

Bases de Dados

PL05/PL06 – Modelação Lógica

Docente: Diana Ferreira

Email: diana.ferreira@algoritmi.uminho.pt

Horário de Atendimento:

4ª feira 18h–19h



Sumário

1

Revisão do Modelo Conceptual

2

Instalação do MySQL Workbench

3

Regras de Derivação

4

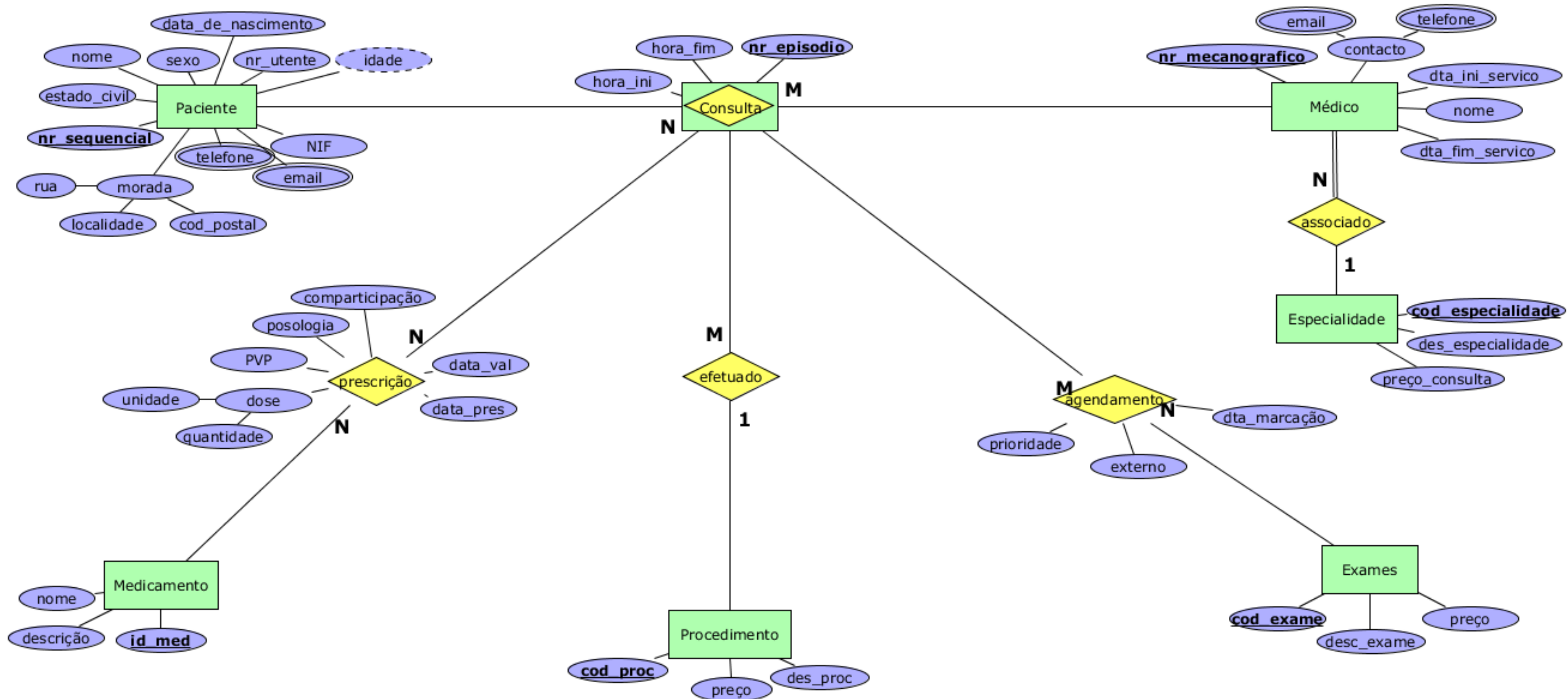
Modelação Lógica

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.

