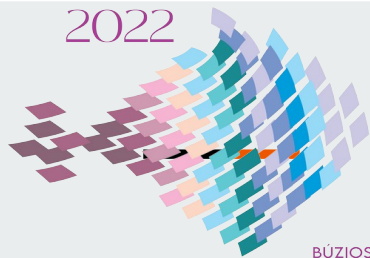


**SBBD**

2022



BÚZIOS  
RIO DE JANEIRO



# Projeto de Bancos de Dados NoSQL

**Angelo Augusto Frozza  
Geomar André Schreiner  
Ronaldo dos Santos Mello**

**GBD/INE/CTC/UFSC**



# Agenda

---

1. Projeto de BD
2. Projeto Lógico de Alto Nível para BD NoSQL Orientado a Agregados
3. Projeto Lógico de Baixo Nível para BD NoSQL Orientado a Agregados
4. Projeto Lógico para BD NoSQL Orientado a Grafos

# Agenda

---

## 1. Projeto de BD

- 2. Projeto Lógico de Alto Nível para BD NoSQL Orientado a Agregados
- 3. Projeto Lógico de Baixo Nível para BD NoSQL Orientado a Agregados
- 4. Projeto Lógico para BD NoSQL Orientado a Grafos

# Projeto de Banco de Dados (BD)

- **Parte integrante do desenvolvimento de um sistema informatizado para uma Organização**

- representação adequada dos dados operacionais
- armazenamento e acesso eficientes

- **Metodologia de projeto de BD**

- definição de esquemas de dados em diferentes níveis de abstração
  - atividades de modelagem de dados
- níveis comumente considerados na literatura
  - **conceitual**
  - **lógico**
  - **físico**



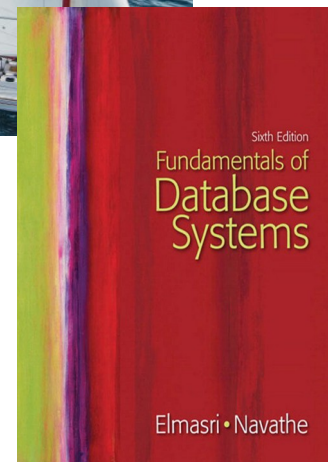
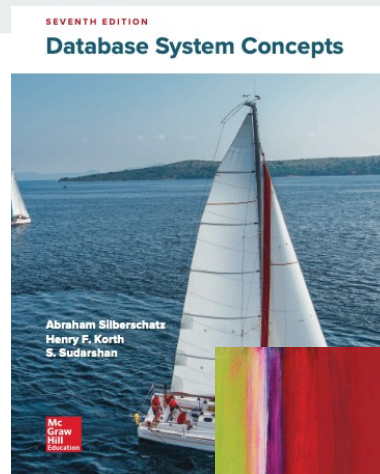
# Metodologia de Projeto de BD

1. Levantamento de requisitos

2. Projeto conceitual

3. Projeto lógico

4. Projeto físico



# Metodologia de Projeto de BD

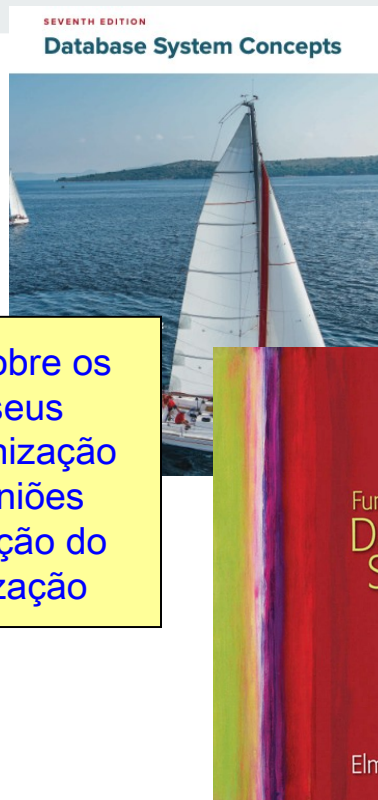
## 1. Levantamento de requisitos

## 2. Projeto conceitual

## 3. Projeto lógico

## 4. Projeto físico

- coleta de informações sobre os dados, suas restrições e seus relacionamentos na Organização
- forma de realização: reuniões com os usuários; observação do funcionamento da Organização



# Metodologia de Projeto de BD

## 1. Levantamento de requisitos

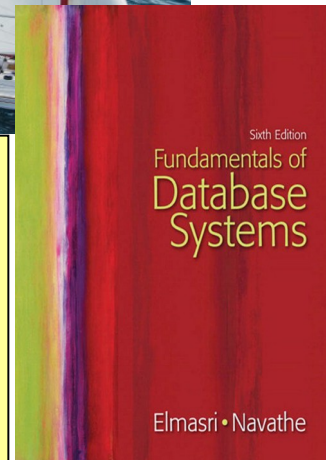
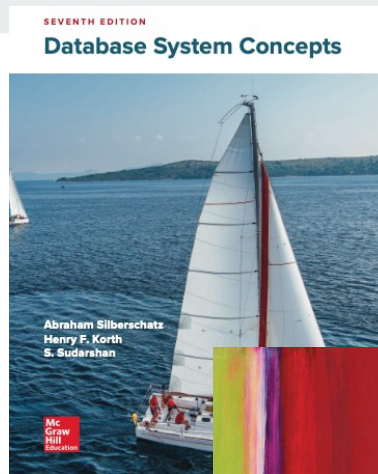
## 2. Projeto conceitual

## 3. Projeto lógico

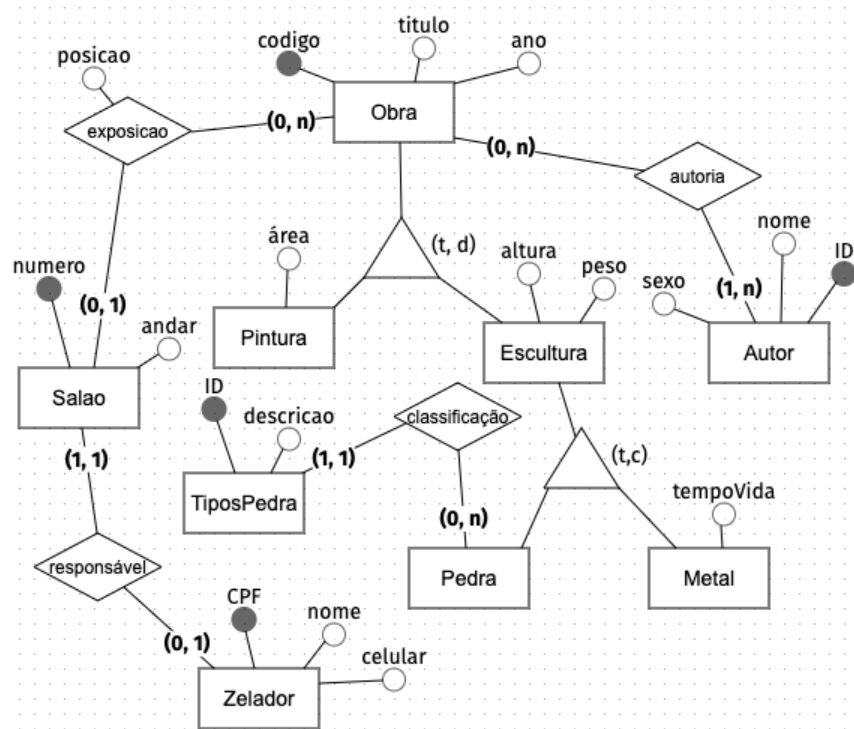
## 4. Projeto físico

↙ ↘  
*Especificação de requisitos*

- modelagem de alto nível dos dados e seus relacionamentos (*independente do modelo de dados do SGBD alvo – esquema conceitual*)
- forma de realização: análise da especificação de requisitos de dados
- objetivo: entendimento dos dados da Organização e validação deste entendimento junto aos usuários da Organização



# Modelo Entidade-Relacionamento (ER)





# Metodologia de Projeto de BD

## 1. Levantamento de requisitos

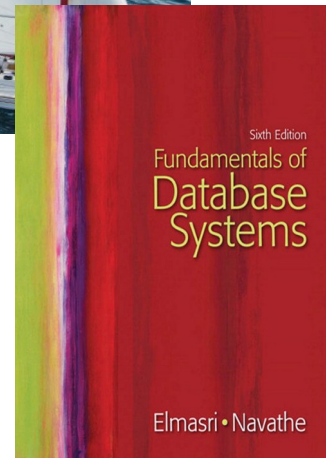
## 2. Projeto conceitual

## 3. Projeto lógico

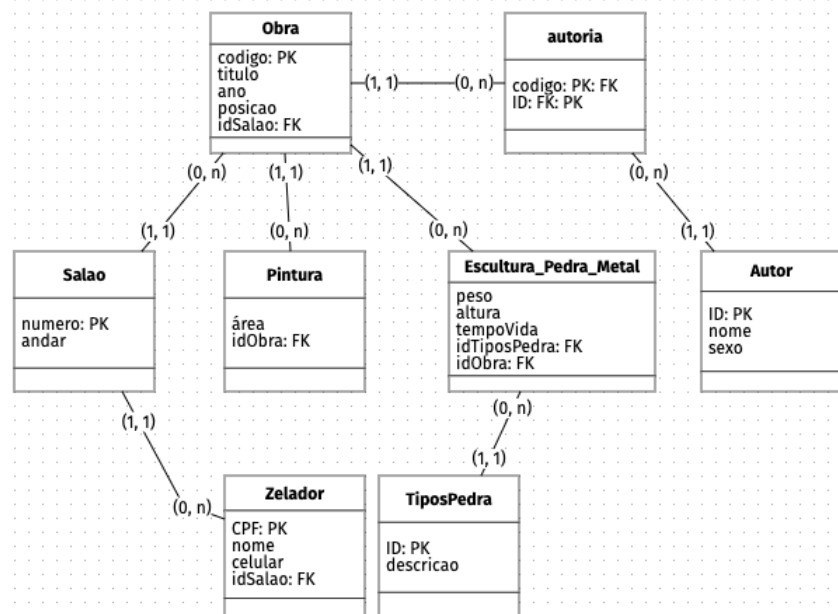
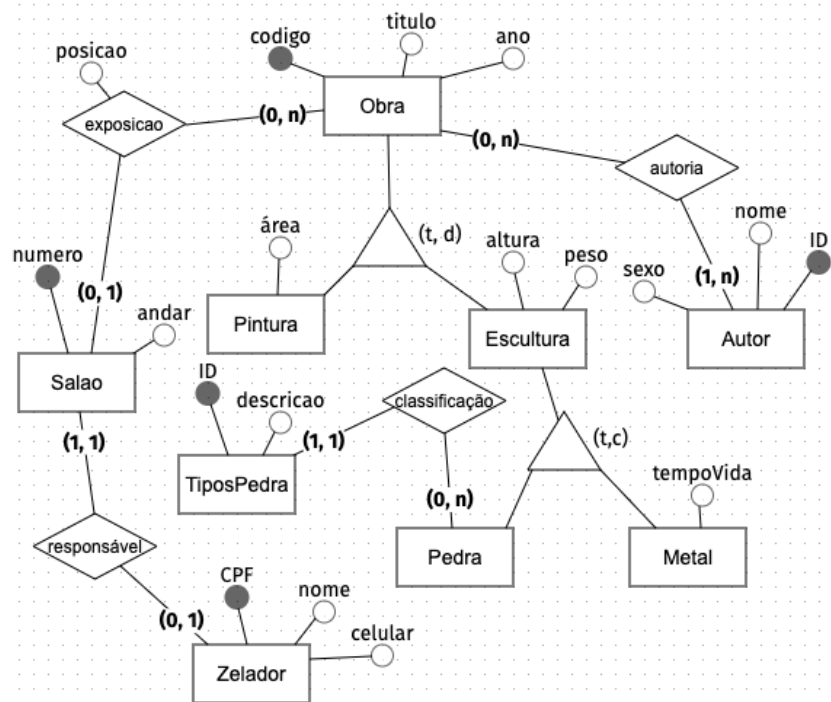
## 4. Projeto físico

- conversão do esquema conceitual em um *esquema válido no modelo de dados do SGBD alvo (esquema lógico)*
- forma de realização: aplicação de regras de conversão
- objetivo: geração de um esquema de dados eficiente para fins de armazenamento e acesso

*Esquema conceitual*



# Conversão ER-Relacional



# Metodologia de Projeto de BD

## 1. Levantamento de requisitos

## 2. Projeto

- especificação do esquema lógico no SGBD alvo
- forma de realização: **instruções DDL**
- objetivo: implementação: (i) da estrutura dos dados e; (ii) de recursos para garantia de integridade, segurança e otimização de acesso (p.ex.: *checks*, gatilhos, visões, índices)

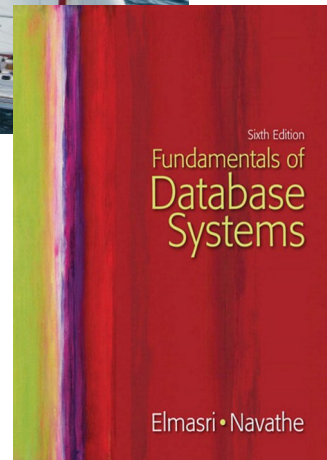
## 3. Projeto

*Esquema lógico*

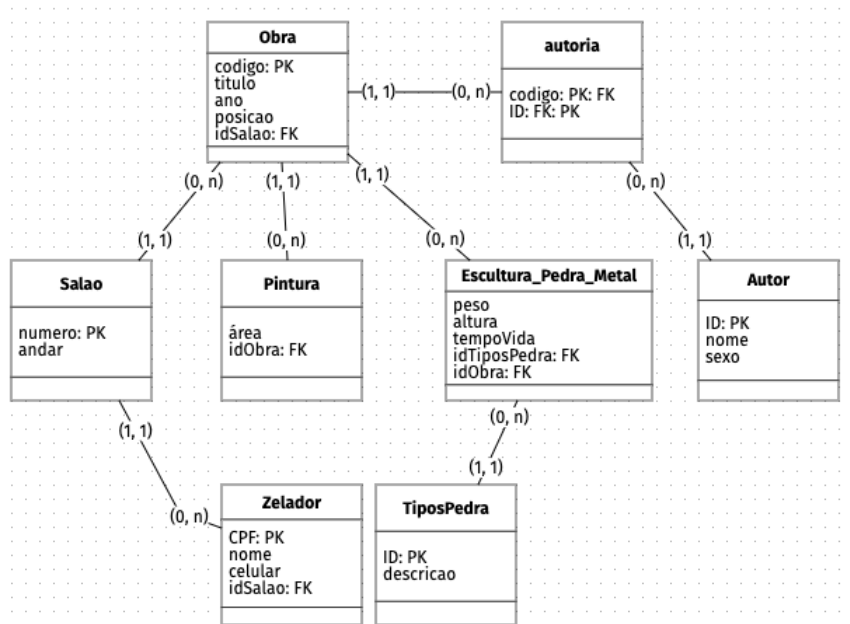
## 4. Projeto físico

SEVENTH EDITION

Database System Concepts



# Implementação SQL/DDL



```

CREATE TABLE Salao (
    numero int,
    andar smallint CHECK (andar > 0
    AND andar < 10),
    PRIMARY KEY(numero));
...
CREATE INDEX Obra_ano
ON Obra(ano)
USING btree;
...
CREATE VIEW ObrasAutor AS
SELECT o.titulo, o.ano, a.nome
FROM Obra_... o JOIN Autor a
ON o.idAutor = a.codigo
...
  
