

# Banco de Dados

CENTRO UNIVERSITÁRIO NEWTON PAIVA

---

Prof. Dr. João Paulo Aramuni





# Banco de Dados

2º período

Prof. Dr. João Paulo Aramuni



# Projeto Lógico de Banco de Dados

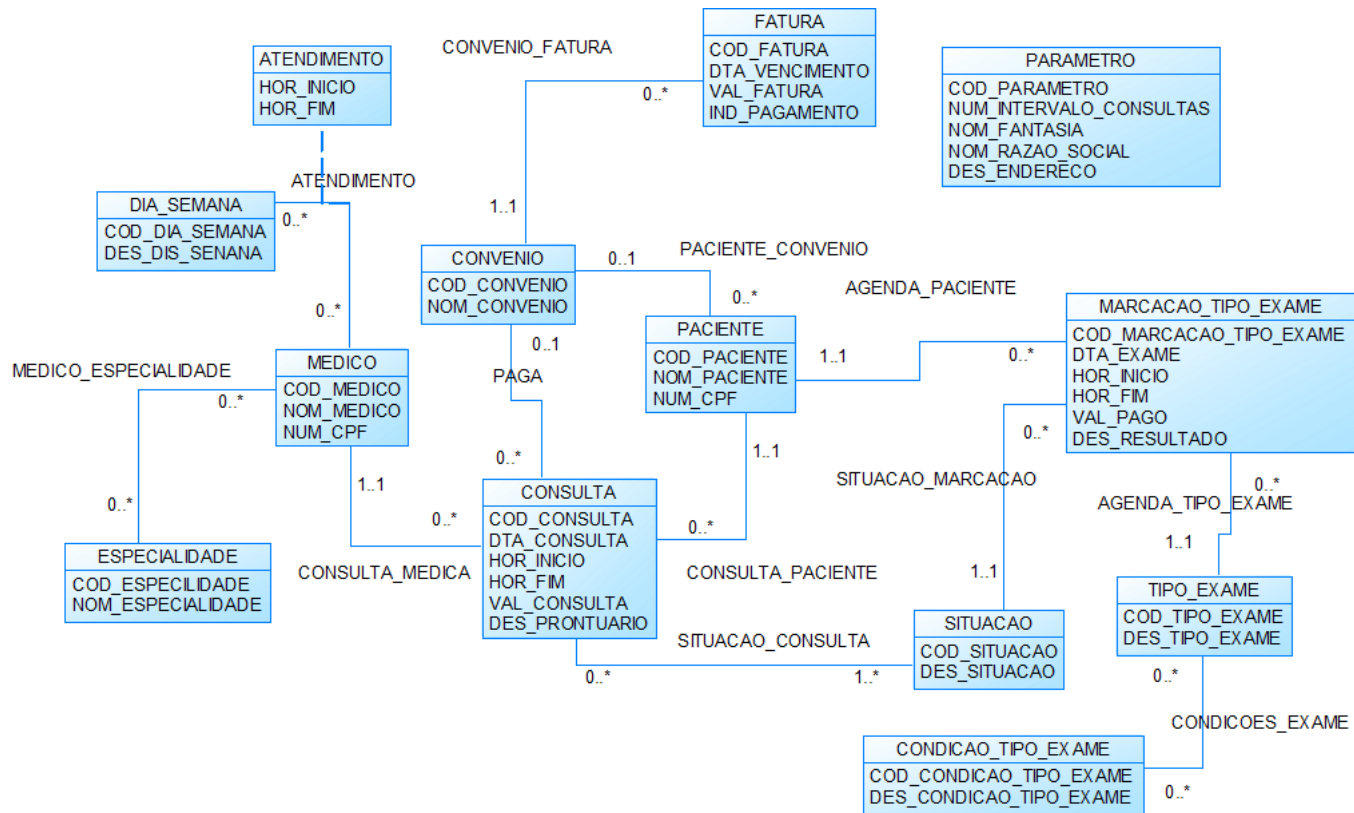
Aula 05

# Projeto Conceitual de BD

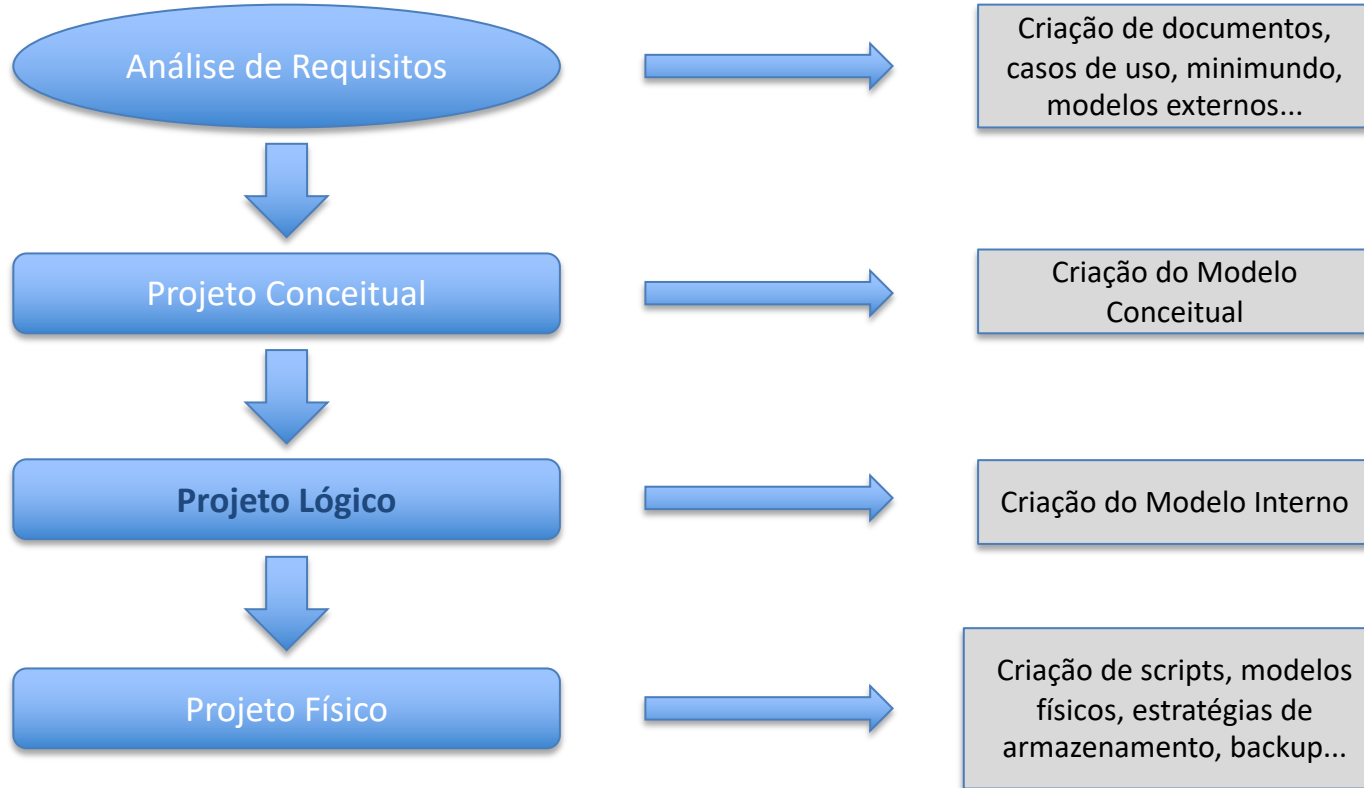
## Atividade: Projete um DCP para um sistema de Clínica Médica.

