

# Bases de Dados

PL05 – Modelação Lógica

**Docente:** Cristiana Neto

**Email:** [cristiana.neto@algoritmi.uminho.pt](mailto:cristiana.neto@algoritmi.uminho.pt)

**Horário de Atendimento:**

6ª feira 09h–10h



# Sumário

1

Revisão do Modelo Conceptual

2

Instalação do MySQL Workbench

3

Regras de Derivação

4

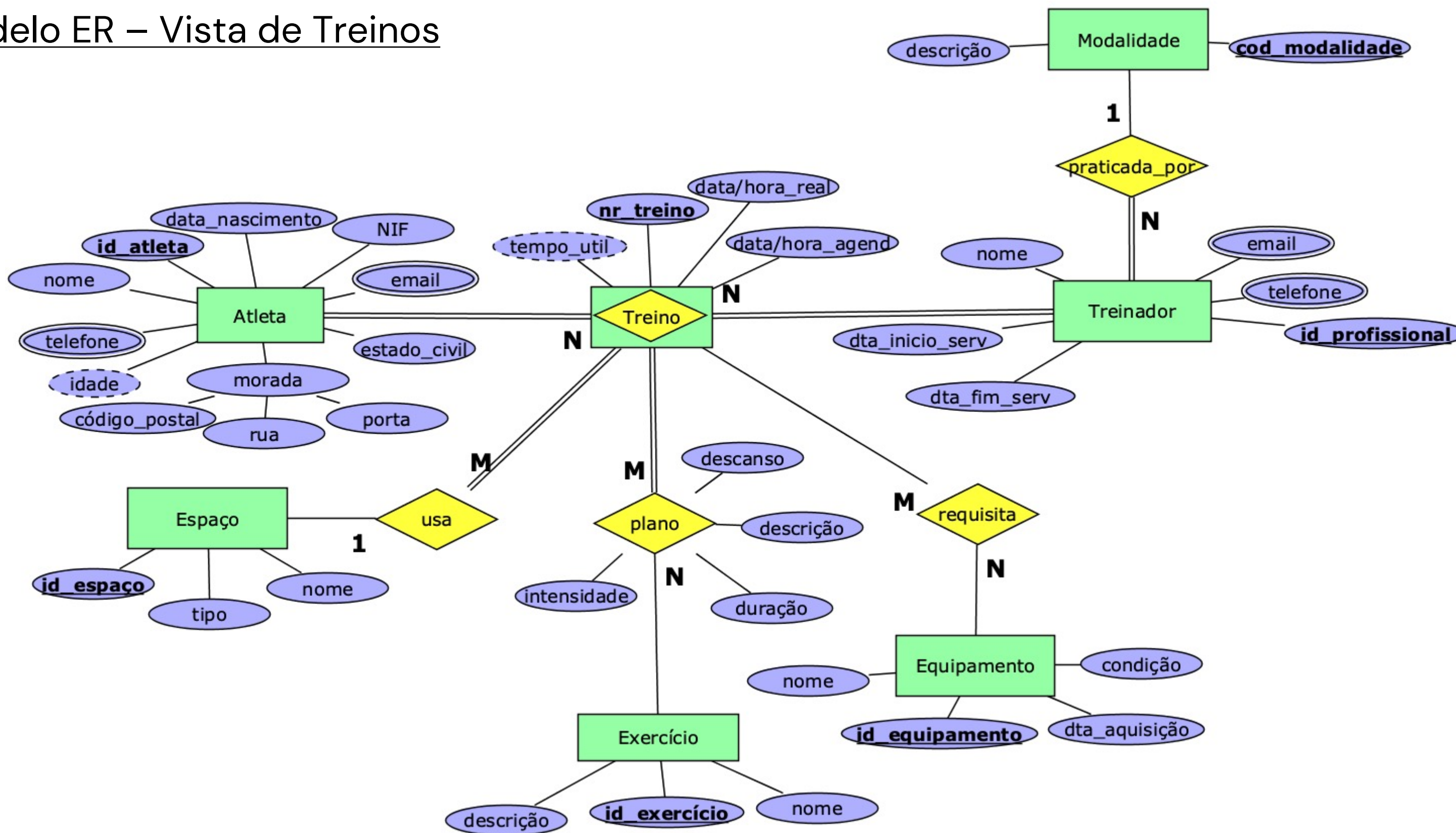
Modelação Lógica

## **Bibliografia:**

- Connolly, T., Begg, C., Database Systems, A Practical Approach to Design, Implementation, and Management , Addison-Wesley, 4a Edição, 2004. **(Chapter 17)**
- Teorey, T., Database Modeling and Design: The Fundamental Principles, II Edição, Morgan Kaufmann, 1994.

