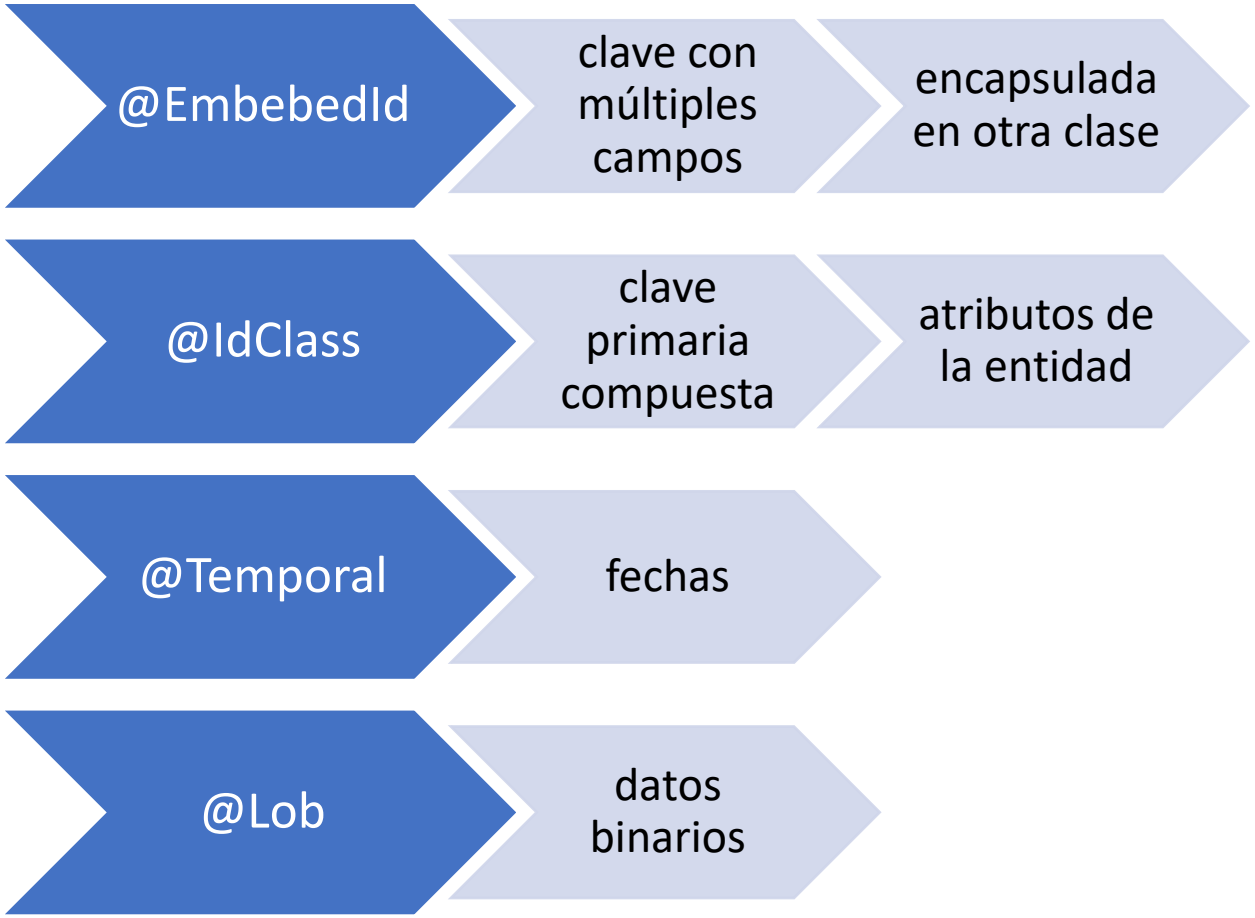
Mapas como atributo

```
@ElementCollection
@MapKeyColumn(name="name")
@Column(name="value")
@CollectionTable(name="example_attributes", joinColumns=@JoinColumn(name="example_id")
Map<String, String> attributes = new HashMap<String, String>(); // maps from attribute
```