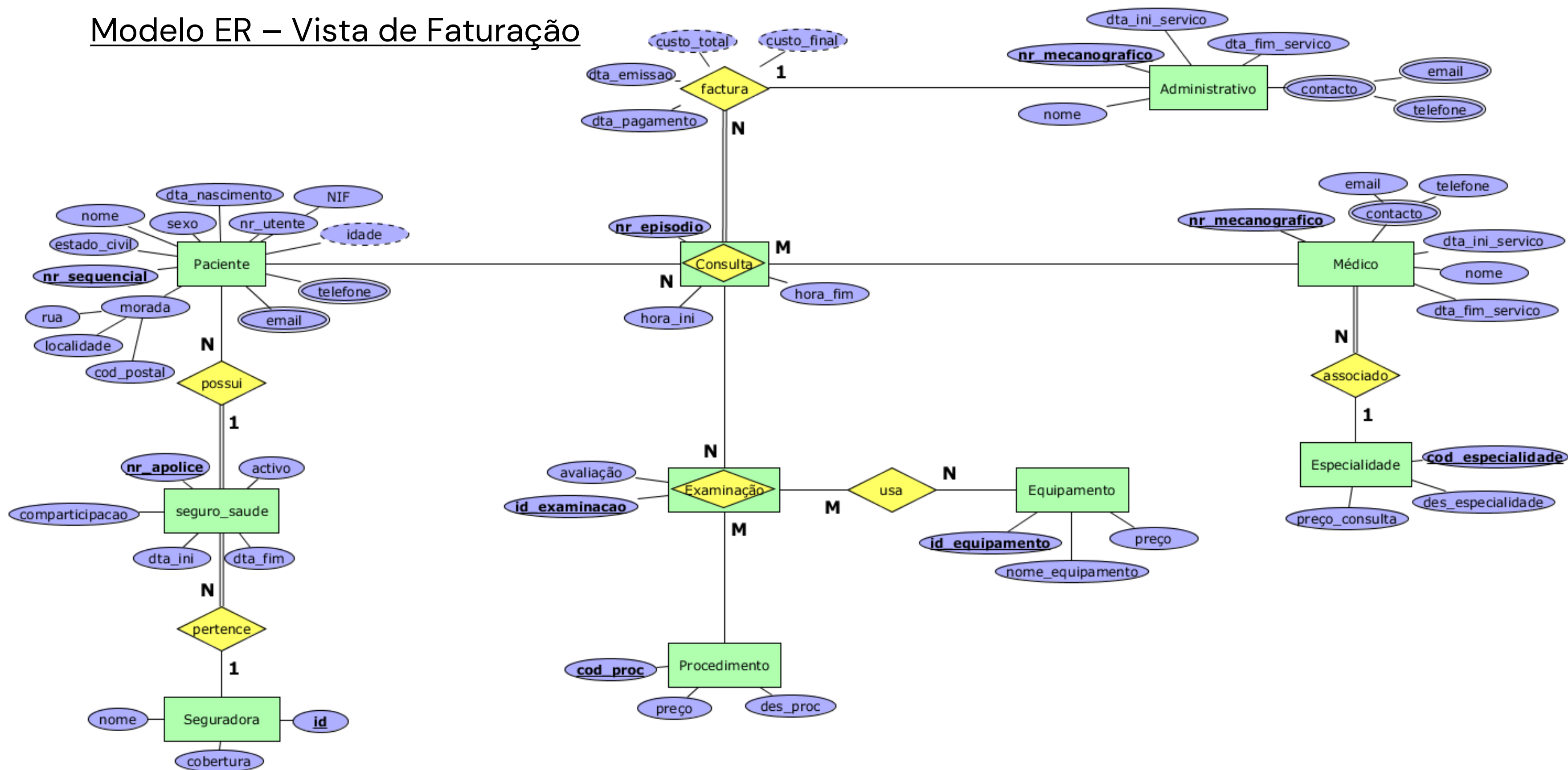
Revisão da aula anterior

Modelo ER – Vista de Consultas



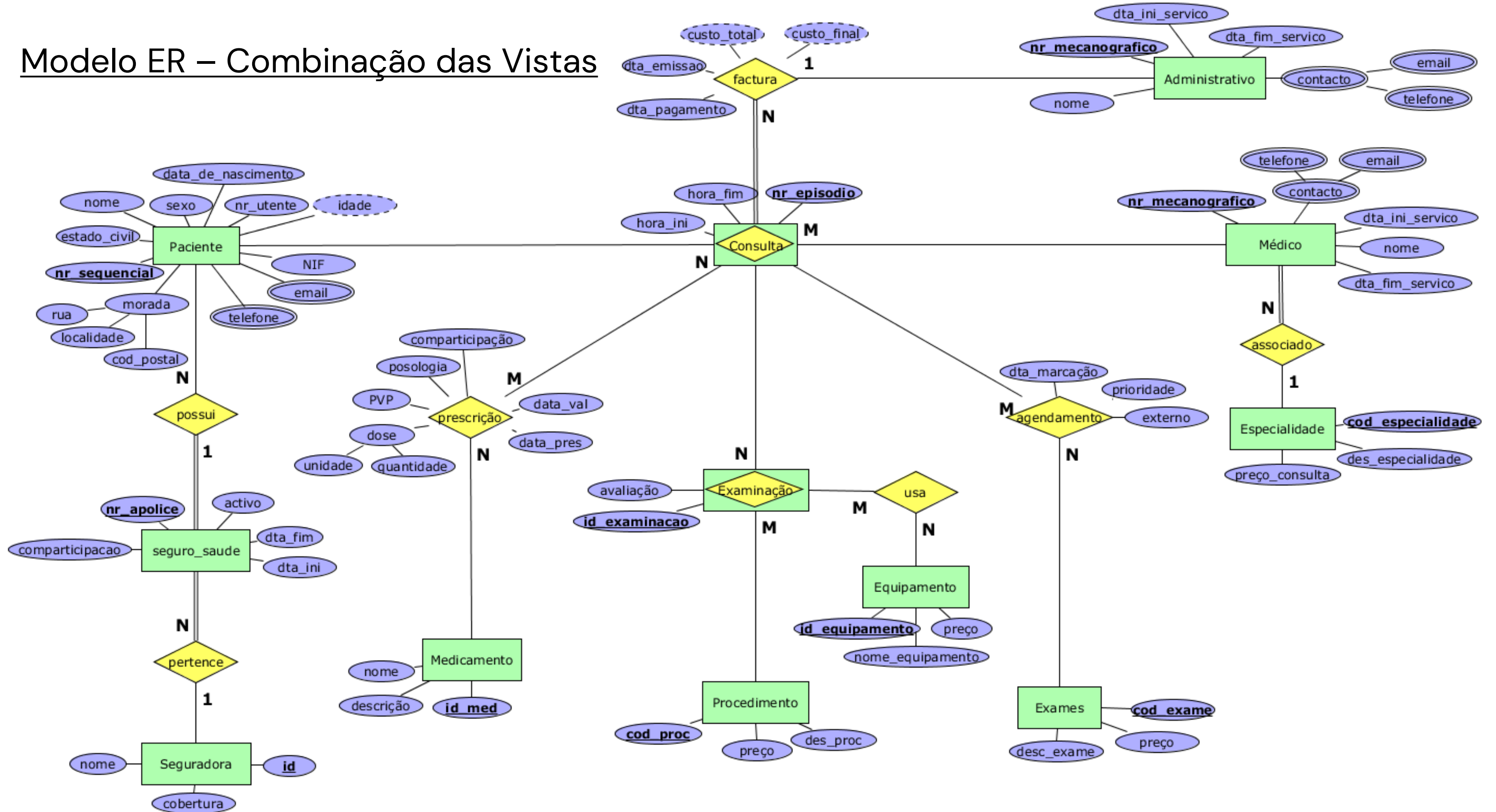
Revisão da aula anterior

Modelo ER – Vista de Faturação



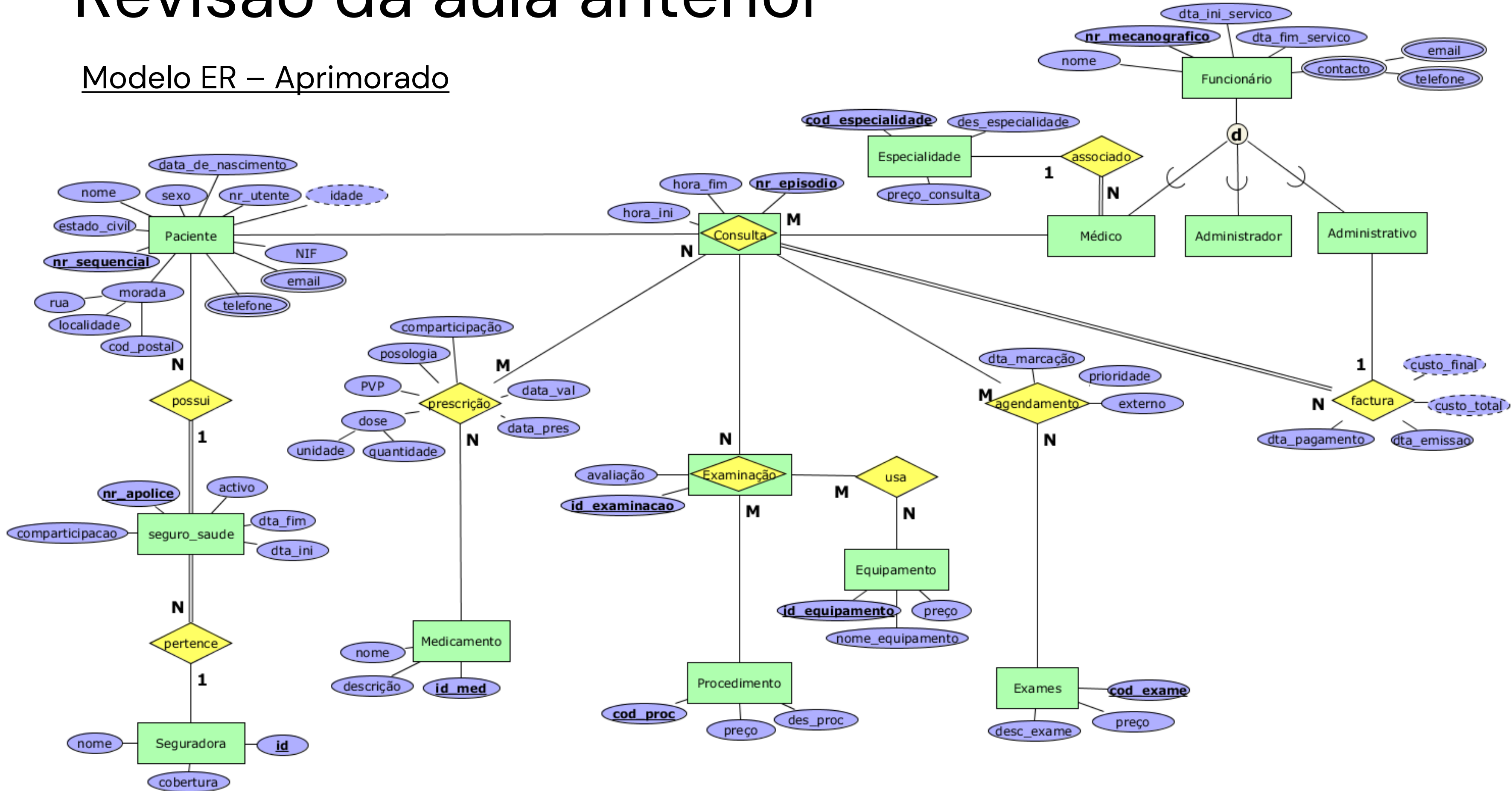
Revisão da aula anterior

Modelo ER – Combinação das Vistas



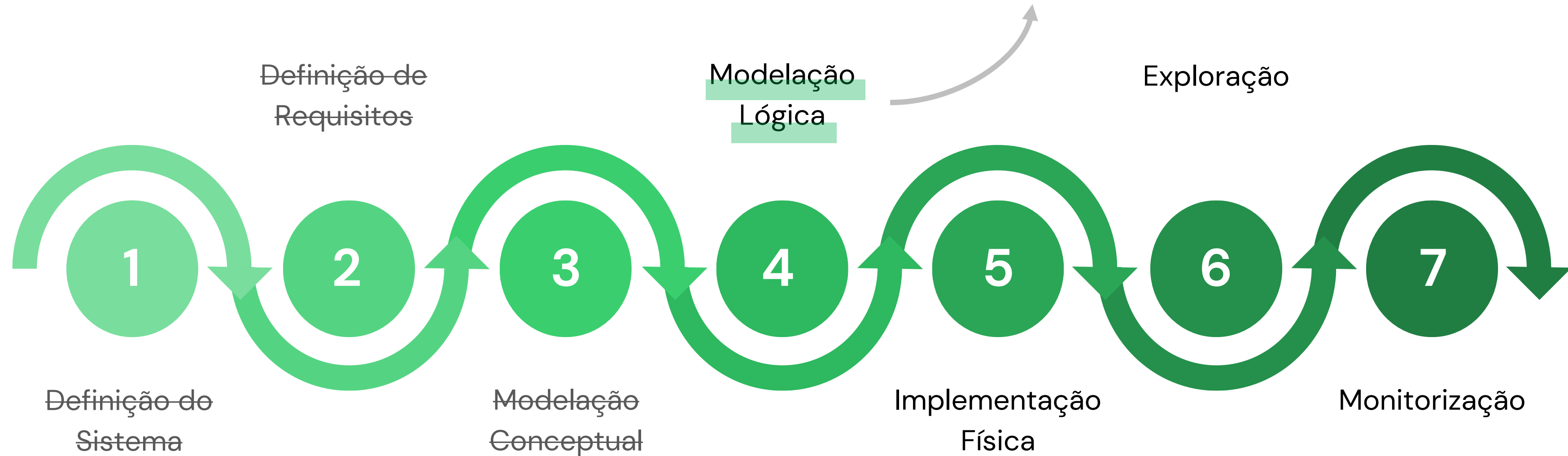
Revisão da aula anterior

Modelo ER – Aprimorado

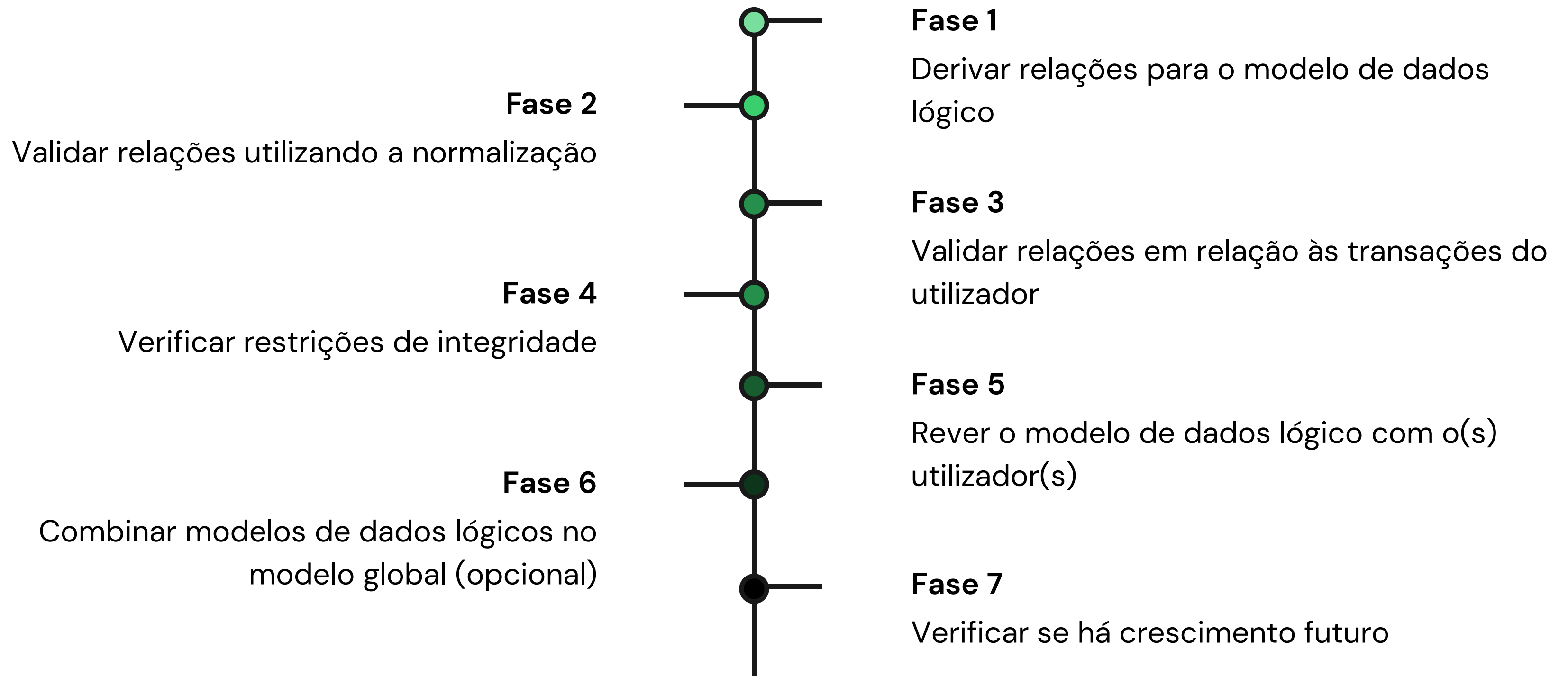


Ciclo de vida de um SBD

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

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

Modelação Lógica

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

Modelação Conceptual

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

→ 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:

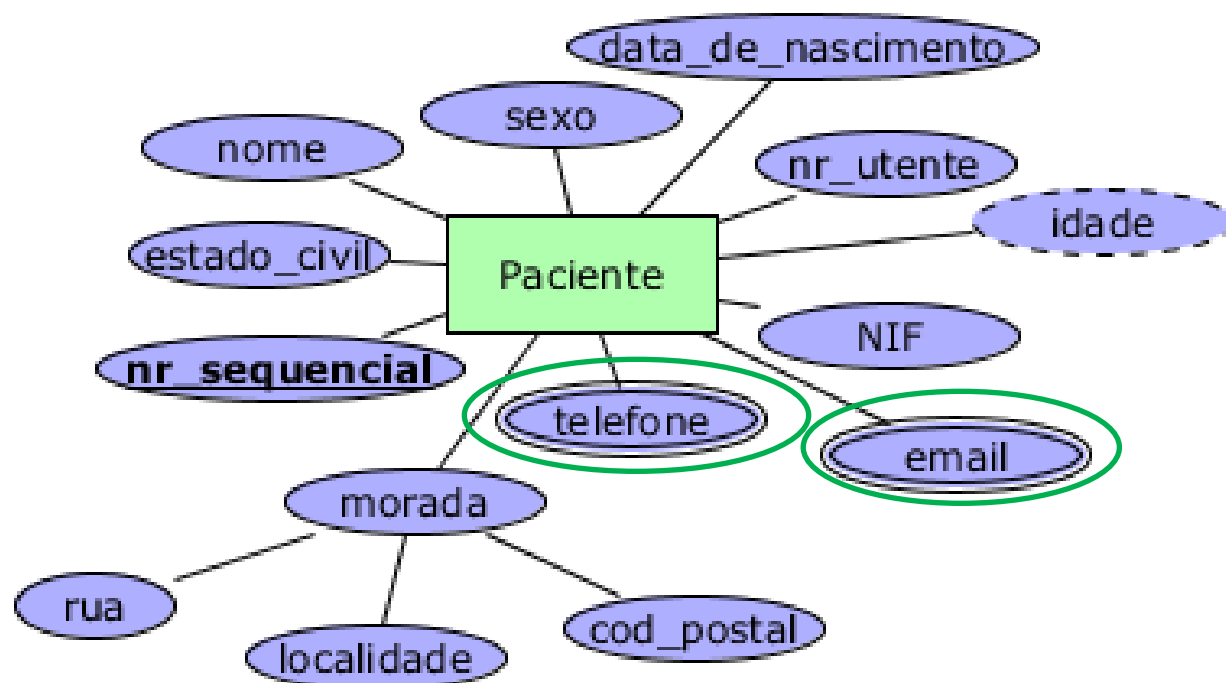
- Entidades Simples
- Atributos multivalor
- Entidades Fracas
- Relacionamentos binários de um-para-muitos (1:N)