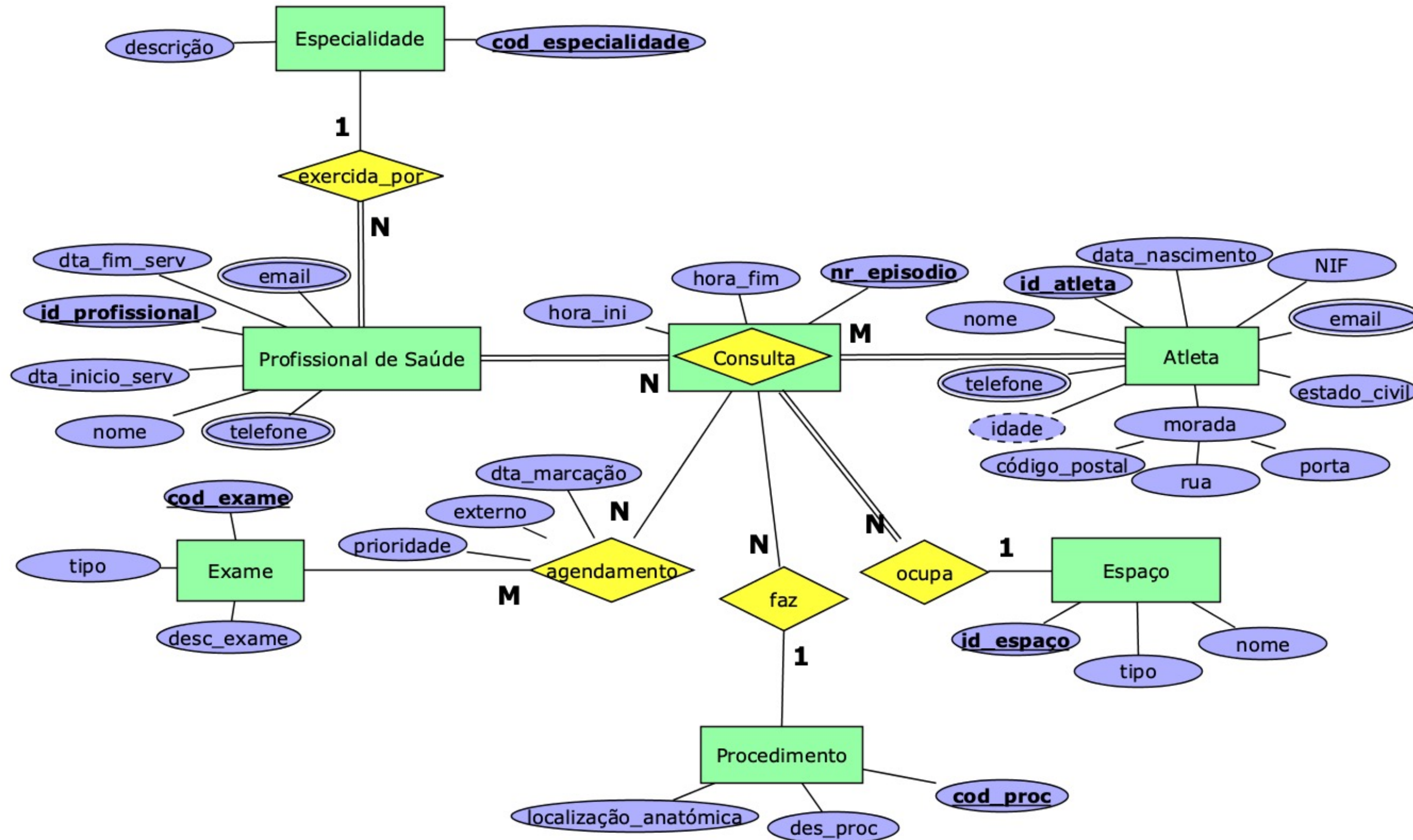
# Revisão das aulas anteriores

## Modelo ER – Vista de Treinos



# Revisão das aulas anteriores

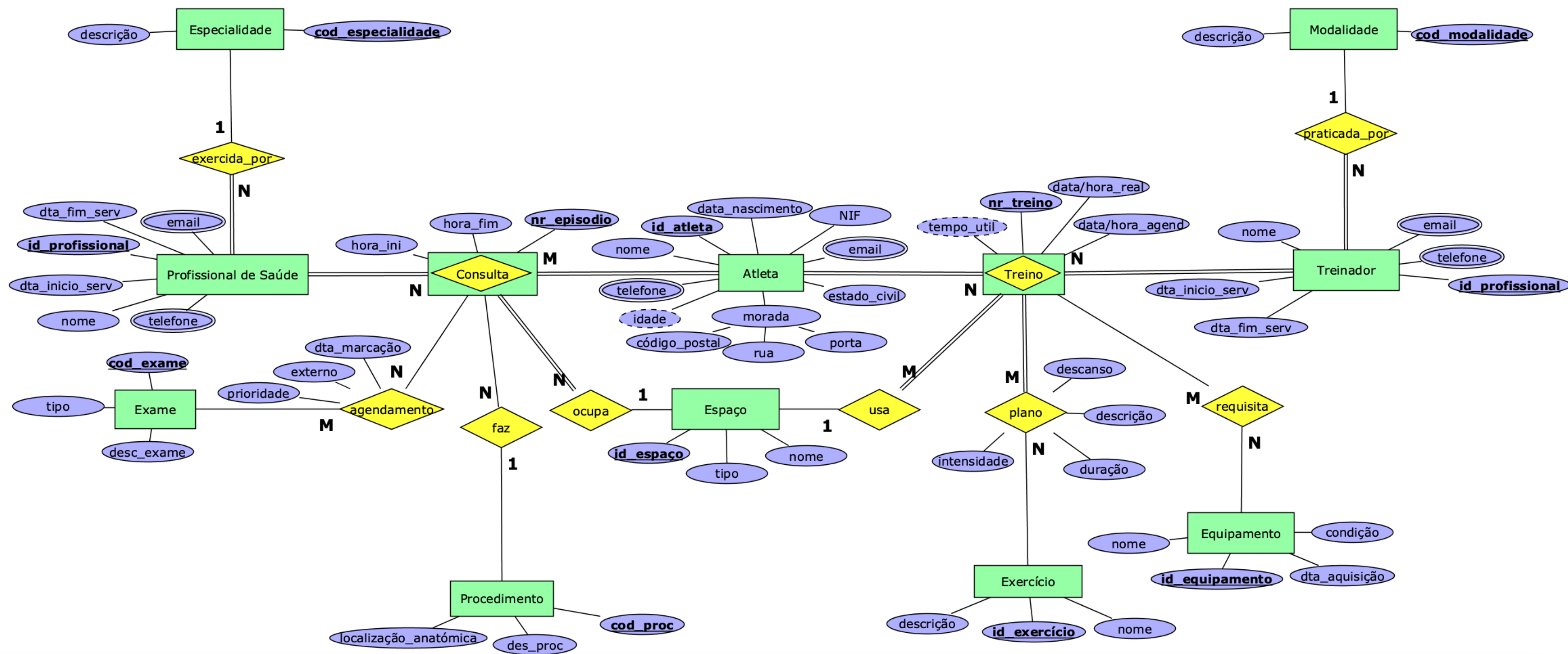
## Modelo ER – Vista de Consultas





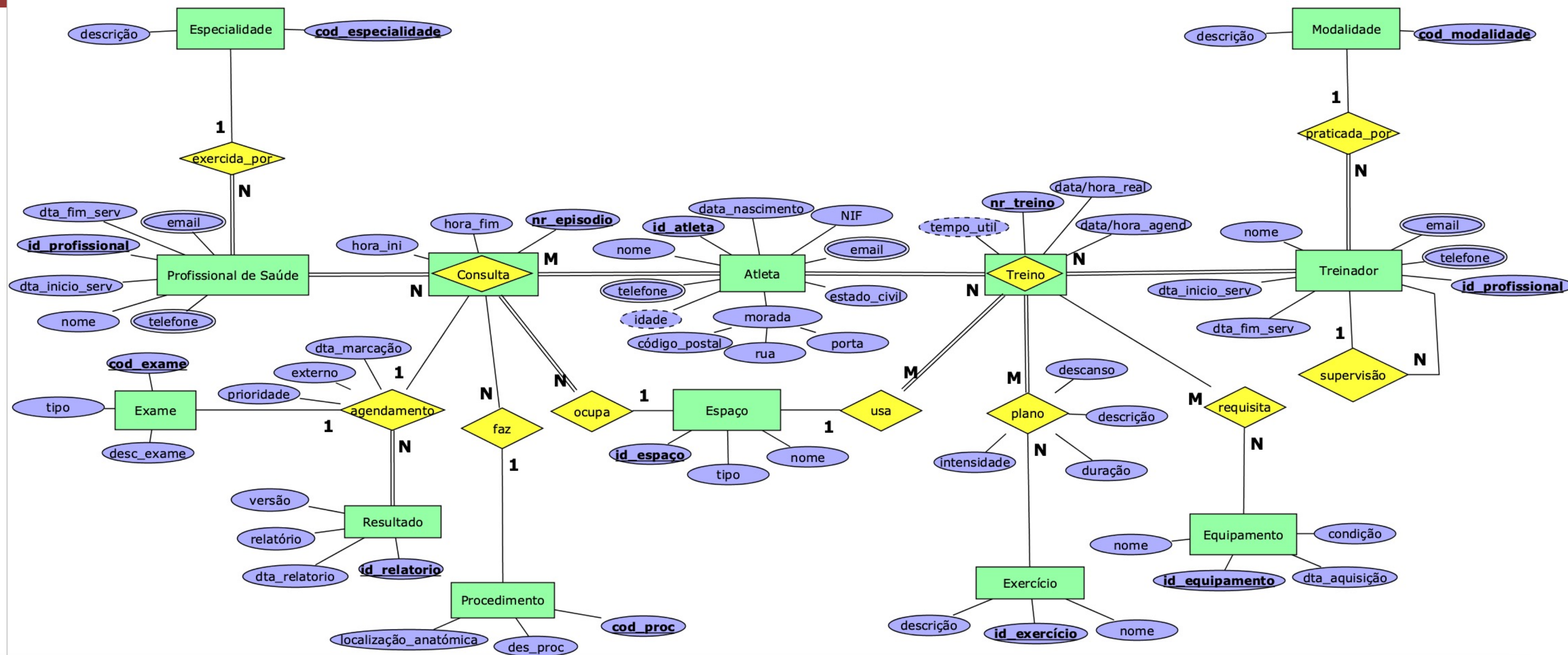
# Revisão das aulas anteriores

## Modelo ER – Combinação das vistas



# Revisão das aulas anteriores

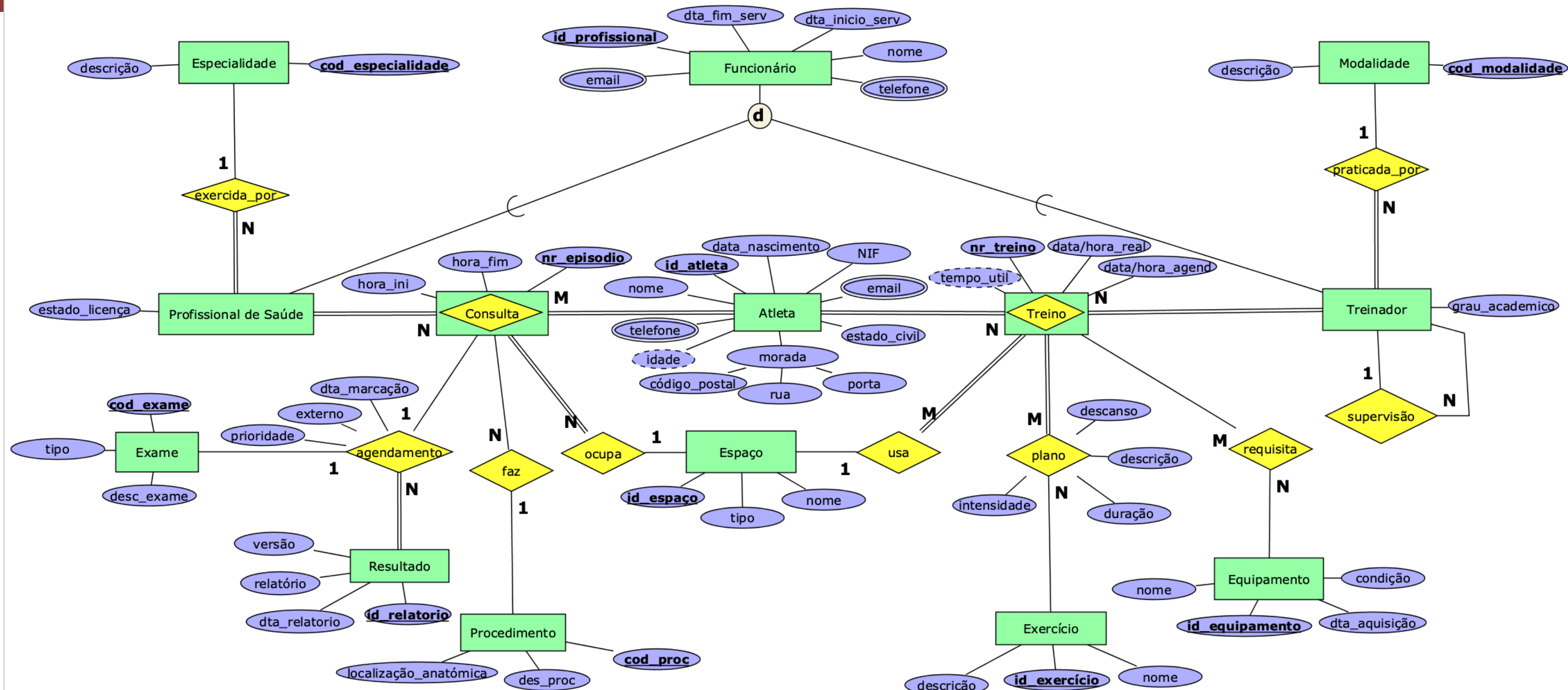
## Modelo ER – Com relacionamento recursivo + relacionamento ternário





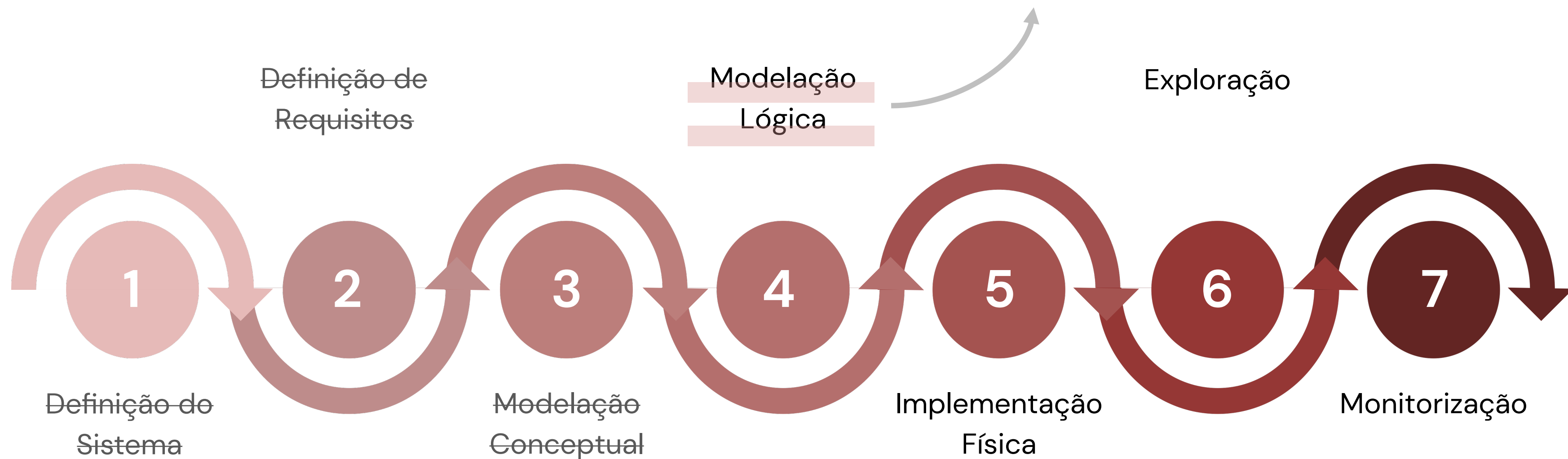
# Revisão das aulas anteriores

## Modelo ER – Aprimorado



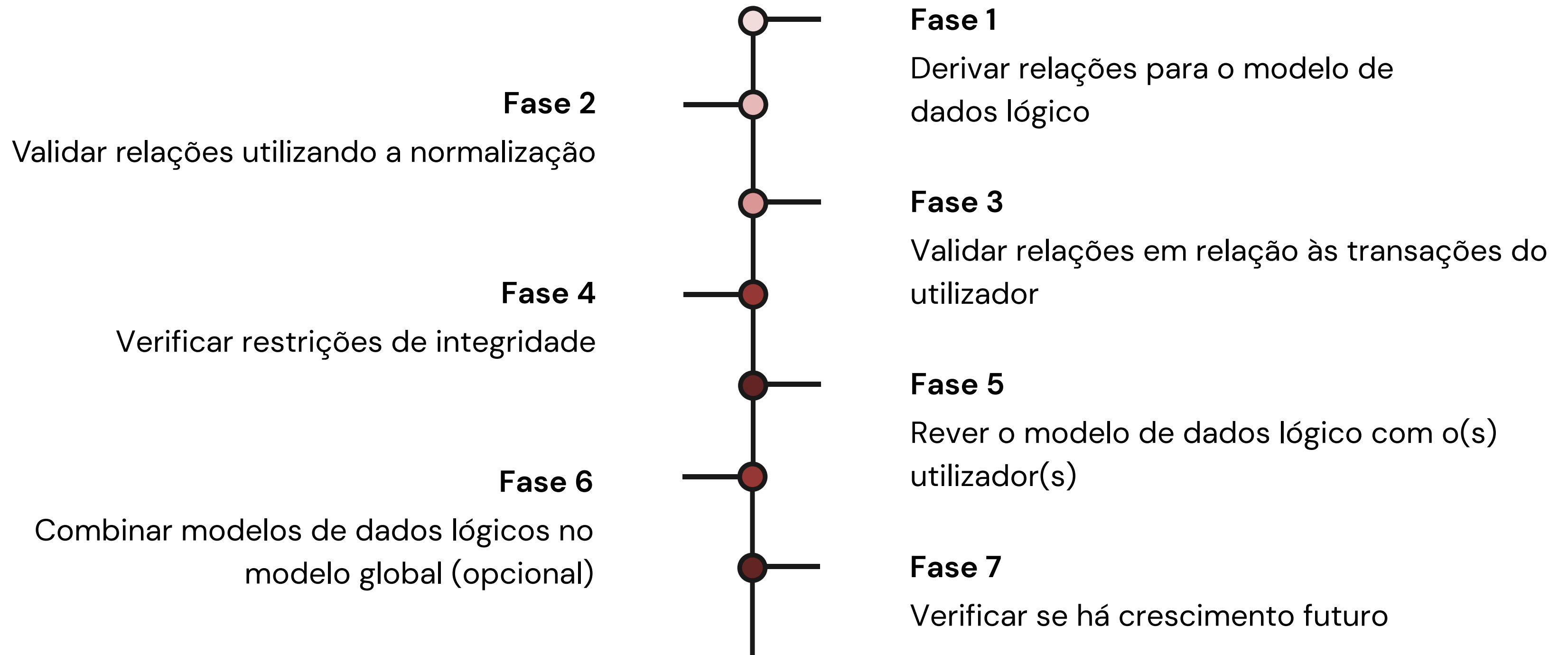
# Modelação Conceptual

Traduzir o modelo de dados conceptual num modelo de dados lógico e, em seguida, validar o modelo para verificar se este é estruturalmente correto e capaz de suportar as transações necessárias.





# Ciclo de vida de um SBD: Modelação Lógica



# Modelo Relacional

Modelo lógico para BDs relacionais, baseado no conceito de relação, também designado por tabela.

**Modelação Física**  
tabelas físicas

O modelo relacional pode depois ser concretizado num SGBD usando a linguagem SQL.

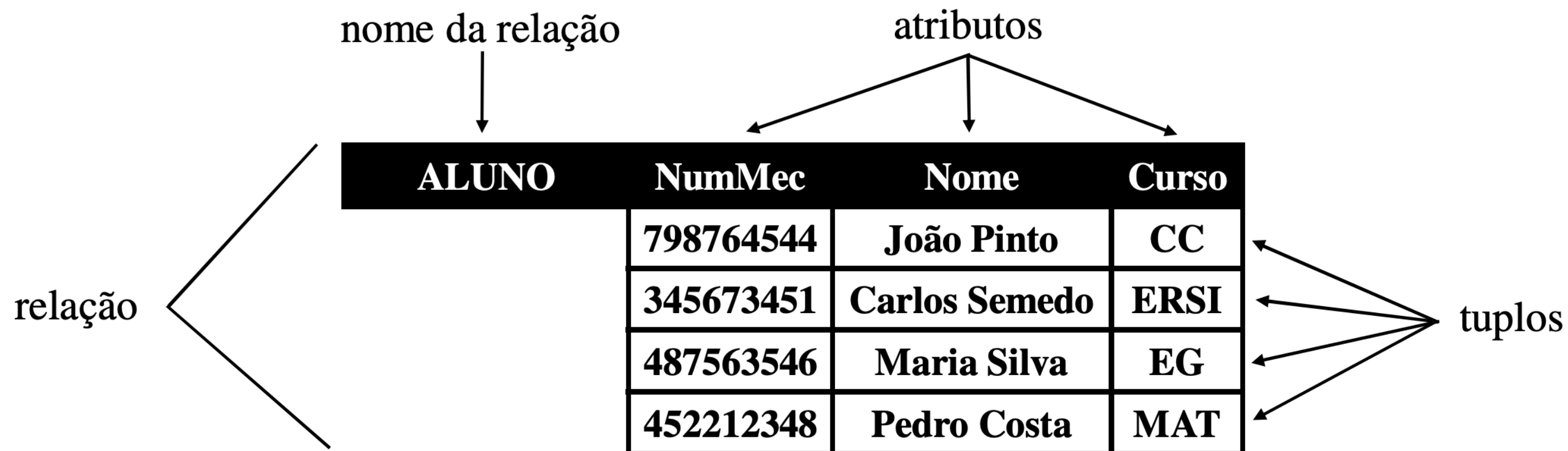
**Modelação Lógica**  
entidades de dados

As entidades-tipo e relacionamentos do modelo ER são mapeados em relações/tabelas no modelo relacional.

**Modelação Conceptual**  
conceitos de negócio

# Modelo Relacional

- É baseado no conceito de **relação**, onde uma relação é uma tabela de valores.
- Uma tabela de valores pode ser vista como um conjunto de linhas ou **tuplos**.
- Cada tuplo é identificado por um conjunto de colunas ou **atributos**.
- Uma base de dados é representada como um conjunto de relações.





# FASE 4: Modelação Lógica

## ➔ Derivar relações

- O relacionamento que uma entidade tem com outra entidade é representado pelo mecanismo de **chave primária/chave estrangeira**.
- Para decidir onde colocar o(s) atributo(s) de chave estrangeira, devemos primeiro identificar as entidades '**pai**' e '**filho**' envolvidas no relacionamento.
- A entidade **pai** refere-se à entidade que **envia uma cópia da sua chave primária** na relação que representa a entidade **filho**, para atuar como a **chave estrangeira**.

# FASE 4: Modelação Lógica

## ➔ Derivar relações

O processo de derivação passa por descrever como as relações são derivadas para as seguintes estruturas que podem ocorrer num modelo de dados concetual:

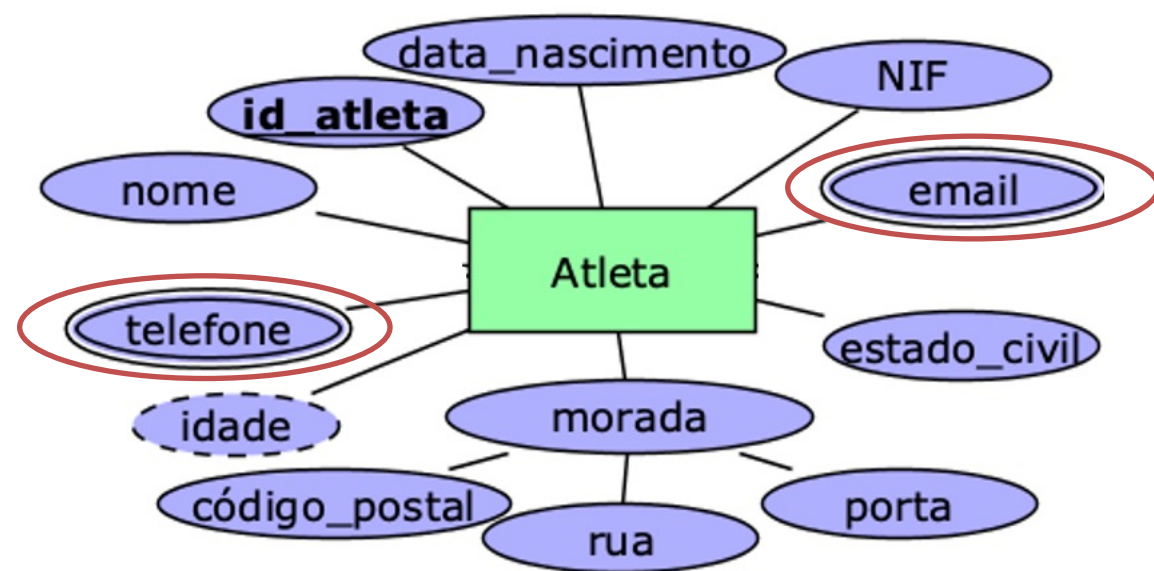
- Entidades Simples
- Atributos multivalor
- Entidades Fracas
- Relacionamentos binários de um-para-muitos (1:N)
- Relacionamentos binários de muitos-para-muitos (N:M)
- Entidade Relacionamento
- Relacionamentos binários de um-para-um (1:1)
- Relacionamentos binários recursivos de um-para-um (1:1)
- Relacionamentos complexos
- Relacionamentos superclasse/subclasse

# FASE 4: Modelação Lógica

## ➔ Derivar relações

### ▪ Entidades Simples

Para cada entidade do modelo de dados, crie **uma relação/tabela** que inclua todos os **atributos simples** dessa entidade. Os atributos derivados devem ser analisados e no caso dos atributos compostos, são apenas incluídos os atributos simples constituintes.