```

# Agenda

---

1. Projeto de BD

**2. Projeto Lógico de Alto Nível para BD NoSQL Orientado a Agregados**

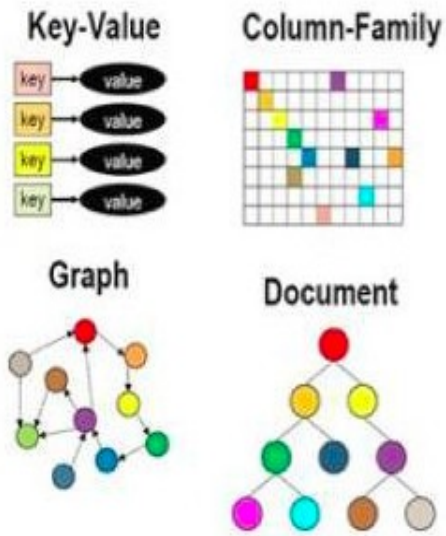
3. Projeto Lógico de Baixo Nível para BD NoSQL Orientado a Agregados

4. Projeto Lógico para BD NoSQL Orientado a Grafos

# Projeto Lógico BD NoSQL

## ● BDs NoSQL

- família de BDs não-relacionais com modelos lógicos diferentes
- metodologia tradicional de projeto de BD deve ser revista



# Projeto Lógico BD NoSQL

## ● BDs NoSQL

- família de BDs não-relacionais com modelos lógicos diferentes
- metodologia tradicional de projeto de BD deve ser revista
- **proposta:**

Levantamento de requisitos



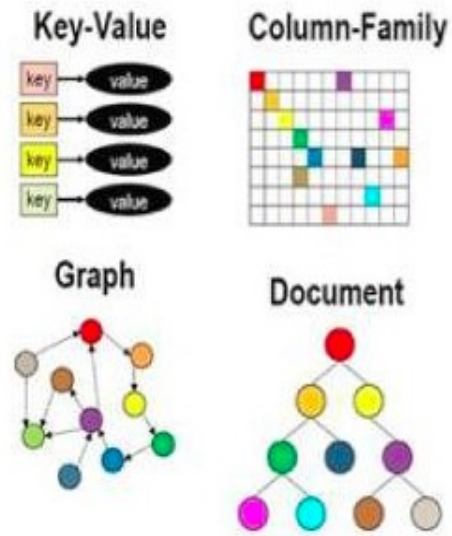
Projeto conceitual



Projeto lógico



Projeto físico



# Projeto Lógico BD NoSQL

## ● BDs NoSQL

- família de BDs não-relacionais com modelos lógicos diferentes
- metodologia tradicional de projeto de BD deve ser revista
- **proposta:**

Levantamento de requisitos



Projeto conceitual



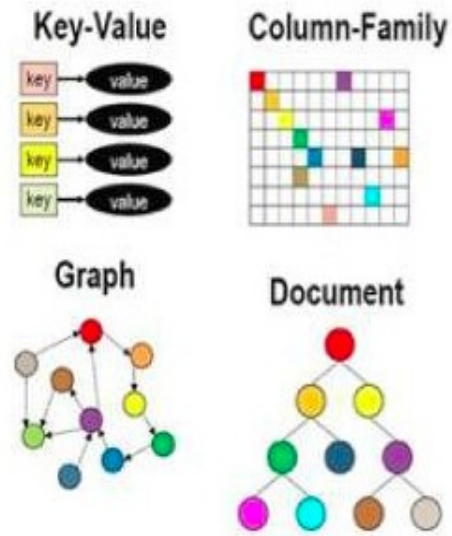
Projeto lógico



Projeto físico

Projeto lógico de alto nível

Projeto lógico de baixo nível





# Projeto Lógico BD NoSQL

## ● BDs NoSQL

- família de BDs não-relacionais com lógicas diferentes
- metodologia tradicional de projeto deve ser revista
- **proposta:**

Levantamento de requisitos



Projeto conceitual



Projeto lógico

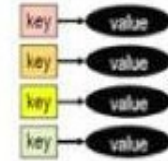


Projeto físico

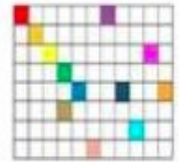
Projeto lógico de alto nível

Projeto lógico de baixo nível

Key-Value



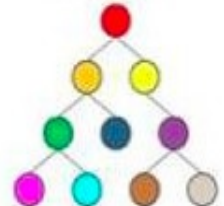
Column-Family



Graph



Document



- abstração em um modelo de dados intermediário baseado no conceito de *Agregado*
- válido para o projeto de BD NoSQL chave-valor, orientado a colunas e orientado a documentos

# Projeto Lógico BD NoSQL

## ● BDs NoSQL

- família de BDs não-relacionais com modelos lógicos diferentes
- metodologia tradicional de projeto de BD deve ser revista
- **proposta:**

Levantamento de requisitos



Projeto conceitual



Projeto lógico



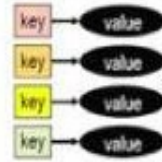
Projeto físico

Projeto lógico de alto nível

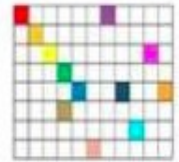
**Projeto lógico de baixo nível**

conversão de um esquema de *Agregados* para os modelos de dados chave-valor, orientado a colunas orientado a documentos (mapeamento mais simples)

Key-Value



Column-Family



Graph



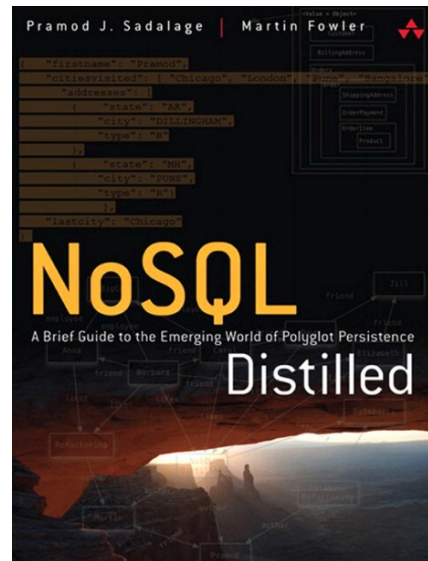
Document



# Modelo de Dados de Agregados

## ● Agregado

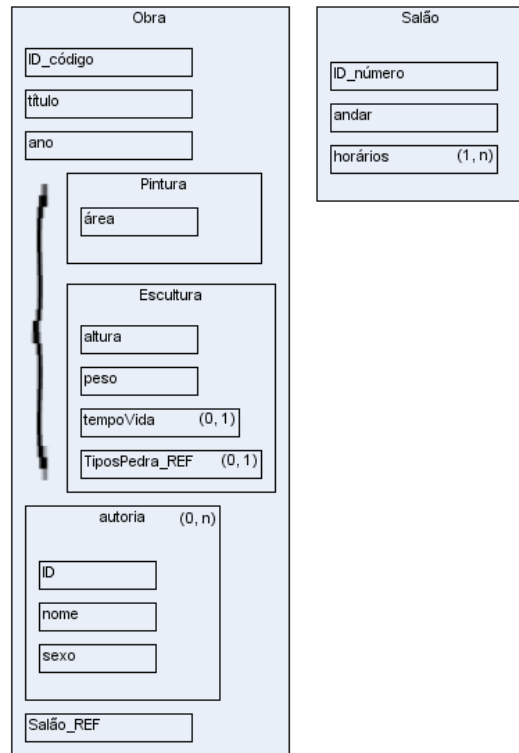
- conceito introduzido por **(Sadalage & Fowler, 2012)**
- abstração de um **objeto identificável** do mundo real cuja estrutura **pode agregar outros objetos**
- similaridade com uma instância de dado representada nos modelos de dados chave-valor, orientado a colunas e orientado a documentos
  - **conteúdo complexo acessado por uma chave**
- evita várias conversões complexas de um esquema conceitual para o esquema de cada um desses 3 modelos de dados NoSQL



# Modelo de Dados de Agregados

## ● Formalização do Modelo de Agregados

- proposto em (Lima & Mello, 2016)
- conceitos:
  - coleção
  - bloco
  - atributo
  - relacionamento de agregação
  - relacionamento de referência
  - restrição de disjunção

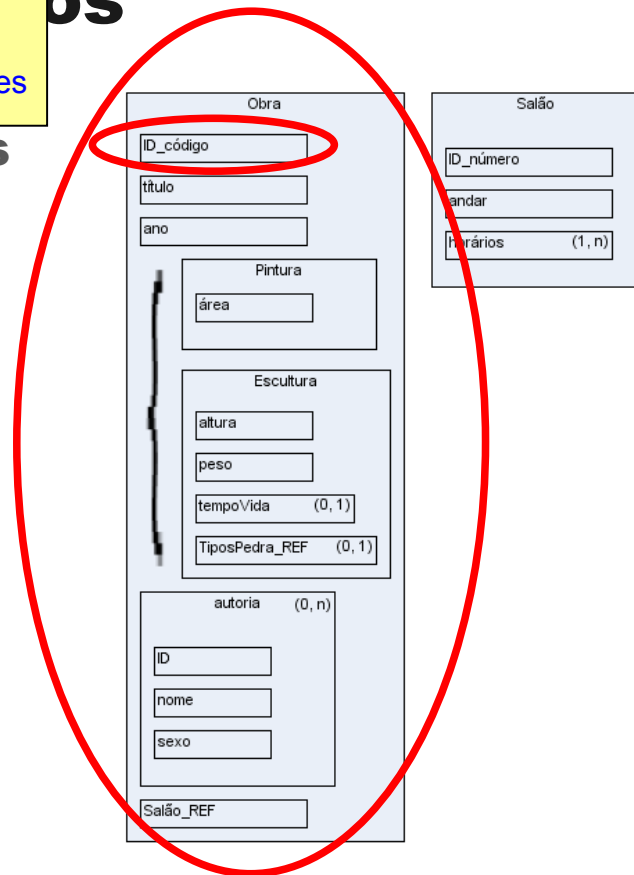


# Modelo de Objetos

- esquema de um objeto do mundo real
- possui um identificador (chave: atributo *ID\_...*)
- pode ser referenciada por outras coleções

## Formalização do Modelo de Agregados

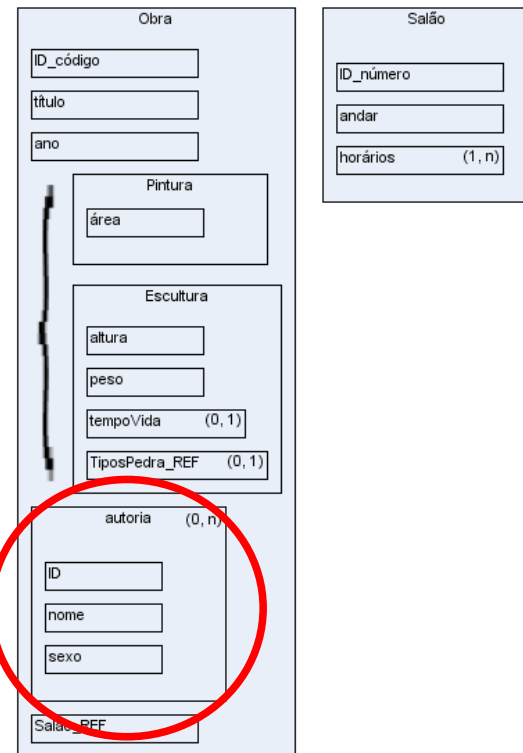
- proposto em (Lima e Mello, 2016)
- conceitos:
  - **coleção**
  - bloco
  - atributo
  - relacionamento de agregação
  - relacionamento de referência
  - restrição de disjunção



# Modelo de Dados de Agregados

- **Formalização**
  - proposto em (L)
  - conceitos:
    - coleção
    - **bloco**
    - atributo
    - relacionamento de agregação
    - relacionamento de referência
    - restrição de disjunção

- componente agregado a uma coleção ou a outro bloco
- pode ter uma cardinalidade associada

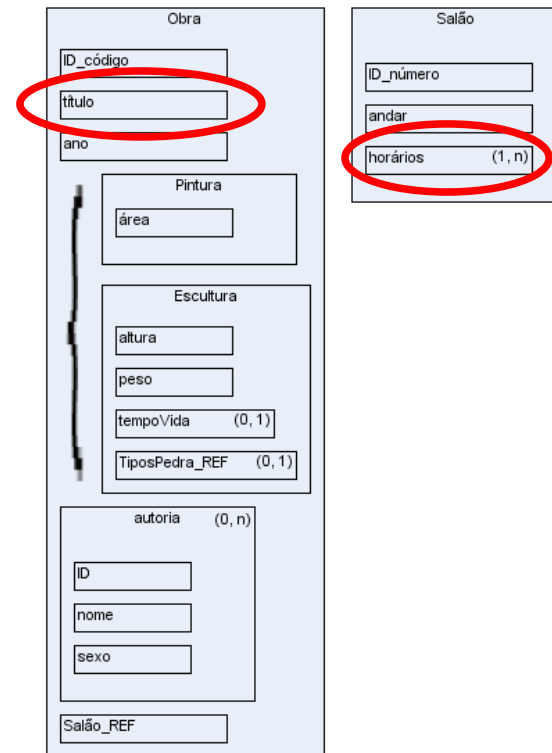


# Modelo de Dados de Agregados

## ● Formalização do Modelo de Agregados

- proposto em (Lima & Mello, 2016)
- conceitos:
  - coleção
  - bloco
  - atributo**
  - relacionamento de agregação
  - relacionamento de referência
  - restrição de disjunção

- propriedade de uma coleção ou bloco
- pode ter uma cardinalidade associada

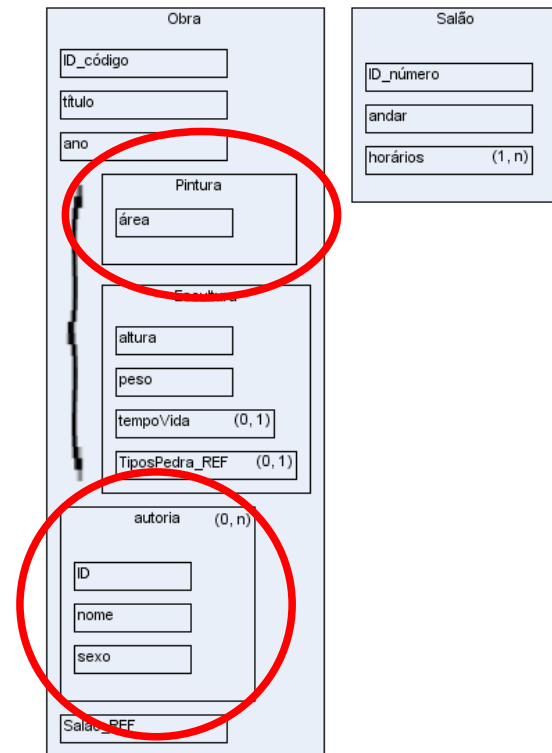


# Modelo de Dados de Agregados

## ● Formalização do Modelo de Agregados

- proposto em (Lima & Mello, 2016)
- conceitos:
  - coleção
  - bloco
  - atributo
  - **relacionamento de agregação**
  - relacionamento de referência
  - restrição de disjunção

blocos contidos dentro de coleções ou de outros blocos (objetos aninhados)



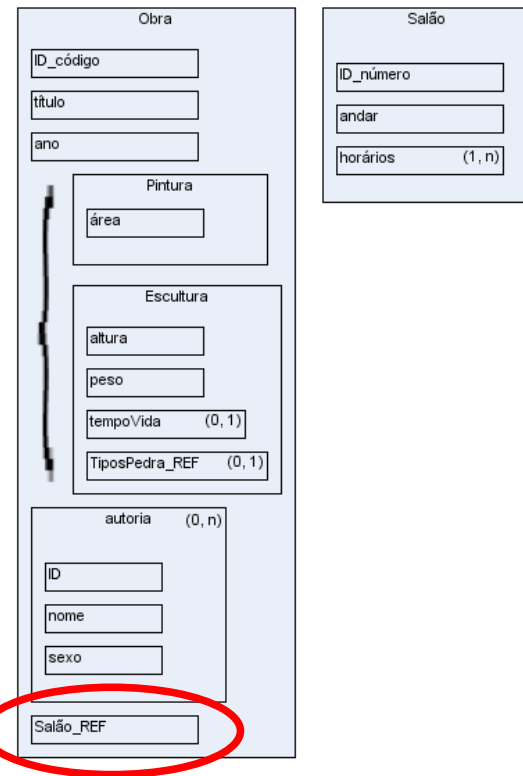


# Modelo de Dados de Agregados

## ● Formalização do Modelo de Agregados

- proposto em (Lima & Mello, 2016)
- conceitos:
  - coleção
  - bloco
  - atributo
  - relacionamento de agregação
  - **relacionamento de referência**
  - restrição de disjunção

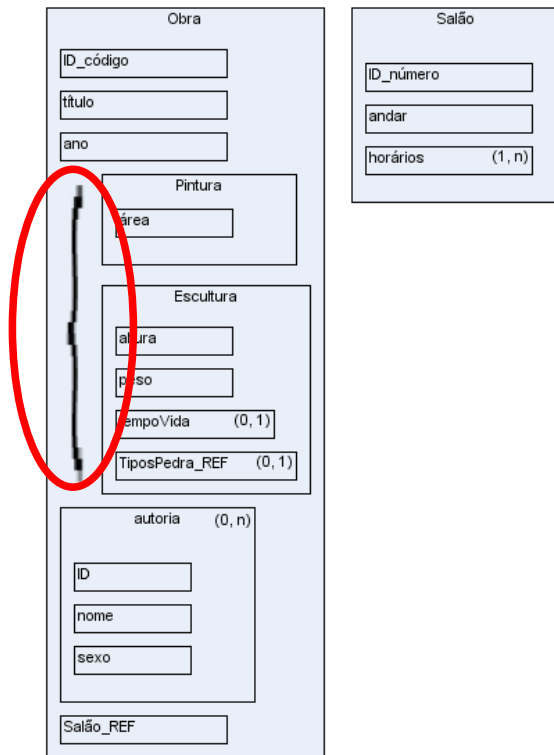
relacionamentos entre coleções  
(atributo ...\_REF)



# Modelo de Dados de Agregados

## ● Formalização do Modelo de Agregados

- proposto em (Lima & Mello, 2016)
- conceitos:
  - coleção
  - bloco
  - atributo
  - relacionamento de **define blocos mutuamente exclusivos**
  - relacionamento de referência
  - **restrição de disjunção**



# Processo de Projeto Lógico de Alto Nível

---

- **Conversão de um esquema conceitual em um esquema de agregados (Lima & Mello, 2016)**
  - baseada nas estratégias para projeto lógico de BD relacionais
- **Etapas**
  - 1) Conversão de Hierarquias
  - 2) Conversão de Relacionamentos

