

UFSC-CTC-INE-PPGCC

INE 410131 – Gerência de Dados para Big Data

Aula 5 – BD NoSQL:
Teoremas / Projeto Lógico

UFSC-CTC-INE-PPGCC

INE 410131 – Gerência de Dados para Big Data

Aula 5 – Teoremas:

Teorema CAP

Propriedades BASE

Teorema PACELC

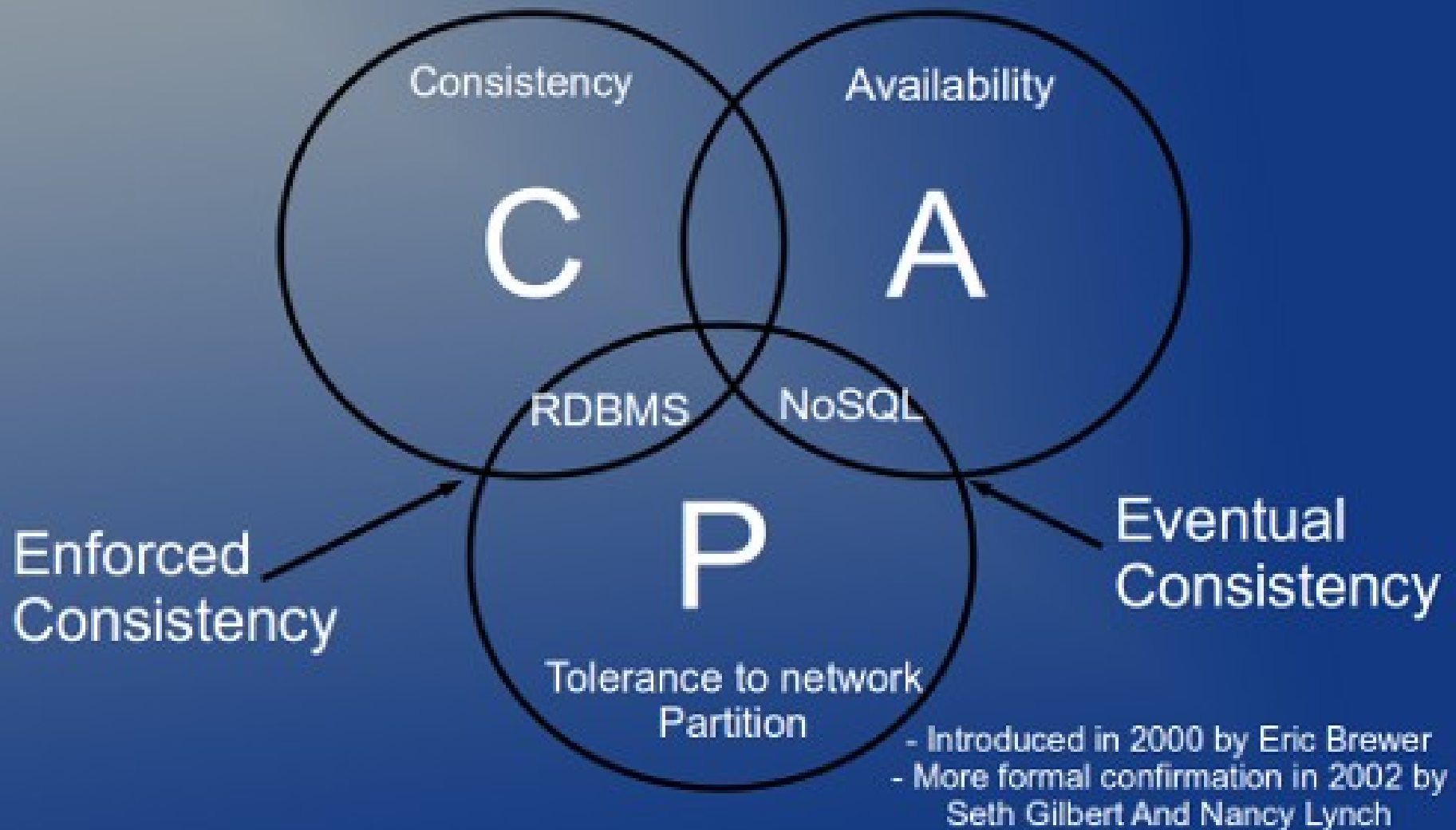
Teorema CAP

- CAP = Consistency, Availability, Partition Tolerance
 - Consistency: operações R sobre um dado d_i devem retornar o resultado da operação W mais recente realizada sobre d_i
 - Availability: toda operação de R / W deve ser realizada com sucesso, ou seja, não retornar erro
 - Partition Tolerance: o sistema deve continuar operando mesmo na ocorrência de falhas que particionem a rede e causem perda ou atraso no envio de mensagens
- Teorema aplicado a sistemas distribuídos

Teorema CAP

- Impossível garantir as 3 propriedades simultaneamente
 - CA: não é possível imaginar que P ocorra (em função de falhas na rede) se quero ter consistência e disponibilidade imediatas
 - CP: não é possível ter disponibilidade contínua (A) se tenho que gerenciar consistência e tolerância a partições na rede
 - AP: não é possível ter consistência imediata (C) se quero ter alta disponibilidade e lidar com problemas de particionamento da rede

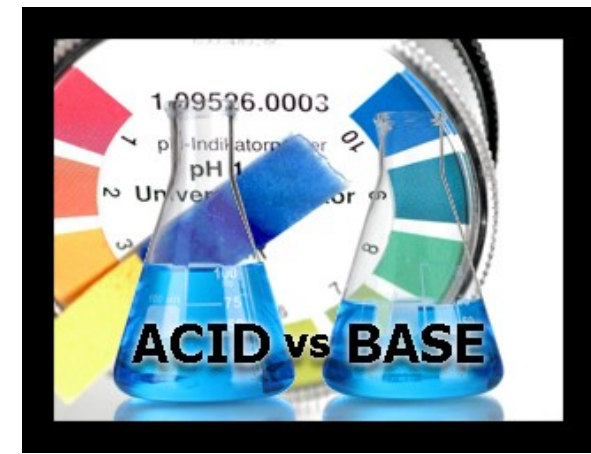
CAP Theorem



Propriedades BASE

- **BASE** = *BA*sically available + *S*oft state + *E*ventually consistent
 - Basically Available: o BD deve estar disponível para *R/W* a maior parte do tempo possível
 - Soft state: o BD não necessita estar consistente todo o tempo, incluindo as diversas réplicas de um mesmo dado
 - Eventually consistent: dados eventualmente inconsistentes estarão consistentes em algum momento futuro
- Propriedades adotadas pelos **BDs NoSQL**
 - Priorizam disponibilidade em detrimento da consistência