**Atleta** (id\_atleta, nome, dta\_nascimento, rua, porta, codigo\_postal, NIF, estado\_civil)

**Chave primária** id\_atleta

**Chave candidata** NIF

**Derivado** idade(dta\_atual – dta\_nascimento)

Atleta
<u>nr_sequencial</u>
<u>id_atleta</u>
dta_nascimento
rua
porta
codigo_postal
NIF
estado_civil

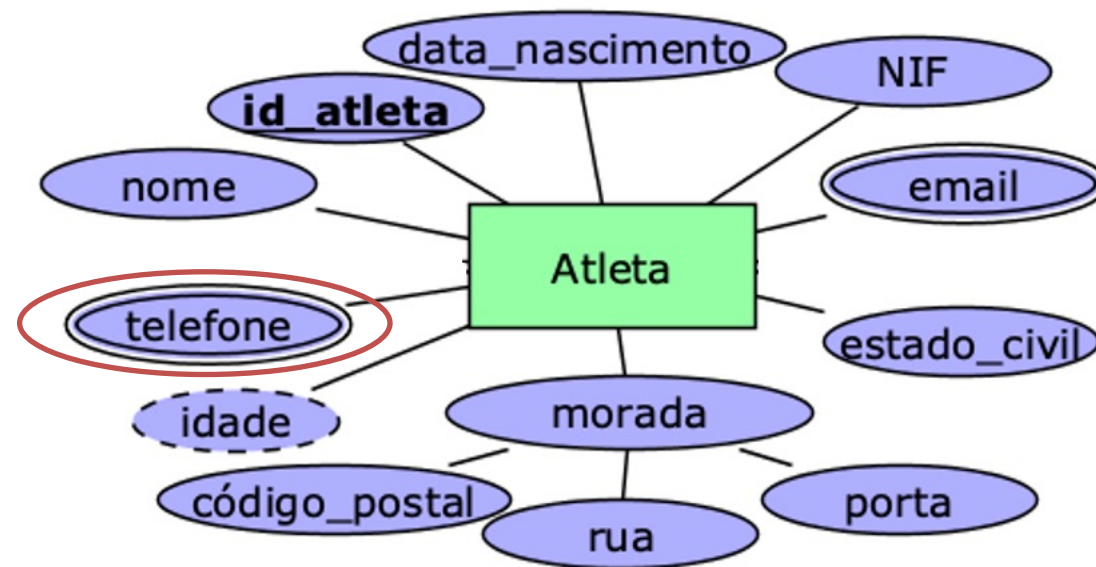


# FASE 4: Modelação Lógica

## ➔ Derivar relações

### ▪ Atributos multivalor

Para cada atributo **multivalor**, crie uma **nova relação** para representar o atributo **multi-valor** com relacionamento de **1:N** com a sua tabela de referência e inclua a **chave primária** da entidade na nova relação, para atuar como **chave estrangeira**.



**Atleta** (id\_atleta, nome, data\_nascimento, rua, porta, código\_postal, NIF, estado\_civil)

**Chave primária** id\_atleta

**Chave candidata** NIF

**Derivado** idade(data\_atual – data\_nascimento)

**Telefone** (telefone, id\_atleta)

**Chave primária** telefone

**Chave estrangeira** id\_atleta referencia

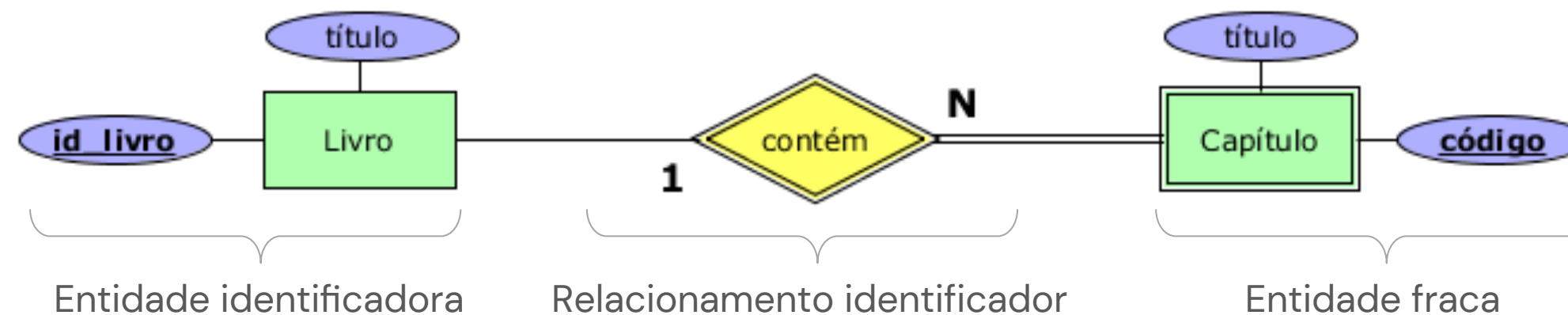
Atleta(id\_atleta)

# FASE 4: Modelação Lógica

## ➔ Derivar relações

### ▪ Entidades Fracas

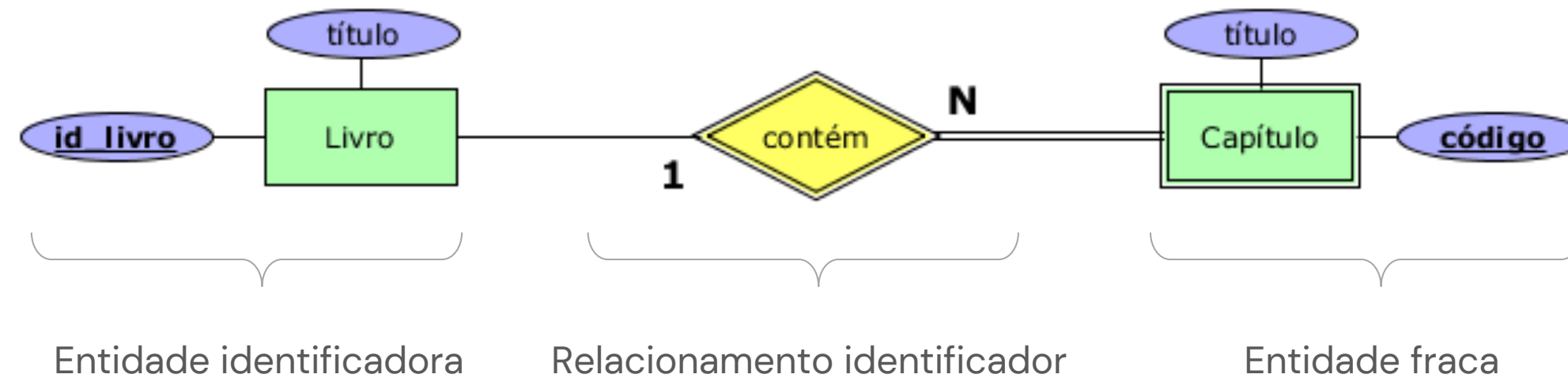
- Para cada entidade fraca do modelo de dados, crie uma relação que inclua todos os atributos simples dessa entidade.
- Se a entidade fraca não possuir atributos que possam constituir chaves candidatas, o conjunto de atributos que permitem identificar univocamente uma ocorrência da entidade fraca, é a **chave parcial** da entidade fraca;
- A chave primária de uma entidade fraca é sempre uma **chave composta** da chave primária da entidade identificadora e da sua chave parcial, portanto, a identificação da chave primária de uma entidade fraca não pode ser feita até que todos os relacionamentos com as entidades proprietárias tenham sido mapeados.



# FASE 4: Modelação Lógica

## ➔ Derivar relações

### ▪ Entidades Fracas



**Livro** (id\_livro, título)  
**Chave primária** id\_livro

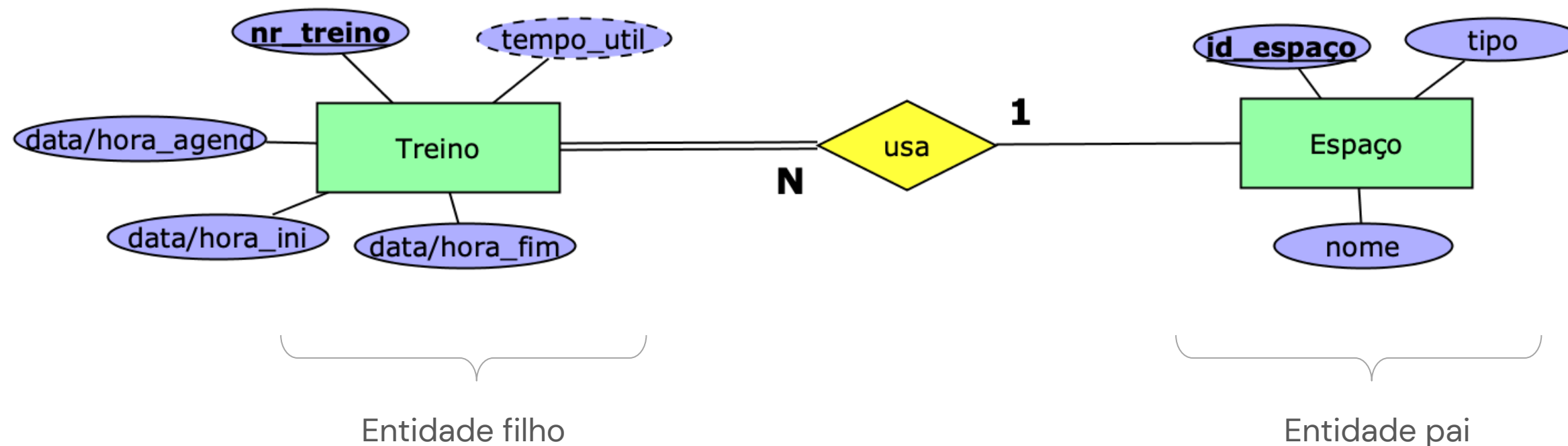
**Capítulo** (id\_livro, código, título)  
**Chave primária** id\_livro, código  
**Chave estrangeira** id\_livro **referencia** Livro(id\_livro)



# FASE 4: Modelação Lógica

## ➔ Derivar relações

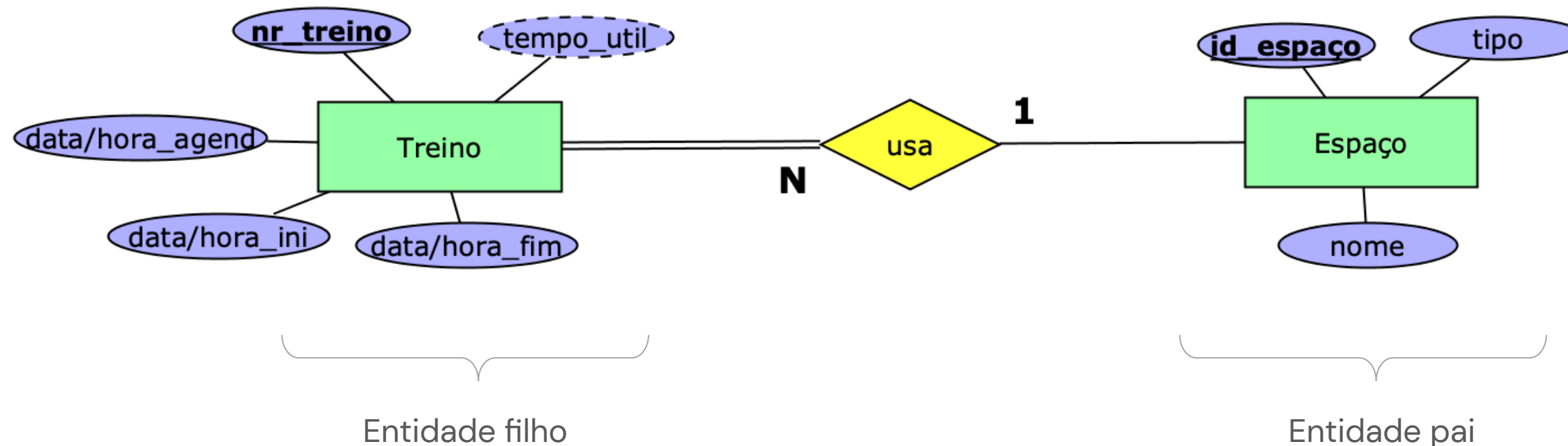
- Relacionamentos binários de um-para-muitos (1:N)
  - Para cada relacionamento binário 1:N, a entidade do lado 'um' do relacionamento é designada como a **entidade pai** e a entidade do lado 'muitos' é designada como a **entidade filho**.
  - Para representar esse relacionamento, cria-se uma **cópia** do(s) atributo(s) de **chave primária** da **entidade pai** na relação que representa a **entidade filho**, para atuar como **chave estrangeira**.



# FASE 4: Modelação Lógica

## ➔ Derivar relações

- Relacionamentos binários de um-para-muitos (1:N)



**Treino** (nr\_treino, tempo\_util, data/hora\_agend, data/hora\_ini, data/hora\_fim)

**Chave primária** nr\_treino

**Chave Estrangeira** id\_espaco **referencia**

Espaço(id\_espaco)

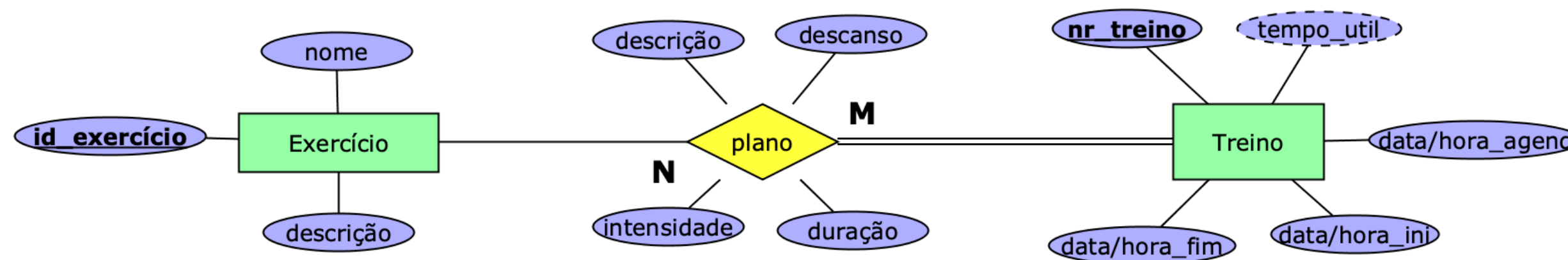
**Espaço** (id\_espaco, tipo, nome)

**Chave primária** id\_espaco

# FASE 4: Modelação Lógica

## ➔ Derivar relações

- Relacionamentos binários de muitos-para-muitos (N:M)
  - Crie uma relação para representar o relacionamento e inclua quaisquer atributos que façam parte do relacionamento.
  - Crie uma **cópia** do(s) atributo(s) de **chave primária** das **entidades** que participam no relacionamento na nova relação, para atuar como **chaves estrangeiras**. A **chave primária** da nova relação é sempre uma chave composta pelas chaves estrangeiras, possivelmente em combinação com outros atributos do relacionamento.