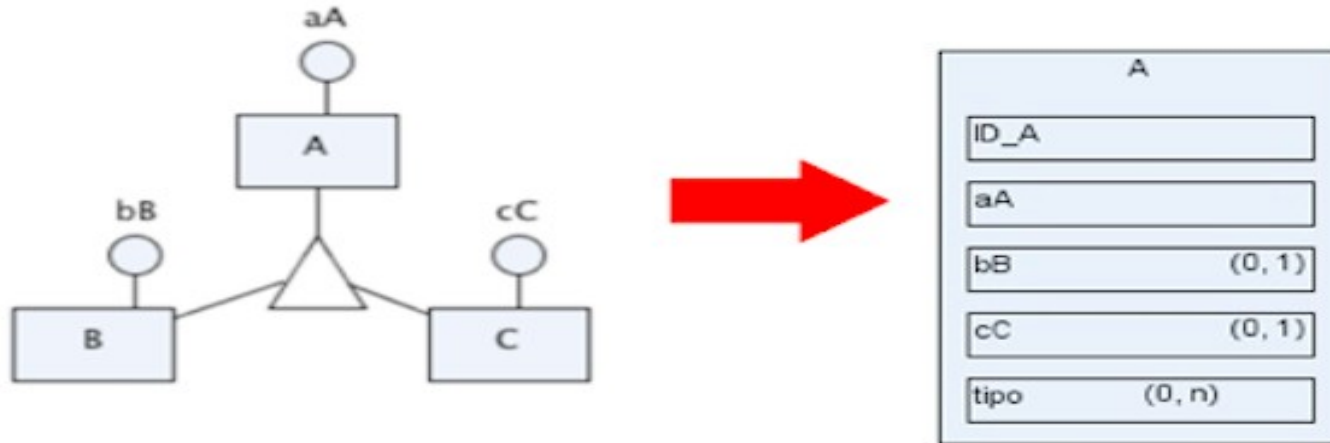
# Etapa 1 – Conversão de Hierarquias

---

- **Processo bottom-up**
- **3 alternativas para a conversão de cada nível hierárquico**
  - ênfase na entidade genérica
  - ênfase nas entidades especializadas
  - ênfase na hierarquia
- **Fatores que influenciam na escolha por uma alternativa**
  - tamanho do esquema
  - transações mais frequentes (*workload* típico)
  - quantidade de restrições de integridade

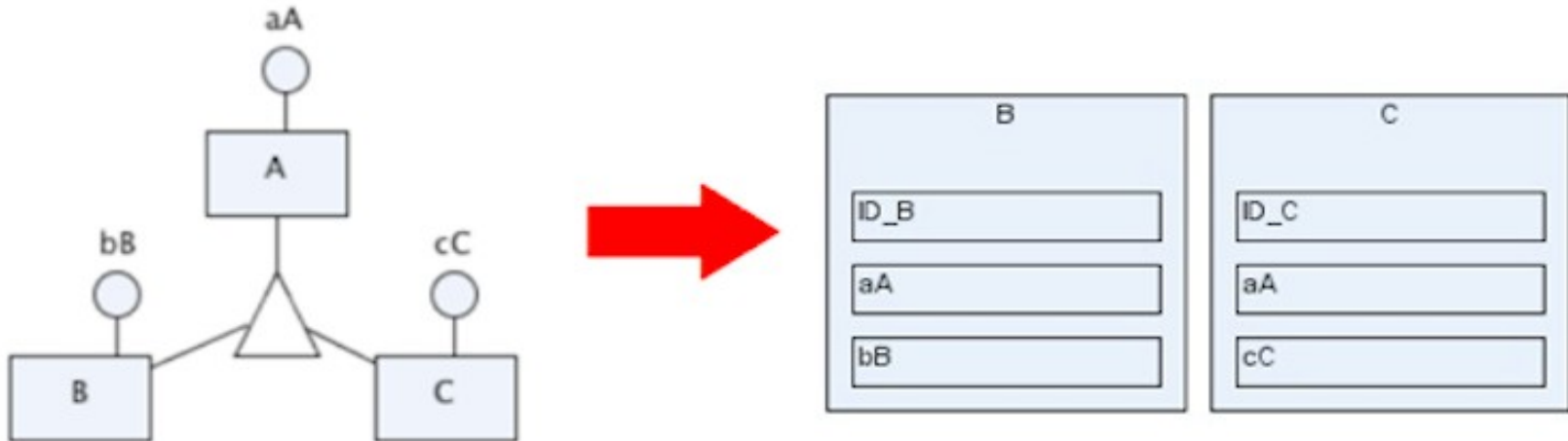
# Etapa 1 – Conversão de Hierarquias

- Processo bottom-up
- 3 alternativas para a conversão de cada nível hierárquico
  - Ênfase na entidade genérica



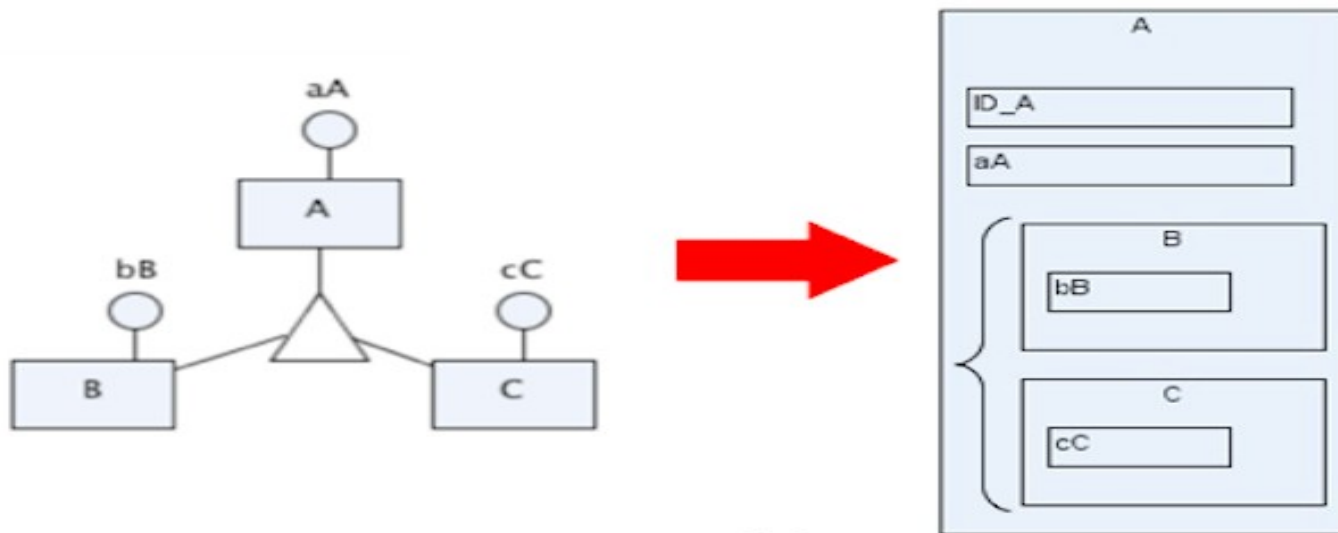
# Etapa 1 – Conversão de Hierarquias

- Processo bottom-up
- 3 alternativas para a conversão de cada nível hierárquico
  - Ênfase nas entidades especializadas



# Etapa 1 – Conversão de Hierarquias

- Processo bottom-up
- 3 alternativas para a conversão de cada nível hierárquico
  - Ênfase na hierarquia



## Etapa 2 – Conversão de Relacionamentos

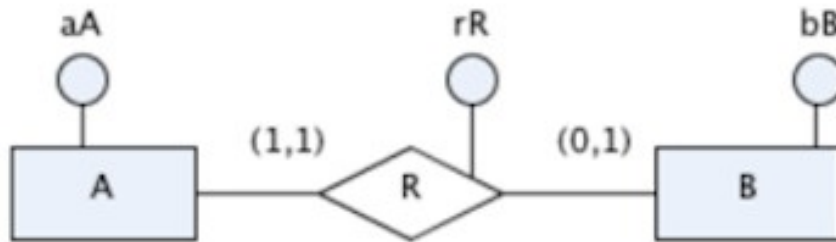
---

- **3 alternativas para a conversão**
  - fusão
  - aninhamento
  - referência
- **Fatores que influenciam na escolha por uma alternativa**
  - cardinalidade do relacionamento
  - tamanho do esquema
  - transações mais frequentes (*workload* típico)
  - quantidade de restrições de integridade



## Etapa 2 – Conversão de Relacionamentos

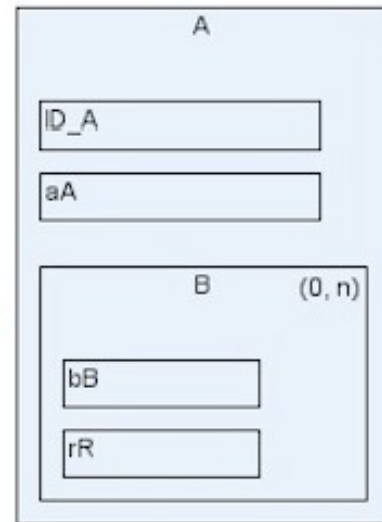
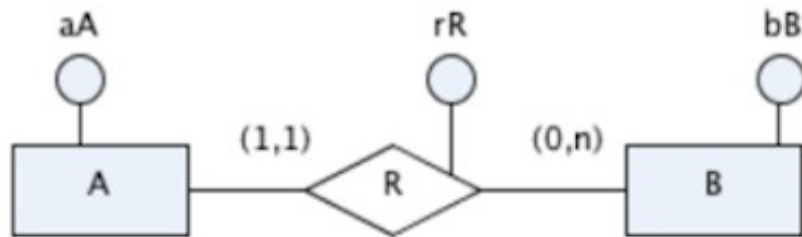
- 3 alternativas para a conversão
  - Fusão



A-B	
ID_A-B	
aA	(0, 1)
bB	(0, 1)
rR	(0, 1)

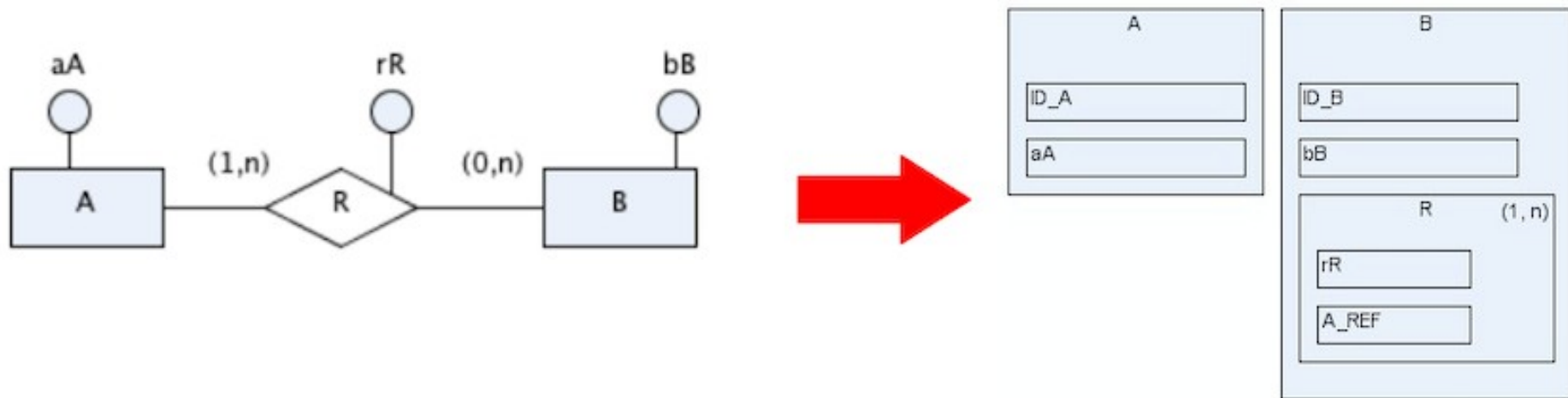
## Etapa 2 – Conversão de Relacionamentos

- 3 alternativas para a conversão
  - Aninhamento



## Etapa 2 – Conversão de Relacionamentos

- 3 alternativas para a conversão
  - Referência



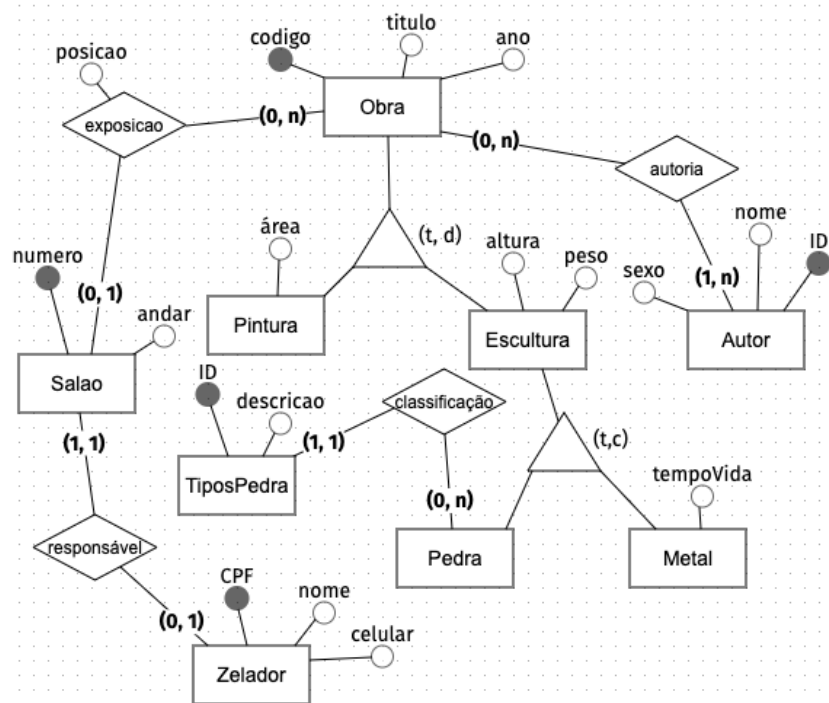
## Etapa 2 – Conversão de Relacionamentos

### ● Algumas recomendações...

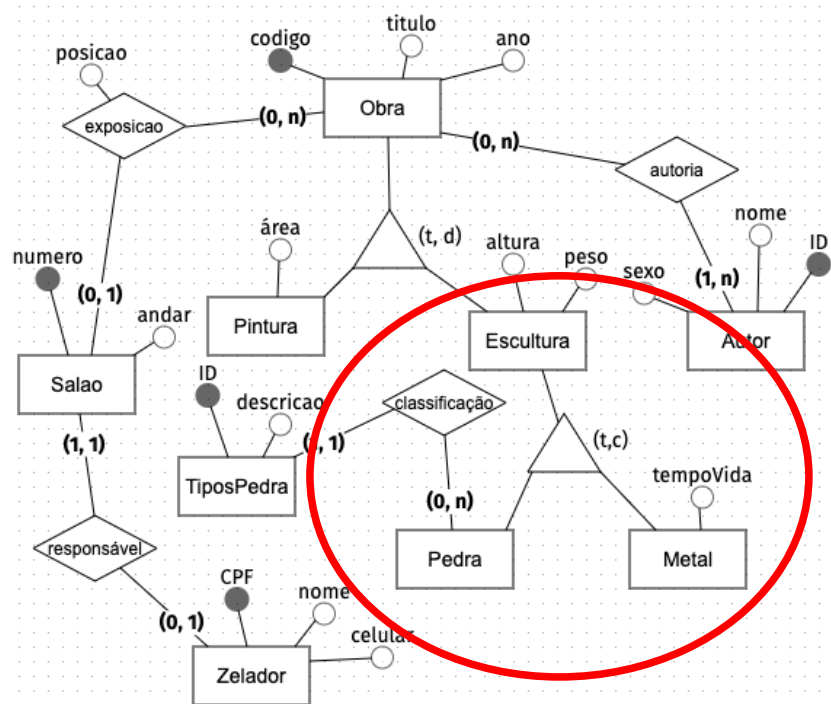
cardinalidade	fusão	aninhamento	referência
(1,1)-(1,1)	✓		
(1,1)-(0,1)		✓	✓
(0,1)-(0,1)			✓
(1,1)-(1,n) / (1,1)-(0,n) / (0,1)-(1,n)		✓	✓
(0,1)-(0,n)			✓
(1,n)-(1,n)/(1,n)-(0,n)		✓	✓
(0,n)-(0,n)			✓