- Relacionamentos binários de muitos-para-muitos (N:M)
- Entidade Relacionamento
- Relacionamentos binários de um-para-um (1:1)
- Relacionamentos binários recursivos de um-para-um (1:1)
-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



Paciente (nr_sequencial, nome, sexo, dta_nascimento, rua, localidade, cod_postal, NIF, nr_utente, estado_civil)

Chave primária nr_sequencial

Chave candidata NIF

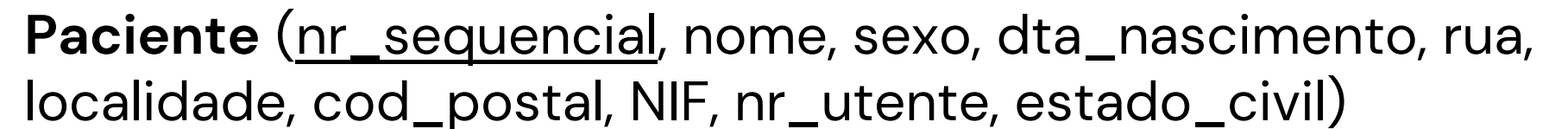
Chave candidata nr_utente

Derivado idade(dta_atual – dta_nascimento)

Paciente
<u>nr_sequencial</u>
nome
sexo
dta_nascimento
rua
localidade
cod_postal
NIF
nr_utente
estado_civil

➡ Derivar relações

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**.



Chave candidata NIF

Chave candidata nr_utente

Derivado idade(dta_atual – dta_nascimento)

Chave primária nr_sequencial, telefone

Chave estrangeira nr_sequencial referencia

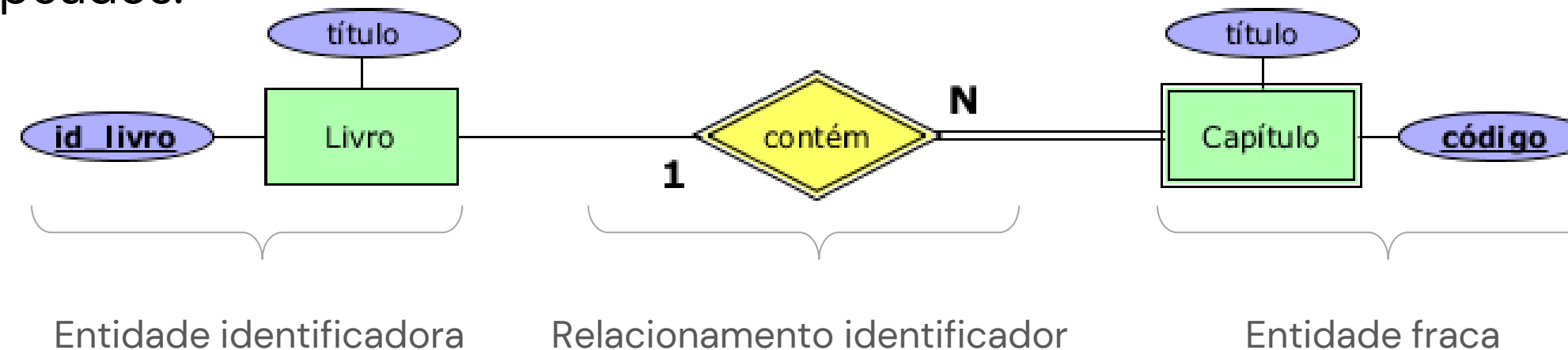
Paciente(nr_sequencial)