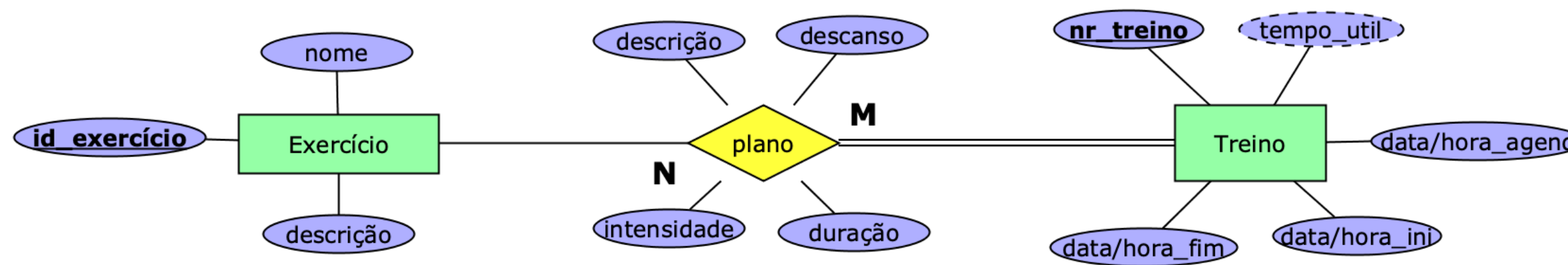




# FASE 4: Modelação Lógica

## ➔ Derivar relações

- Relacionamentos binários de muitos-para-muitos (N:M)



**Exercício** (id\_exercicio, nome, descrição)  
**Chave primária** id\_exercicio

**Treino** (nr\_treino, tempo\_util, data/hora\_ini, data/hora\_fim, data/hora\_agend)  
**Chave primária** nr\_treino

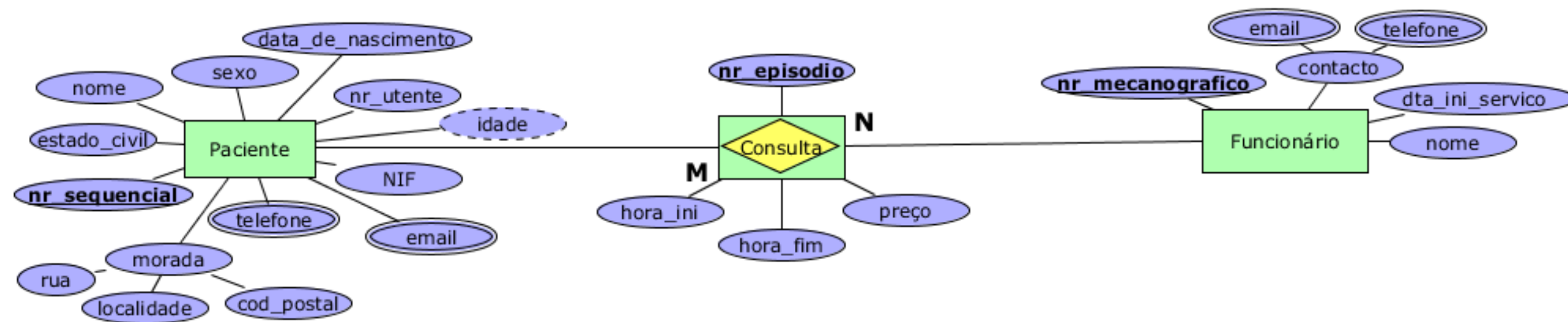
**Plano** (id\_exercicio, nr\_treino, descrição, descanso, intensidade, duração)  
**Chave primária** id\_exercicio, nr\_treino  
**Chave Estrangeira** id\_exercicio **referencia** Exercício(id\_exercicio)  
**Chave Estrangeira** nr\_treino **referencia** Treino(nr\_treino)

# FASE 4: Modelação Lógica

## ➔ Derivar relações

### ▪ Entidade Relacionamento

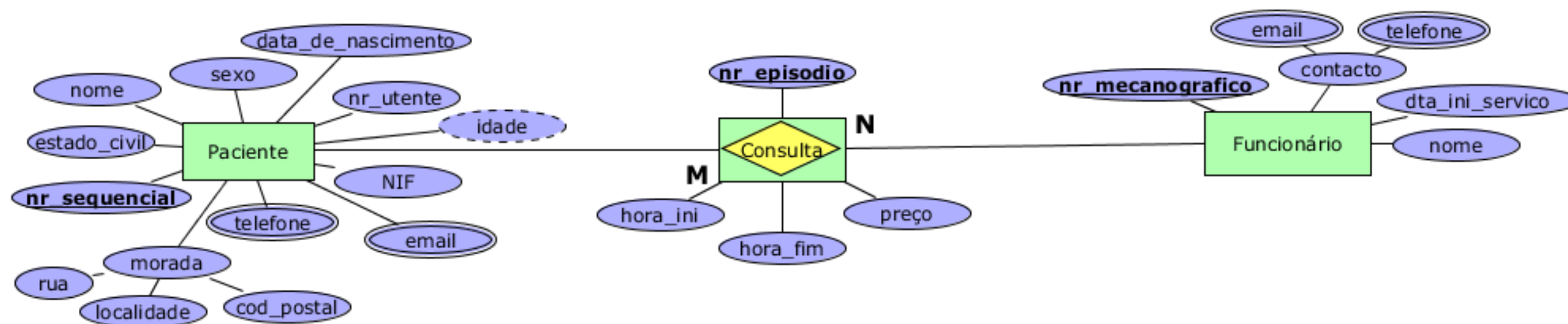
- Crie uma relação para representar a entidade-relacionamento como se fosse uma entidade independente e inclua todos os atributos que fazem parte da entidade-relacionamento.
- Crie uma **cópia** do(s) atributo(s) de **chave primária** das **entidades** que participam na entidade-relacionamento na nova relação, para atuar como **chaves estrangeiras**. Caso a entidade-relacionamento **não** possua chave primária, essas chaves estrangeiras formarão a **chave primária**.



# FASE 4: Modelação Lógica

## ➔ Derivar relações

### ▪ Entidade Relacionamento



**Paciente** (nr\_sequencial, nome, sexo, dta\_nascimento, rua, localidade, cod\_postal, NIF, nr\_utente, estado\_civil)

**Chave primária** nr\_sequencial

**Chave candidata** NIF

**Chave candidata** nr\_utente

**Derivado** idade(dta\_atual – dta\_nascimento)

**Consulta** (nr\_episodio, nr\_sequencial, nr\_mecanografico, hora\_ini, hora\_fim, preco)

**Chave primária** nr\_episodio

**Chave Estrangeira** nr\_sequencial **referencia** Paciente(nr\_sequencial)

**Chave Estrangeira** nr\_mecanografico **referencia** Funcionário(nr\_mecanografico)

**Funcionário** (nr\_mecanografico, nome, dta\_ini\_servico)

**Chave primária** nr\_mecanografico

# FASE 4: Modelação Lógica

## ➔ Derivar relações

- Relacionamentos binários de um-para-um (1:1)
  - Nestes casos, a criação de relações é mais complexa, porque a **cardinalidade** não pode ser usada para identificar as entidades pai e filho num relacionamento.
  - Em vez disso, as restrições de **participação** são usadas para decidir se é preferível combinar as entidades numa só relação ou se é mais adequado criar duas relações e colocar uma cópia da chave primária de uma relação na outra:
    - (a) participação obrigatória em ambos os lados do relacionamento 1:1;
    - (b) participação obrigatória num lado do relacionamento 1:1;
    - (c) participação opcional em ambos os lados do relacionamento 1:1.



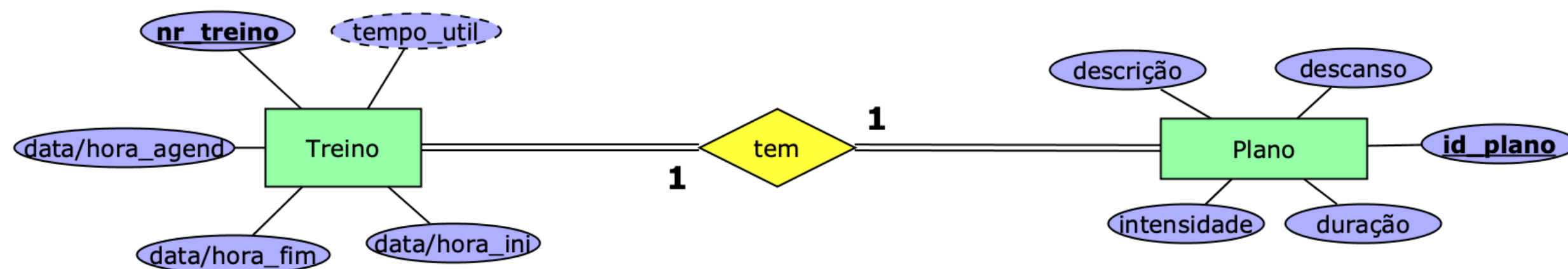
# FASE 4: Modelação Lógica

## ➔ Derivar relações

### ▪ Relacionamentos binários de um-para-um (1:1)

➔ (a) participação obrigatória em ambos os lados do relacionamento 1:1;

– Combinar as entidades envolvidas **numa só relação** e escolher uma das chaves primárias das entidades originais para ser a chave primária da nova relação, enquanto outra (se existir) é usada como chave candidata.



**Treino** (nr\_treino, tempo\_util, data/hora\_ini, data/hora\_fim, data/hora\_fim, id\_plano, descrição, descanso, intensidade, duração)

**Chave primária** nr\_treino

**Chave candidata** id\_plano

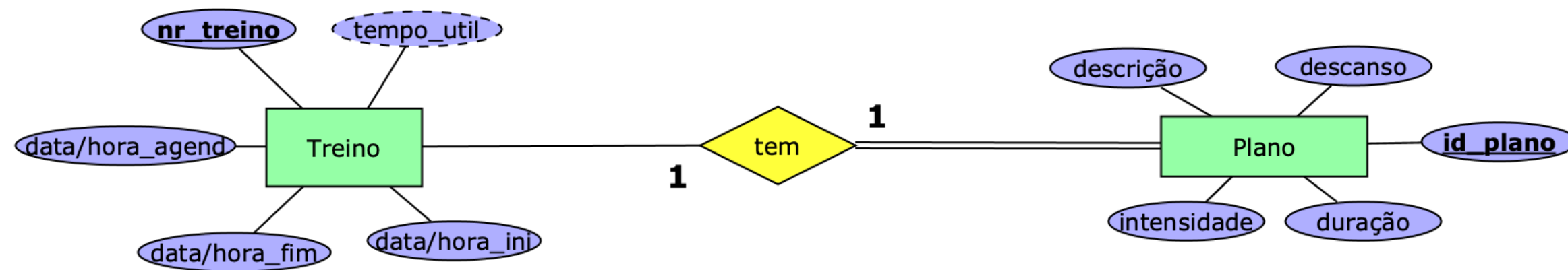
# FASE 4: Modelação Lógica

## ➔ Derivar relações

### ▪ Relacionamentos binários de um-para-um (1:1)

➔ (b) participação obrigatória num lado do relacionamento 1:1;

– Cópia da chave primária da entidade pai colocada na relação que representa a entidade filho.



**Treino** (nr\_treino, tempo\_util, data/hora\_ini, data/hora\_fim, data/hora\_fim)

**Chave primária** nr\_treino

**Plano** (id\_plano, descrição, descanso, intensidade, duração, nr\_treino)

**Chave primária** id\_plano

**Chave estrangeira** nr\_treino **referencia** Treino (nr\_treino)

# FASE 4: Modelação Lógica

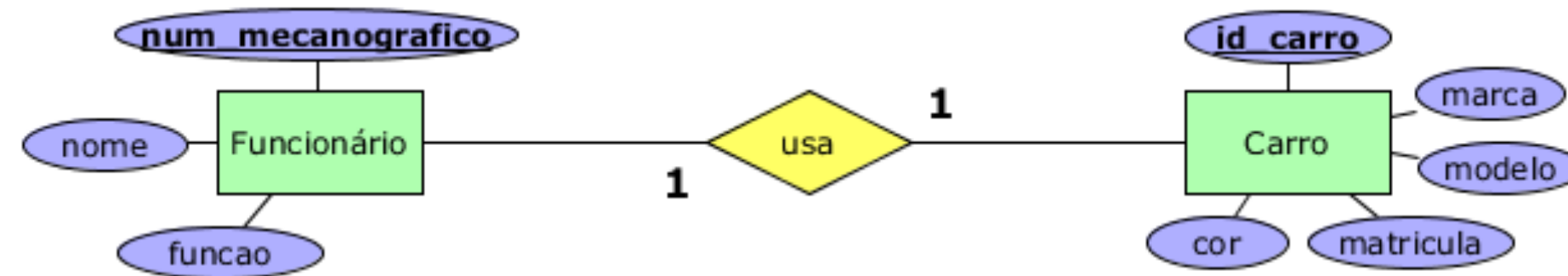
## ➔ Derivar relações

- Relacionamentos binários de um-para-um (1:1)

➔ (c) participação opcional em ambos os lados do relacionamento 1:1.

Cópia da chave primária da entidade pai colocada na relação que representa a entidade filho. A designação das entidades pai e filho é arbitrária, a menos que se possa descobrir mais sobre o relacionamento.

### EXEMPLO:



Suponha que a maioria dos carros, mas não todos, sejam usados pelos funcionários e que apenas uma minoria dos funcionários use carros. A entidade **Carro**, embora opcional, está mais próxima de ser obrigatória do que a entidade **Funcionário**. Portanto, neste caso deveríamos designar o **Funcionário** como **entidade-pai** e o **Carro** como **entidade-filho**.

# FASE 4: Modelação Lógica

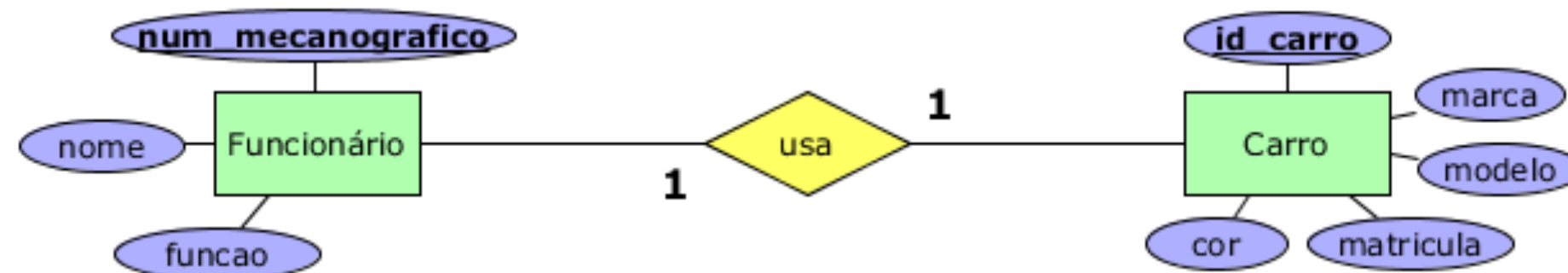
## ➔ Derivar relações

- Relacionamentos binários de um-para-um (1:1)

➔ (c) participação opcional em ambos os lados do relacionamento 1:1.

