



# JPA – JAVA PERSISTENCE API

PROF. ELIANE RODRIGUES MARION SANTA ROSA  
PROF. FÁTIMA

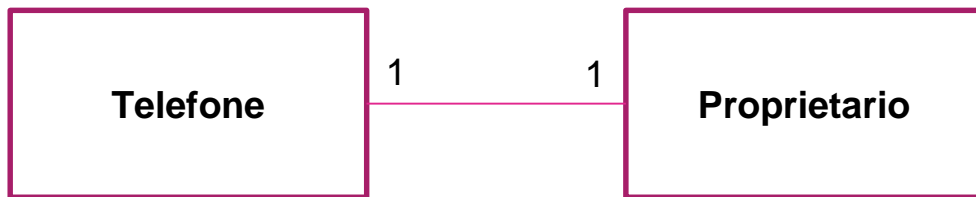
1

# Relacionamentos



## Associações um-para-um

- O relacionamento um-para-um, também conhecido como one-to-one, pode ser usado para dividir uma entidade em duas (criando duas tabelas), para ficar mais normalizado e organizado.
- Esse tipo de associação poderia ser usado entre Telefone e Proprietario.





## Associações um-para-um

- Precisamos anotar a classe Proprietario com @Entity, fazer as anotações dos atributos.

```
@Entity
@Table(name = "TBL_TELEFONE")
public class Telefone {

    @Id
    @GeneratedValue
    @Column(name = "codigo")
    private int id;

    @Column(name = "nro_ddd")
    private int ddd;

    @Column(name = "tx_numero")
    private String numero;
}
```

```
@Entity
public class Proprietario {

    @Id
    @GeneratedValue
    @Column(name = "codigo")
    private int codigo;

    @OneToOne
    private Telefone telefone;
}
```



## Associações um-para-um

- A anotação `@OneToOne` é usada para especificar um relacionamento de banco de dados um-para-um.
- Por padrão, o nome da coluna no banco de dados é definido com o nome do atributo da associação `nomeAtributoIdDestino`, exemplo `COD_TELEFONE`, por isso precisamos fazer essa indicação através da anotação `@JoinColumn`.

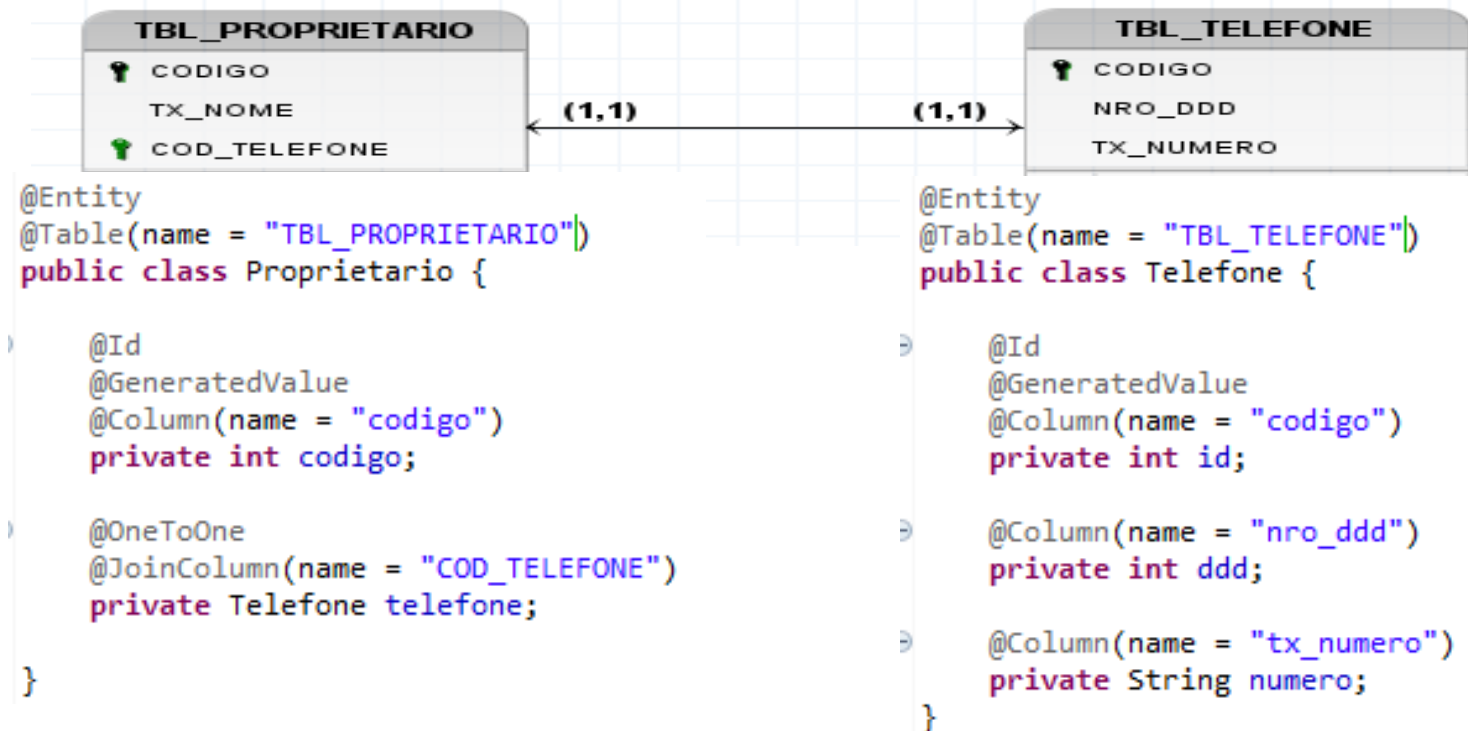
```
@OneToOne  
@JoinColumn(name = "COD_TELEFONE")  
private Telefone telefone;
```