BASE vs. ACID



- BASE é **conflitante** com o **ACID** dos BDRs distribuídos
 - ACID introduz um **overhead** no processamento de transações distribuídas para ser garantido
- Mas como os BDs NoSQL garantem a consistência global adotando o BASE?
 - Protocolos de controle de concorrência **Multiversão**
 - Protocolos de **reconciliação de réplicas** em 2 etapas
 1. Intercâmbio de versões de dados atualizados entre os nodos que possuem esses dados
 2. Consenso sobre qual versão será considerada o estado final do dado (última atualização, transação com maior prioridade, ...)
 - Protocolos de reconciliação são ativados quando o SGBD detecta (periodicamente) inconsistências nos dados atualizados ou quando um próximo R/W vai ser executado no dado

Teorema PACELC

- Evolução do teorema CAP
 - Afirma que todo sistema distribuído deve ser capaz de gerenciar particionamento de rede (P) para funcionar corretamente
 - Desta forma, um SGBD nas nuvens em situação de particionamento da rede (P) deve escolher entre (A) e (C) ou, caso contrário (else – (E)), se ele estiver executando na ausência de particionamento na rede, ele deve escolher entre baixa latência (L) ou consistência (C)
- SGBDs com foco em baixa latência (L) priorizam a disponibilidade (AP)
 - Maioria dos BDs NoSQL
- SGBDs com foco em consistência (C) são CP
 - Tentam resolver o mais breve possível inconsistências (alguns BDs NoSQL) ou impedir inconsistências (BDs NewSQL)

Teorema PACELC e SGBDs nas Nuvens

DDBS	P+A	P+C	E+L	E+C
DynamoDB	Yes		Yes ^[a]	
Cassandra	Yes		Yes ^[a]	
Cosmos DB	Yes		Yes	
Riak	Yes		Yes ^[a]	
VoltDB/H-Store		Yes		Yes
Megastore		Yes		Yes
BigTable/HBase		Yes		Yes
MongoDB	Yes			Yes
PNUTS		Yes	Yes	

UFSC-CTC-INE-PPGCC

INE 410131 – Gerência de Dados para Big Data

**Aula 5 – BDs NoSQL:
Projeto Lógico**

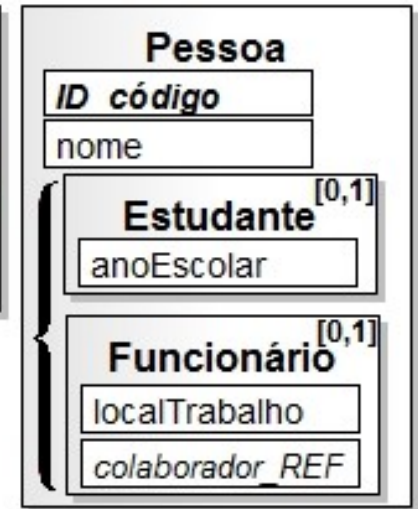
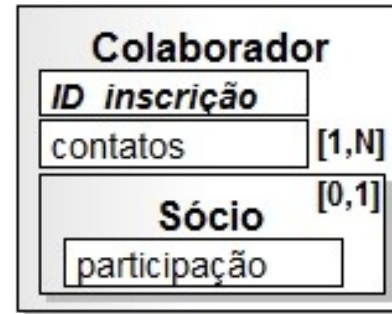
Projeto Lógico de BDs NoSQL

- Tema de pesquisa atual
- Modelagem de BDs NoSQL é um tópico pouco explorado na academia e na indústria
 - nem todo BD NoSQL exige um esquema para os dados, porém...
 - a existência de um esquema requer decisões de projeto para evitar desempenho ruim no acesso à *Big Data*
- Conceito de **Agregado** (Sadalage & Fowler, 2012)
 - base para a modelagem lógica de BDs de documento, colunar e chave-valor
 - representação de dados relacionados de forma (preferencialmente) aninhada