Cópia da chave primária da entidade pai colocada na relação que representa a entidade filho. A designação das entidades pai e filho é arbitrária, a menos que se possa descobrir mais sobre o relacionamento.

EXEMPLO:



**Funcionário** (num\_mecanografico, nome, funcao)  
**Chave primária** num\_mecanografico

**Carro** (id\_carro, marca, modelo, matricula, cor, num\_mecanografico)  
**Chave primária** id\_carro  
**Chave estrangeira** num\_mecanografico **referencia** Funcionário(num\_mecanografico)



# FASE 4: Modelação Lógica

## ➔ Derivar relações

Os relacionamentos recursivos de 1:N e N:M seguem as regras de participação de um relacionamento binário de 1:N e N:M, respectivamente.

- Relacionamentos binários recursivos de um-para-um (1:1)

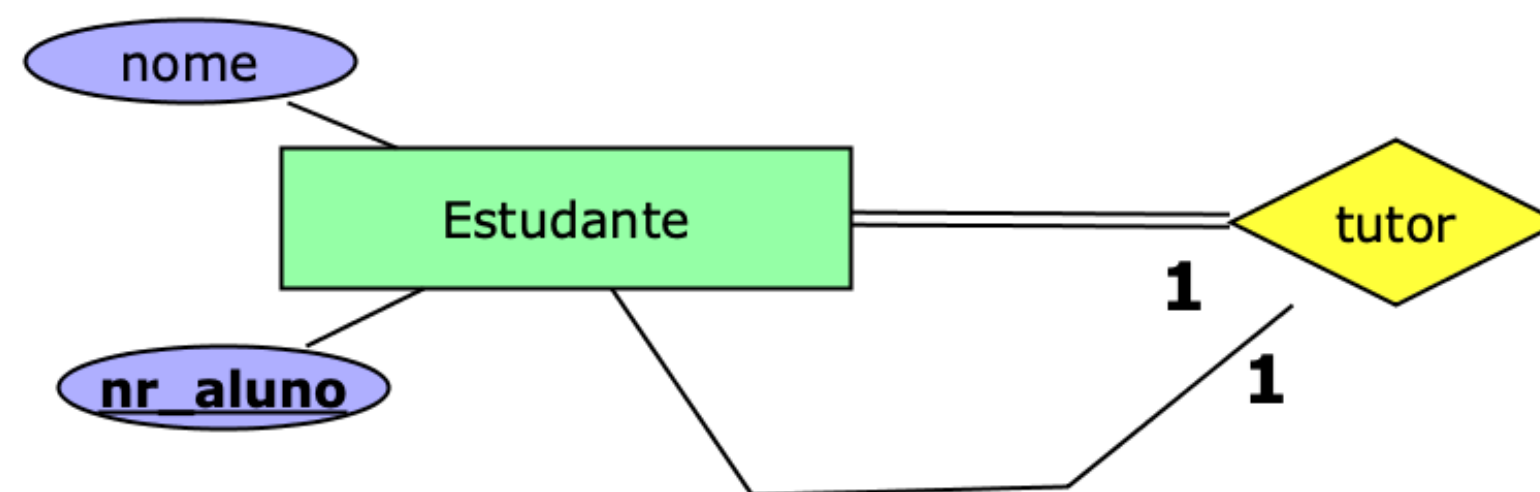
Os relacionamentos recursivos de 1:1 seguem as regras:

- **participação obrigatória de ambos os lados:** relação única com uma cópia da chave primária a agir como chave estrangeira que deve ser renomeada para facilitar a interpretação e não pode ser nula (semelhante ao relacionamento recursivo 1:N).
- **participação opcional de ambos os lados:** criar uma nova relação para representar o relacionamento recursivo que teria apenas dois atributos a funcionar com chave primária composta pelas duas chaves primárias que devem ser renomeadas para facilitar a interpretação e que agem também como chaves estrangeiras (semelhante ao relacionamento recursivo M:N).
- **participação obrigatória em apenas um lado:** opção de seguir qualquer uma das duas abordagens anteriores.

# FASE 4: Modelação Lógica

## ➔ Derivar relações

- Relacionamentos binários recursivos de um-para-um (1:1)



**Estudante** (nr\_aluno, nome, tutor)  
**Chave primária** nr\_aluno  
**Chave estrangeira** tutor **referencia**  
 Estudante(nr\_aluno)

ou

**Estudante** (nr\_aluno, nome)  
**Chave primária** nr\_aluno

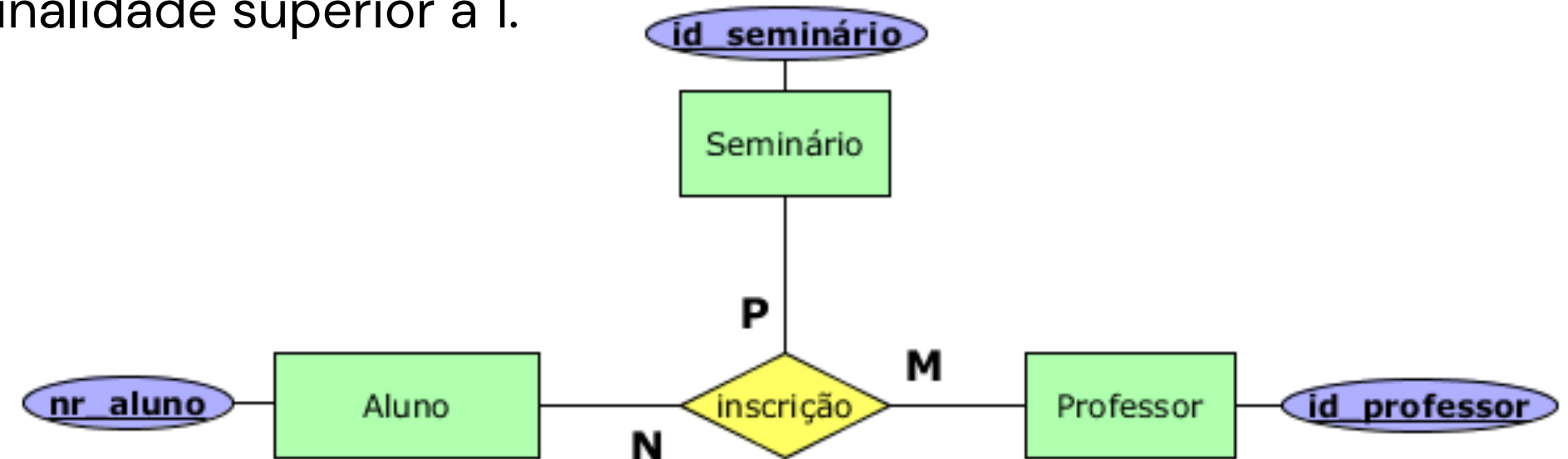
**Tutor** (nr\_aluno, nr\_aluno\_tutor)  
**Chave primária** nr\_aluno,  
 nr\_aluno\_tutor  
**Chave estrangeira** nr\_aluno\_tutor  
**referencia** Estudante(nr\_aluno)  
**Chave estrangeira** nr\_aluno  
**referencia** Estudante(nr\_aluno)

# FASE 4: Modelação Lógica

## ➔ Derivar relações

### ▪ Relacionamentos complexos

- Para cada relacionamento complexo, criar **uma relação** para representar o **relacionamento** e incluir quaisquer atributos que façam parte do relacionamento.
- Colocamos uma **cópia** da(s) **chave(s) primária(s)** das entidades que participam no relacionamento complexo na nova relação, para atuar como **chaves estrangeiras**.
- A determinação da chave primária da nova relação depende da cardinalidade do relacionamento complexo. Passa a ser composta pelas **chaves primárias** das entidades que participam no relacionamento complexo e que têm cardinalidade superior a 1.



# FASE 4: Modelação Lógica

## ➔ Derivar relações

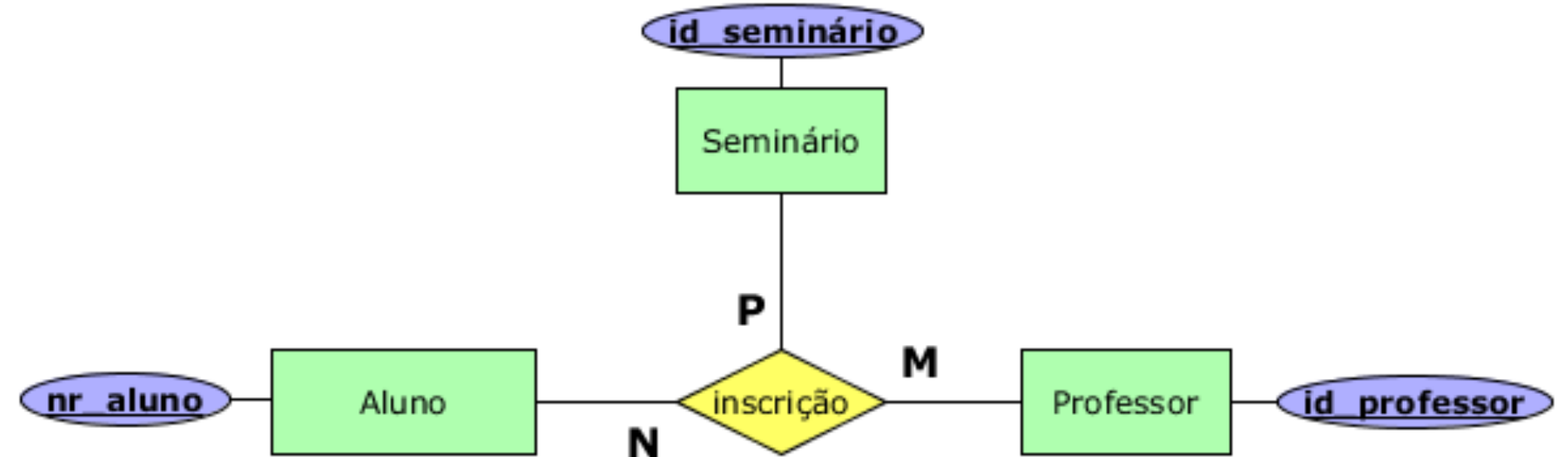
### ▪ Relacionamentos complexos N:M:P

- A nova relação tem uma chave primária composta pelas **chaves primárias** das entidades que participam no relacionamento complexo.

**Aluno** (nr\_aluno, ...)  
**Chave primária** nr\_aluno

**Professor** (id\_prof, ...)  
**Chave primária** id\_prof

**Seminário** (id\_semi, ...)  
**Chave primária** id\_semi



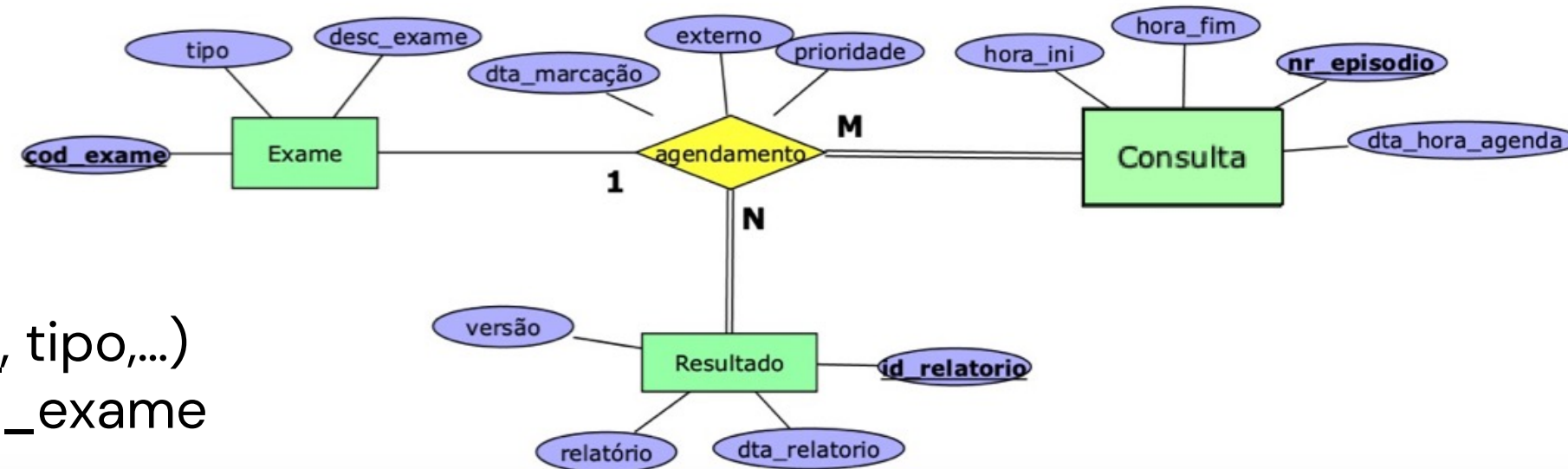
**Inscrição** (id\_semi, id\_prof, nr\_aluno)  
**Chave primária** id\_semi, id\_prof, nr\_aluno  
**Chave estrangeira** id\_semi **referencia** Seminário(id\_seminário)  
**Chave estrangeira** id\_prof **referencia** Professor(id\_prof)  
**Chave estrangeira** nr\_aluno **referencia** Aluno(nr\_aluno)

# FASE 4: Modelação Lógica

## ➔ Derivar relações

### ▪ Relacionamentos complexos 1:N:M

- A nova relação tem uma chave primária composta pelas chaves primárias das entidades que participam no relacionamento complexo com cardinalidade N.