Nome do campo na tabela  
TBL\_Proprietario



## Associações um-para-um



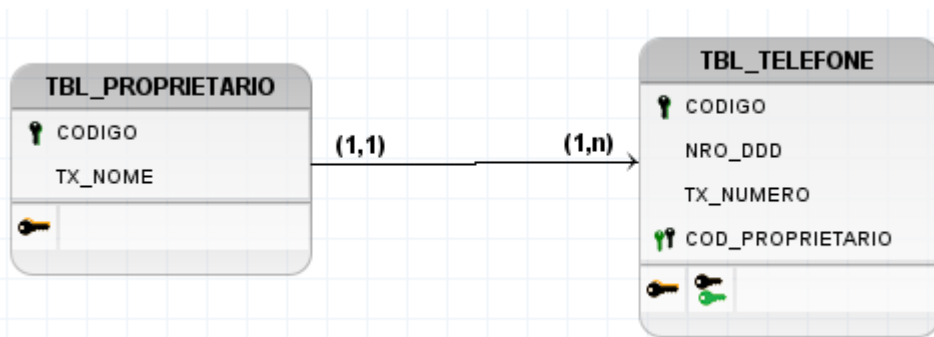
# 2

## **Anotações JPA Relacionamento OneToMany e ManyToOne**



## Associações muitos-para-um

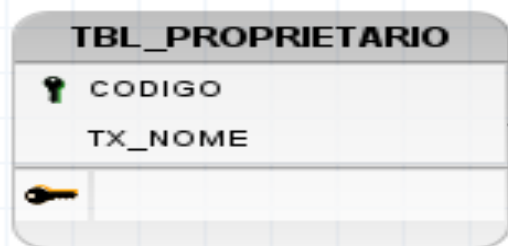
- Vamos alterar o mapeamento que fizemos anteriormente, Mudaremos para o relacionamento **ManyToOne**, indica que pode haver muitos objetos do telefone para um proprietário.
- A anotação @ManyToOne indica a multiplicidade do relacionamento entre telefone e proprietário.







## Associações muitos-para-um



(1,1)

(1,n)



```
@Entity
@Table(name = "TBL_PROPRIETARIO")
public class Proprietario {

    @Id
    @GeneratedValue
    @Column(name = "codigo")
    private int codigo;

    @Column(name = "tx_nome")
    private String nome;
}
```

```
@Entity
@Table(name = "TBL_TELEFONE")
public class Telefone {

    @Id
    @GeneratedValue
    @Column(name = "codigo")
    private int id;

    @Column(name = "nro_ddd")
    private int ddd;

    @Column(name = "tx_numero")
    private String numero;

    @ManyToOne
    @JoinColumn(name = "COD_PROPRIETARIO")
    private Proprietario proprietario;
}
```