Modelo Lógico de Agregados

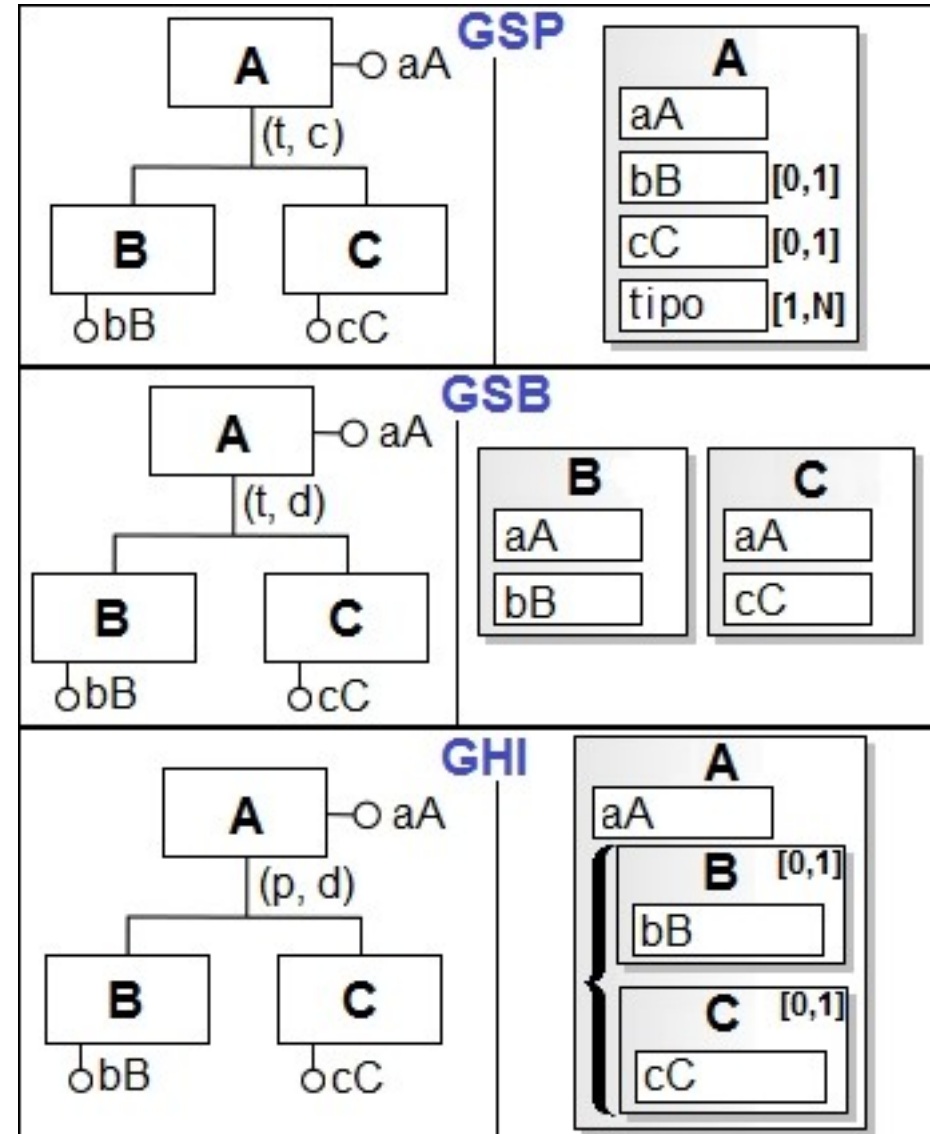
- Conceitos básicos

- coleções, blocos e atributos
- um esquema de BD possui uma ou mais coleções
- uma coleção é composta por um bloco raiz (unidade de acesso)
- um bloco raiz é composto por um ID e um conjunto não-vazio de atributos e/ou blocos aninhados (mono ou multivalorados)
- tipos de relacionamento
 - Hierárquicos (blocos aninhados) (pode haver disjunção)
 - Referência (atributo)



Mapeamento Conceitual-Lógico

- Três alternativas para a conversão de generalizações
 - GSP** – Ênfase na **Superclasse**
 - GSB** – Ênfase nas **Subclasses**
 - GHI** – Ênfase na **Hierarquia**
- Decisão por uma das alternativas deve considerar
 - tamanho da porção do esquema gerado
 - prioridades de acesso (*workload*)
 - restrições (*totalidade* e *disjunção*)



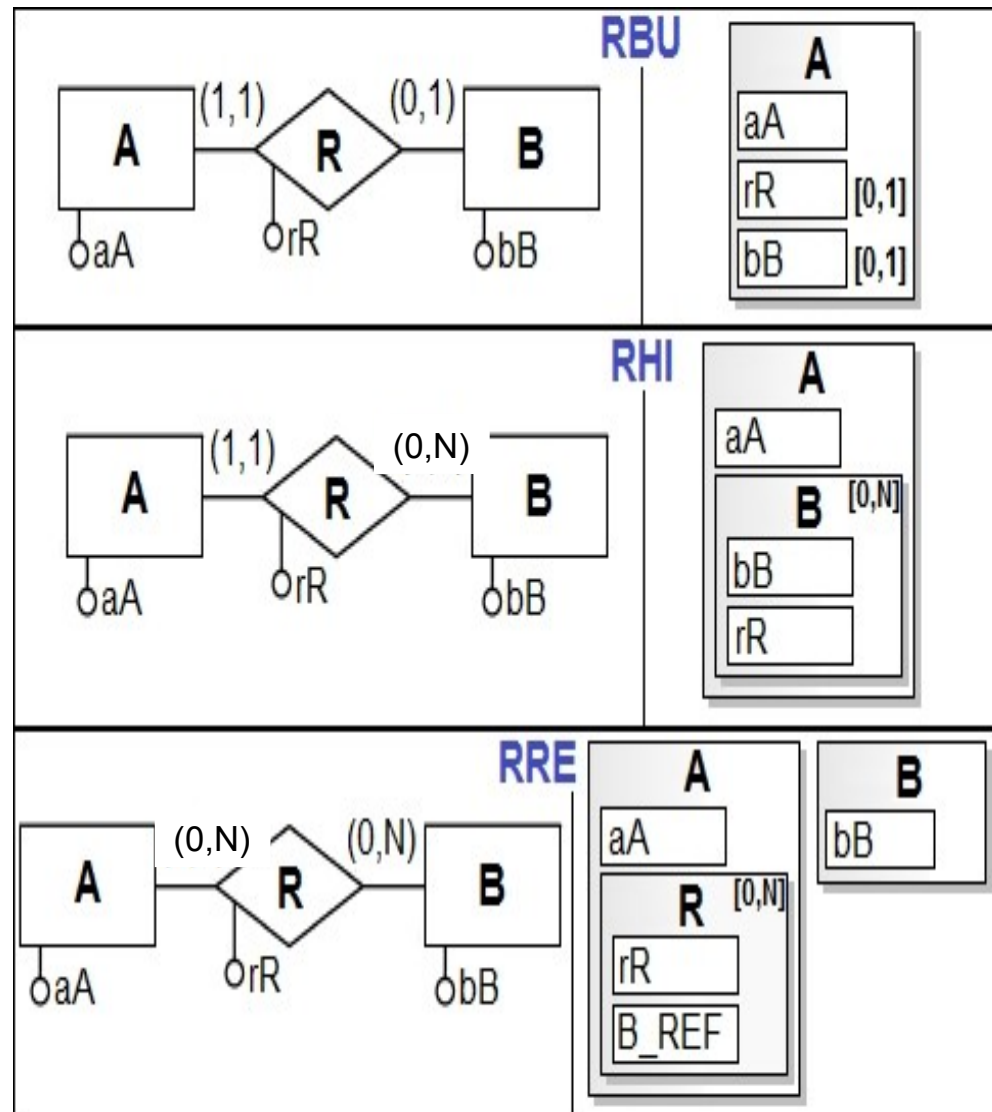
Mapeamento Conceitual-Lógico

- Três alternativas para a conversão de relacionamentos

- RBU** – Modelado por **Bloco Único**
- RHI** – Modelado por **Hierarquia**
- RRE** – Modelado por **Referências**