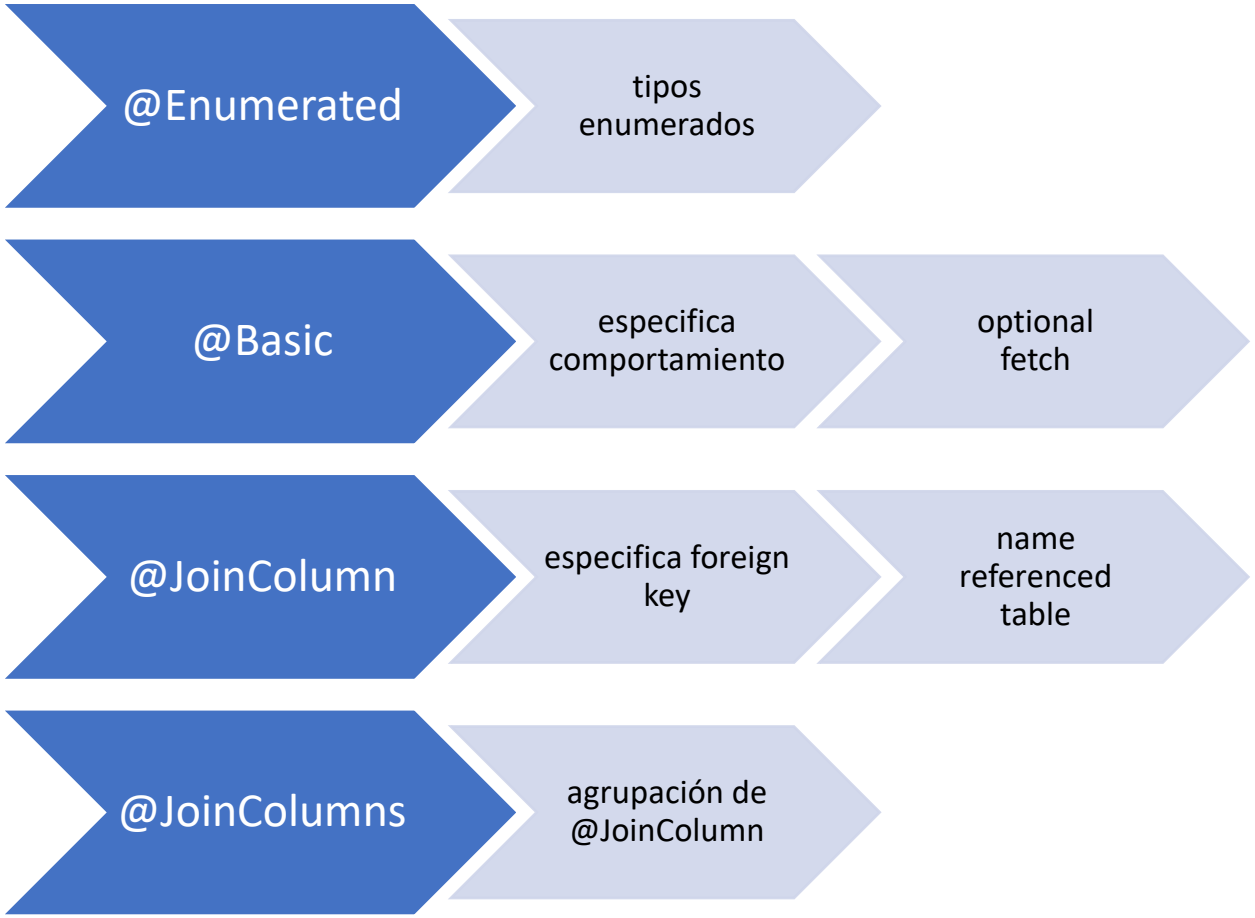
Mapeo de atributos



Mapeo de atributos

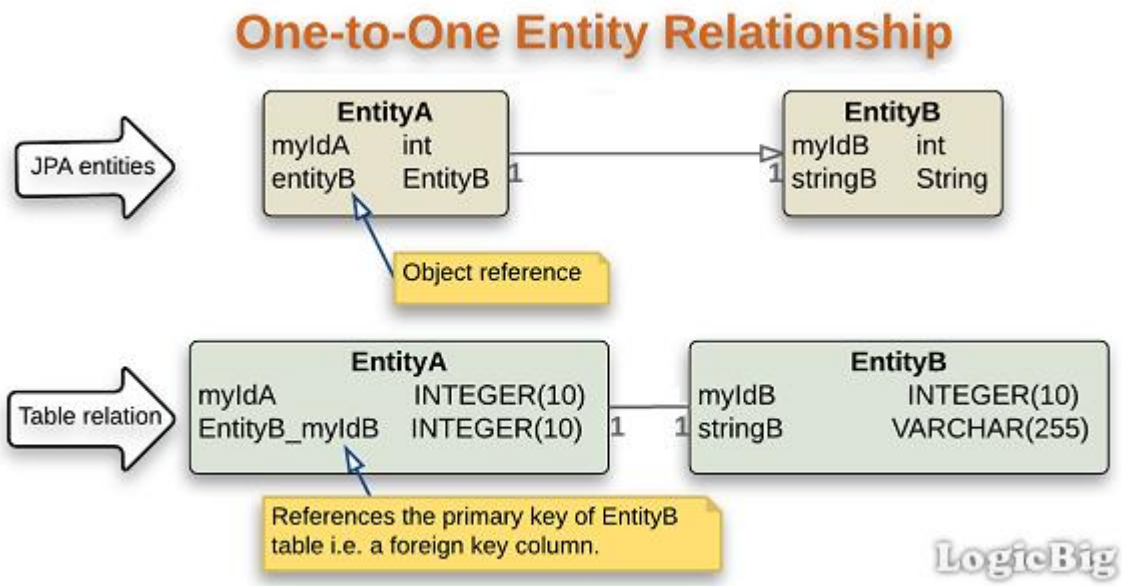


Mapeo de atributos



@OneToOne

Relaciones



https://www.logicbig.com/tutorials/java-ee-tutorial/jpa/entity-relationships-topic-contents.html

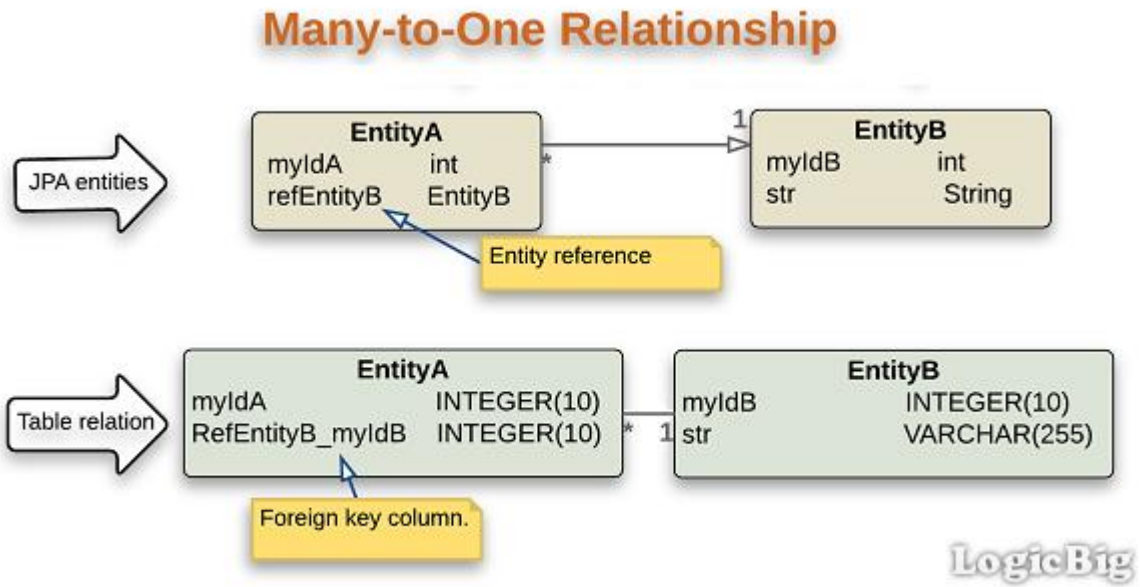
```java
@Entity
public class EntityA {
  @Id
  @GeneratedValue
  private int myIdA;
  @OneToOne
  private EntityB entityB;

  public EntityB getEntityB() {
      return entityB;
  }

  public void setEntityB(EntityB entityB) {
      this.entityB = entityB;
  }
}
```

```java
@Entity
public class EntityB {
  @Id
  @GeneratedValue
  private int myIdB;
  private String stringB;

  public String getStringB() {
      return stringB;
  }

  public void setStringB(String stringB) {
      this.stringB = stringB;
  }
}
```

```java
@Entity
public class EntityA {
  @Id
  @GeneratedValue
  private int myIdA;
  @OneToOne
  @JoinColumn(name = "MY_JOIN_COLUMN")
  private EntityB entityB;

  public EntityB getEntityB() {
      return entityB;
  }

  public void setEntityB(EntityB entityB) {
      this.entityB = entityB;
  }
}
```

```java
@Entity
public class EntityB {
  @Id
  @GeneratedValue
  private int myIdB;
  private String stringB;

  public String getStringB() {
      return stringB;
  }

  public void setStringB(String stringB) {
      this.stringB = stringB;
  }
}
```