- iniciar a conversão pelos casos 1-1, seguido de 1-n e depois m-n

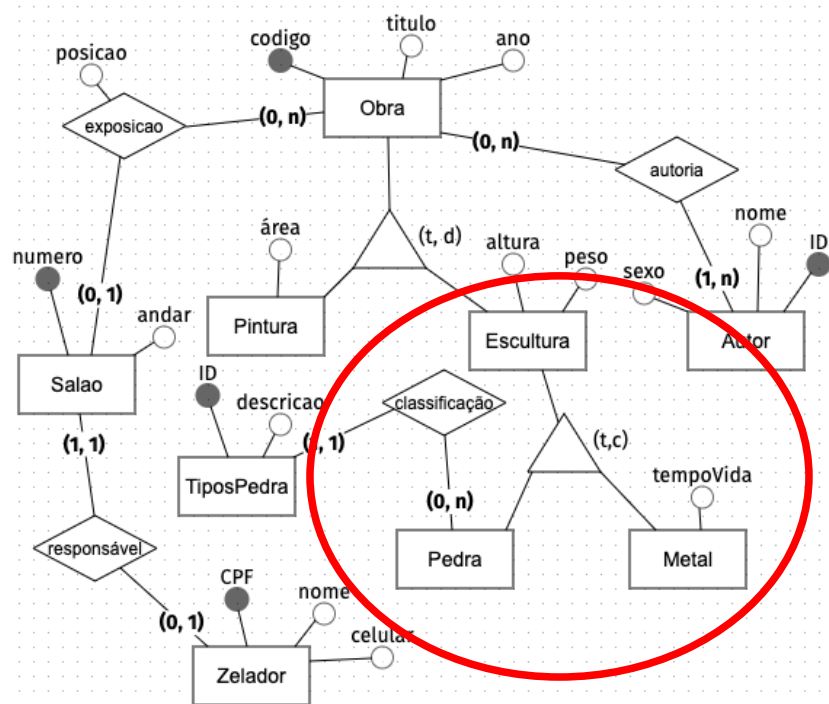
# Running Example – Museu



# Passo 1 – Conversão de Hierarquias



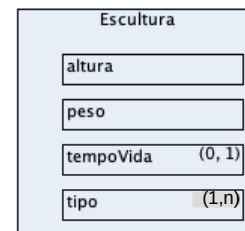
# Passo 1 – Conversão de Hierarquias



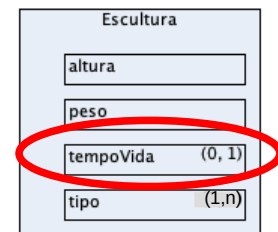
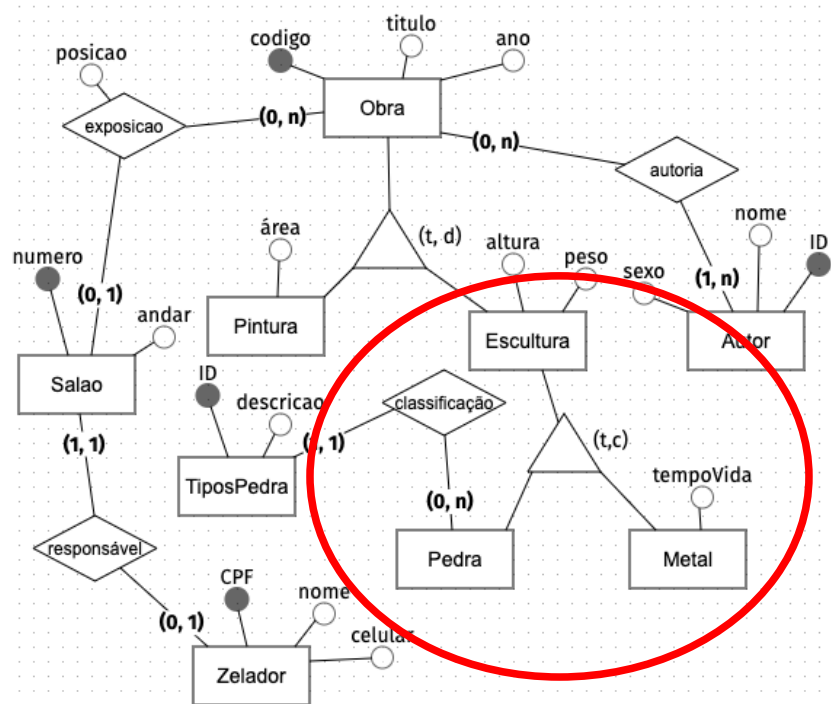
possível alternativa



ênfase na entidade  
genérica



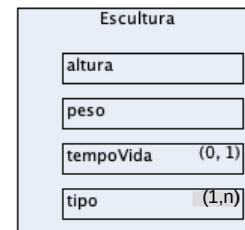
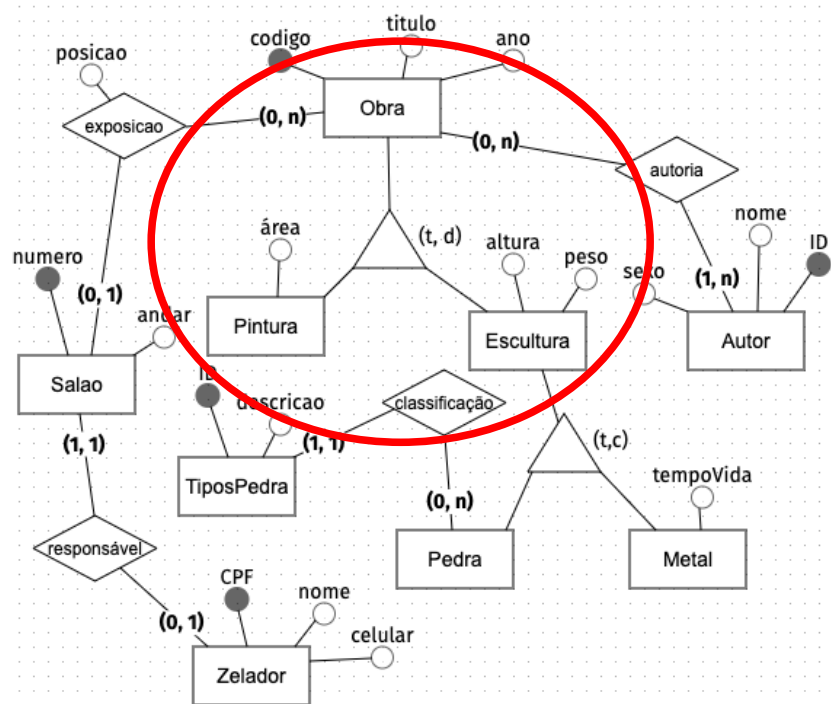
# Passo 1 – Conversão de Hierarquias



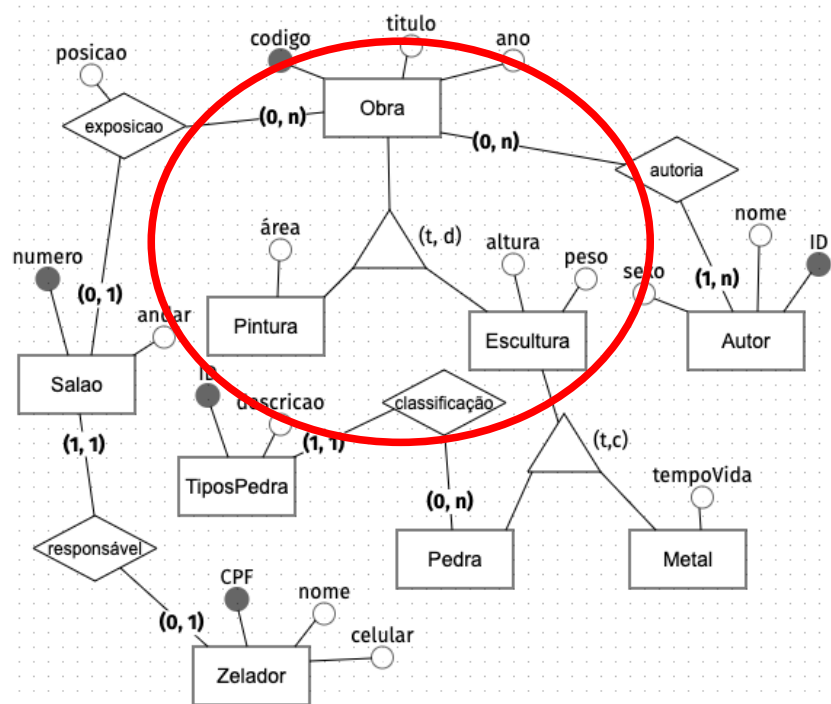
\* **RI:** tempoVida é **obrigatório** quando tipo  $\subseteq$  'Metal'



# Passo 1 – Conversão de Hierarquias



# Passo 1 – Conversão de Hierarquias



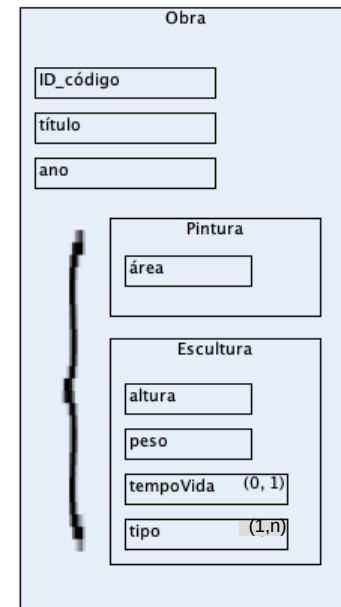
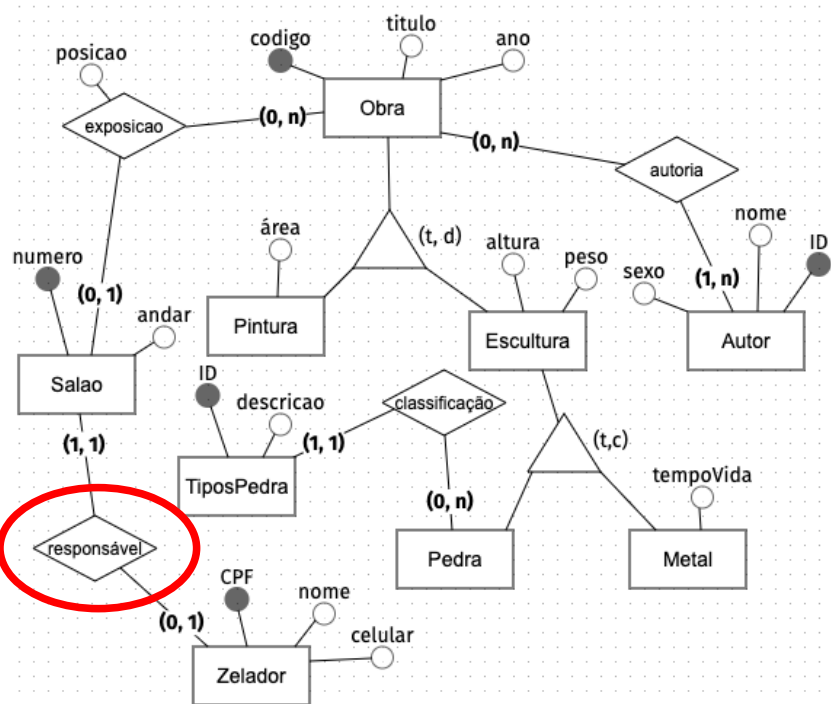
possível alternativa



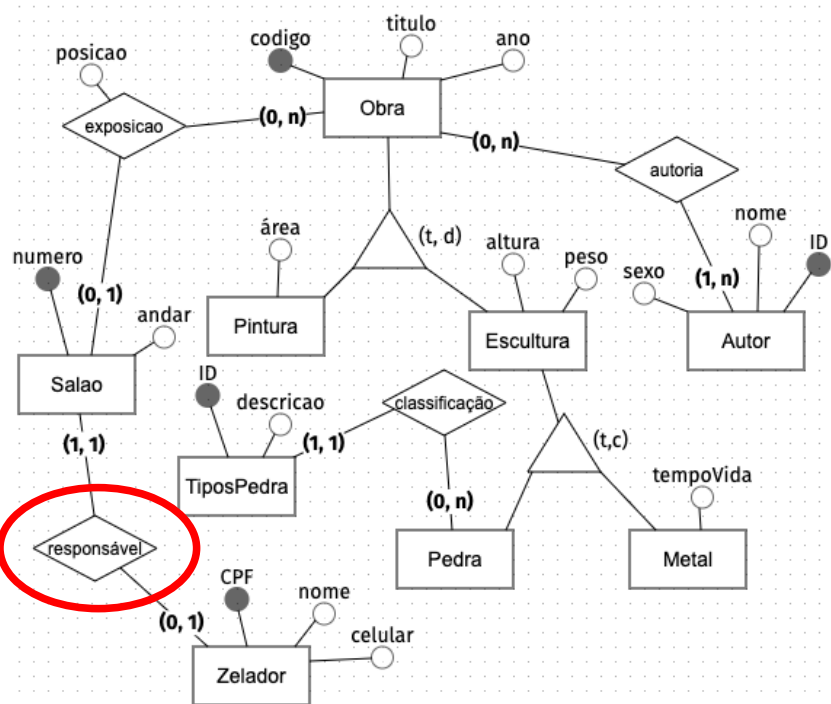
ênfase na hierarquia

Obra	
ID_código	
título	
ano	
<div>Pintura</div> <div>área</div>	
<div>Escultura</div> <div>altura</div> <div>peso</div> <div>tempoVida (0, 1)</div> <div>tipo (1,n)</div>	

## Passo 2 – Conversão de Relacionamentos



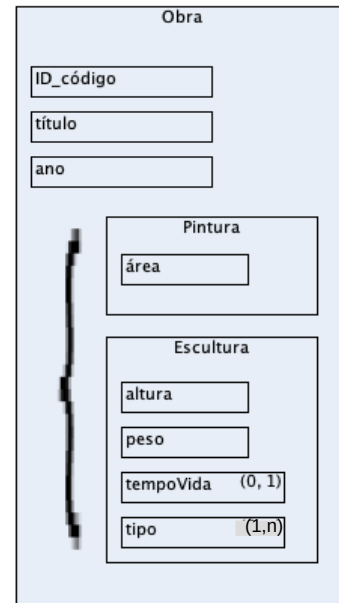
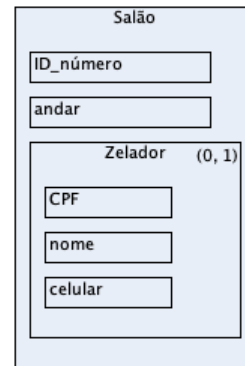
## Passo 2 – Conversão de Relacionamentos



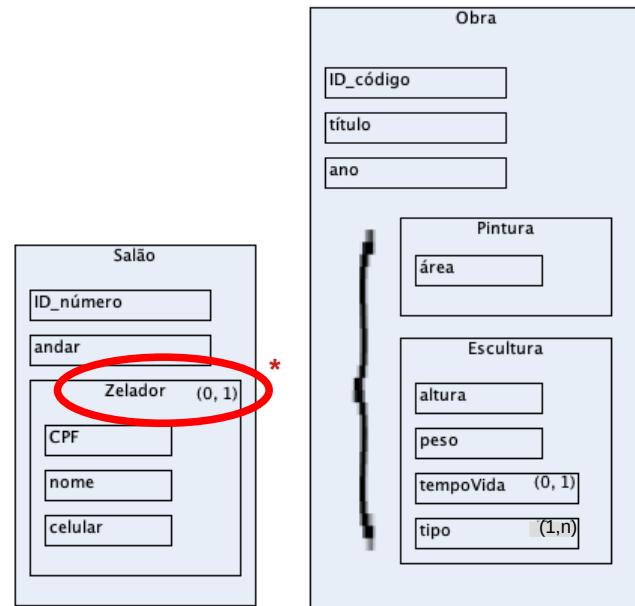
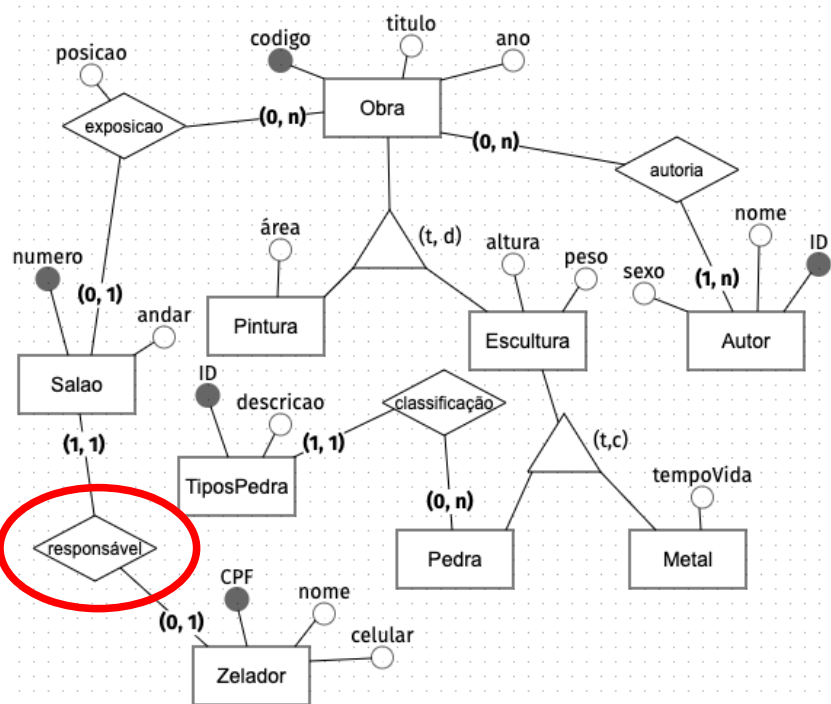
possível alternativa



aninhamento

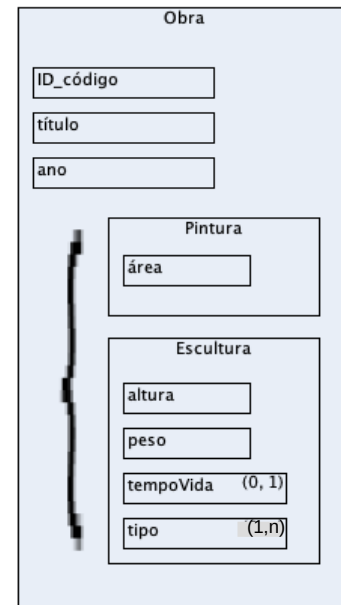
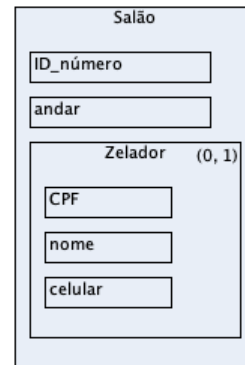
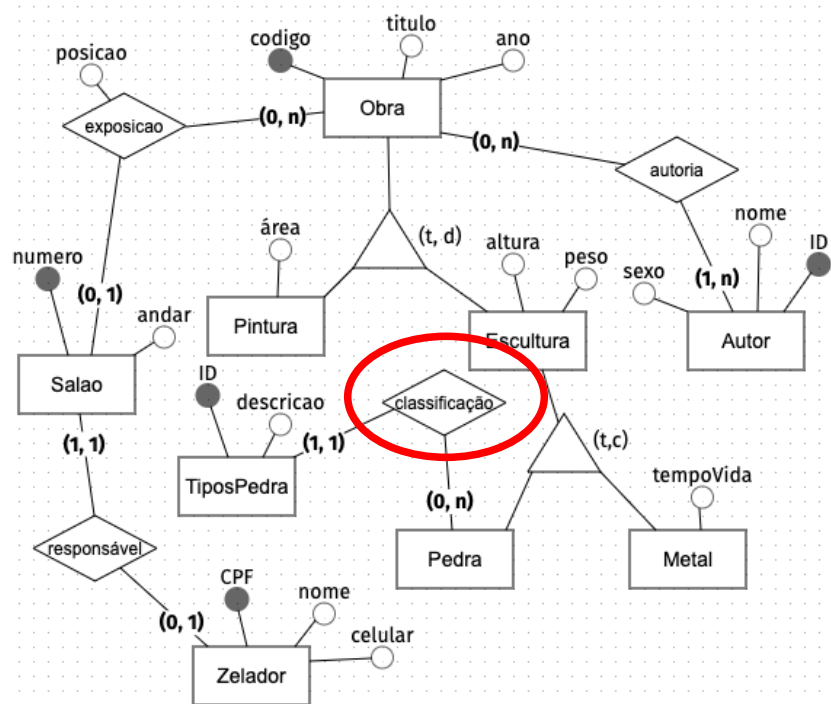