- Decisão por uma das alternativas deve considerar as cardinalidades do relacionamento e as prioridades de acesso

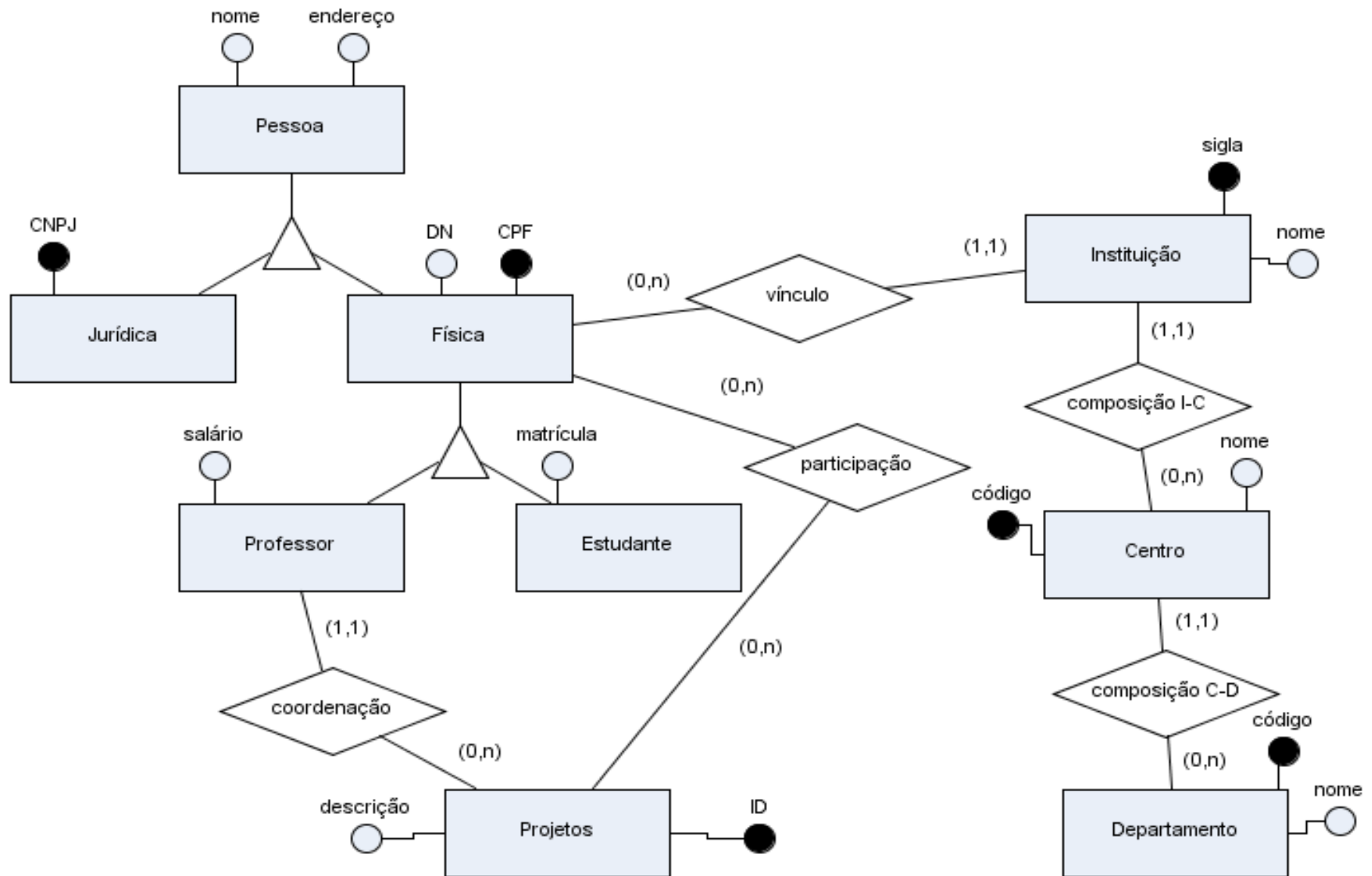
- RBU : (1:1)**
- RHI : (1:N) e (1:1)**
- RRE : (M:N), (1:N) e (1:1)**