Uma clínica médica possui vários médicos atuantes em diversas especialidades como clínico geral, cardiologia, pediatria, oftalmologia, etc. Um médico pode atuar em diferentes especialidades e pode atender em diversos horários do dia e da semana. Um paciente só pode marcar uma consulta se estiver cadastrado. Sempre que uma consulta é marcada, é aberto um prontuário automatizado do cliente que conterá os históricos das consultas já realizadas pelo paciente na clínica. A marcação de consultas para os pacientes está relacionada à agenda do médico. Esta agenda deve ser emitida diariamente para controle das consultas do médico. Diferentes tipos de exames também podem ser marcados e realizados na Clínica. Cada tipo de exame requer determinadas condições do paciente, como jejum de 24 horas e vasilhames específicos. Os históricos de todos os exames marcados devem ficar armazenados no sistema. O resultado final do exame deverá estar automatizado contendo a conclusão final do exame. Atualmente, a clínica atende também através de convênios, que cobrem consultas e exames. O paciente só pode estar vinculado a um convênio. É importante saber quais os tipos de exames da clínica uma determinada empresa conveniada cobre. Um paciente pode marcar, cancelar e remarcar consultas e exames. Isto determina o status da consulta ou do exame. A fatura aos conveniados é emitida ao final do mês e para consultas estratégicas os dados das faturas devem ficar armazenados no banco de dados do sistema. Diversas consultas podem ainda ser realizadas como médicos de uma dada especialidade com horários disponíveis para um dado dia e hora, dias e horários de atendimentos de um dado médico, horários de consultas e exames de um dado paciente, especialidade mais procurada, faturamento no mês com consultas e exames e conveniadas em débitos com a clínica.

# Solução DCP Clínica Médica



# Projeto Lógico de BD



# Projeto Lógico de BD

## 1) Escolher o modelo

**Relacional** vs Não relacional

## 2) Escolher o SGBD

**DBeaver**

## 3) Escolher o BD dentro do SGBD

**MariaDB**, MySQL, SQLite, Oracle, etc

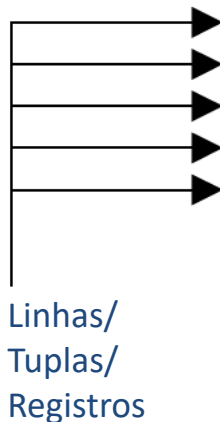
## 4) Converter DCP em DER

Transformar o

Diagrama de Classes Persistentes em

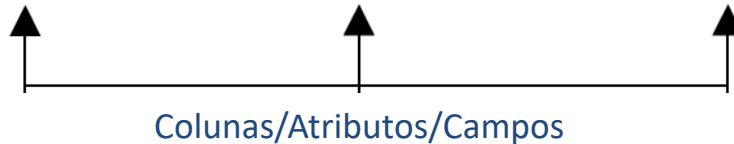
**Diagrama Entidade-Relacionamento**

## 5) Partir para o Projeto Físico (SQL)



Tabela/Relação/Arquivo

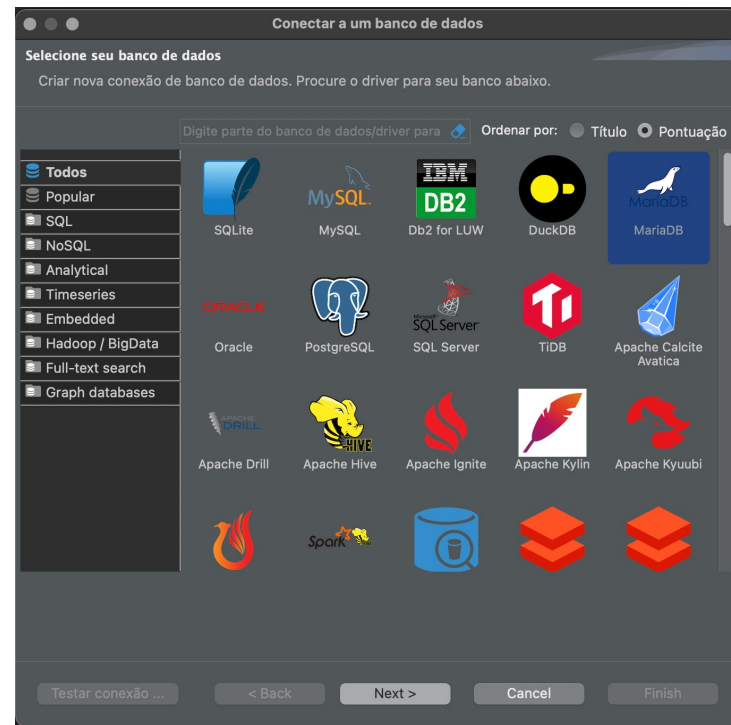
| <b>MATRICULA</b> | <b>NOME</b>         | <b>IDADE</b> |
|------------------|---------------------|--------------|
| 9912333-4        | Luis Fernando Matos | 26           |
| 8822266-9        | Sandra Barbosa      | 13           |
| 7777990-0        | Júlio Vieira        | 22           |
| 2234567-8        | Antônio Rodrigues   | 21           |
| 1400985-3        | Pedro Lemos         | 19           |