## Passo 2 – Conversão de Relacionamentos

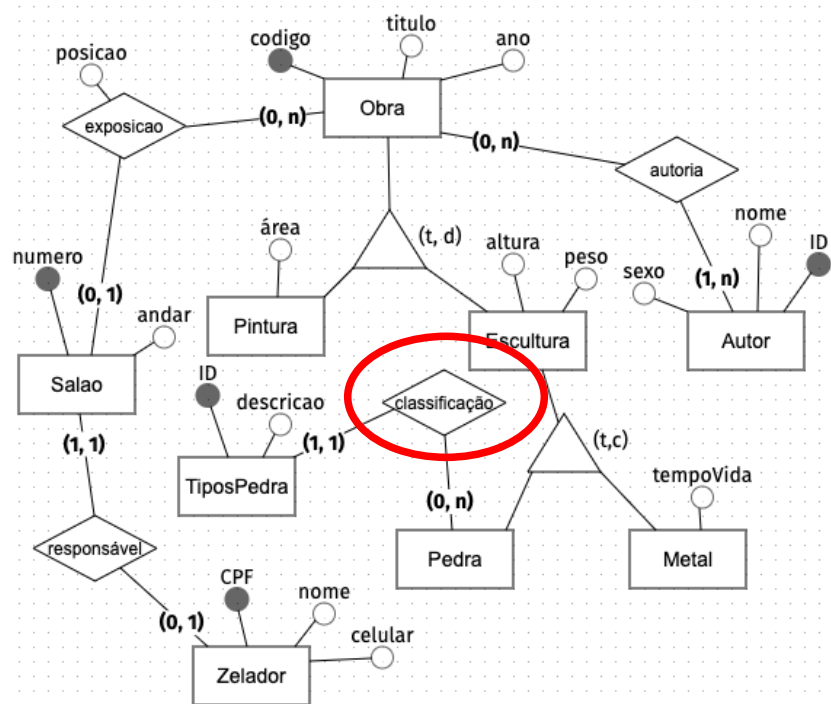


\* **RI**: zelador deve ser único em todo o conjunto de salões

## Passo 2 – Conversão de Relacionamentos



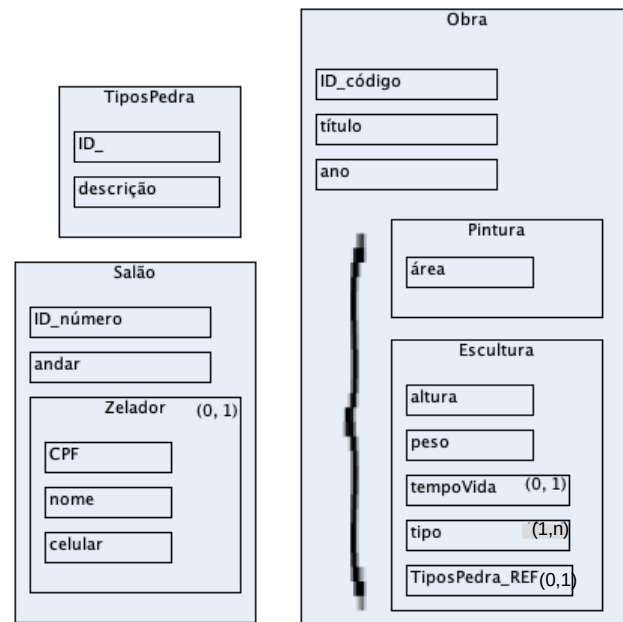
## Passo 2 – Conversão de Relacionamentos



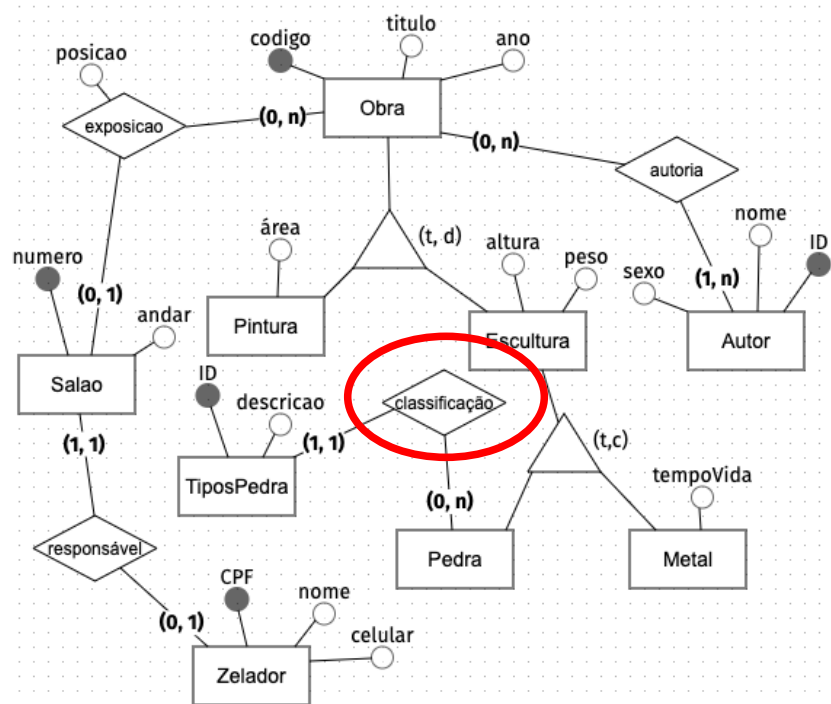
possível alternativa



referência



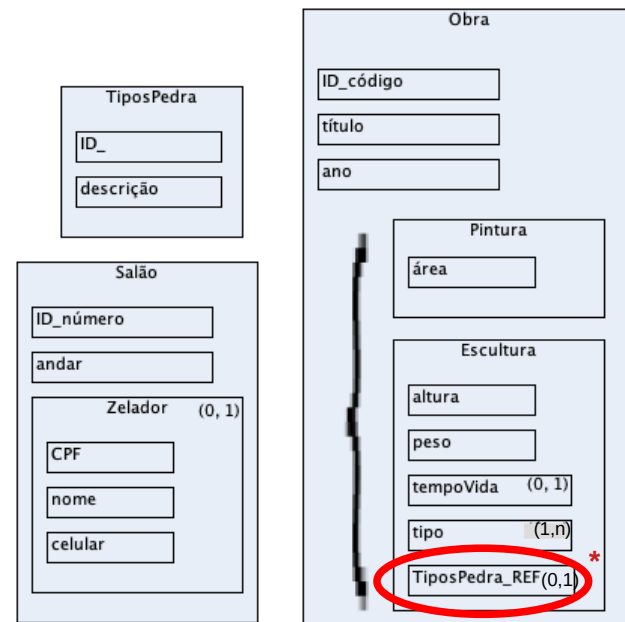
## Passo 2 – Conversão de Relacionamentos



possível alternativa



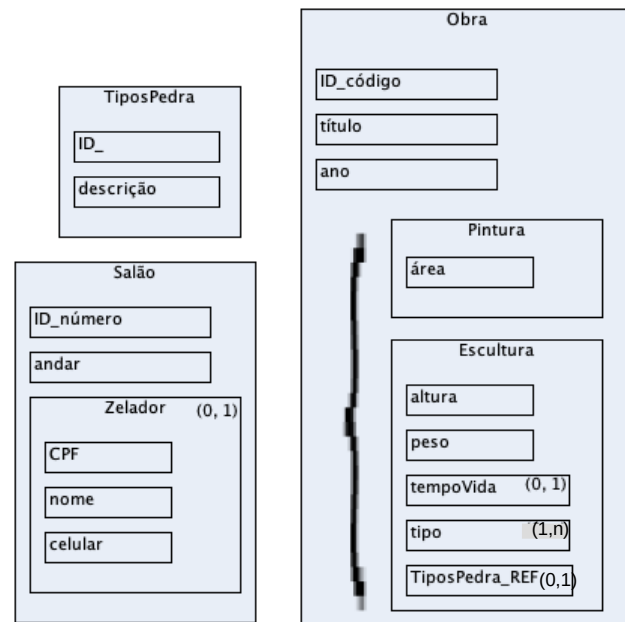
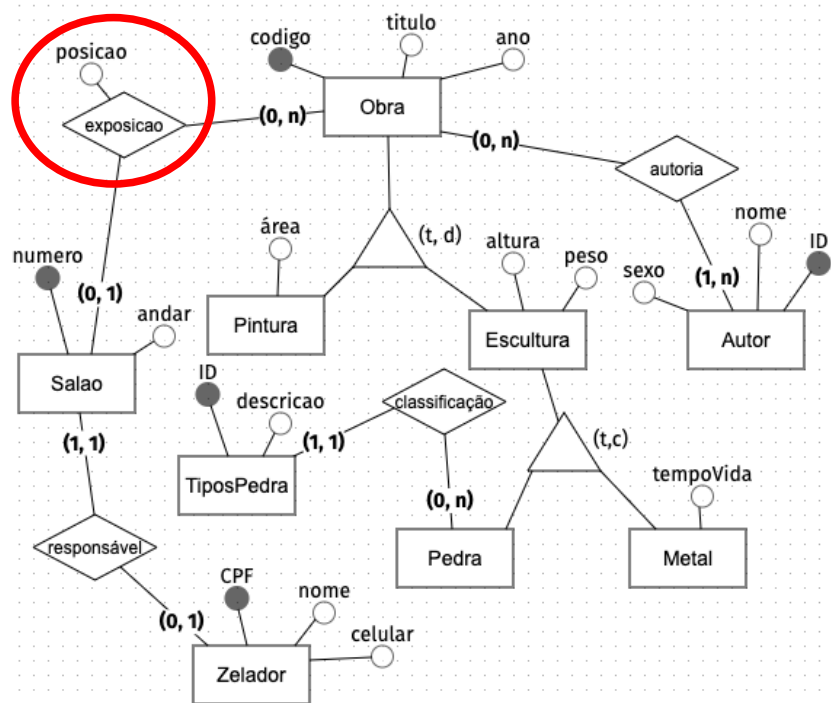
referência



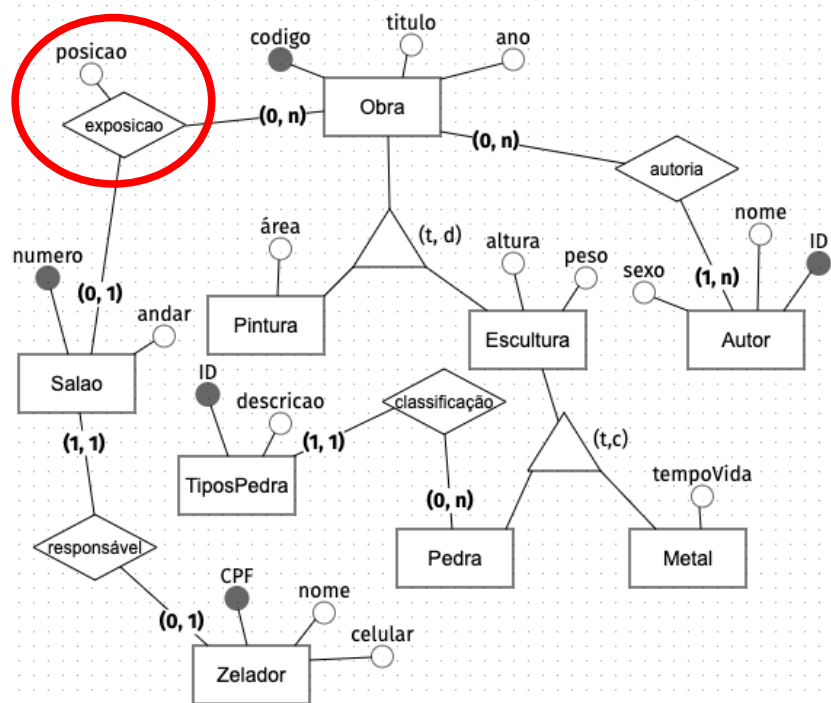
*\* RI: TiposPedra\_REF é obrigatória quando  $\text{tipo} \subseteq \text{'Pedra'}$*



## Passo 2 – Conversão de Relacionamentos



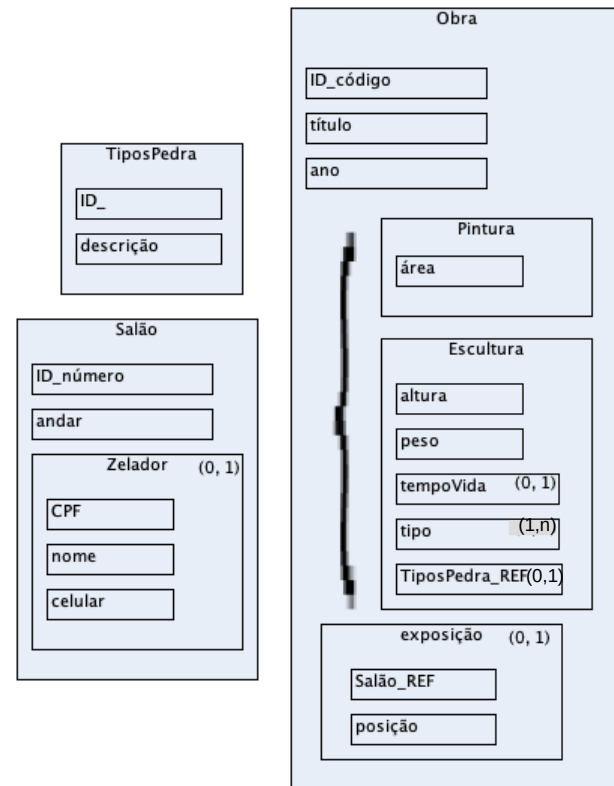
## Passo 2 – Conversão de Relacionamentos



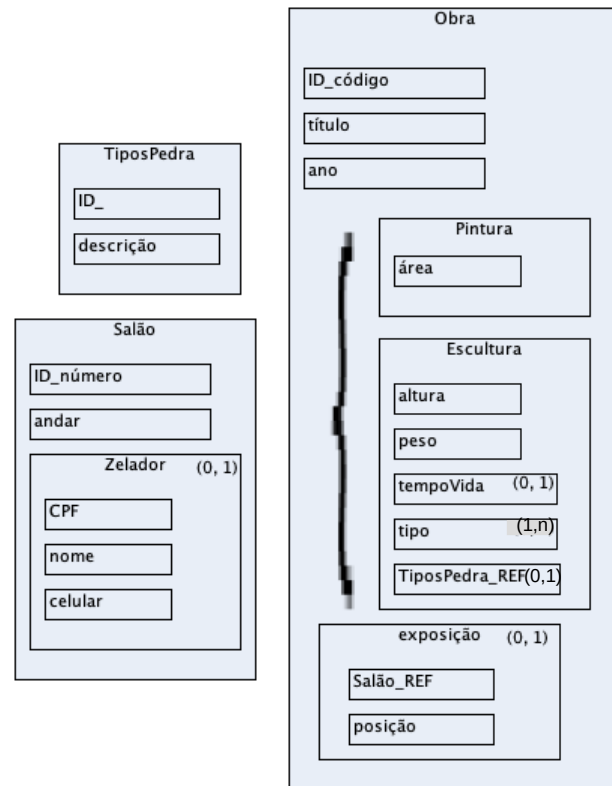
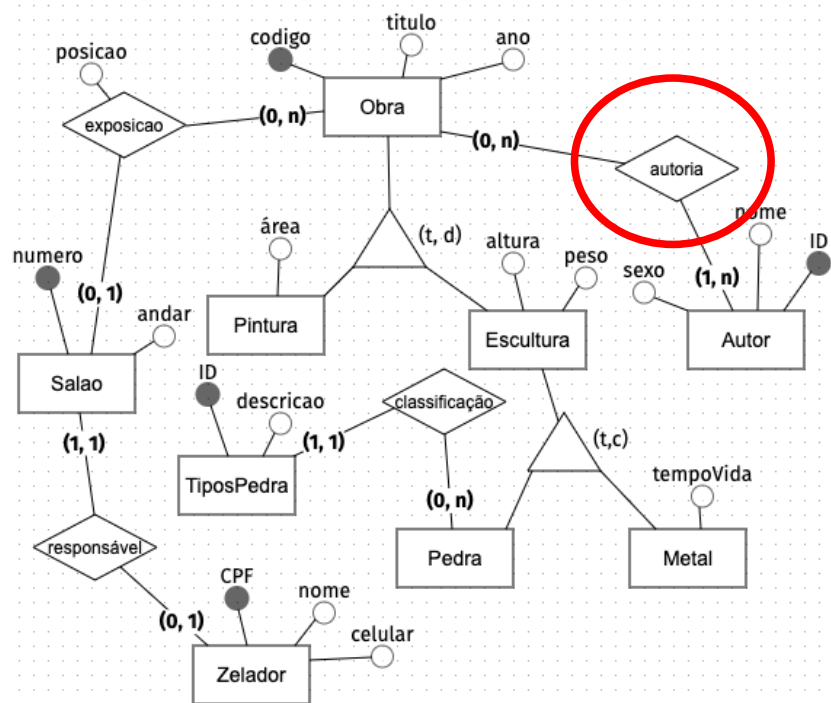
possível alternativa