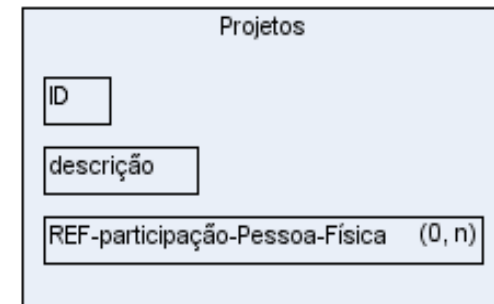
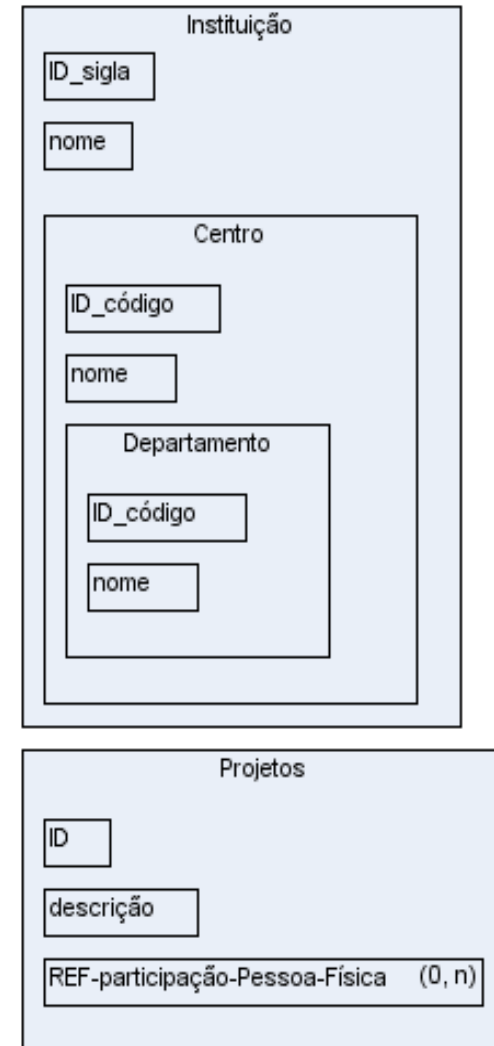
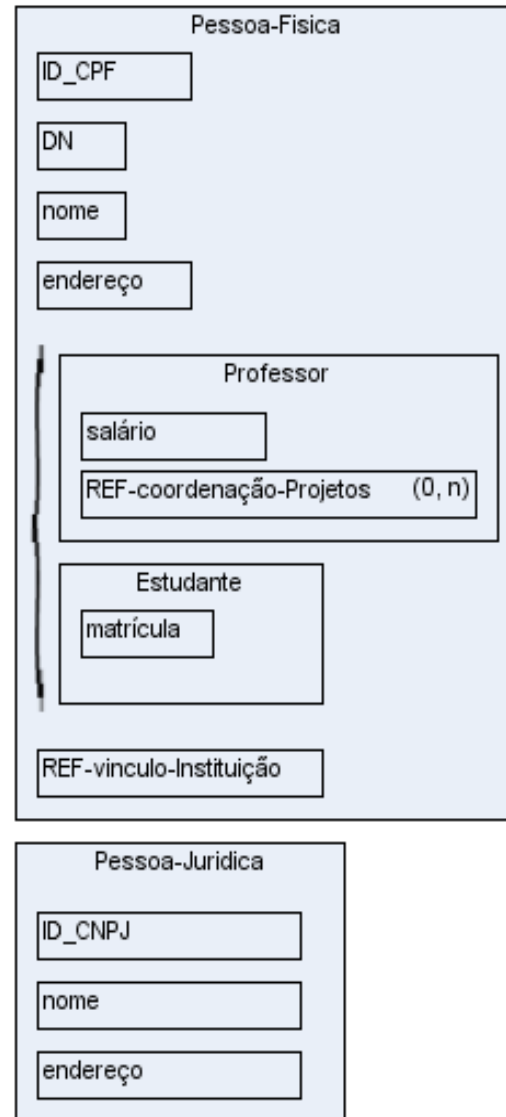
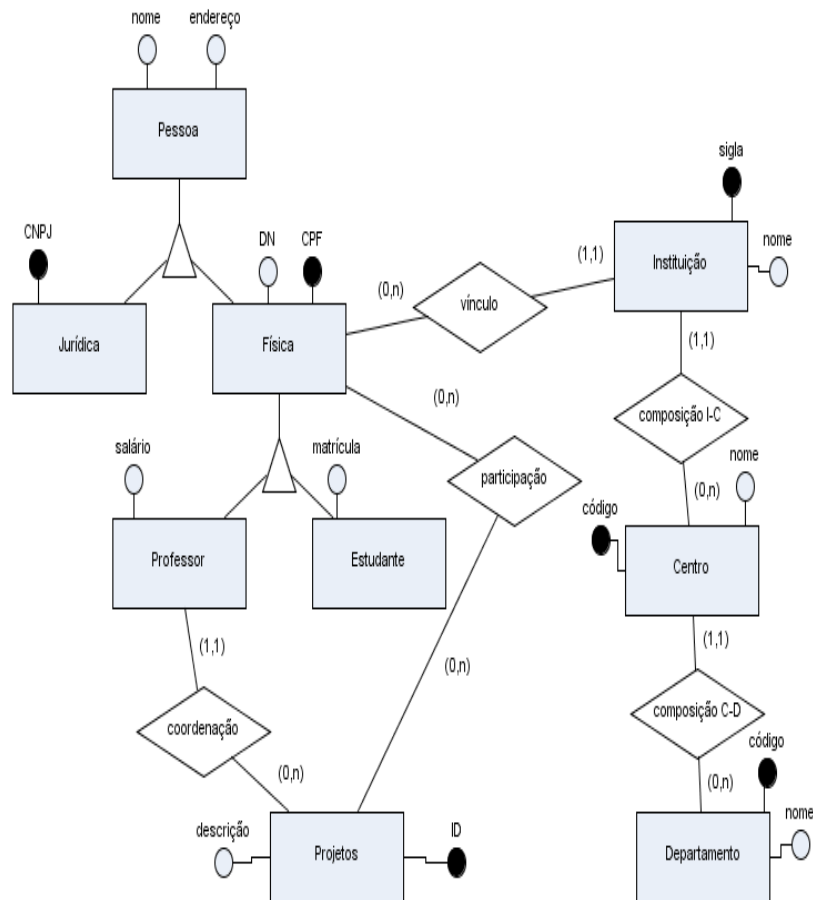
Processo de Projeto Lógico

- Conversão de Hierarquias
 - prioridade para a geração da **menor porção de esquema**
 - estratégia **bottom-up** de conversão para hierarquias de vários níveis
 - regra **GSB** não se aplica a generalizações parciais (p)
- Conversão de Relacionamentos
 - prioridade para a conversão de **cadeias mais longas de relacionamentos com cardinalidade 1-1 ou 1-N**
 - promove um maior nível de aninhamento de dados