# Projeto Lógico de BD

## Dica:

- Cada Banco de Dados terá suas especificidades.
- Os tipos de dados, bem como a sintaxe dos comandos SQL que iremos escrever no projeto físico, podem ser diferentes.
- Apesar da maioria dos bancos de dados seguirem o padrão **ANSI SQL**, podem existir extensões ou recursos específicos que são exclusivos dele, além de possuir algumas diferenças na implementação dos comandos e tipos de dados padrão.
- Vejamos um exemplo para o banco de dados **MariaDB**.





# MariaDB - Datatypes

| Dados Numéricos Inteiros |  |                                |
|--------------------------|--|--------------------------------|
| Tipo                     | Escopo com sinal                                       | Escopo sem sinal               |
| Tinyint                  | -128 a 127   | 0 a 255                        |
| Smallint                 | -32.768 a 32.767                                       | 0 a 65.535                     |
| Mediumint                | -8.388.608 a 8.388.607                                 | 0 a 16.777.215                 |
| Int                      | -2.147.483.648 a 2.147.483.647                         | 0 a 4.294.967.295              |
| Bigint                   | -9.223.372.036.854.775.808 a 9.223.372.036.854.775.807 | 0 a 18.446.744.073.709.551.615 |

| Dados Numéricos (Bit e Boolean) |                        |
|---------------------------------|------------------------|
| Tipo                            | Numero máximo de bytes |
| bit                             | 1                      |
| bool ou boolean                 | 1                      |

| Dados de Texto Não-Binário |                        |
|----------------------------|------------------------|
| Tipo de texto              | Numero máximo de bytes |
| Tinytext                   | 255                    |
| Text                       | 65.535                 |
| MediumText                 | 16.777.215             |
| LongText                   | 4.294.967.295          |
| Varchar                    | 65.535                 |
| Char                       | 255                    |

| Dados Temporais |                     |   |
|-----------------|---------------------|---|
| Tipo            | Formato padrão      | Valores permitidos                        |
| Date            | AAAA-MM-DD          | 1000-01-01 a 9999-12-31                   |
| Datetime        | AAAA-MM-DD HH:MI:SS | 1000-01-01 00:00:00 a 9999-12-31 23:59:00 |
| Timestamp       | AAAA-MM-DD HH:MI:SS | 1970-01-01 00:00:00 a 2037-12-31 23:59:00 |
| Year            | AAAA                | 1901 a 2155                               |
| Time            | HHH:MI:SS           | -838:59:59 a 838:59:59                    |

| Dados de Texto Binário |                        |
|------------------------|------------------------|
| Tipo de texto          | Numero máximo de bytes |
| Tinyblob               | 255                    |
| Blob                   | 65.535                 |
| Mediumblob             | 16.777.215             |
| Longblob               | 4.294.967.295          |
| Varbinary              | 65.535                 |
| Binary                 | 255                    |

| Dados Numéricos de Ponto Flutuante e Ponto Fixo |  |  |
|---|--|--|
| Tipo  | Escopo numérico  |  |
| Float(p,e)                                      | -3,402823466E+38 a -1,175494351E-38 e de 1.175494351E-38 a 3,402823466E+38                                 |  |
| Double(p,e)                                     | -1,7976931348623157E+308 a -2,2250738585072014E-308 e de 2,2250738585072014E-308 a 1.7976931348623157E+308 |  |
| Decimal(p,e)                                    | -1,7976931348623157E+308 a -2,2250738585072014E-308 e de 2,2250738585072014E-308 a 1.7976931348623157E+308 |  |