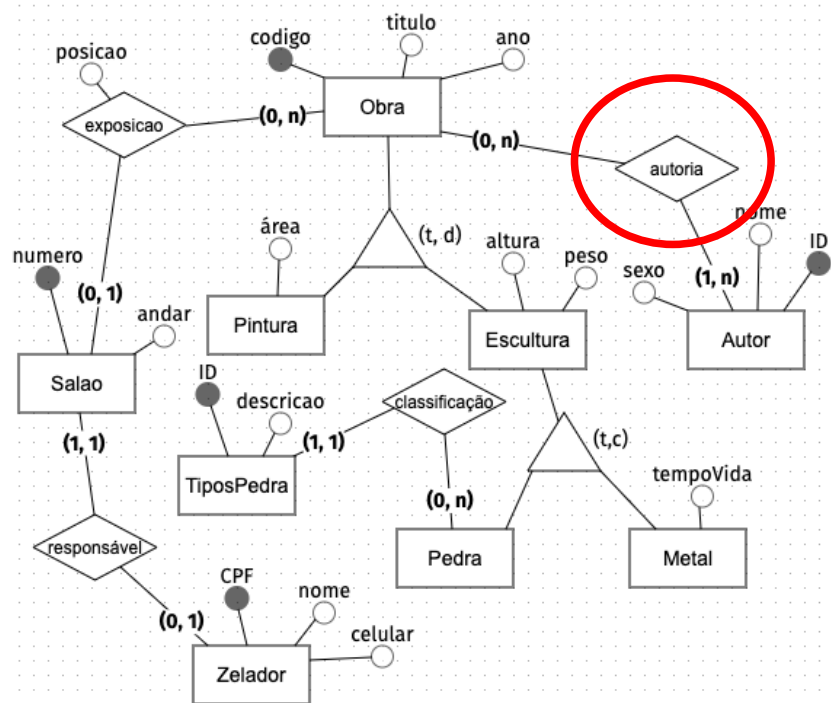
referência



## Passo 2 – Conversão de Relacionamentos



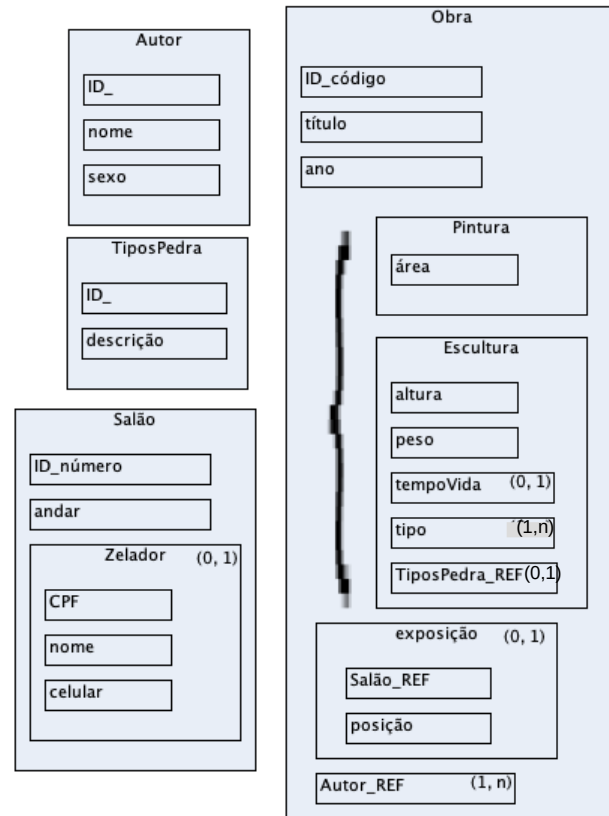
## Passo 2 – Conversão de Relacionamentos



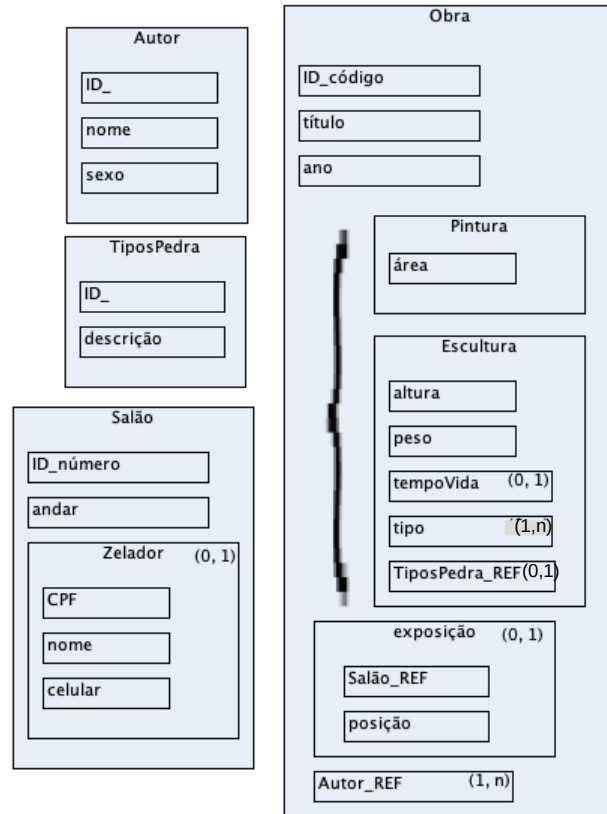
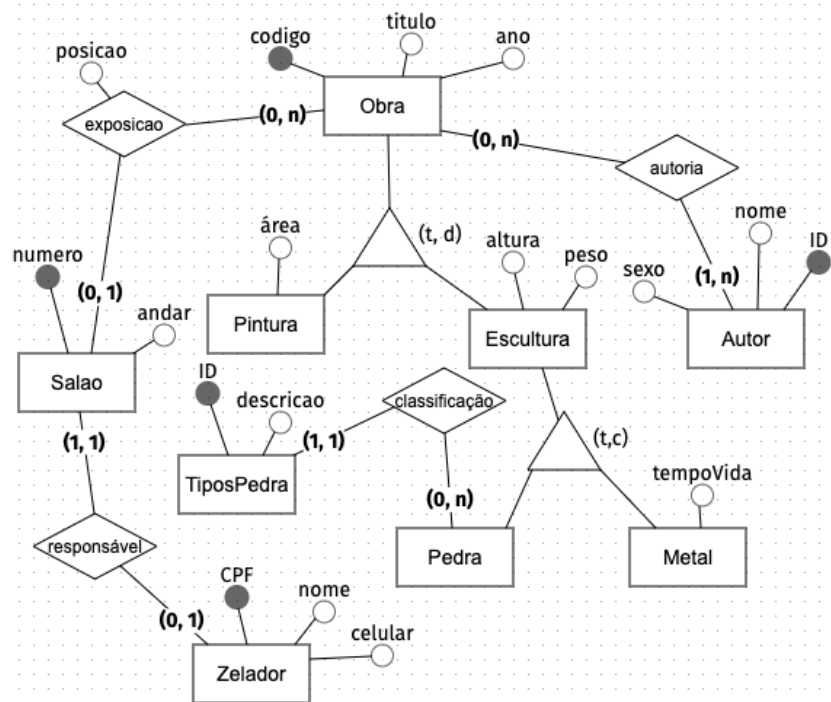
possível alternativa



referência



# (possível) Conversão Concluída!



# Agenda

---

1. Projeto de BD
2. Projeto Lógico de Alto Nível para BD NoSQL Orientado a Agregados
- 3. Projeto Lógico de Baixo Nível para BD NoSQL Orientado a Agregados**
4. Projeto Lógico para BD NoSQL Orientado a Grafos

# Projeto Lógico de Baixo Nível

---

- **Objetivo**

- Conversão do **esquema lógico de alto nível baseado em agregados** para **esquemas representados nos modelos chave-valor, orientado a colunas e orientado a documentos**

- **Esses 3 modelos de dados NoSQL**

- São centrados em acesso via chave
- Podem manter um conteúdo com estruturas aninhadas

# Projeto Lógico - **BD Chave-Valor**

Modelo de agregados	Modelo chave-valor
coleção	esquema chave-valor
ID da coleção	chave
conteúdo da coleção	valor *

\* conteúdos estruturados devem ser desaninhados e serializados



# Exemplo

**Obra**

ID\_código

título

ano

**Pintura**

área

**Escultura**

altura

peso

tempoVida (0, 1)

tipo (1,n)

TiposPedra\_REF(0,1)

**exposição (0, 1)**

Salão\_REF

posição

Autor\_REF (1, n)

**Salão**

ID\_número

andar

horários (1, n)



**Chave:** O145

**Valor:** {título: 'Abaporu'; ...; Pintura.área: 15;...; exposição.Salão\_REF: S2; ...; Autor\_REF: A34}

**Chave:** S2

**Valor:** {andar: 1; horários: '8:00','14:00','19:00'}

## Projeto Lógico - **BD Orientado a Colunas**

Modelo de agregados	Modelo de colunas
coleção	família de colunas
ID da coleção	chave da família de colunas
atributo simples	coluna
atributo de referência	coluna
atributo multivalorado	coluna multivalorada ou coluna com conteúdo serializado
bloco	supercoluna ou conteúdo desaninhado e serializado

# Exemplo

**Obra**

ID\_código

título

ano

**Pintura**

área

**Escultura**

altura

peso

tempoVida (0, 1)

tipo (1,n)

TiposPedra\_REF(0,1)

**exposição (0, 1)**

Salão\_REF

posição

Autor\_REF (1, n)

**Salão**

ID\_número

andar

horários (1, n)



## família de colunas Obra

O145

<b>título</b>	...	<b>exposição</b>	
Abaporu	...	<b>Salão_REF</b>	<b>posição</b>
	...	S2	P38
<b>Autor_REF</b>			
A34			

## família de colunas Salão

S2

<b>andar</b>	<b>horários</b>
1	'8:00','14:00','19:00'

# Projeto Lógico - **BD Orientado a Documentos**

Modelo de agregados	Modelo de documento
coleção	documento
ID da coleção	chave do documento
atributo simples	atributo simples
atributo de referência	atributo simples
atributo multivalorado	atributo do tipo lista
bloco	atributo do tipo objeto

# Exemplo

**Obra**

ID\_código

título

ano

**Pintura**

área

**Escultura**

altura

peso

tempoVida (0, 1)

tipo (1,n)

TiposPedra\_REF(0,1)

**exposição (0, 1)**

Salão\_REF

posição

Autor\_REF (1, n)

**Salão**

ID\_número

andar

horários (1, n)



## Documento Obra

```
{ "_id": "O145",
  "título": "Abaporu", ...
  "exposição": {
    "Salão_REF": "S2",
    "posição": "P38"}, ...
  "Autor_REF": ["A34"] }
```

## Documento Salão

```
{ "_id": "S2",
  "andar": "1",
  "horários": ["8:00", "14:00", "19:00"] }
```

# Agenda

---

1. Projeto de BD
2. Projeto Lógico de Alto Nível para BD NoSQL Orientado a Agregados
3. Projeto Lógico de Baixo Nível para BD NoSQL Orientado a Agregados
- 4. Projeto Lógico para BD NoSQL Orientado a Grafos**

# Projeto Lógico – **BD Orientado a Grafo**

---

- **Modelo orientado a grafos**
  - buscas centradas em relacionamentos ao invés de acessos via chave
- **Esquema conceitual é uma estrutura de grafo**
  - mapeamento direto ER → grafo é mais intuitivo

# Processo de Projeto Lógico

---

- **Carência de uma metodologia detalhada na literatura**
- **Sugestões**
  - adotar a mesma metodologia vista anteriormente
    - 1) **conversão de hierarquias**
    - 2) **conversão de relacionamentos**
  - é possível **fundir** entidades conectadas em uma hierarquia ou relacionadas com cardinalidade 1-1



# Regras de Conversão

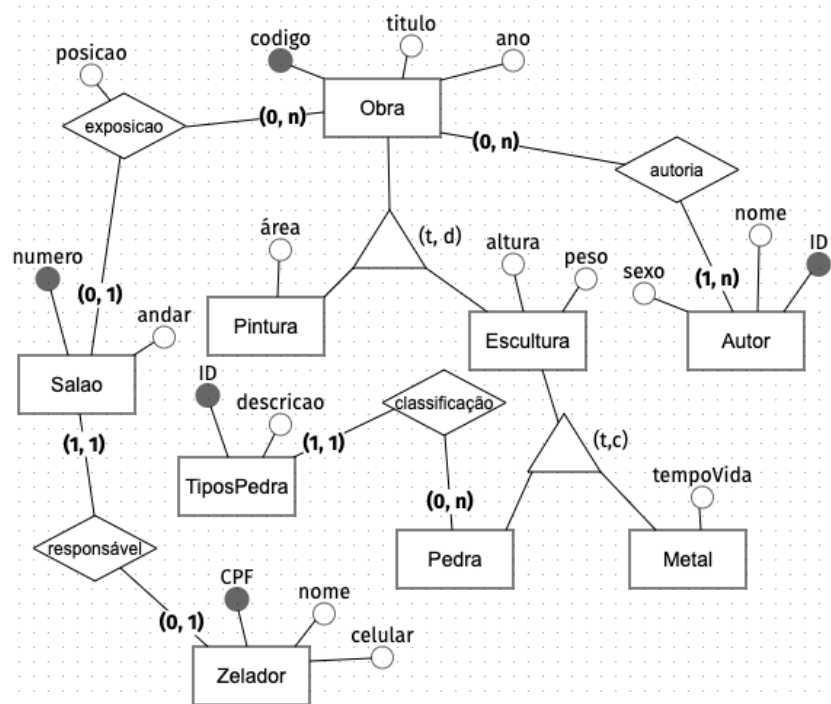
---

ER	Modelo de Grafo
entidade	vértice
atributo simples	propriedade
atributo composto	vértice
atributo multivalorado	propriedade <i>array</i> ou vértice
relacionamento	aresta *
relacionamento n-ário	vértice

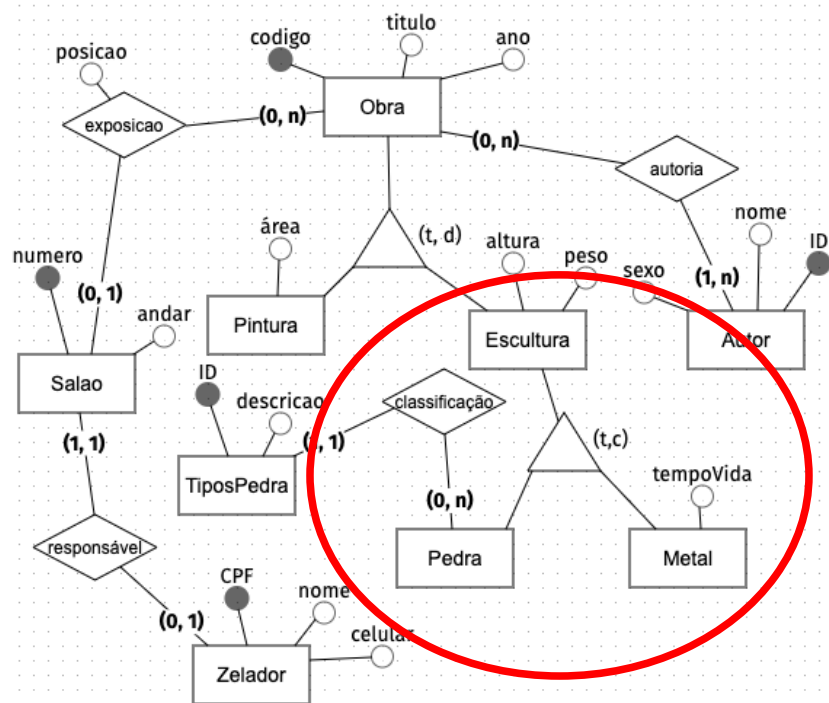
\* pode ser bidirecional dependendo do *workload* típico