@ManyToOne



Relaciones

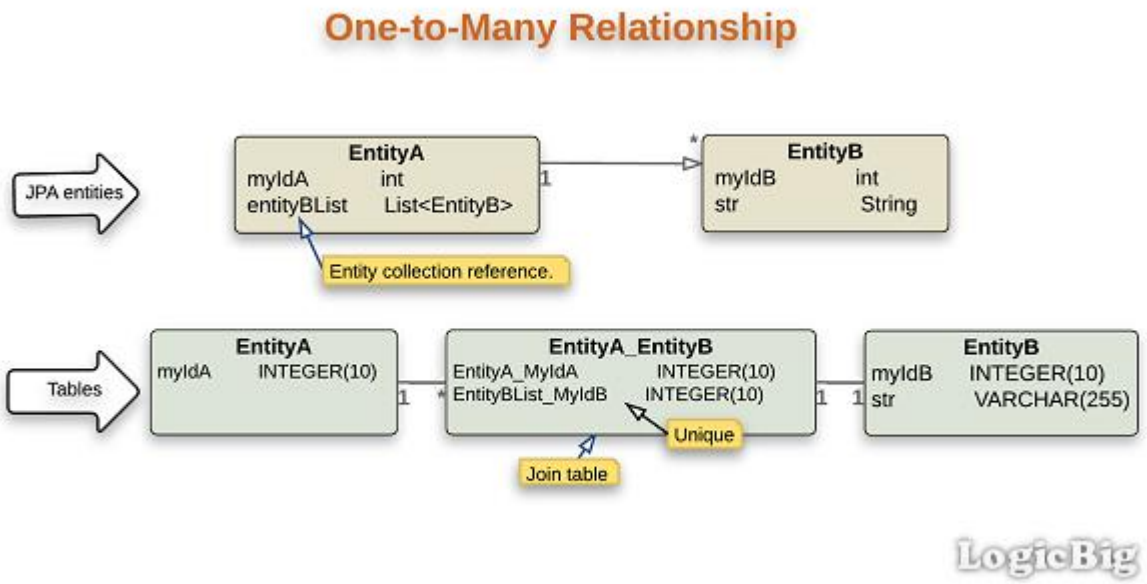
https://www.logicbig.com/tutorials/java-ee-tutorial/jpa/entity-relationships-topic-contents.html

```java
@Entity
public class EntityA {
  @Id
  @GeneratedValue
  private int myIdA;
  @ManyToOne
  private EntityB refEntityB;
    ............
}
```

```java
@Entity
public class EntityB {
  @Id
  @GeneratedValue
  private int myIdB;
  private String str;
    ............
}
```

@OneToMany

Relaciones



https://www.logicbig.com/tutorials/java-ee-tutorial/jpa/entity-relationships-topic-contents.html

```java
@Entity
public class EntityA {
  @Id
  @GeneratedValue
  private int myIdA;
  @OneToMany
  private List<EntityB> entityBList;
    ..............
}
```

```java
@Entity
public class EntityB {
  @Id
  @GeneratedValue
  private int myIdB;
  private String str;
    ..............
}
```

```java
@Entity
public class EntityA {
  @Id
  @GeneratedValue
  private int myIdA;
  @OneToMany
  @JoinTable(name = "MY_JOIN_TABLE",
          joinColumns = {@JoinColumn(name = "MY_ENTITY_A_FK")},
          inverseJoinColumns = {@JoinColumn(name = "MY_ENTITY_B_FK")}
  )
  private List<EntityB> entityBList;
```

```java
@Entity
public class EntityB {
  @Id
  @GeneratedValue
  private int myIdB;
  private String str;
    ..............
}
```

Parámetros comunes

cascade

- ALL
- PERSIST
- MERGE
- REMOVE
- REFRESH

Parámetros comunes

fetch

- LAZY
- EAGER

Parámetros comunes

optional

mappedBy

Ejemplo mapeo O/RI