**Exame** (cod\_exame, tipo,...)  
**Chave primária** cod\_exame

**Consulta** (nr\_episodio, hora\_ini, ...)  
**Chave primária** nr\_episodio

**Resultado** (id\_relatorio, relatório, ...)  
**Chave primária** id\_relatorio

**Agendamento** (nr\_episodio, id\_relatorio, cod\_exame)  
**Chave primária** nr\_episodio, id\_relatorio  
**Chave estrangeira** nr\_episodio **referencia** Consulta(nr\_episodio)  
**Chave estrangeira** id\_relatorio **referencia** Resultado(id\_relatorio)  
**Chave estrangeira** cod\_exame **referencia** Exame(cod\_exame)

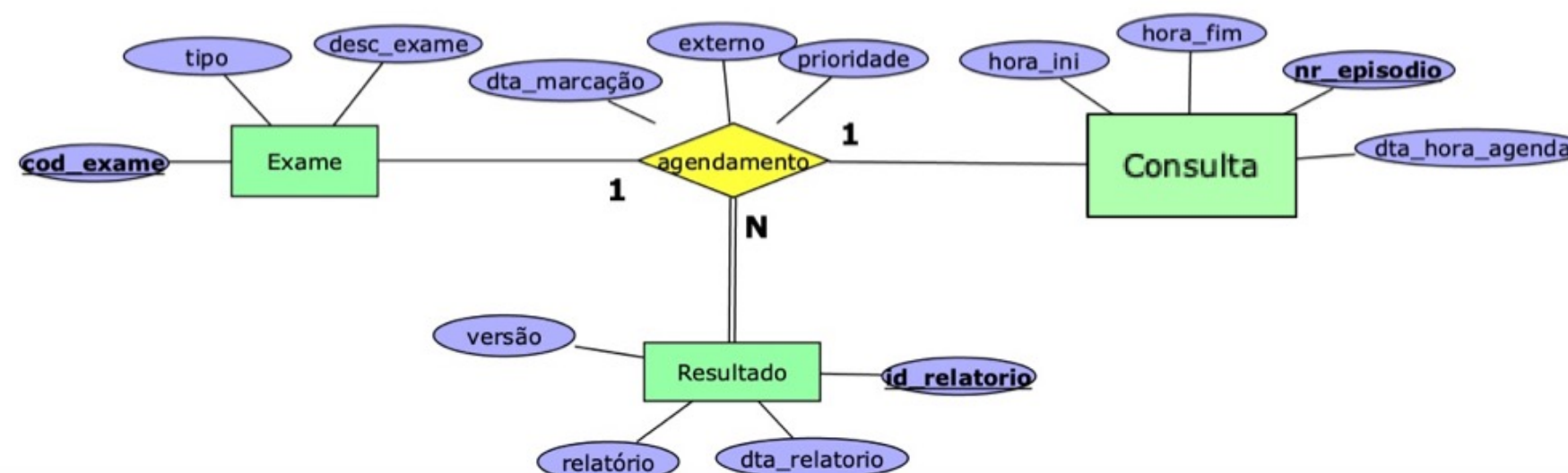


# FASE 4: Modelação Lógica

## ➔ Derivar relações

### ▪ Relacionamentos complexos 1:1:N

- A nova relação tem uma chave primária composta pela chave primária da entidade que participa no relacionamento complexo com cardinalidade N e a chave primária de uma das outras duas entidades, definida de forma arbitrária. Para além disso, o outro par deve ser único.



**Agendamento** (nr\_episodio, id\_relatorio, cod\_exame)

**Chave primária** nr\_episodio, id\_relatorio

**Chave estrangeira** nr\_episodio **referencia** Consulta(nr\_episodio)

**Chave estrangeira** id\_relatorio **referencia** Resultado(id\_relatorio)

**Chave estrangeira** cod\_exame **referencia** Exame(cod\_exame)

ou

**Agendamento** (nr\_episodio, cod\_exame, id\_relatorio)

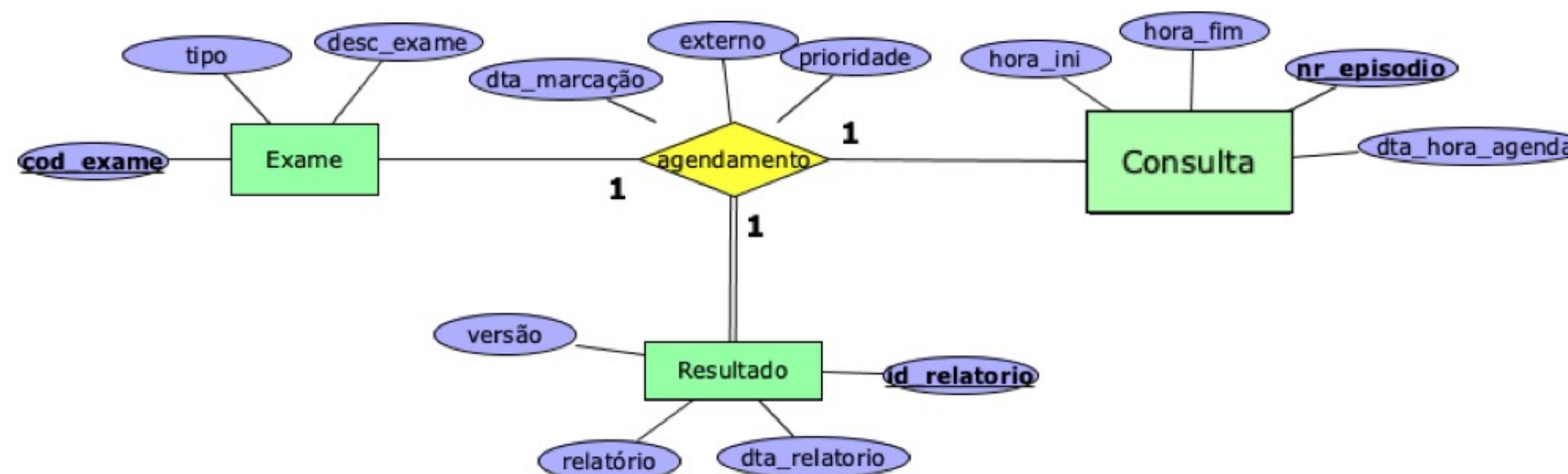
**Chave primária** nr\_episodio, cod\_exame

# FASE 4: Modelação Lógica

## ➔ Derivar relações

### ▪ Relacionamentos complexos 1:1:1

- A nova relação tem uma chave primária composta pelas chaves primárias de duas das entidades que participam no relacionamento complexo, definidas de forma arbitrária. Para além disso, o outro par deve ser único.



**Agendamento** (nr\_episodio, id\_relatorio, cod\_exame)  
**Chave primária** nr\_episodio, id\_relatorio  
**Chave estrangeira** nr\_episodio **referencia** Consulta(nr\_episodio)  
**Chave estrangeira** id\_relatorio **referencia** Resultado(id\_relatorio)  
**Chave estrangeira** cod\_exame **referencia** Exame(cod\_exame)

ou

**Agendamento** (nr\_episodio, cod\_exame, id\_relatorio)  
**Chave primária** nr\_episodio, cod\_exame

ou

**Agendamento** (id\_relatorio, cod\_exame, nr\_episodio)  
**Chave primária** id\_relatorio, cod\_exame

# FASE 4: Modelação Lógica

## ➔ Derivar relações

- Relacionamentos superclasse/subclasse
  - Identifique a **superclasse** como entidade pai e a **subclasse** como entidade filho.
  - A representação mais adequada de um relacionamento deste tipo depende do número de:
    - restrições de disjunção e participação no relacionamento superclasse/subclasse;
    - se as subclasses estão envolvidas em relacionamentos distintos;
    - número de participantes no relacionamento superclasse/subclasse.

# FASE 4: Modelação Lógica

## ➔ Derivar relações

Restrições de Participação	Restrições de Disjunção	Relações Requeridas
Obrigatória	Não disjunto {And}	Relação <b>única</b> com um <b>atributo</b> para cada subclasse (flag)
Opcional	Não disjunto {And}	<b>Duas</b> relações: uma relação para a <b>superclasse</b> e uma relação para <b>todas as subclASSES</b> com um <b>atributo</b> para cada subclasse (flag)
Obrigatória	Disjunto {Or}	<b>Muitas</b> relações (uma relação para <b>cada</b> combinação superclasse/subclasse)
Opcional	Disjunto {Or}	<b>Muitas</b> relações (uma relação para a <b>superclasse</b> e uma para <b>cada subclasse</b> )

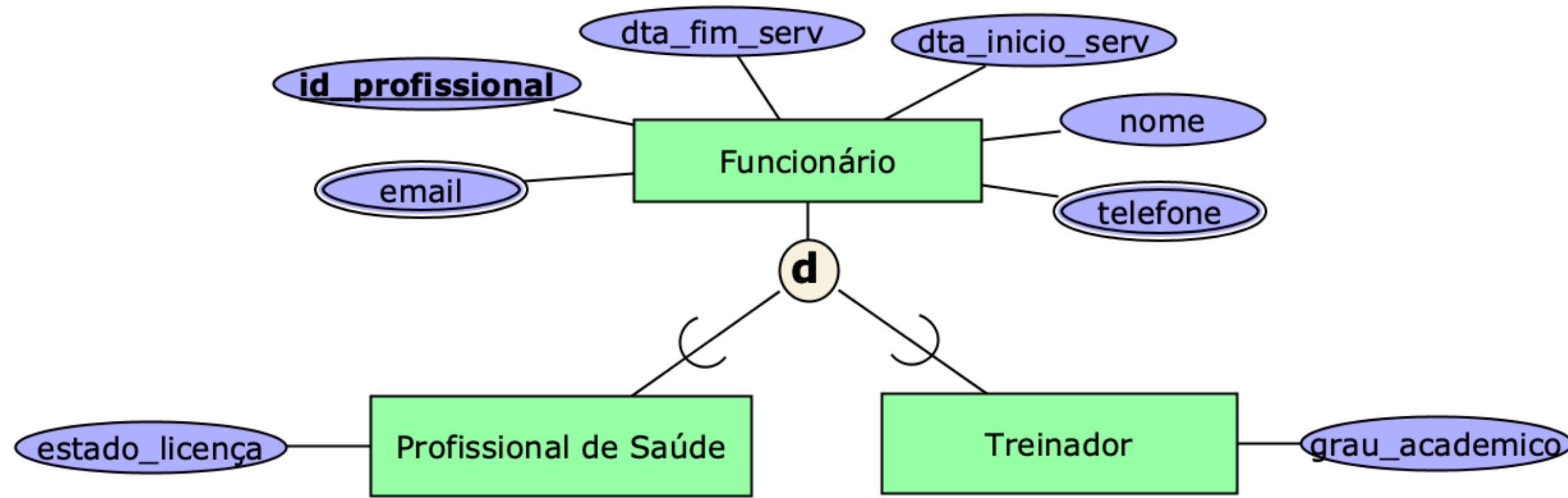
# FASE 4: Modelação Lógica

## ➔ Derivar relações

Opcional

Disjunto {Or}

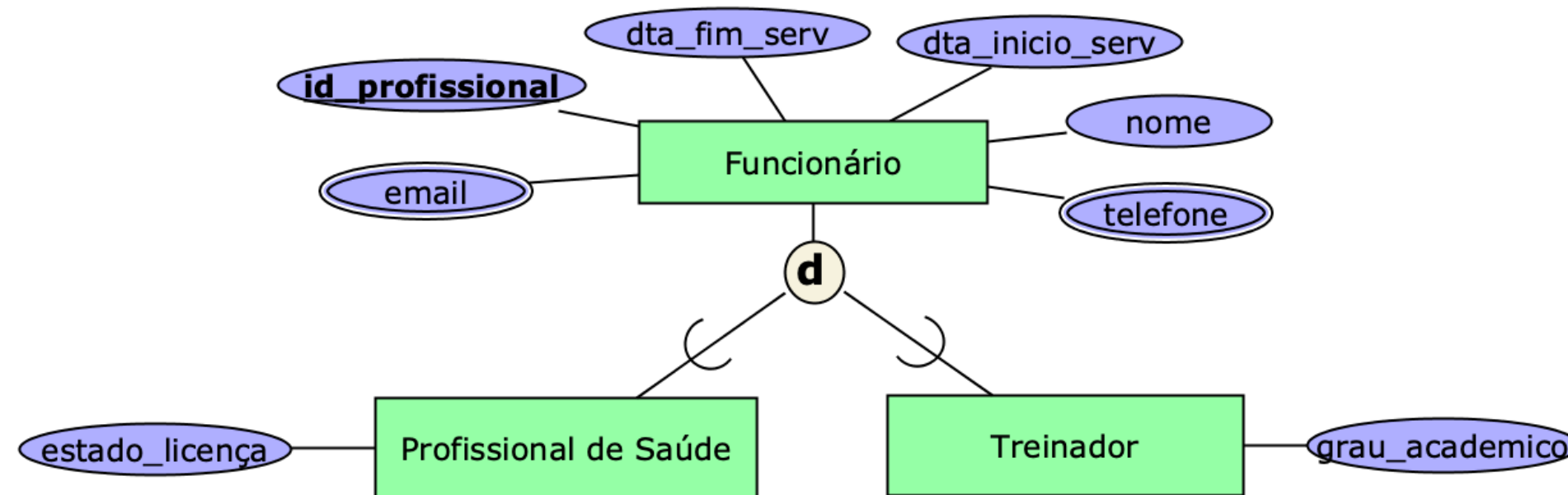
Muitas relações (uma relação para a superclasse e uma para cada subclasse)





# FASE 4: Modelação Lógica

## ➔ Derivar relações



**Funcionário** (nr\_mecanografico, nome, dta\_ini\_serviço ...)

**Chave primária** nr\_mecanografico

**Profissional de Saúde** (nr\_mecanografico, estado\_licença)

**Chave primária** nr\_mecanografico

**Chave estrangeira** nr\_mecanografico **referencia** Funcionário(nr\_mecanografico)

**Trreinador** (nr\_mecanografico, grau\_academico)

**Chave primária** nr\_mecanografico

**Chave estrangeira** nr\_mecanografico **referencia** Funcionário(nr\_mecanografico)

# FASE 4: Modelação Lógica

## ➔ Exercícios

### Ficha de Exercícios PL05:

Questão 1

# Material p/ a aula

MySQL Workbench + MySQL Community Server

<https://dev.mysql.com/downloads/workbench/>

*Windows*

<https://dev.mysql.com/downloads/installer/>

<https://dev.mysql.com/doc/mysql-installation-excerpt/5.7/en/windows-installation.html>

*Linux*

<https://www.digitalocean.com/community/tutorials/how-to-install-mysql-on-ubuntu-22-04>

*MacOS*

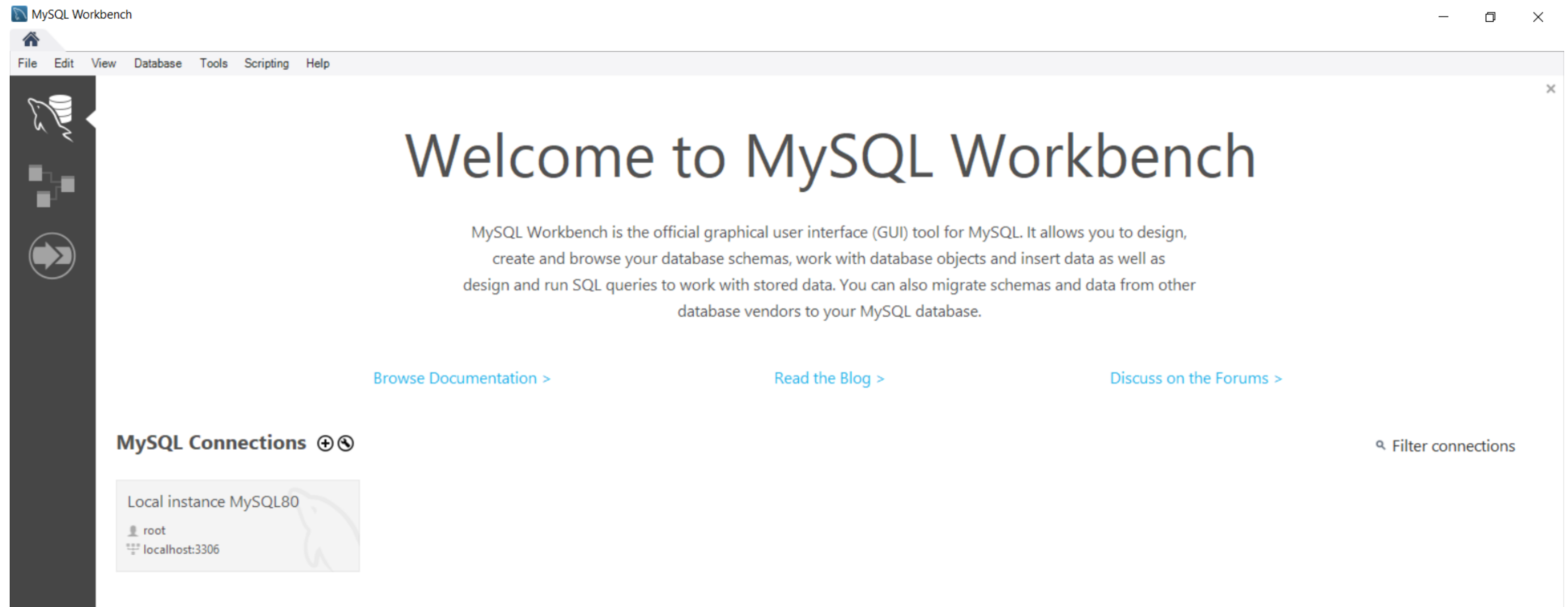
<https://dev.mysql.com/downloads/mysql/>