# Projeto Lógico de BD

## Primary Key (PK) vs Foreign Key (FK) - Chave primária vs Chave estrangeira

**Primary Key** (Chave Primária): A Primary Key é uma coluna (ou conjunto de colunas/Composite Primary Key) em uma tabela de banco de dados que serve como identificador exclusivo para cada registro na tabela. Ela garante que não haja valores duplicados na coluna e é usada para indexar a tabela, o que torna as consultas mais eficientes. Cada tabela deve ter uma chave primária.

- **Exemplo:** Considere uma tabela chamada "Clientes" com as colunas "ID", "Nome" e "Email". A coluna "ID" pode ser definida como a chave primária, garantindo que cada cliente tenha um ID único na tabela.

**Foreign Key** (Chave Estrangeira): A Foreign Key é uma coluna em uma tabela que estabelece uma relação com a chave primária de outra tabela. Ela é usada para criar vínculos entre diferentes tabelas e garantir a integridade referencial dos dados, ou seja, garantir que não haja valores "órfãos" ou inválidos nas colunas que fazem referência a outras tabelas.

- **Exemplo:** Além da tabela "Clientes", vamos supor que temos outra tabela chamada "Pedidos" com as colunas "PedidoID", "Data" e "ClienteID". A coluna "ClienteID" na tabela "Pedidos" é uma foreign key que se relaciona com a coluna "ID" na tabela "Clientes". Isso significa que cada pedido está associado a um cliente específico, e a chave estrangeira "ClienteID" na tabela "Pedidos" aponta para a chave primária "ID" na tabela "Clientes".

# Projeto Lógico de BD

## Primary Key (PK) vs Foreign Key (FK) - Chave primária vs Chave estrangeira

### Obrigatoriedade da chave estrangeira:

Em alguns casos, a chave estrangeira pode ser definida como **obrigatória** (NOT NULL), o que significa que sempre é necessário fornecer um valor válido para a chave estrangeira no momento da inserção de um novo registro na tabela.

Por outro lado, a chave estrangeira também pode ser definida como **opcional** (NULL), o que permite que o valor da chave estrangeira seja deixado em branco durante a inserção. Nesse caso, o registro será inserido sem uma referência à tabela relacionada.

| DEPARTAMENTO     |                  |
|------------------|------------------|
| COD_DEPARTAMENTO | NOM_DEPARTAMENTO |
| 1                | Pessoal          |
| 2                | Vendas           |
| 3                | Produção         |

Relação = Tabela

Tuplas = Linhas

| EMPREGADO     |                   |              |             |                          |
|---------------|-------------------|--------------|-------------|--------------------------|
| COD_EMPREGADO | NOM_EMPREGADO     | NUM_TELEFONE | VAL_SALARIO | COD_DEPARTAMENTO_LOTACAO |
| 212           | José Antônio      | 33326262     | 1967,98     | 2                        |
| 321           | Antônio Rodrigues | 32124589     | 3045,84     | 3                        |
| 109           | Lucas da Silva    | 32124589     | 3605,00     | 3                        |
| 450           | Maria Fernandes   | 33129860     | 2398,45     | 1                        |
| 867           | Carlos Henrique   | 35634011     | 6099,00     | 3                        |

Primary Key = Chave Primária

| PROJETO     |                        |            |            |
|-------------|------------------------|------------|------------|
| COD_PROJETO | NOM_PROJETO            | DTA_INICIO | DTA_FIM    |
| 001         | Carro supersônico      | 02/03/2002 |            |
| 002         | Implantação ERP        | 23/01/2001 | 19/07/2001 |
| 003         | Gestão de energia 2003 | 01/06/2002 |            |