Exemplo de Modelagem Conceitual



Possível Modelagem Lógica NoSQL

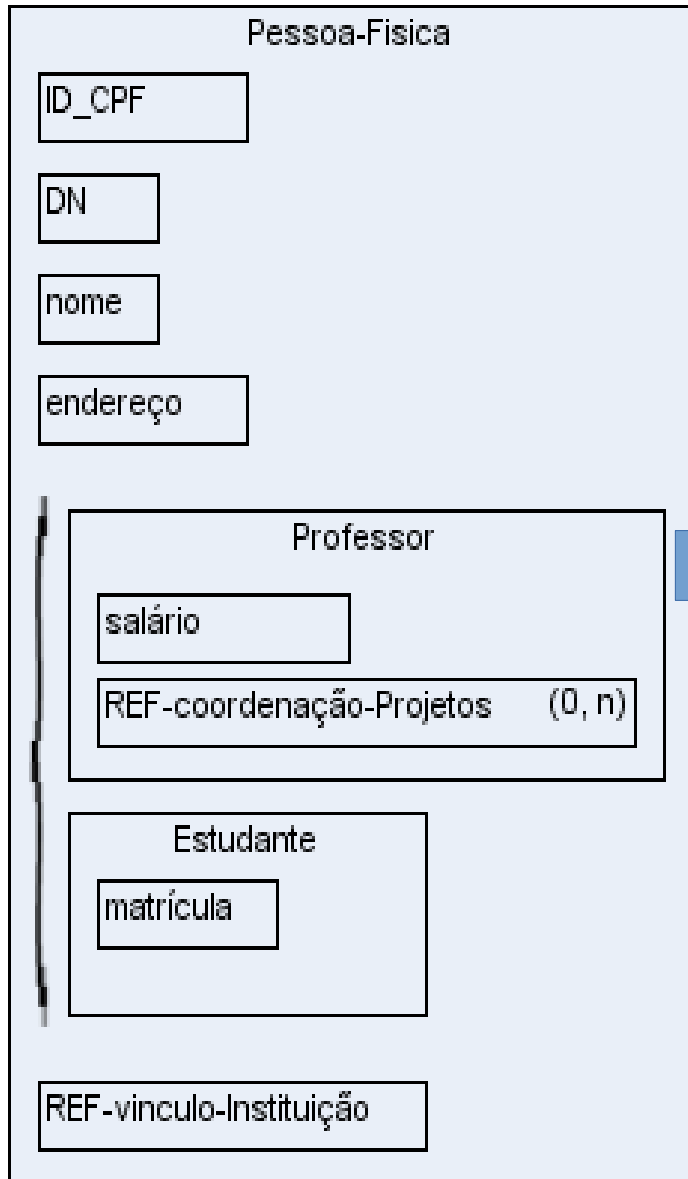


Modelagem Física NoSQL – BD Chave-Valor

- Coleção \Rightarrow Par Chave-Valor
- Regras de Mapeamento
 - ID da coleção \Rightarrow Chave
 - Conteúdo da coleção \Rightarrow Valor *