FASE 4: Modelação Lógica

→ Derivar relações

▪ Entidades Fracas

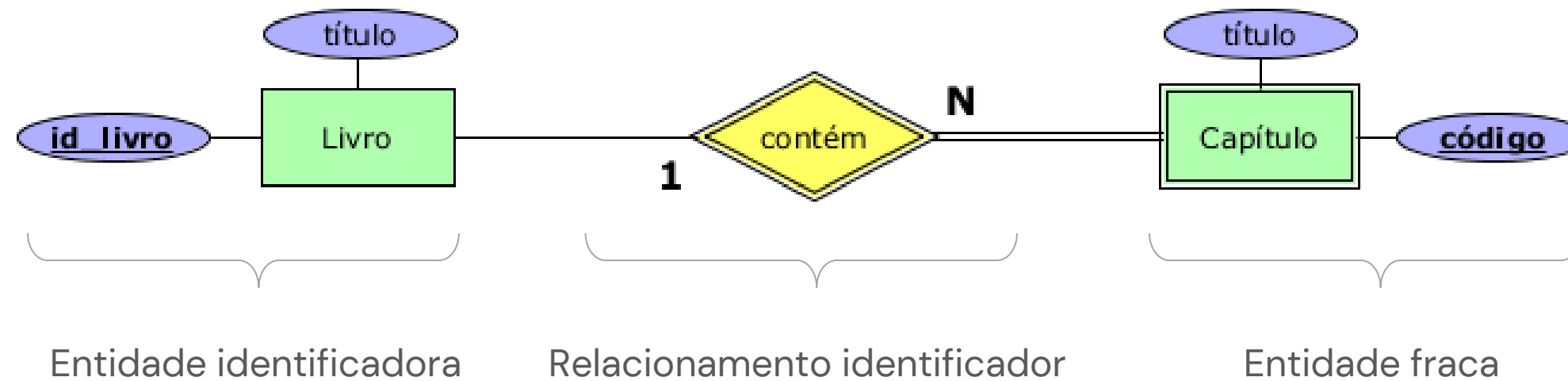
- Para cada entidade fraca do modelo de dados, crie uma relação que inclua todos os atributos simples dessa entidade.
- Se a entidade fraca não possuir atributos que possam constituir chaves candidatas, o conjunto de atributos que permitem identificar univocamente uma ocorrência da entidade fraca, é 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



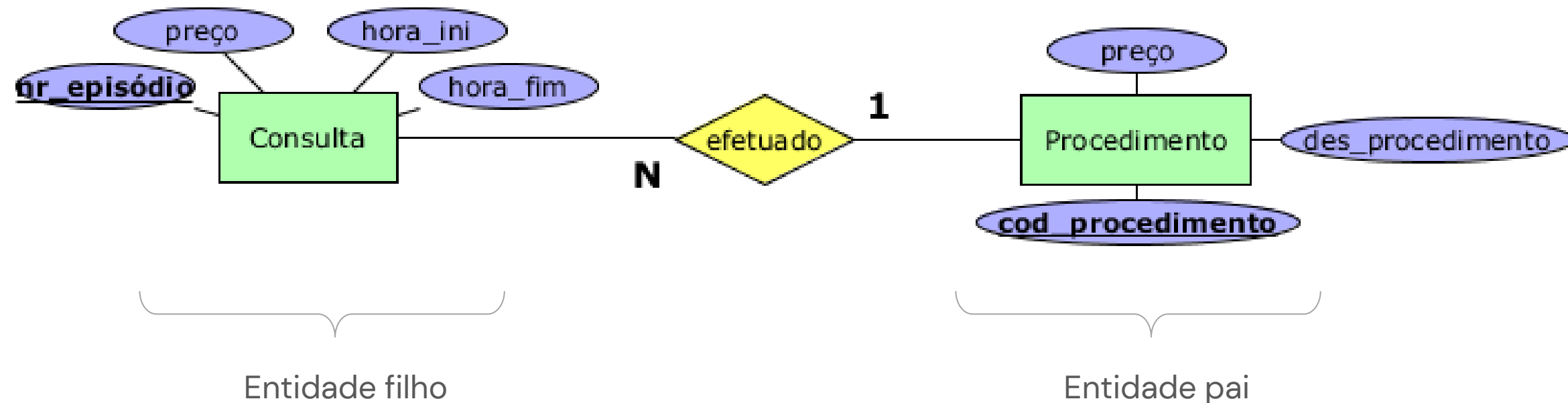
Livro (id_livro, título)
Chave primária id_livro

Capítulo (id_livro, codigo, título)
Chave primária id_livro, codigo
Chave estrangeira id_livro **referencia** Livro(id_livro)

FASE 4: Modelação Lógica

→ Derivar relações

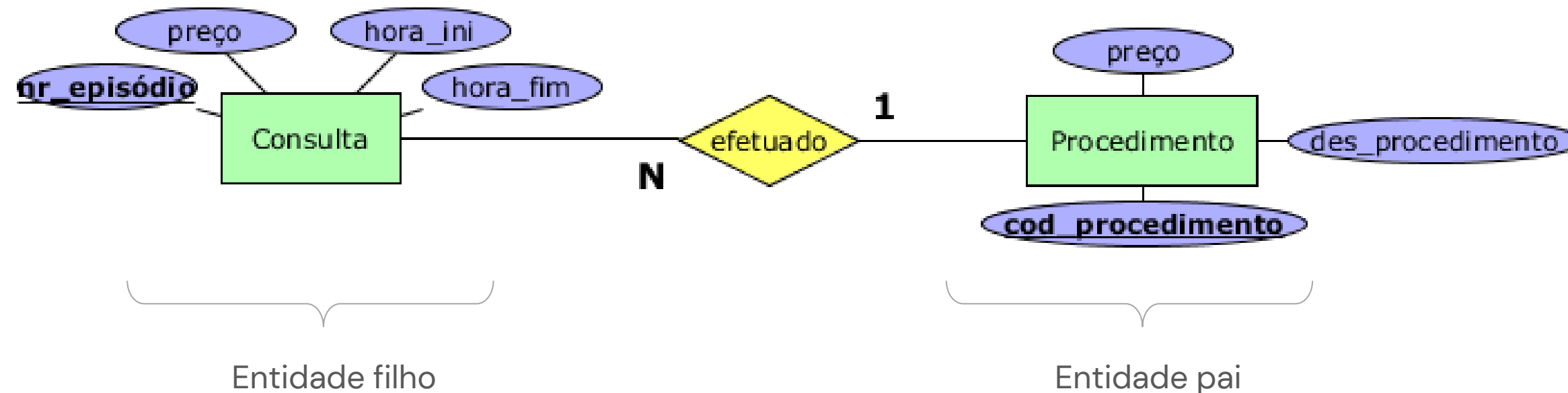
- Relacionamentos binários de um-para-muitos (1:N)
 - Para cada relacionamento binário 1:N, a entidade do lado 'um' do relacionamento é designada como a entidade pai e a entidade do lado 'muitos' é designada como a entidade filho.
 - Para representar esse relacionamento, cria-se uma **cópia** do(s) atributo(s)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



Consulta (nr_episodio, preço, hora_ini, hora_fim, cod_procedimento)

Chave primária nr_episodio

Chave Estrangeira cod_procedimento referencia

Procedimento(cod_procedimento)

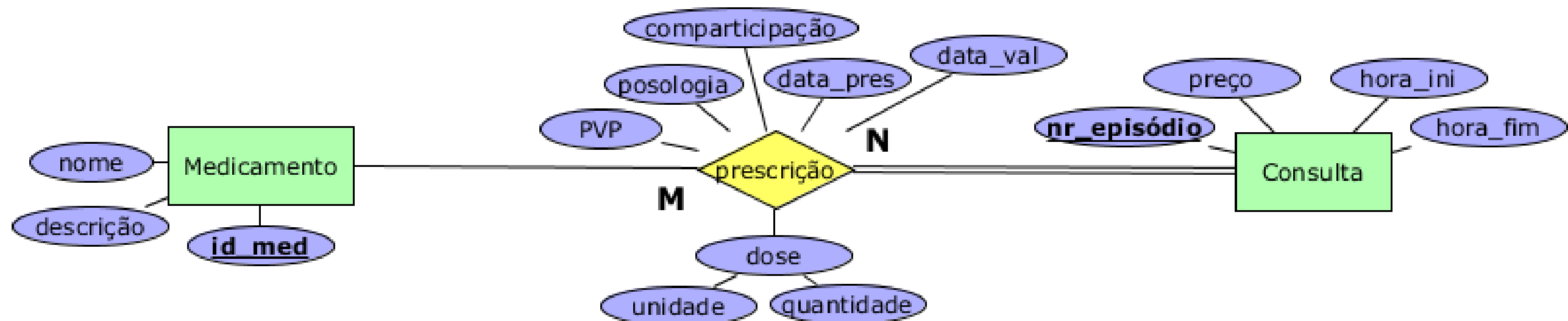
Procedimento (cod_procedimento, des_procedimento, preço)