Foreign Key = Chave Estrangeira

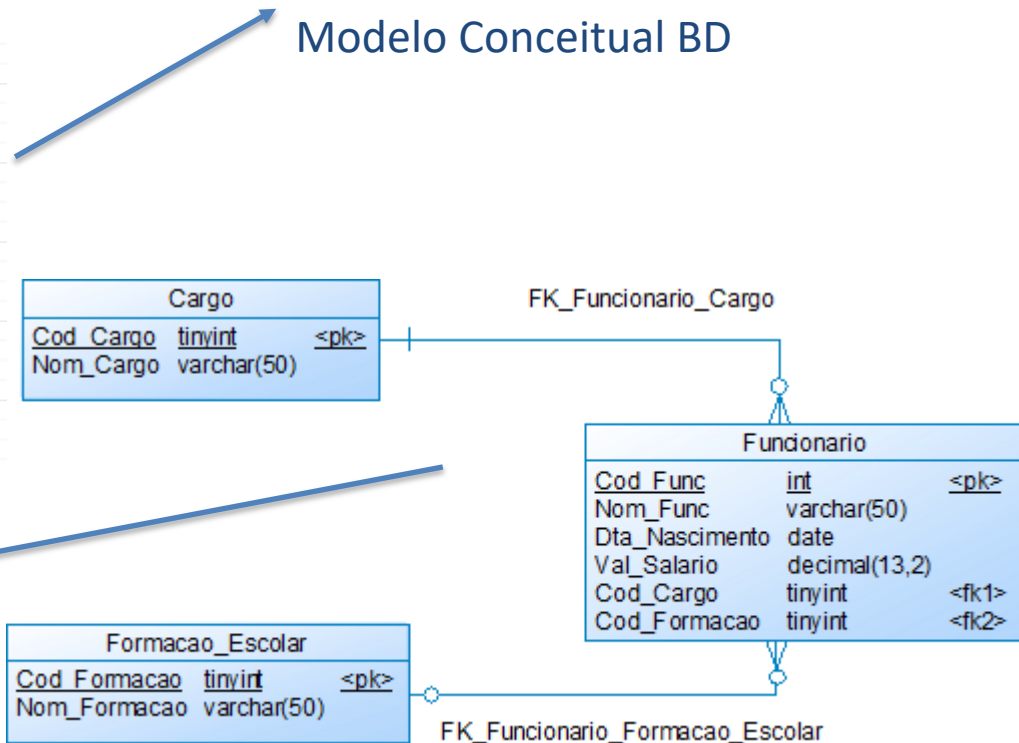
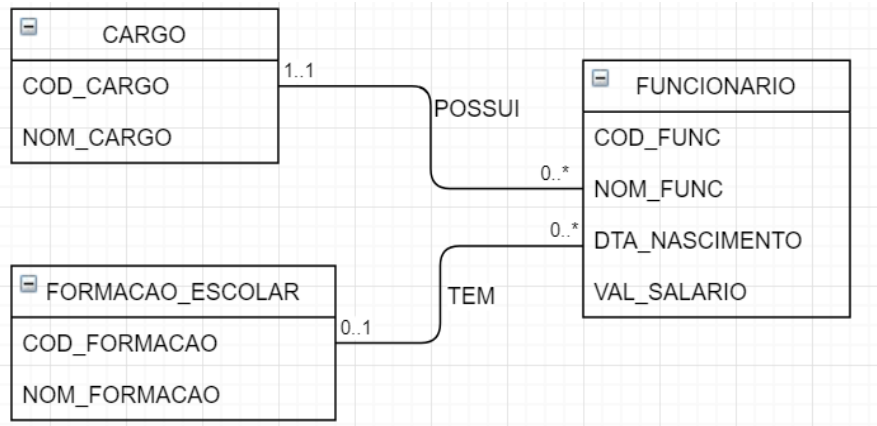
| EMPREGADO_PROJETO |               |                       |
|-------------------|---------------|-----------------------|
| COD_PROJETO       | COD_EMPREGADO | NUM_HORAS_TRABALHADAS |
| 001               | 321           | 89                    |
| 001               | 867           | 31                    |
| 002               | 212           | 29                    |
| 003               | 450           | 8                     |
| 003               | 212           | 14                    |