<https://dev.mysql.com/doc/refman/8.0/en/macos-installation.html>

# FASE 4: Modelação Lógica

## ➔ MySQL Workbench

- 1) Após a instalação, o GUI vai abrir com a configuração ao MySQL server já efetuada (assinalado na figura)



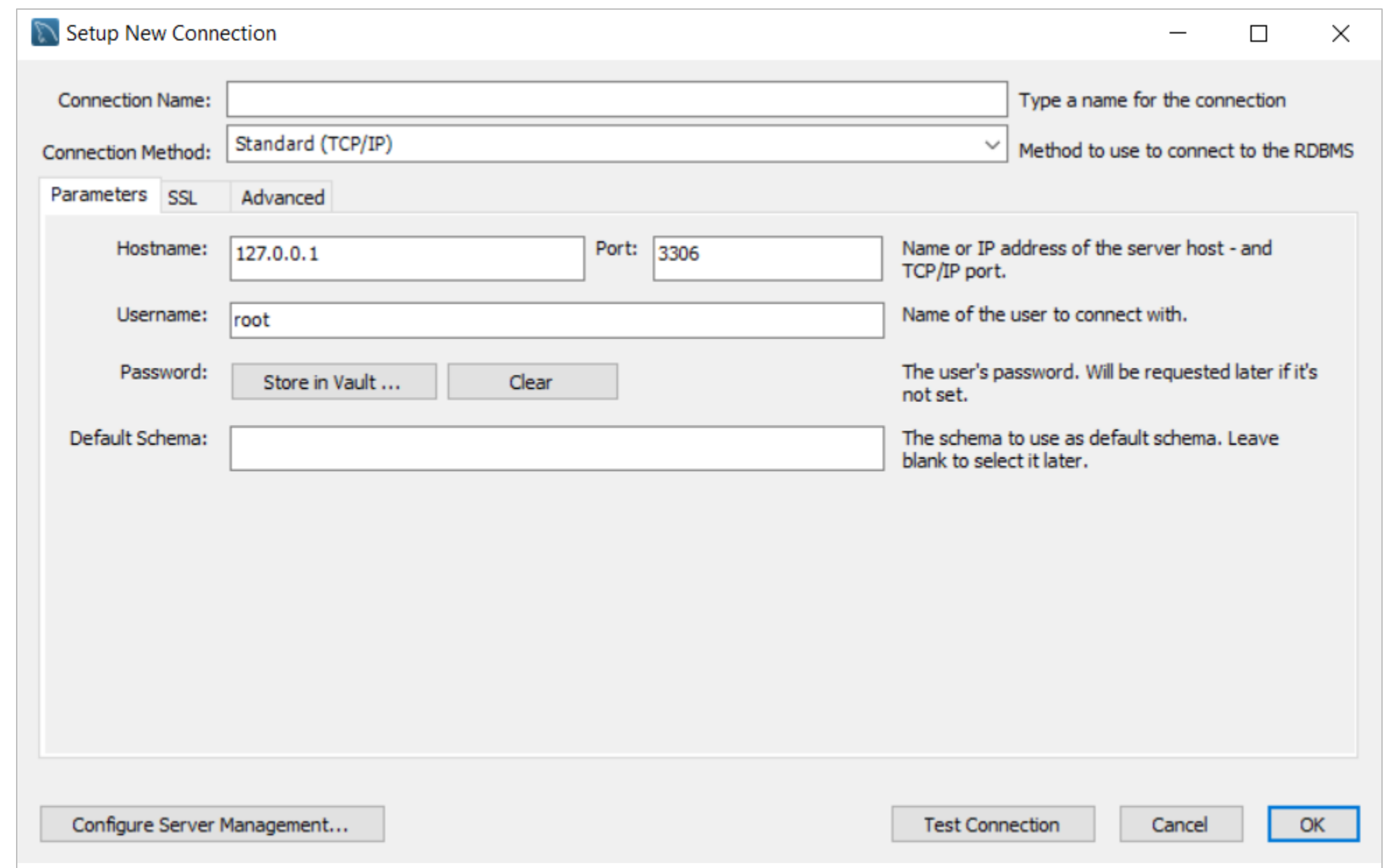
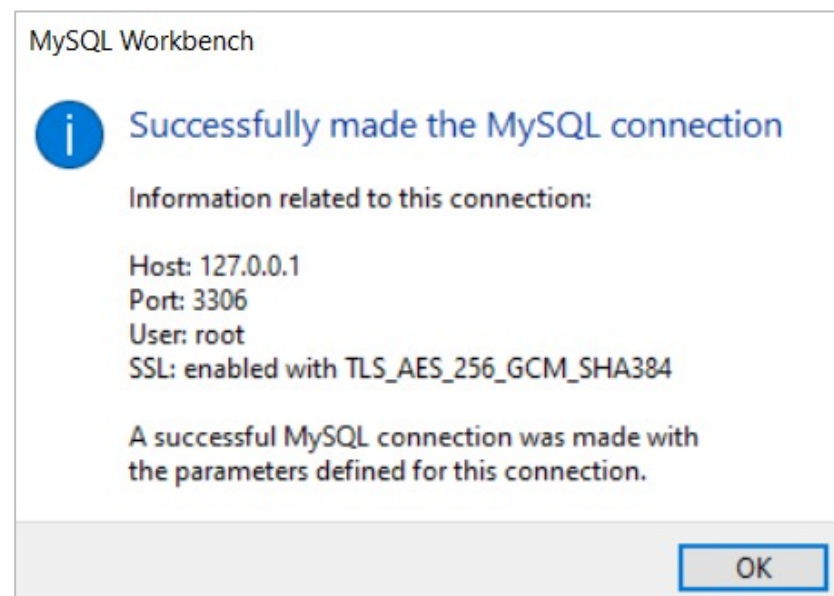
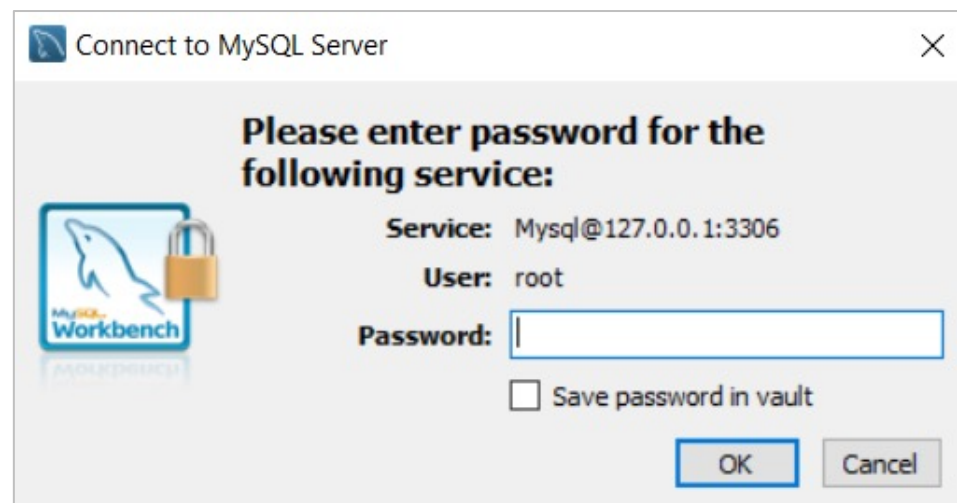
**NOTA:** se a conexão não aparecer, é provável que falte ou tenha falhado alguma etapa do guia de instalação

# FASE 4: Modelação Lógica

## ➔ MySQL Workbench

2) Para criar uma nova conexão:

- clicar no botão + junto de 'MySQL Connections'.
- configurar a nova conexão;
- testar a conexão ao MySQL Server;
- inserir a palavra-passe.

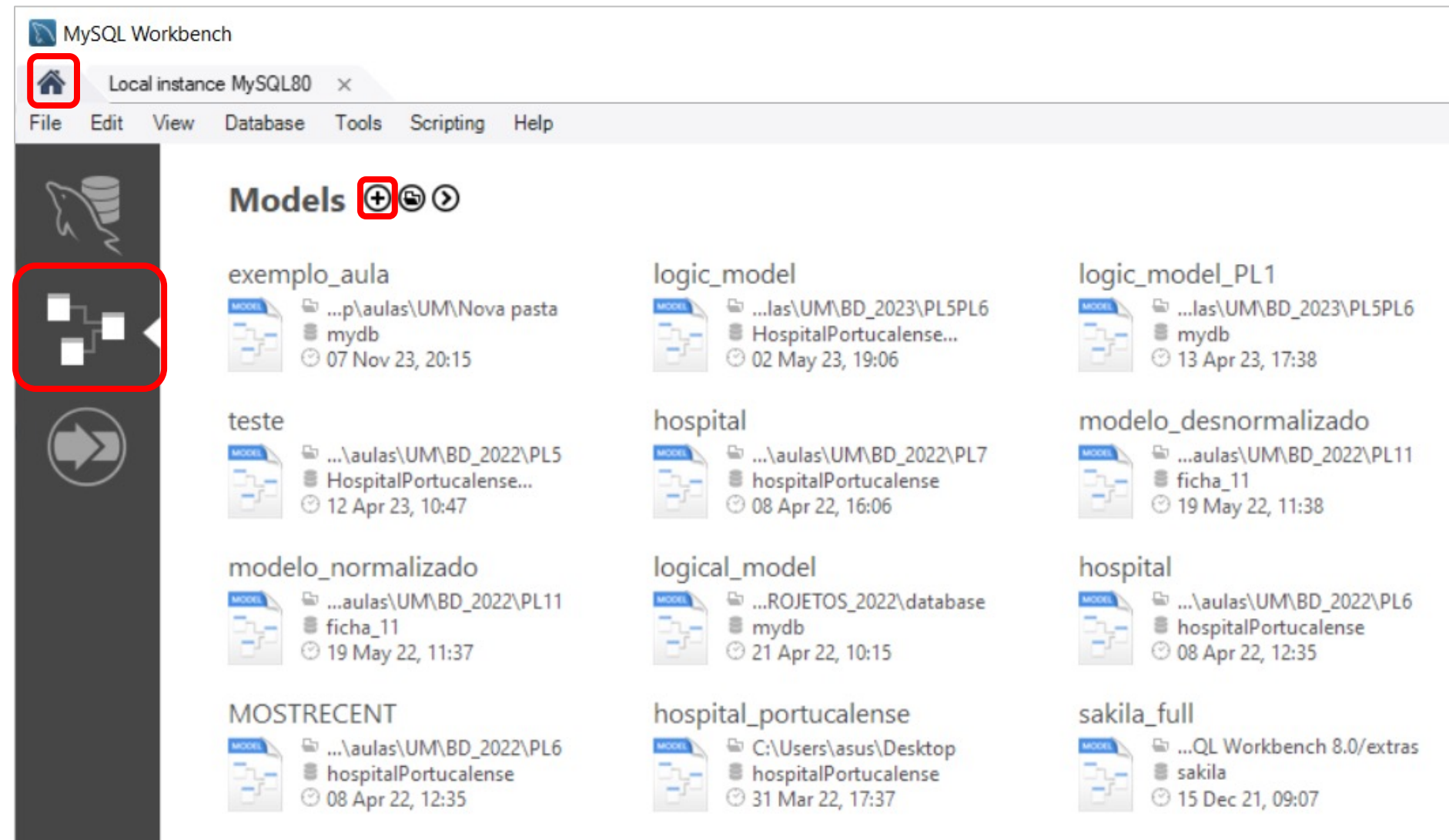




# FASE 4: Modelação Lógica

## ➔ MySQL Workbench

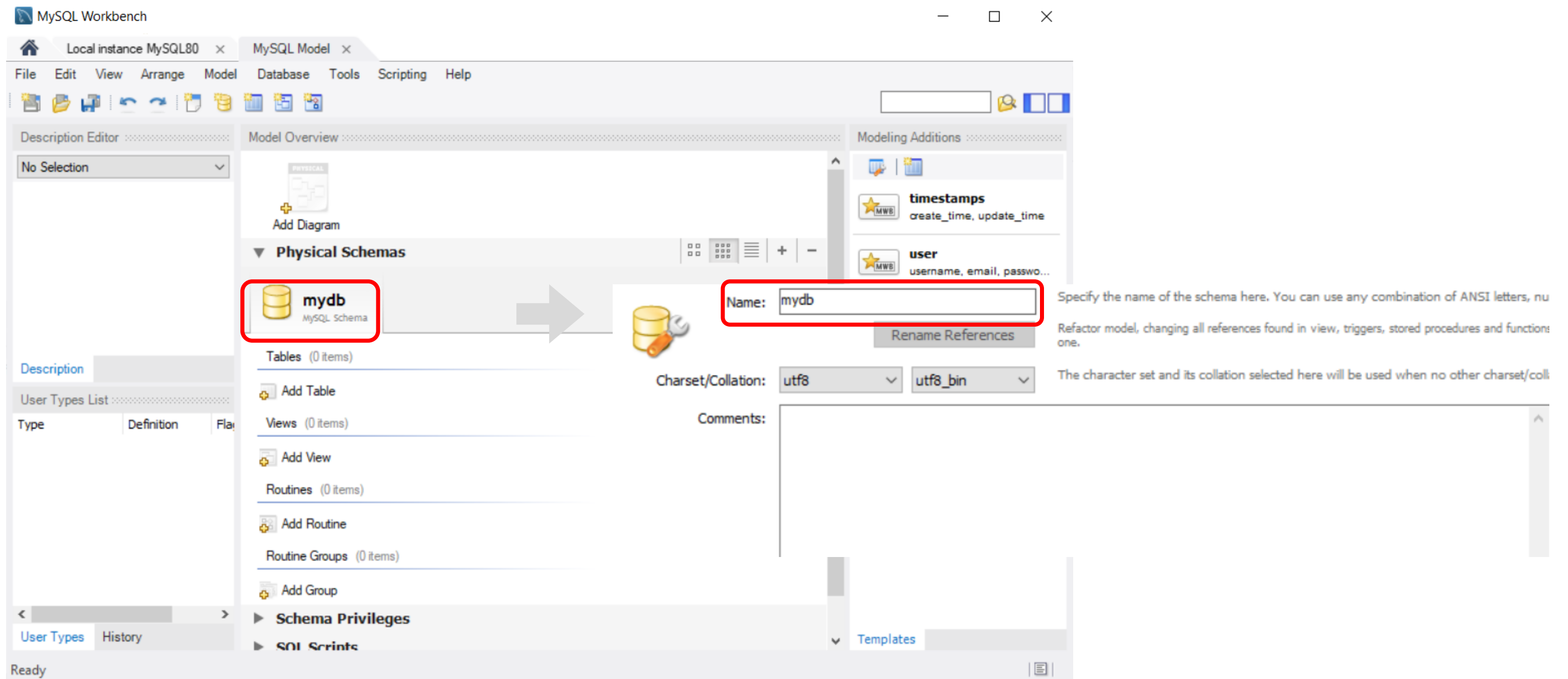
Ir ao menu inicial e clicar no separador “Models” para criar um novo esquema.