Chave primária cod_procedimento

FASE 4: Modelação Lógica

→ Derivar relações

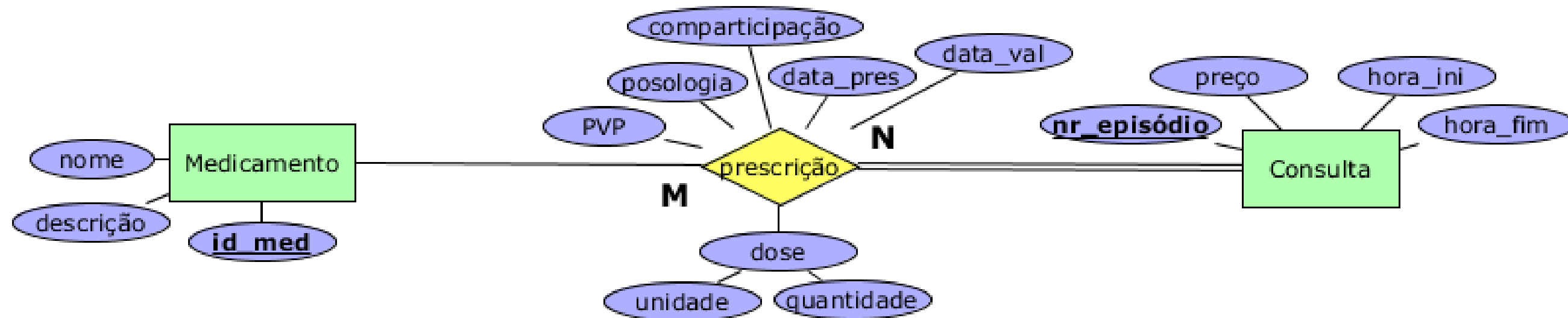
- Relacionamentos binários de muitos-para-muitos (N:M)
 - Crie uma relação para representar o relacionamento e inclua quaisquer atributos que façam parte do relacionamento.
 - Crie uma **cópia** do(s) atributo(s) de **chave primária** das **entidades** que participam no relacionamento na nova relação, 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)



Medicamento (id_med, nome, descrição)
Chave primária id_med

Consulta (nr_episodio, preço, hora_ini, hora_fim)
Chave primária nr_episodio

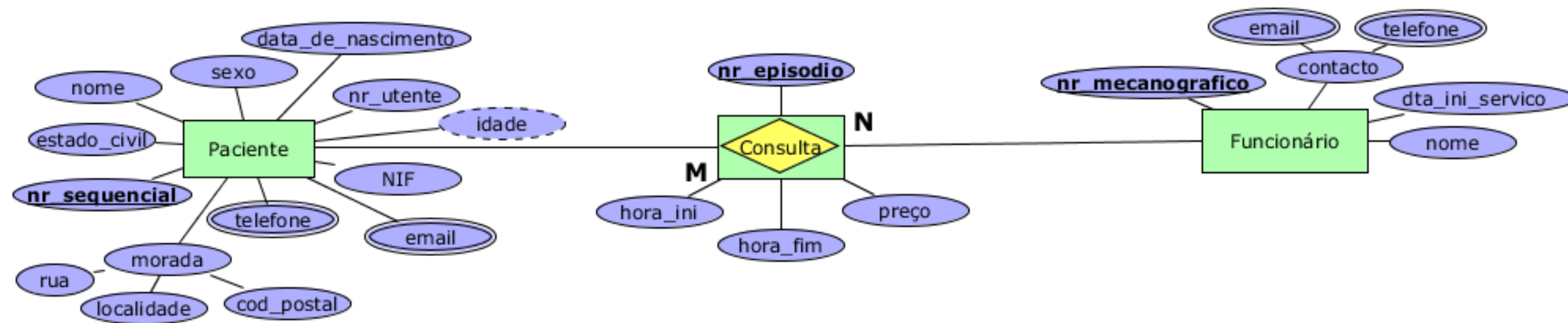
Prescrição (id_med, nr_episodio, unidade, quantidade, posologia, PVP, participação, data_val, data_pres)
Chave primária id_med, nr_episodio
Chave Estrangeira id_med **referencia** Medicamento(id_med)
Chave Estrangeira nr_episodio **referencia** Consulta(nr_episodio)

FASE 4: Modelação Lógica

➔ Derivar relações

▪ Entidade Relacionamento

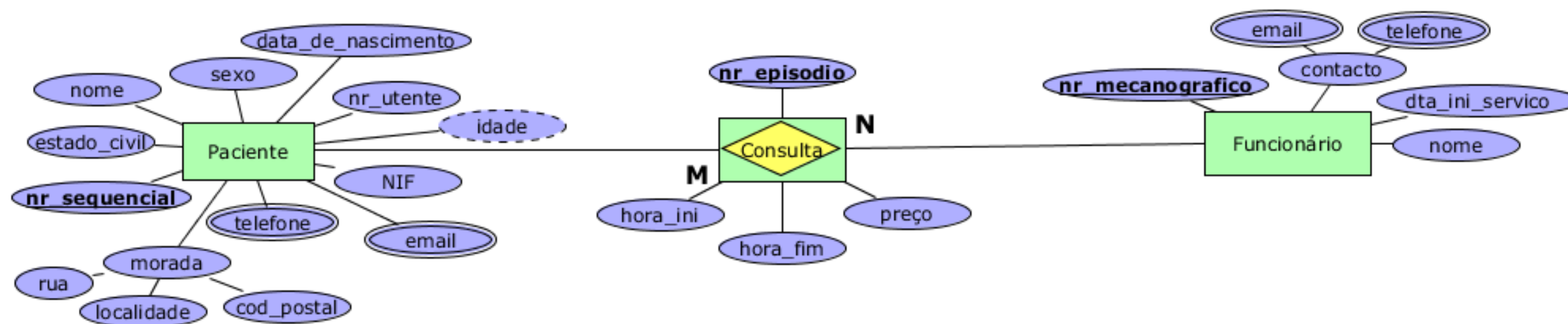
- Crie uma relação para representar a entidade-relacionamento como se fosse uma entidade independente e inclua todos os atributos que fazem parte da entidade-relacionamento.
- Crie uma **cópia** do(s) atributo(s) de **chave primária** das **entidades** que participam na entidade-relacionamento na nova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

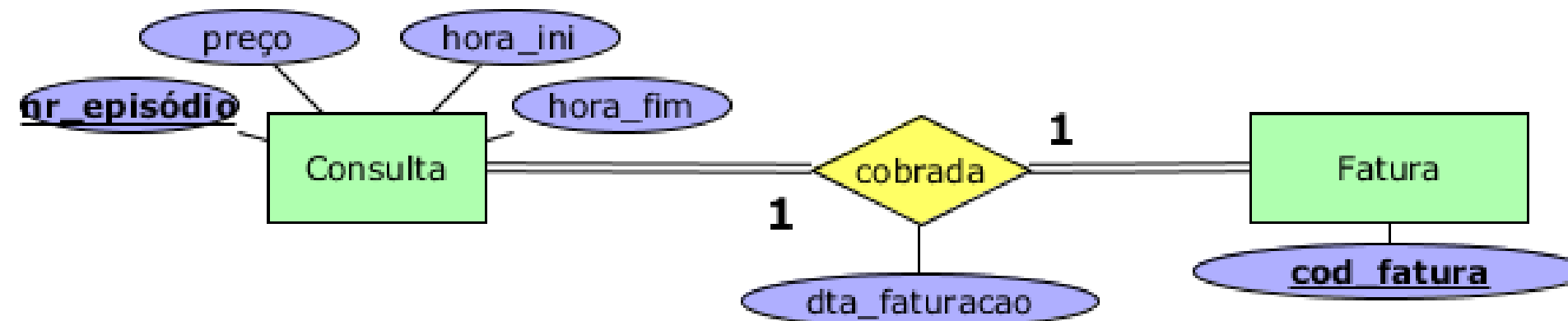
FASE 4: Modelação Lógica

→ 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.



Consulta (nr_episodio, preço, hora_ini, hora_fim, dta_faturacao, cod_fatura)

Chave primária nr_episodio

Chave candidata cod_fatura

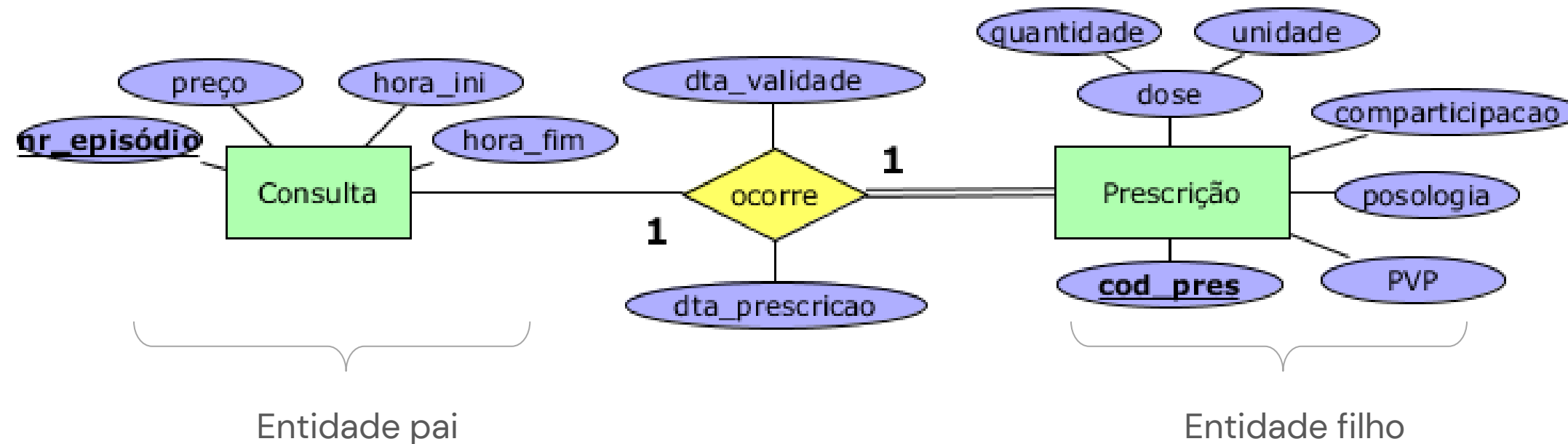
FASE 4: Modelação Lógica

→ Derivar relações

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

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

- A entidade com **participação opcional** é designada como **entidade-pai** e a entidade com **participação obrigatória** como **entidade-filho**.
- **Cópia** da **chave primária** da **entidade pai** colocada na relação que representa a **entidade filho**.