* conteúdos estruturados devem ser desaninhados e serializados

Exemplo Modelagem Física BD NoSQL Chave-Valor



BD Chave-Valor

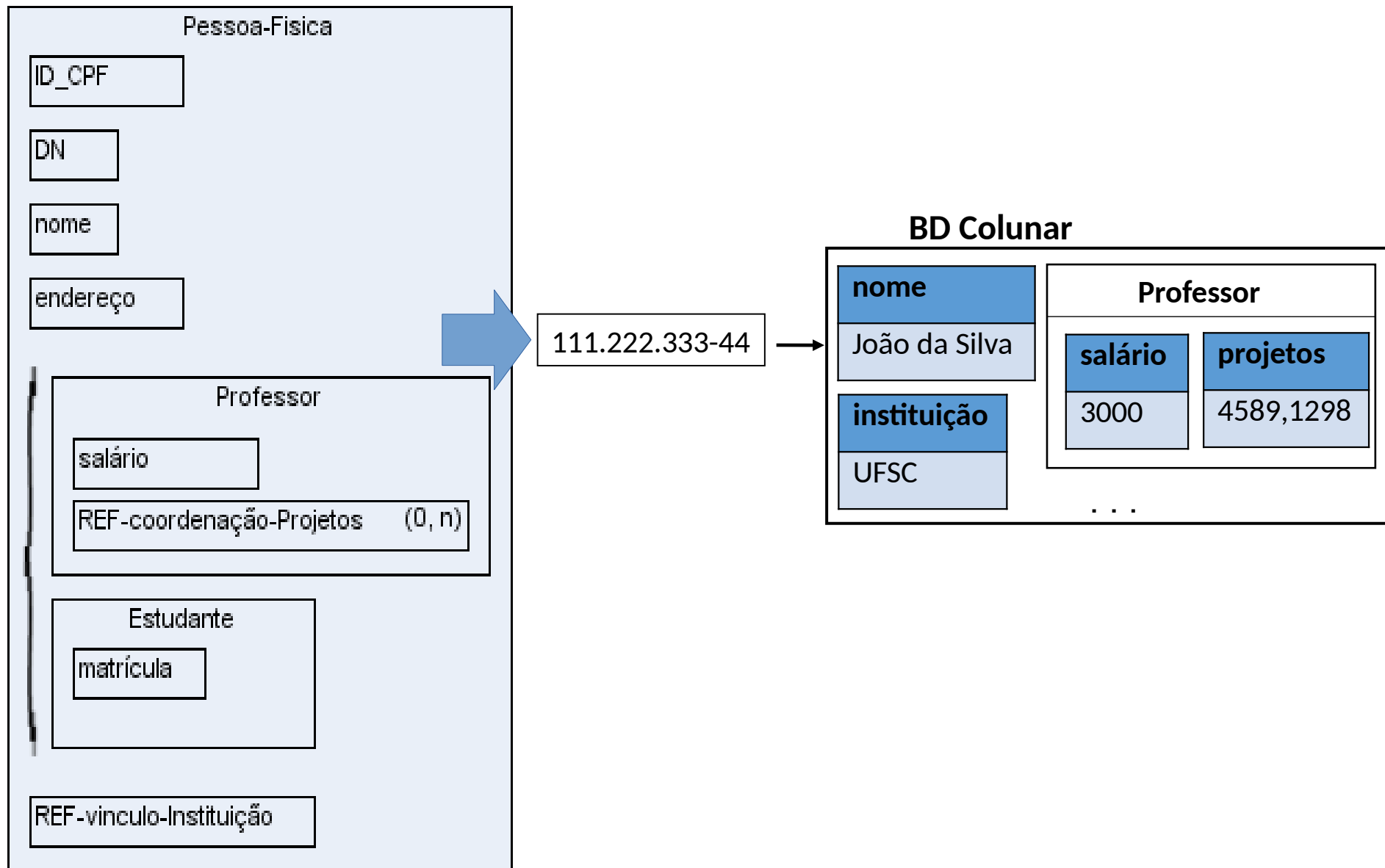
Chave: 111.222.333-44

Valor: "nome: João da Silva; ...; salário: 3000; projetos: 4589,1298; instituição: UFSC"

Modelagem Física NoSQL – BD Colunar

- Coleção \Rightarrow Família de Colunas
- Regras de Mapeamento
 - ID da coleção \Rightarrow Chave da família de colunas
 - Atributo simples \Rightarrow Coluna
 - Atributo multivalorado \Rightarrow Coluna multivalorada ou coluna com conteúdo serializado
 - Atributo de referência \Rightarrow Chave da coleção referenciada
 - Bloco \Rightarrow Supercoluna

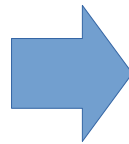
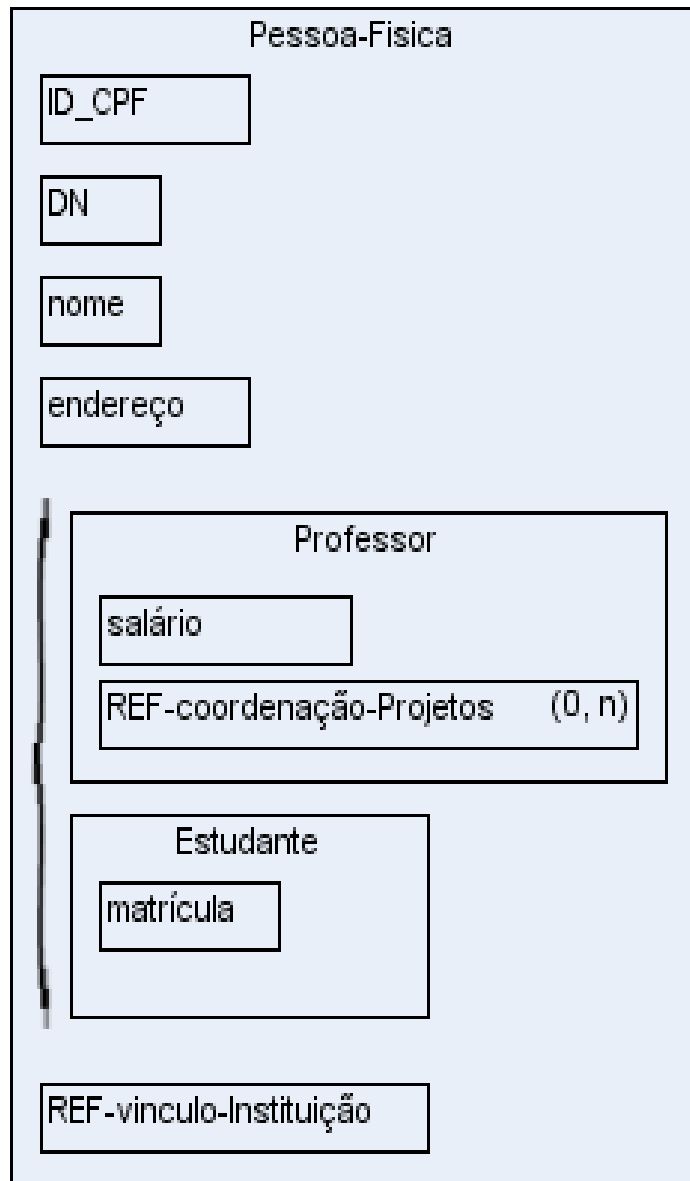
Exemplo Modelagem Física BD NoSQL Colunar



Modelagem Física NoSQL – BD Documento

- Coleção \Rightarrow Documento
- Regras de Mapeamento
 - ID da coleção \Rightarrow Chave do documento
 - Atributo simples \Rightarrow Atributo simples
 - Atributo multivalorado \Rightarrow Atributo do tipo Lista
 - Atributo de referência \Rightarrow Chave da coleção referenciada
 - Bloco \Rightarrow Atributo do tipo Objeto