\* possui uma cardinalidade associada

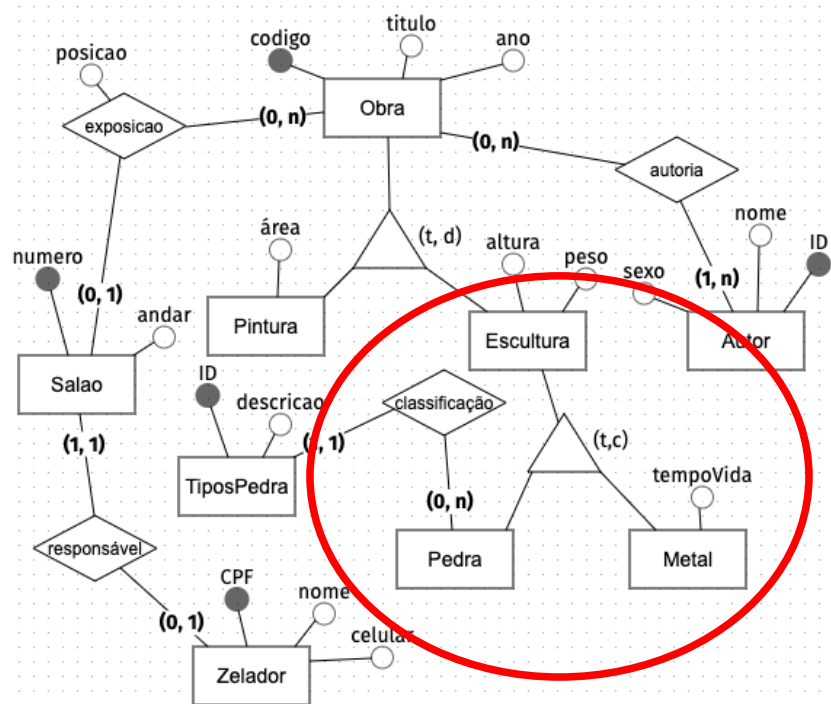
# Running Example - Museu



# Passo 1 – Conversão de Hierarquias



# Passo 1 – Conversão de Hierarquias



possível alternativa

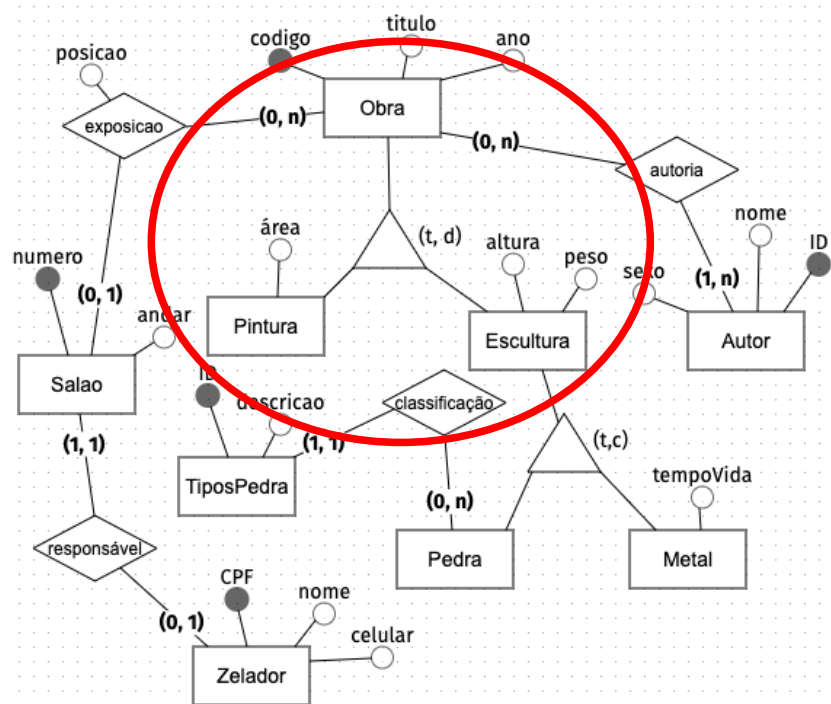


ênfase na entidade  
genérica

Escultura

altura  
peso  
...  
tempoVida  
tipo (1, n)

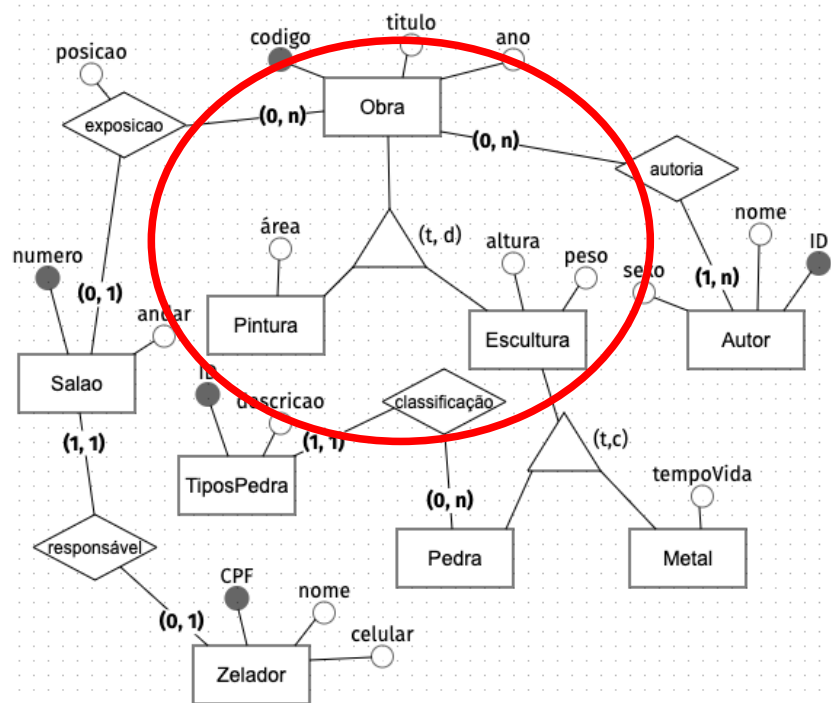
# Passo 1 – Conversão de Hierarquias



Escultura

altura  
peso  
...  
tempoVida  
tipo (1,n)

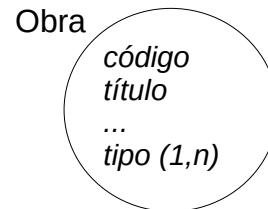
# Passo 1 – Conversão de Hierarquias



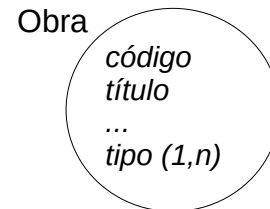
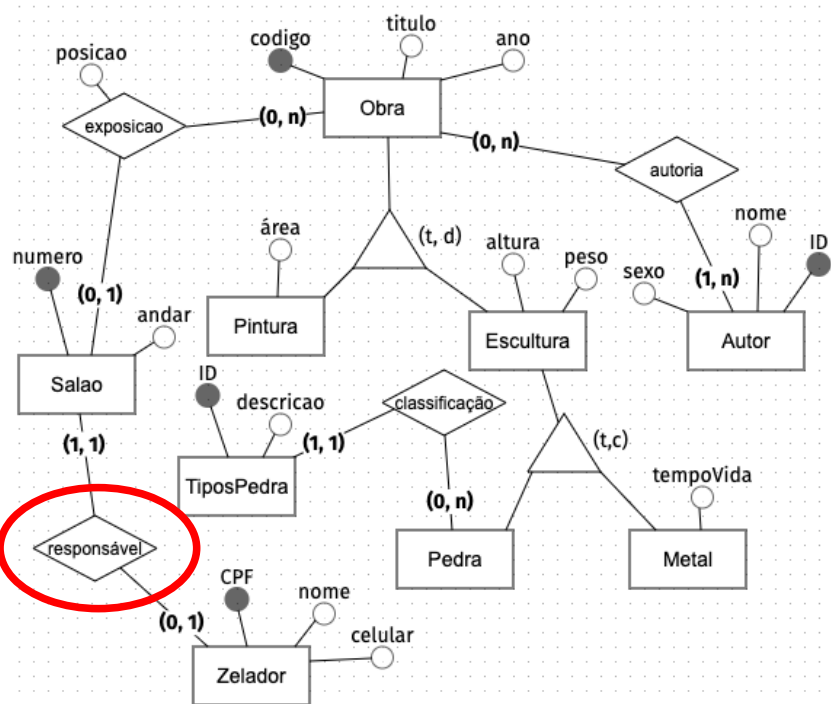
possível alternativa



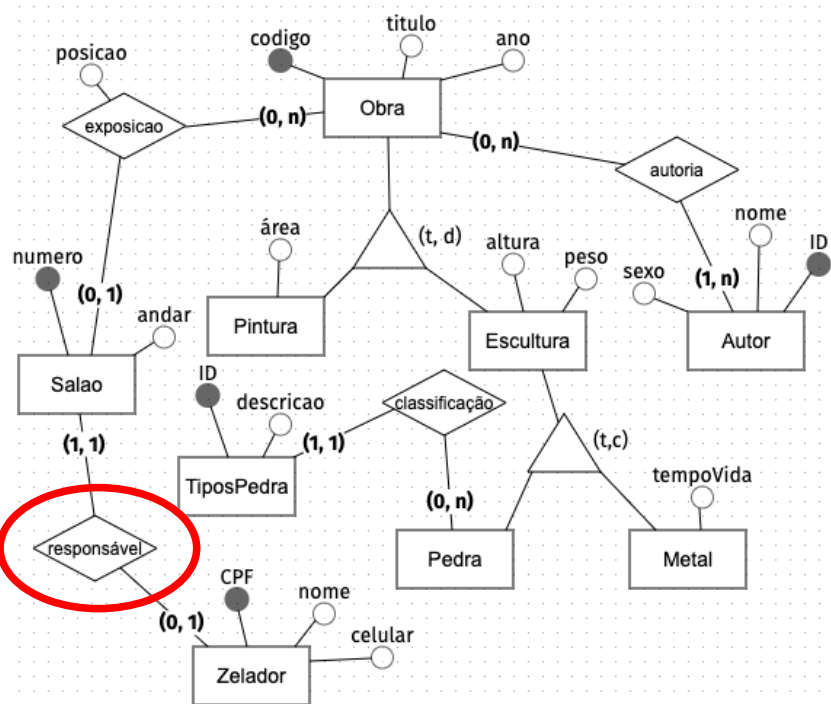
ênfase na entidade  
genérica



## Passo 2 – Conversão de Relacionamentos



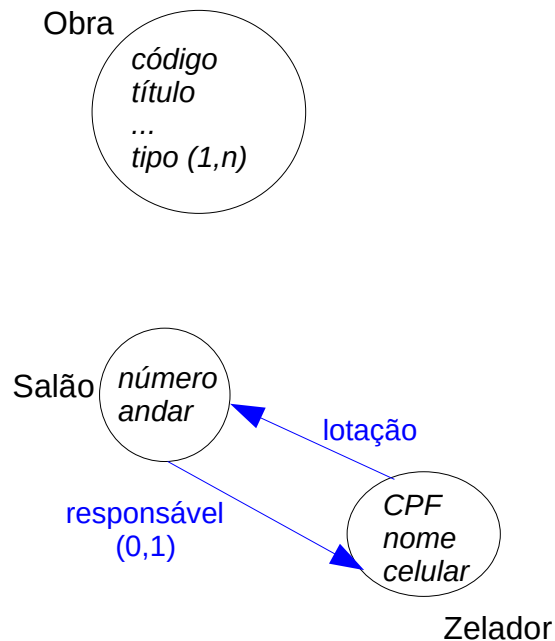
## Passo 2 – Conversão de Relacionamentos



possível alternativa

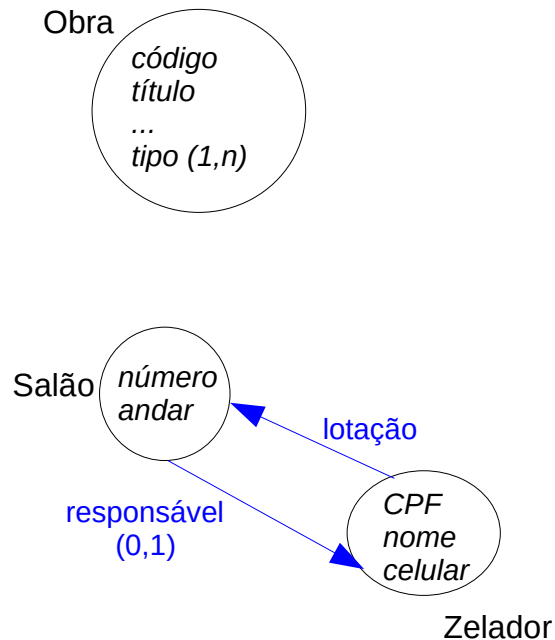
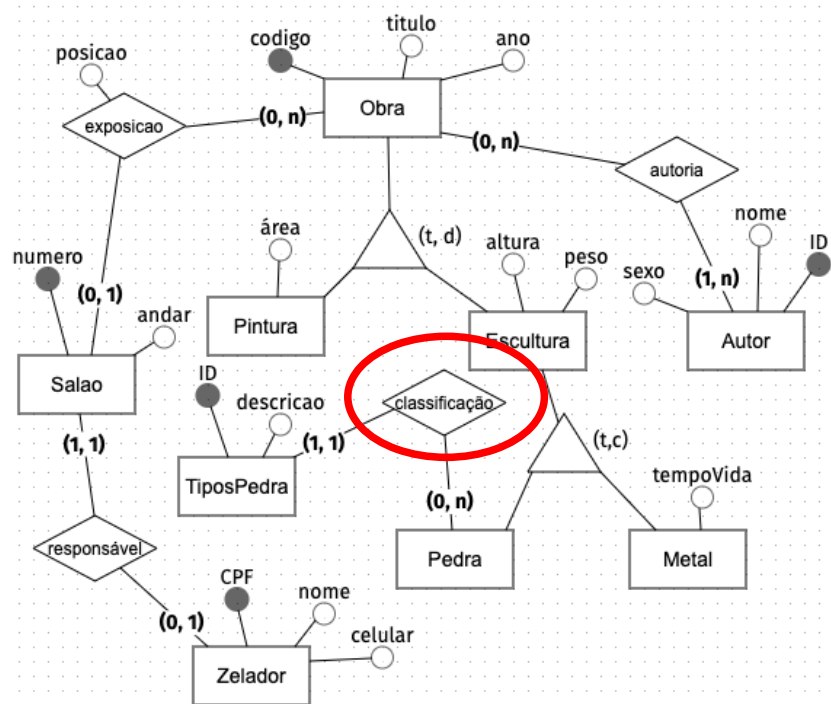


2 vértices + 2 arestas  
(bidirecional)

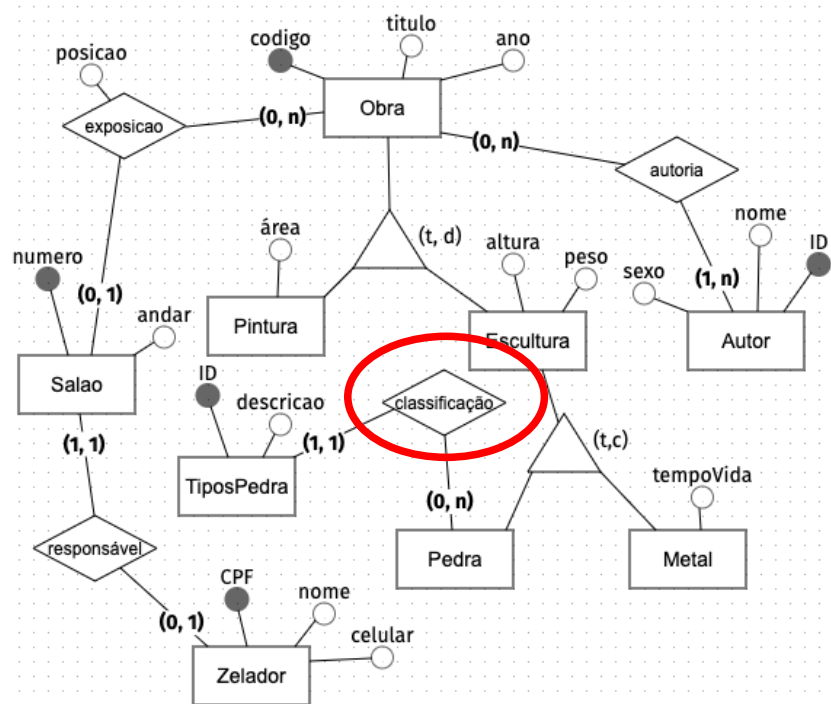




## Passo 2 – Conversão de Relacionamentos



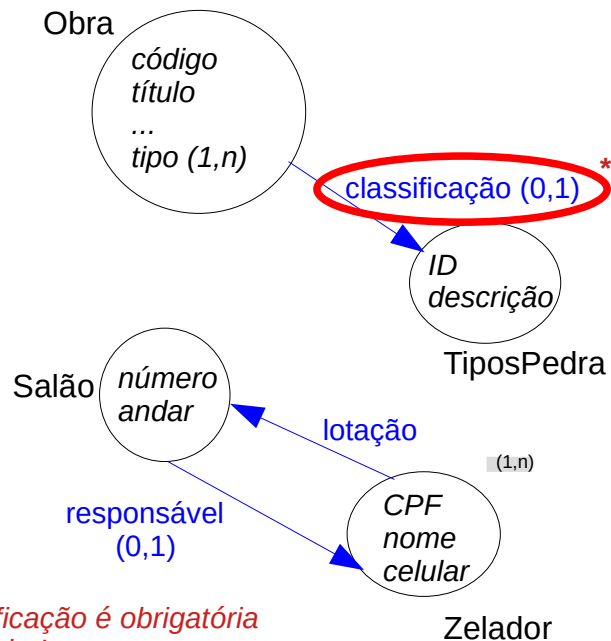
## Passo 2 – Conversão de Relacionamentos