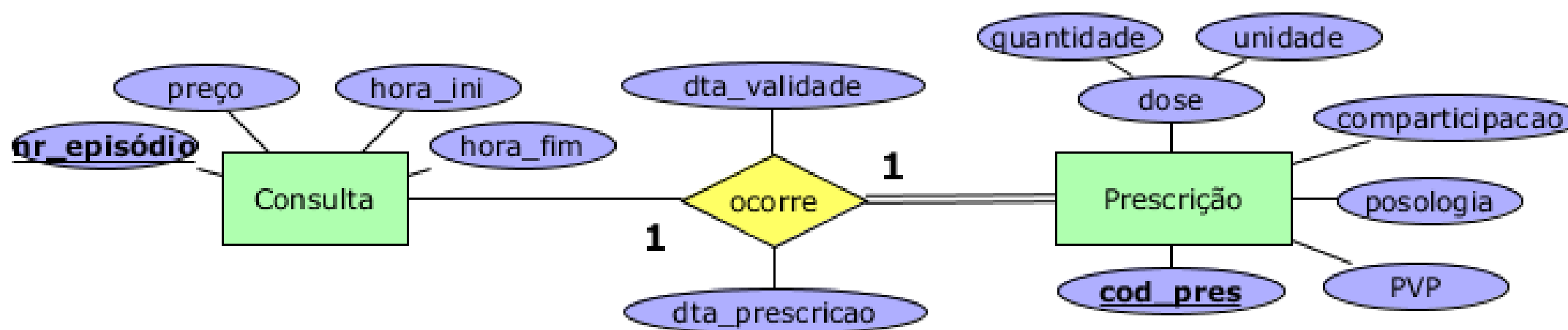
FASE 4: Modelação Lógica

➔ 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.



Consulta (nr_episodio, preço, hora_ini, hora_fim)

Chave primária nr_episodio

Prescricao (cod_pres, quantidade, unidade, posologia, PVP, comparticipação, dta_prescrição, dta_validade, nr_episodio)

Chave primária cod_pres

Chave estrangeira nr_episodio **referencia** Consulta(nr_episodio)

FASE 4: Modelação Lógica

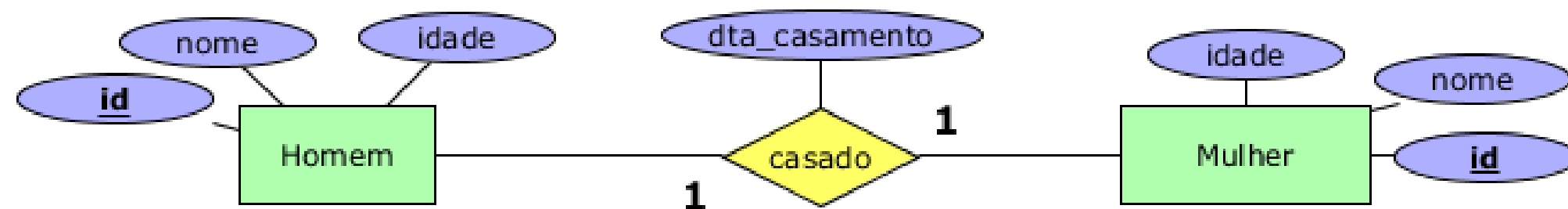
→ Derivar relações

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

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

Opção 1. Criar uma nova relação para representar o relacionamento.

Opção 2. Cópia da chave primária da entidade pai colocada na relação que representa a entidade filho.



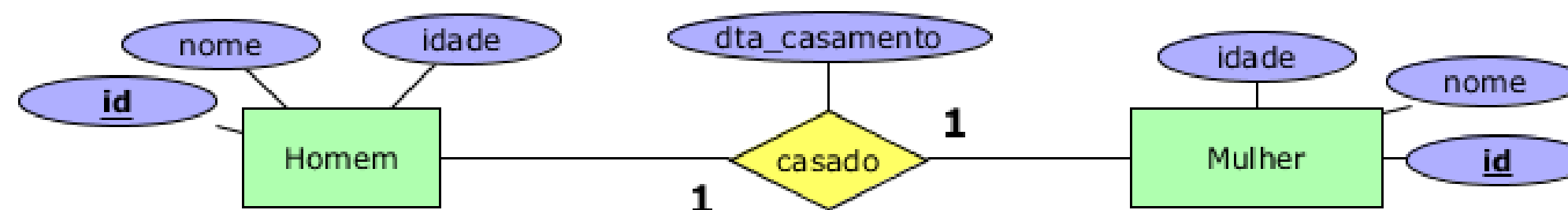
FASE 4: Modelação Lógica

→ Derivar relações

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

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

Opção 1. Criar uma nova relação para representar o relacionamento.



Homem (id, nome, idade)
Chave primária id

Casamento (id_homem, id_mulher, dta_casamento)
Chave primária id_homem, id_mulher
Chave estrangeira id_homem **referencia** Homem(id_homem)
Chave estrangeira id_mulher **referencia** Mulher(id_mulher)

Mulher (id, nome, idade)
Chave primária id

FASE 4: Modelação Lógica

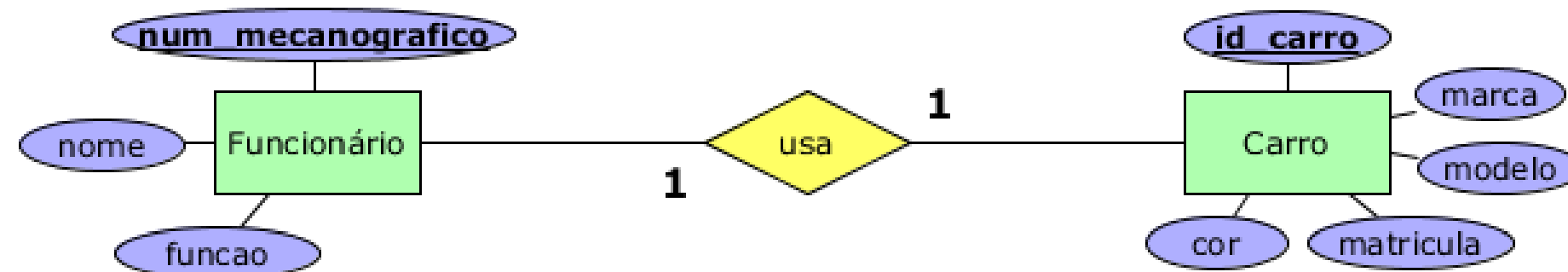
→ Derivar relações

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

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

Opção 2. 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

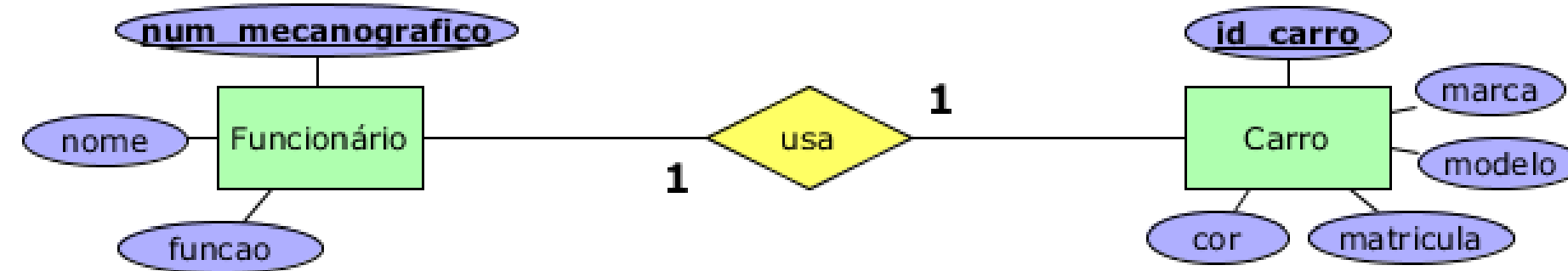
→ Derivar relações

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

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

Opção 2. 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:

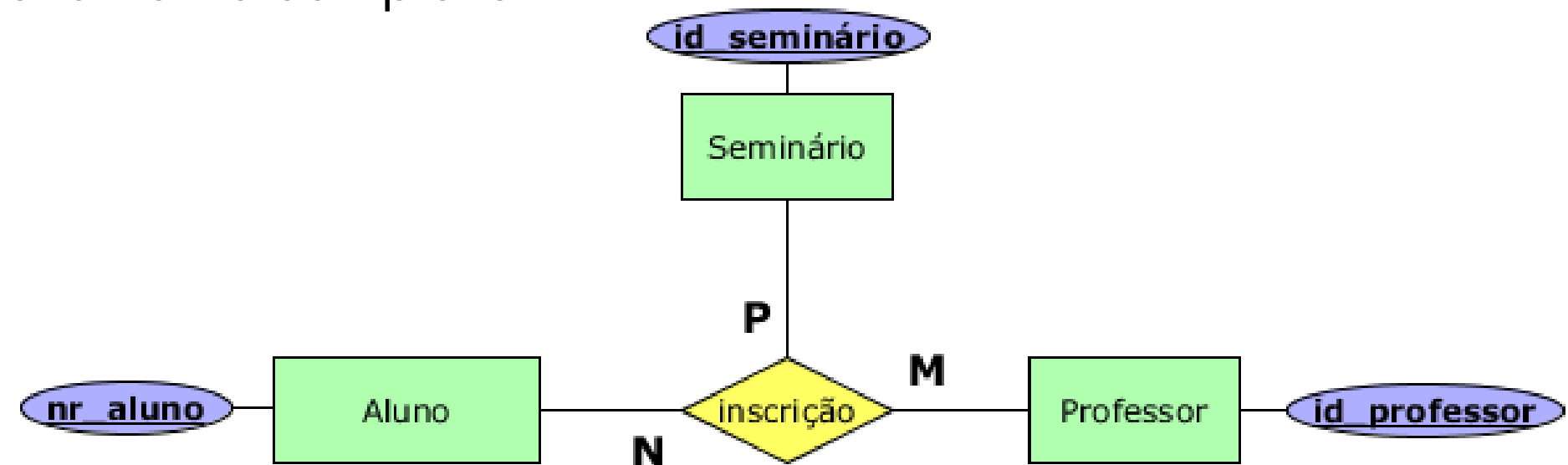
- **participação obrigatória de ambos os lados:** relação única com duas cópias da chave primária.
- **participação obrigatória em apenas um lado:** opção de criar uma relação única com duas cópias da chave primária, ou criar uma nova relação para representar o relacionamento. A nova relação teria apenas dois atributos, ambos cópias da chave primária.
- **participação opcional de ambos os lados:** criar uma nova relação conforme descrito acima.

FASE 4: Modelação Lógica

→ Derivar relações

▪ Relacionamentos complexos

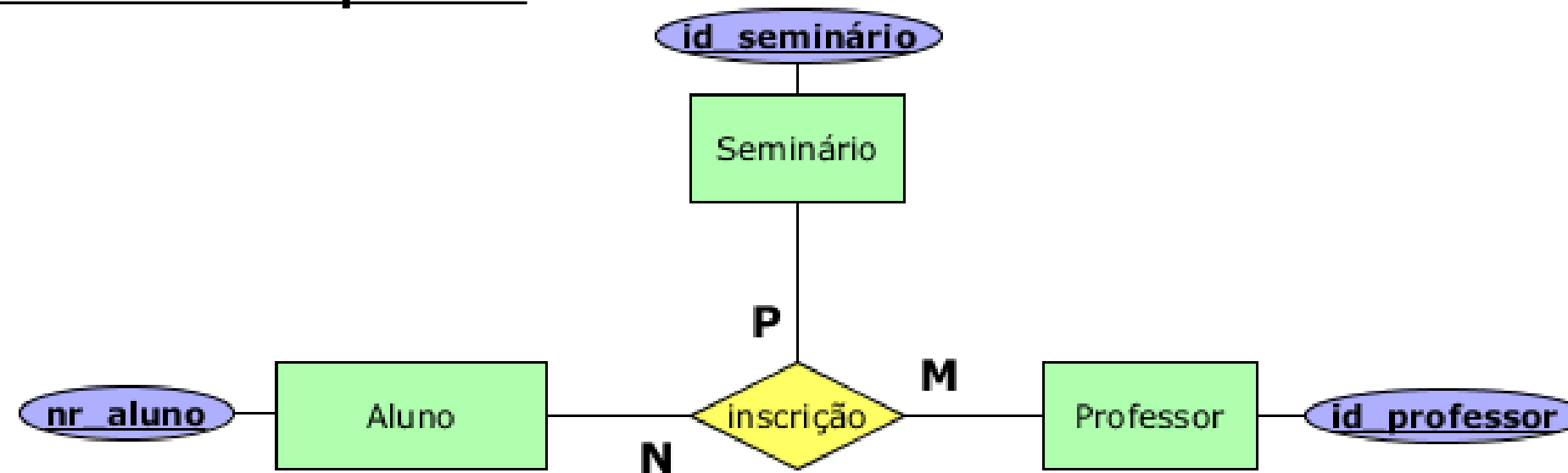
- Para cada relacionamento complexo, criar **uma relação** para representar o **relacionamento** e incluir quaisquer atributos que façam parte do relacionamento.
- Colocamos uma **cópia** da(s) **chave(s) primária(s)** das entidades que participam no relacionamento complexo na nova relação, para atuar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



FASE 4: Modelação Lógica

→ Derivar relações

▪ Relacionamentos complexos



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 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 única
Opcional	Não disjunto {And}	Duas relações: uma relação para a superclasse e uma relação para todas as subclasses
Obrigatória	Disjunto {Or}	Muitas relações (uma relação para cada combinação superclasse/subclasse)
Opcional	Disjunto {Or}	Muitas relações (uma relação para a superclasse e uma para cada subclasse)