Projeto Lógico de BD

# Projeto Lógico de BD

DER: Transformar classes em Tabelas - Regra 1:N

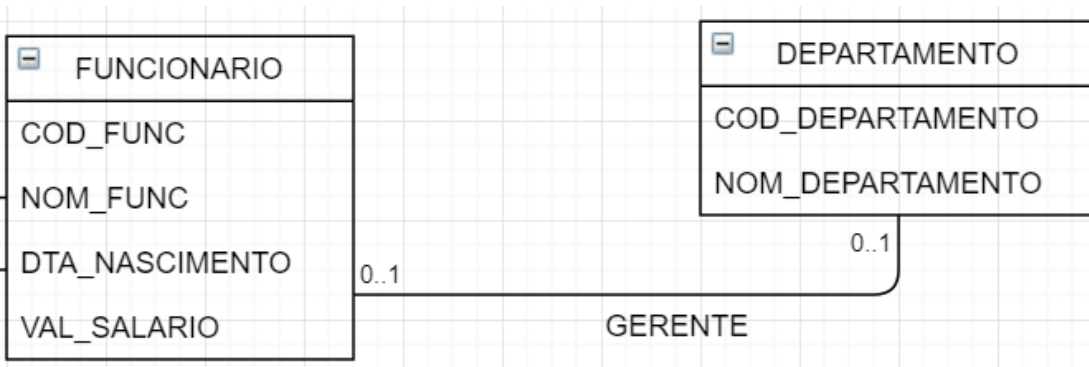
DCP:  
Modelo Conceitual BD



DER:  
Modelo Lógico BD

# Projeto Lógico de BD

DER: Transformar classes em Tabelas - Regra 1:1



DCP:  
Modelo Conceitual BD

DER:  
Modelo Lógico BD

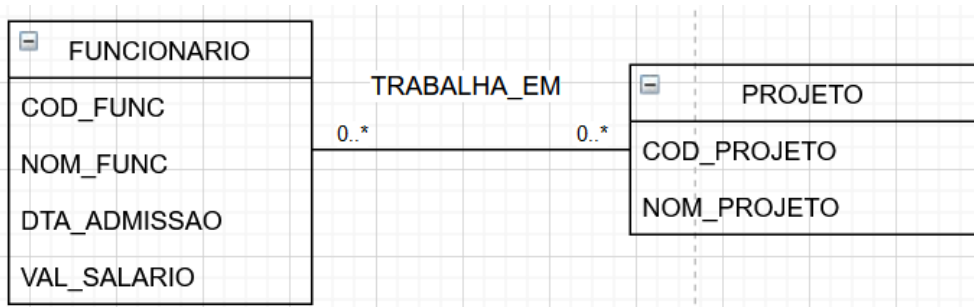
| Funcionario     |               |      |
|-----------------|---------------|------|
| <u>Cod_Func</u> | int           | <pk> |
| Nom_Func        | varchar(50)   |      |
| Dta_Nascimento  | date          |      |
| Val_Salario     | decimal(13,2) |      |

FK\_GERENTE

| Departamento            |             |      |
|-------------------------|-------------|------|
| <u>Cod_Departamento</u> | int         | <pk> |
| Nom_Departamento        | varchar(30) |      |
| Cod_Func_Gerente        | int         | <fk> |