possível alternativa

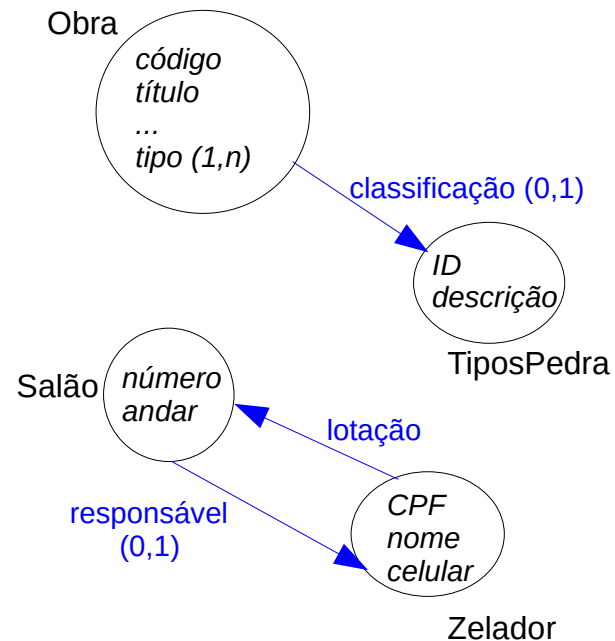
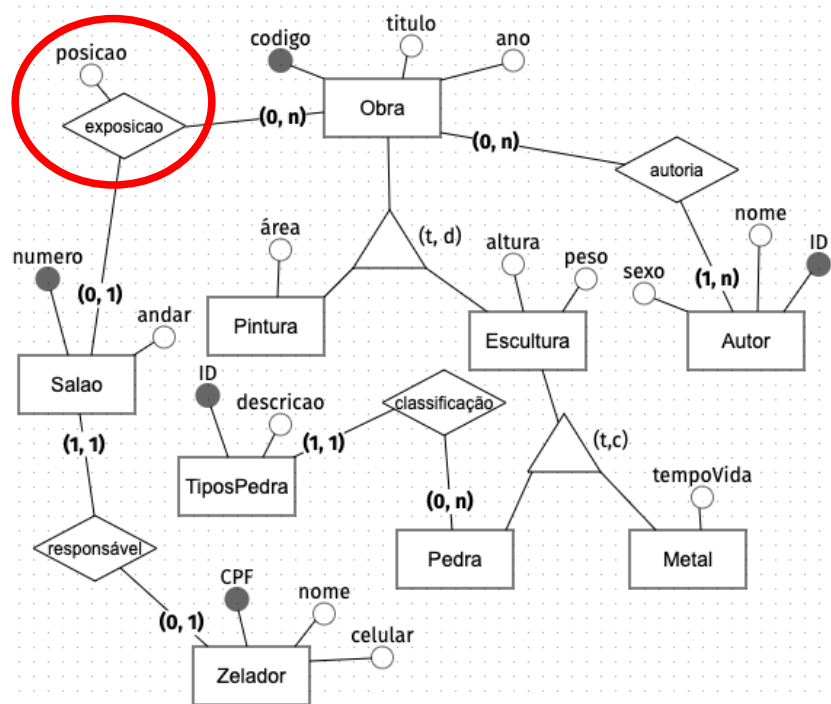


vértice + aresta  
(unidirecional)

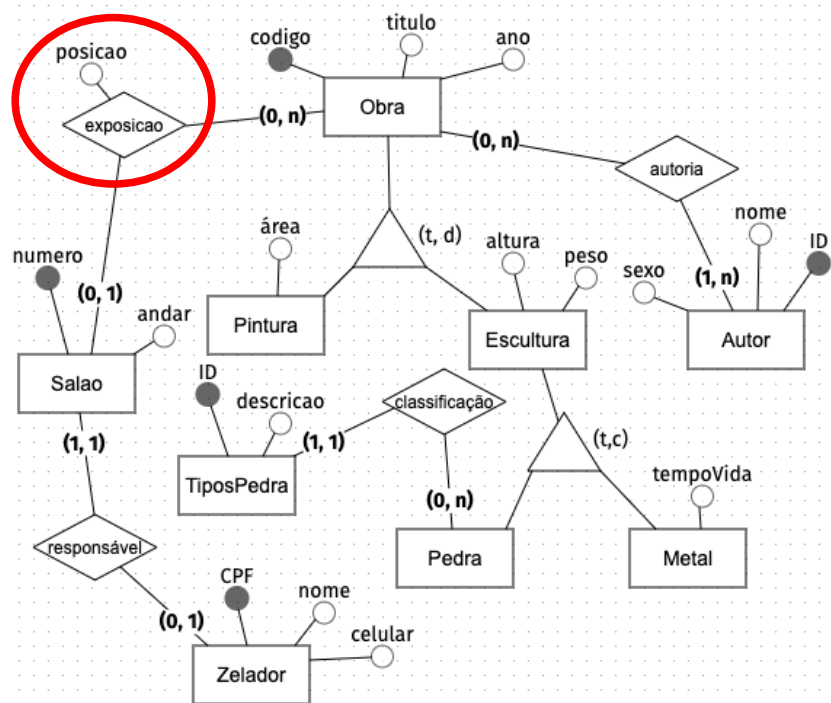


\* RI: aresta classificação é obrigatória  
quando  $\text{tipo} \subseteq \text{'Pedra'}$

## Passo 2 – Conversão de Relacionamentos



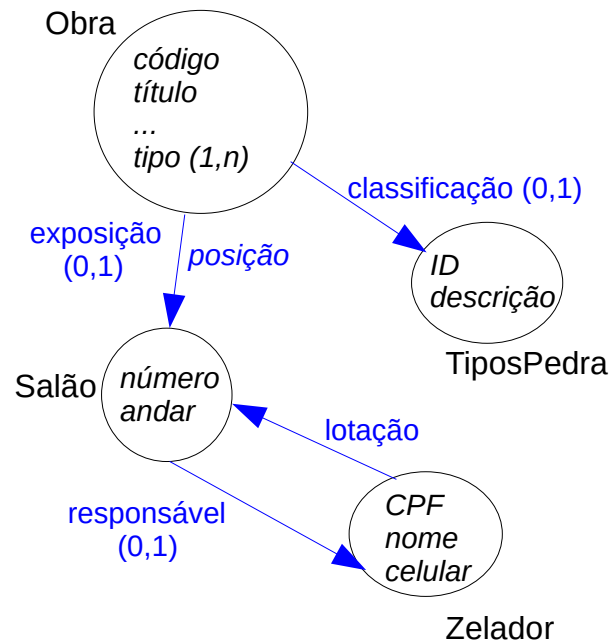
## Passo 2 – Conversão de Relacionamentos



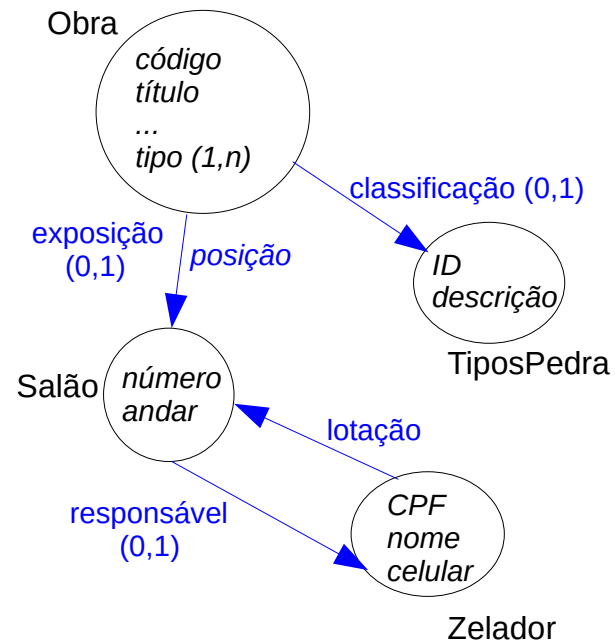
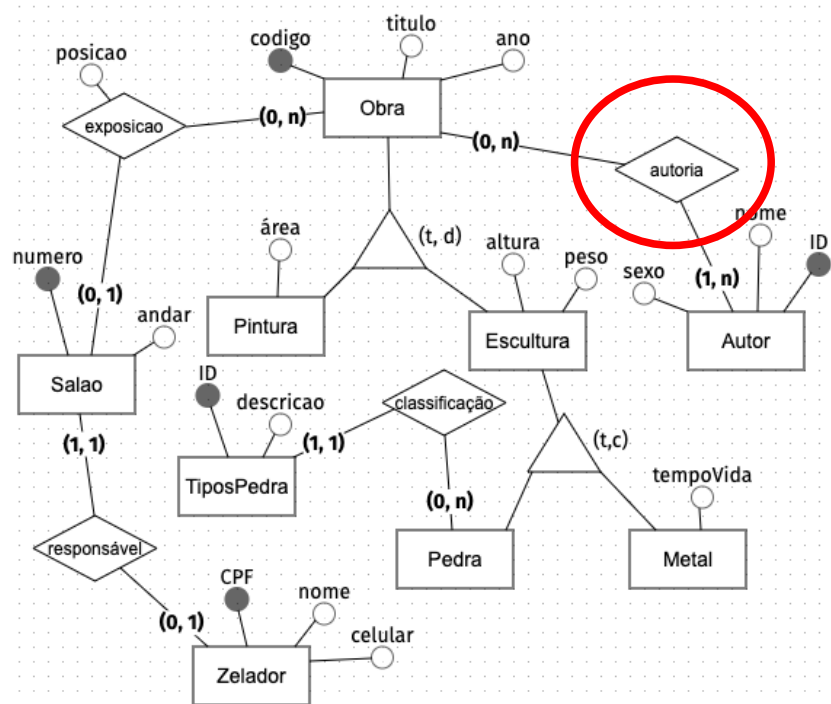
possível alternativa



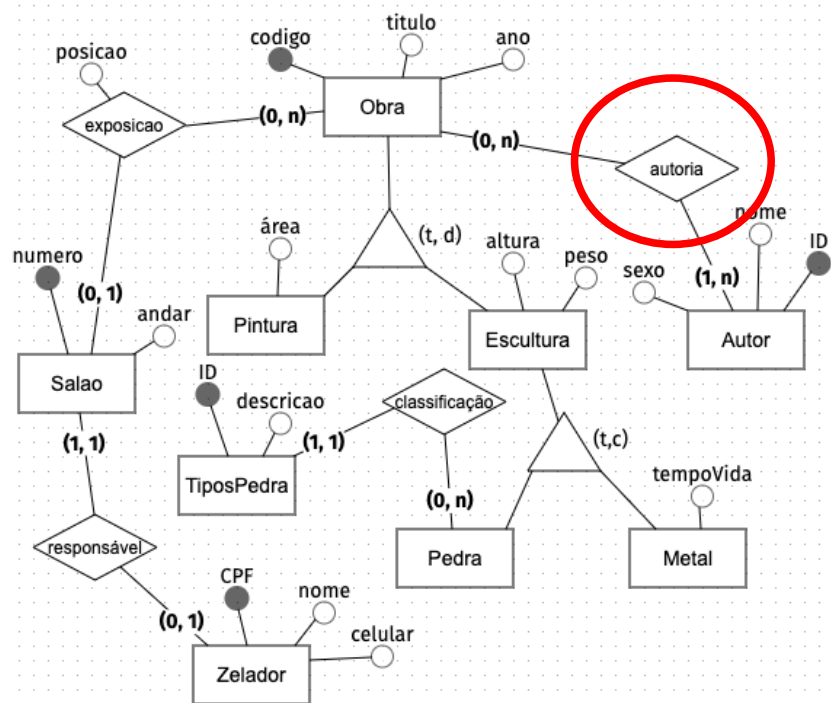
aresta  
(unidirecional e  
com propriedade)



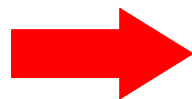
## Passo 2 – Conversão de Relacionamentos



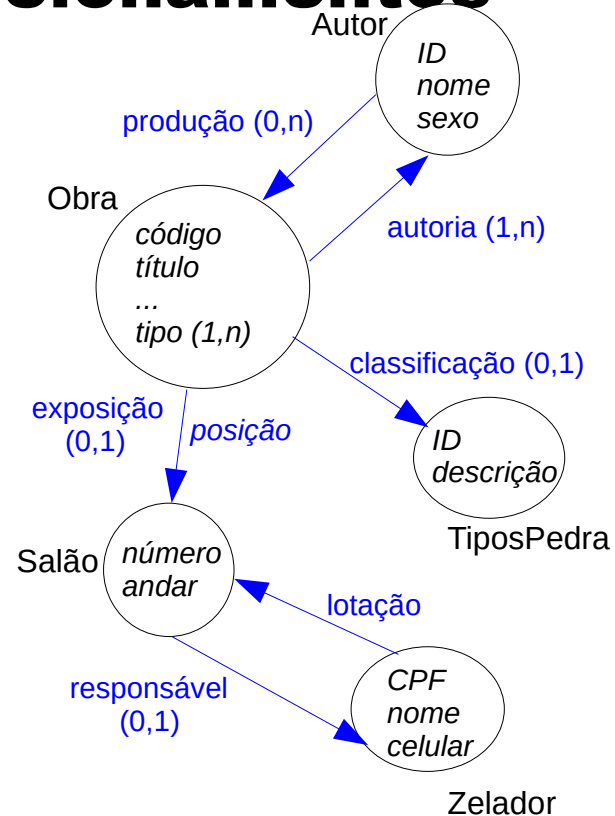
## Passo 2 – Conversão de Relacionamentos



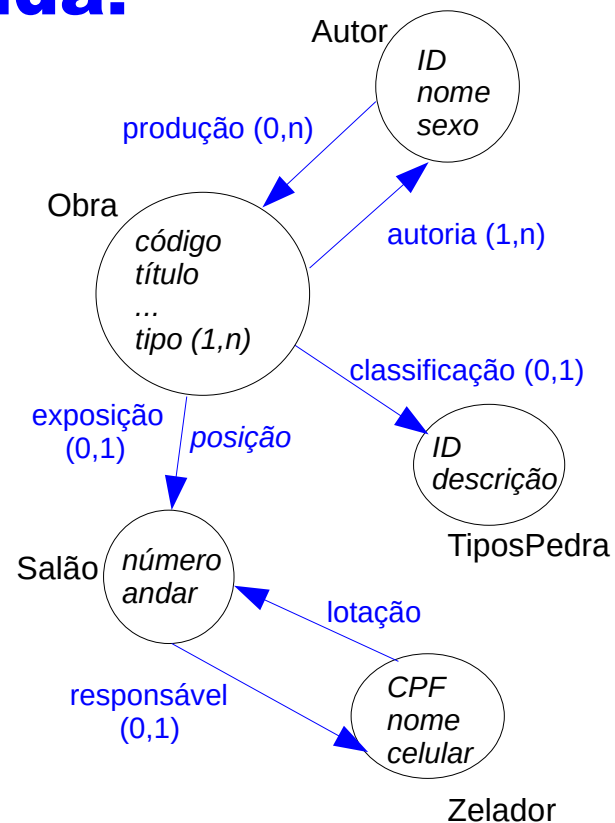
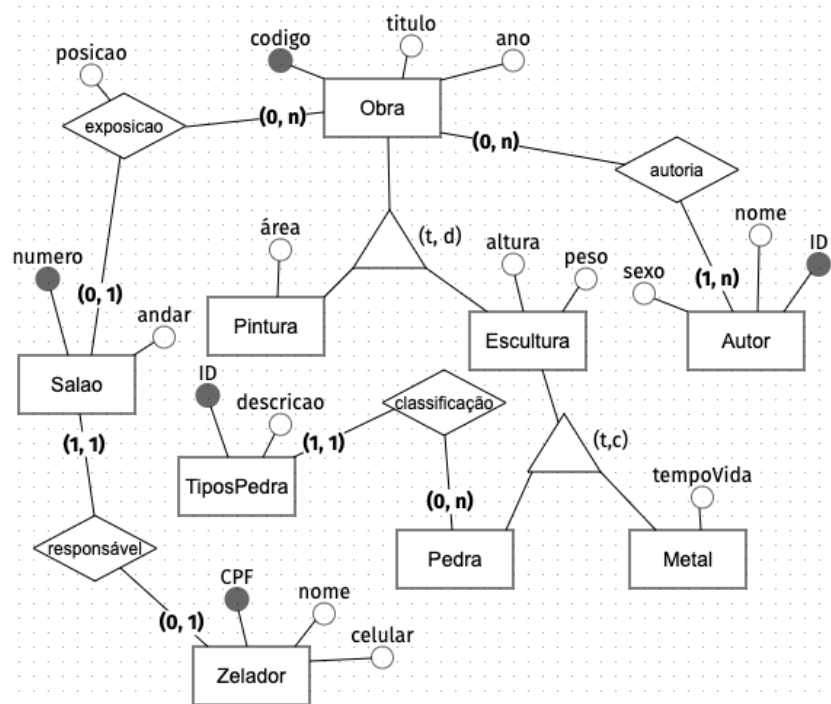
possível alternativa



vértice + 2 arestas  
(bidirecional)



# (possível) Conversão Concluída!



# Atividade 2 – Projeto NoSQL

Proponha um projeto lógico NoSQL baseado em agregados e um projeto lógico para BD NoSQL grafo, para o domínio abaixo de um **Zoológico**.