Exemplo Modelagem Física BD NoSQL Documento



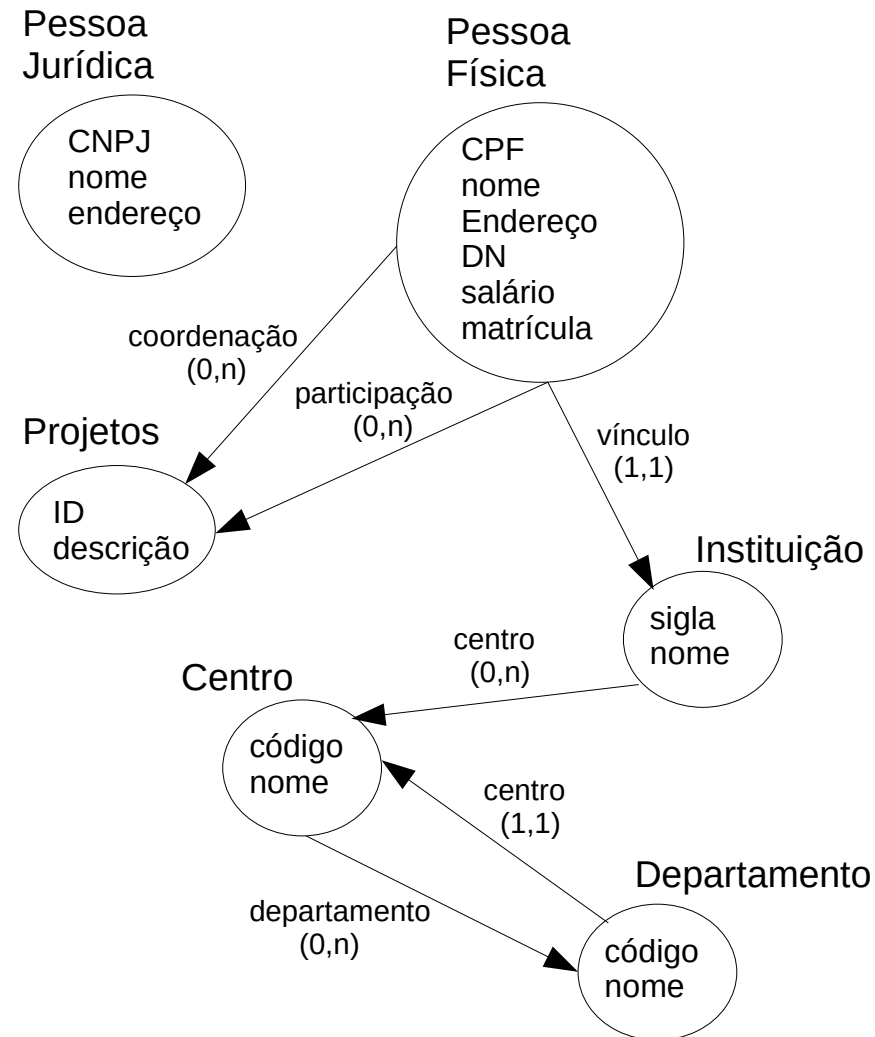
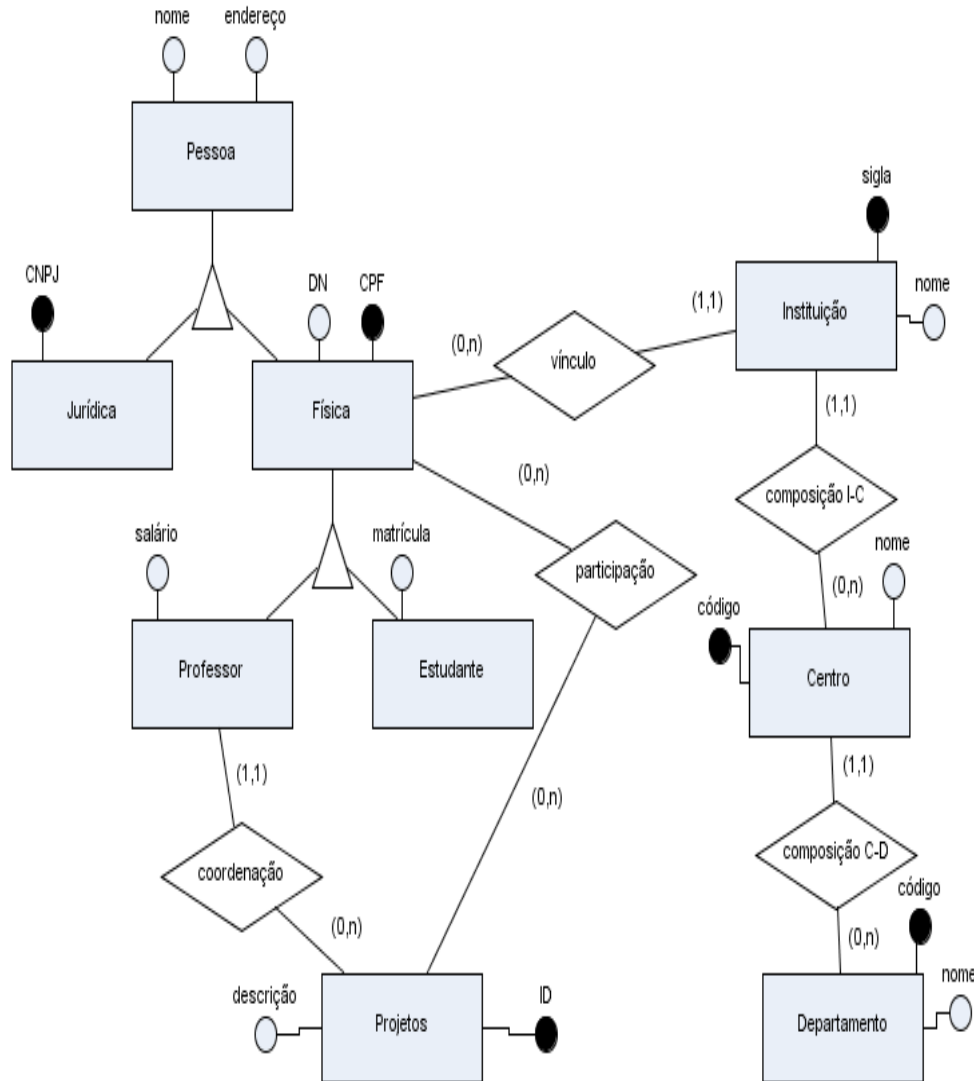
BD Documento

```
{ "_id": "111.222.333-44",  
  "nome": "João da Silva", ...  
  "Professor": {  
    "salário": "3000",  
    "projetos": [ "4589", "1298" ],  
    "instituição": "UFSC" }  
}
```

BD NoSQL Grafo

- Não segue uma abordagem orientada a agregados
 - Ênfase nos relacionamentos entre quaisquer tipos de dados e não em uma organização hierárquica deles
- Neste caso, basicamente entidades tornam-se nodos e relacionamentos tornam-se arestas
 - Decisão por arestas unidirecionais ou bidirecionais depende do *workload*
 - Pode-se fundir entidades fracamente relacionadas à entidades mais conectadas

Exemplo Modelagem Lógica BD NoSQL Grafo



Atividade 4 – Projeto NoSQL

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