# FASE 4: Modelação Lógica

## ➔ MySQL Workbench

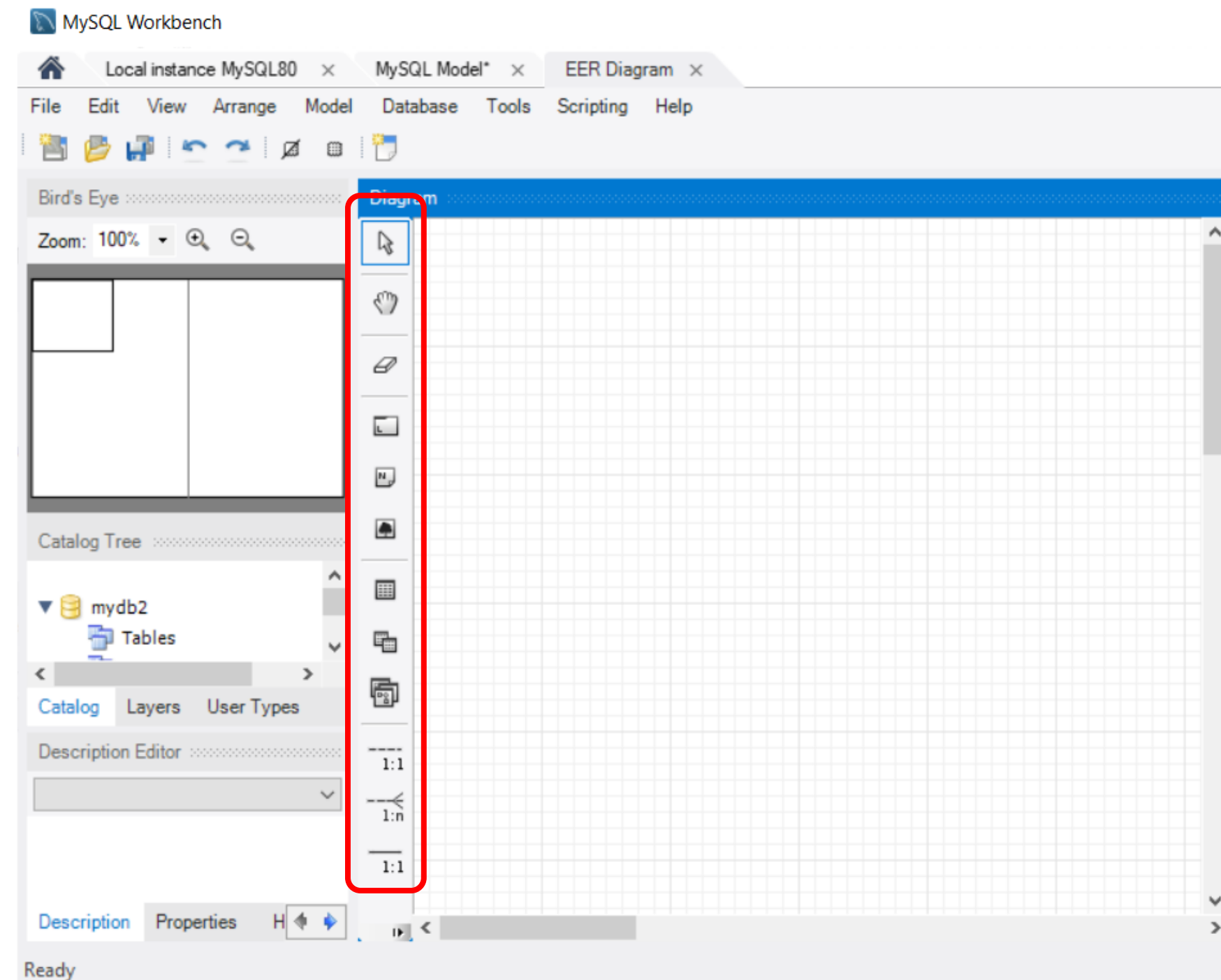
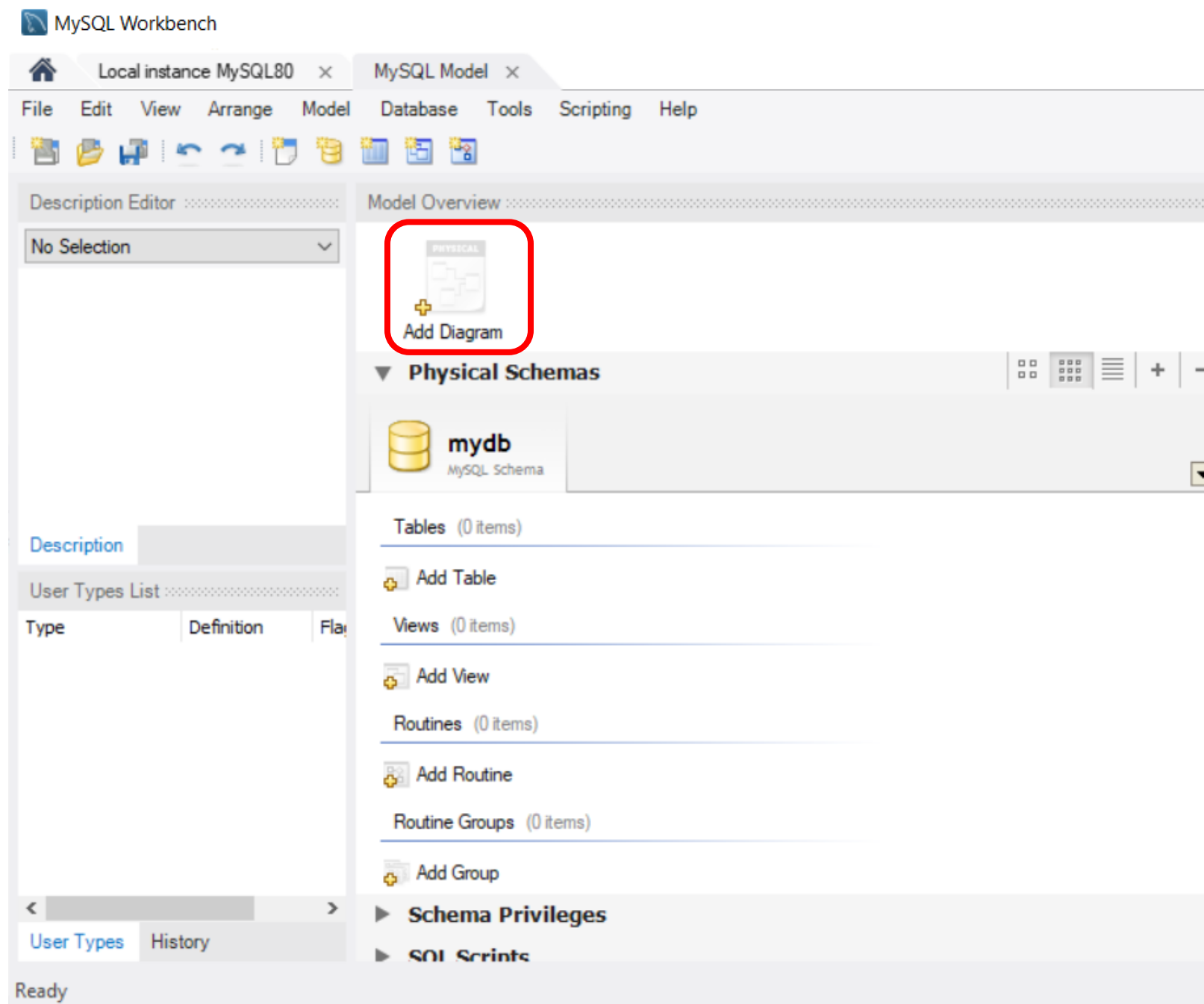
O Workbench cria então um novo esquema com o nome 'mydb'. Para alterar o nome do esquema, basta clicar duas vezes em cima de 'mydb'



# FASE 4: Modelação Lógica

## ➔ MySQL Workbench

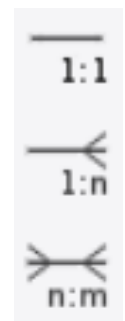
Depois de configurar o nome, clicar no botão 'Add Diagram'. Uma nova janela é criada chamada 'EER Diagram'.



# Modelação Lógica – MySQL

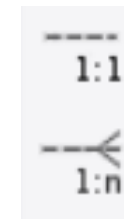
Quando estamos a construir o modelo lógico de dados no MySQL, é importante ter em consideração os seguintes aspetos:

- Tipo de relacionamento:



**Relacionamentos identificadores** (linha cheia) Quando a chave primária da entidade pai é incluída na chave primária da entidade filho.

🔑 – Chave estrangeira e chave primária.



**Relacionamentos não identificadores** (linha tracejada)

Quando a chave primária da entidade pai é incluída na entidade filho, mas não como parte da sua chave primária.

🔑 – Chave estrangeira NOT NULL – participação obrigatória no modelo conceptual

◊ – Chave estrangeira – participação opcional no modelo conceptual

- Direcção do relacionamento:

Os relacionamentos devem começar na relação/tabela que deve alocar a chave estrangeira.

# Modelação Lógica – MySQL

- Valores padrão/por defeito: Devem ser usados caso se queira considerar um valor por *default*.

Default/Expression

estado\_civil CHAR(1) ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ 'S'

Data Type: CHAR(1)

Default: 'S'

estado\_civil CHAR(1) NULL DEFAULT 'S',

- PK (Primary Key), NN (Not Null), UQ (Unique Index), B (Binary), UN (Unsigned), ZF (Zero Fill), AI (auto increment), G (generated)
  - PK – deve ser usado para atributos que são chave primária;
  - NN – deve ser usado em todos os atributos de chave primária e todos os atributos que não possam ser NULL;
  - UQ – deve ser aplicado sempre que há chaves candidatas, faz com que não hajam valores duplicados na tabela;
  - UN – define que não podem ser inseridos valores negativos nessa coluna.
  - ZF – preenche o valor definido para o campo com zeros até a largura de exibição especificada na definição da coluna.
  - AI – deve ser usado para gerar automaticamente quando um novo registo é inserido numa tabela.
  - G – deve ser usado para gerar atributos a partir de outros usando uma expressão.



# Tipos de Dados no MySQL

## ➔ Dados Alfanuméricos

<https://dev.mysql.com/doc/refman/8.0/en/data-types.html>

**VARCHAR** (strings de tamanho **variável**) vs. **CHAR** (strings de tamanho **fixo**)

Valor	CHAR(4)		VARCHAR(4)	
"	'____'	4 bytes	"	1 byte
'AB'	'AB__'	4 bytes	'AB'	3 bytes
'ABC'	'ABC_'	4 bytes	'ABC'	4 bytes
'ABCD'	'ABCD'	4 bytes	'ABCD'	5 bytes

O VARCHAR usa 1 ou 2 bytes de memória adicionais para tamanho ou para marcar o fim dos dados.

Para armazenar textos mais longos:

- TEXT
- TINYTEXT
- MEDIUMTEXT
- LONGTEXT

# Tipos de Dados no MySQL

## ➔ Dados Alfanuméricos

<https://dev.mysql.com/doc/refman/8.0/en/data-types.html>

- O tipo **ENUM** é um objeto de string cujo valor é seleccionado a partir de um conjunto de valores permitidos que são definidos explicitamente no momento de criação da coluna.

### EXEMPLO:

prioridade ENUM('Não Urgente', 'Pouco Urgente', 'Urgente', 'Muito Urgente', 'Emergente') NOT NULL );

A coluna prioridade aceitará apenas a inserção de um dos cinco valores definidos. O MySQL mapeia cada membro de enumeração para um índice numérico. Neste caso, 'Não Urgente', 'Pouco Urgente', 'Urgente', 'Muito Urgente' e 'Emergente' são mapeados para 1, 2, 3, 4 e 5 respectivamente.

- O tipo **SET** é um objeto string que pode ter zero ou mais valores, cada um dos quais deve ser escolhido a partir de um conjunto de valores especificados quando a tabela é criada.

### EXEMPLO:

tipo SET('A', 'B') NOT NULL );

A coluna tipo aceitará a inserção de "", 'A', 'B' ou 'A,B'. O MySQL armazena valores SET numericamente, com o bit de ordem inferior do valor armazenado correspondendo ao primeiro membro do conjunto.

# Tipos de Dados no MySQL

## ➔ Dados de Data/Hora

<https://dev.mysql.com/doc/refman/8.0/en/data-types.html>

Tipo de Dados	Notação
<u>DATE</u>	YYYY-MM-DD
<u>TIME</u>	hh:mm:ss
<u>DATETIME</u> *	YYYY-MM-DD hh:mm:ss
<u>TIMESTAMP</u> **	YYYY-MM-DD hh:mm:ss
<u>YEAR</u>	YYYY

\* O intervalo suportado varia de '1000-01-01 00:00:00' a '9999-12-31 23:59:59'.

\*\* O intervalo suportado varia de '1970-01-01 00:00:01' a '2038-01-19 03:14:07'.

# Tipos de Dados no MySQL

## ➔ Dados de Data/Hora



<https://dev.mysql.com/doc/refman/8.0/en/data-types.html>

- Fixed-Point Types (Exact Value) – DECIMAL

Os tipos DECIMAL armazenam valores de dados numéricos exatos. Este tipo de dados é usado quando é importante preservar a precisão exata, por exemplo, com dados monetários.

DECIMAL(n,m)

n – precisão – representa o número de dígitos significativos que são armazenados.

m – escala – representa o número de dígitos que podem ser armazenados após o ponto decimal.

**Exemplo:** 105,98€ → DECIMAL (5,2)

# Bases de Dados

PL05 – Modelação Lógica

**Docente:** Cristiana Neto

**Email:** [cristiana.neto@algoritmi.uminho.pt](mailto:cristiana.neto@algoritmi.uminho.pt)

**Horário de Atendimento:**

6ª feira 09h–10h





# FASE 4: Modelação Lógica

## ➔ Exercícios

Ficha de Exercícios PL05:

Questão 2