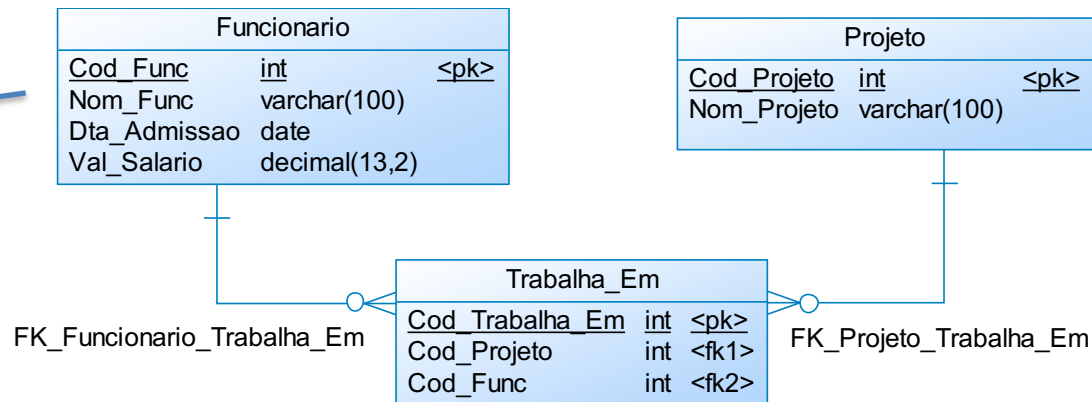
# Projeto Lógico de BD

DER: Transformar classes em Tabelas - Regra N:M



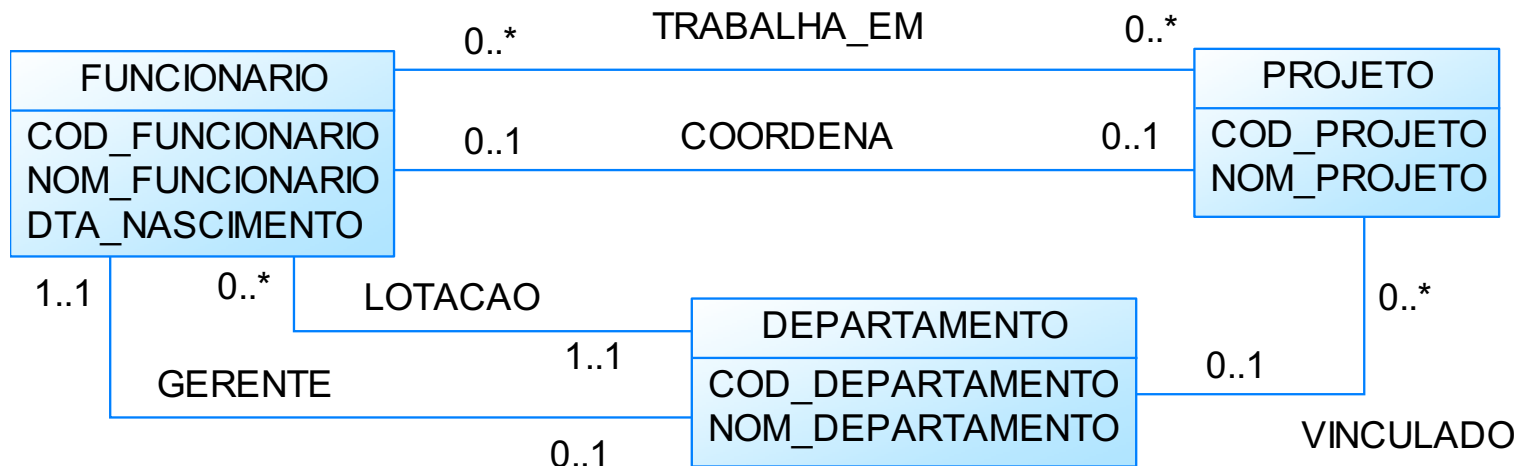
DCP:  
Modelo Conceitual BD

DER:  
Modelo Lógico BD



# Projeto Lógico de BD

Desenhe o **DER** para o DCP abaixo:  
Utilize o draw.io ou o Power Designer







Obrigado!

joao.aramuni@newtonpaiva.br