FASE 4: Modelação Lógica

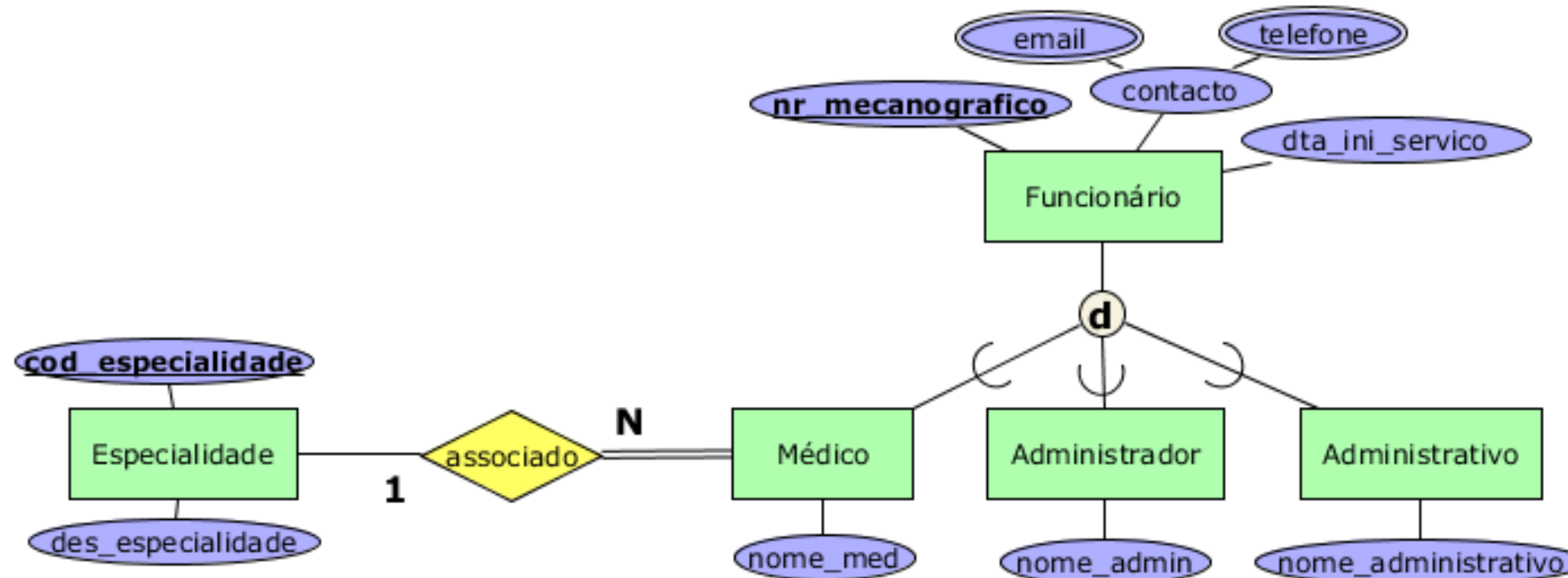
→ Derivar relações

- Relacionamentos superclasse/subclasse

Opcional

Disjunto {Or}

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



FASE 4: Modelação Lógica

→ Derivar relações

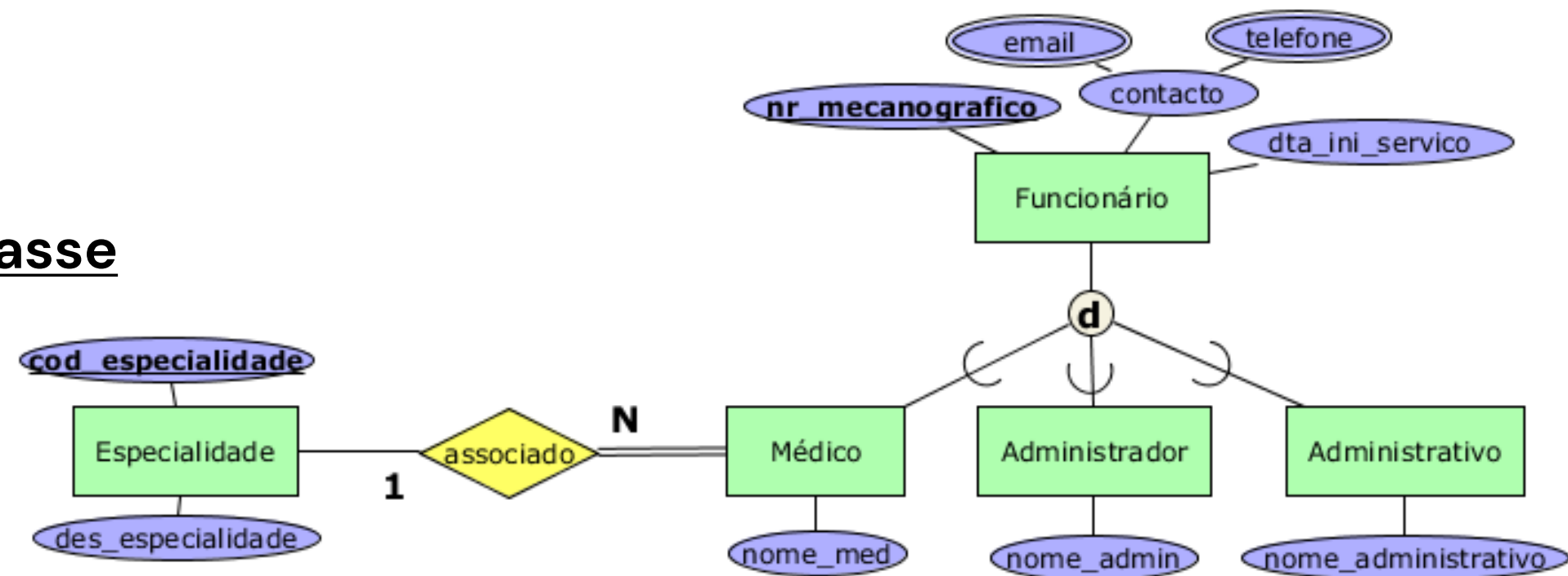
▪ Relacionamentos superclasse/subclasse

Funcionário (nr_mecanografico, dta_ini_servico)
Chave primária nr_mecanografico

Médico (nr_mecanografico, cod_especialidade, nome_med)
Chave primária nr_mecanografico
Chave estrangeira nr_mecanografico **referencia** Funcionario(nr_mecanografico)

Administrador (nr_mecanografico, nome_admin)
Chave primária nr_mecanografico
Chave estrangeira nr_mecanografico **referencia** Funcionario(nr_mecanografico)

Administrativo (id_administrativo, nome_administrativo)
Chave primária nr_mecanografico
Chave estrangeira nr_mecanografico **referencia** Funcionario(nr_mecanografico)



FASE 4: Modelação Lógica

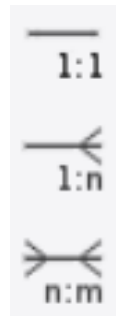
→ Derivar relações

Questão 1: Crie relações para o modelo de dados lógico de modo a representar as entidades, relacionamentos e atributos que foram identificados.

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)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'S'
--------------	---------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	-----

Data Type:

Default:

`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**)

- O comprimento de dados do tipo CHAR e VARCHAR indica o nº máximo de caracteres que é possível armazenar;
- Os dados do tipo CHAR são preenchidos à direita com **espaços em branco** para o comprimento especificado.

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

Para armazenar dados JSON:

- JSON

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 Numéricos

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

- Fixed-Point Types (Exact Value) – DECIMAL, NUMERIC

Os tipos DECIMAL e NUMERIC armazenam valores de dados numéricos exatos. Estes tipos de dados são usados 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)

* DEC e FIXED são sinónimos de DECIMAL

FASE 4: Modelação Lógica

➔ Validar o modelo em relação às transações do utilizador

Questão 2: Assegure que as relações no modelo lógico de dados suportam todas as transações necessárias.

Nota:

- Se conseguirmos assegurar todas as transações, validamos o modelo de dados lógico contra as transações do utilizador.
- No entanto, se não formos capazes de realizar uma transação manualmente, deve haver um problema com o modelo de dados, que tem de ser resolvido. Neste caso, é provável que tenha sido introduzido um erro durante a criação das relações, e devemos voltar atrás e verificar as áreas do modelo de dados a que a transação está a aceder para identificar e resolver o problema

FASE 4: Modelação Lógica

➔ Verificar as restrições de integridade

Devem ser considerados os seguintes tipos de restrições de integridade:

- **Dados necessários/obrigatórios:** Alguns atributos devem conter sempre um valor válido, ou seja, não podem conter valores "Null". Estas restrições deveriam ter sido identificadas quando documentamos os atributos no dicionário de dados (Aula 3 – Fase 3).
- **Restrições de domínio de atributos:** Cada atributo tem um domínio, ou seja, um conjunto de valores que são possíveis. Por exemplo, o sexo de uma pessoa ou é 'M' ou 'F' ou 'I'. Estas restrições deveriam ter sido identificadas quando escolhemos os domínios de atributos para o modelo de dados (Aula 3 – Fase 4).
- **Multiplicidade:** A multiplicidade representa as restrições que são colocadas nos relacionamentos entre os dados da BD. Estas restrições deveriam ter sido identificadas quando definimos os relacionamentos entre as entidades (Aula 3 – Fase 2).

FASE 4: Modelação Lógica

➔ Verificar as restrições de integridade

- **Integridade de entidade:** O valor da chave primária de uma tabela/relação não pode ser “Null” nem igual a outro já existente (caso contrário não conseguiríamos identificar registos). Estas restrições deveriam ter sido consideradas quando identificamos as chaves primárias para cada tipo de entidade (Aula 3 – Fase 5).
- **Restrições gerais/regras de negócio:** As atualizações de entidades podem ser controladas por restrições que regem as transações “do mundo real” que são representadas pelas atualizações. Por exemplo: Uma receita não pode conter mais do que 5 medicamentos.
- **Integridade referencial:** Um valor definido (diferente de “Null”) para um atributo que seja chave estrangeira deve referir-se a uma chave primária da tabela a que a chave estrangeira se refere, ou seja, a uma tupla existente na relação pai. Há duas questões que devem ser abordadas:
 - A primeira considera se os nulos são permitidos para a chave estrangeira. Em geral, se a participação do filho na relação for:
 - obrigatória → nulos não são permitidos;
 - opcional → nulos são permitidos.

FASE 4: Modelação Lógica

➔ Verificar as restrições de integridade

- A segunda define como garantir a integridade referencial. Para fazer isso, especificamos restrições de existência que definem as condições sob as quais uma chave estrangeira pode ser inserida, atualizada ou excluída.
 - Inserção ou atualização de uma tupla na relação filha – Para garantir a integridade referencial, verifique se o atributo de chave estrangeira da nova tupla está definido como nulo ou com um valor de uma tupla existente.
 - Remoção de uma tupla da relação pai – Se uma tupla de uma relação pai é excluída, a integridade referencial é perdida se existir uma tupla filho referenciando a tupla pai. Existem várias estratégias que podemos considerar:
 - NO ACTION — Impede a remoção da tupla da relação pai se houver alguma tupla filho referenciada.
 - SET NULL — Quando uma tupla pai é excluída, os valores de chave estrangeira em todas as tuplas filho correspondentes são automaticamente definidos como nulos. Esta estratégia só pode ser aplicada se os atributos que constituem a chave estrangeira aceitarem nulos.

FASE 4: Modelação Lógica

➔ Verificar as restrições de integridade

- SET DEFAULT — Quando uma tupla pai é excluída, os valores de chave estrangeira em todas as tuplas filho correspondentes devem ser automaticamente configurados para os seus valores padrão. Esta estratégia só pode ser aplicada se os atributos que constituem a chave estrangeira tiverem valores padrão definidos.
- CASCADE — Quando a tupla pai é excluída, exclui automaticamente todas as tuplas filhas referenciadas. Se qualquer tupla filha excluída atuar como pai noutro relacionamento, a operação de exclusão deverá ser aplicada às tuplas nessa relação filha e assim por diante em cascata. No caso do Hospital Portucalense, “Excluir um médico exclui automaticamente todas as consultas realizadas por ele”. Nesta situação, esta estratégia não seria sábia.
- NO CHECK — Quando uma tupla pai é excluída, nada é feito para garantir a integridade referencial.

FASE 4: Modelação Lógica

➔ Verificar as restrições de integridade

Paciente				
<u>nr_sequencial</u>	nome	sexo	dta_nascimento	...
323431	Ana Luísa Dias Gomes	F	20/12/1990	
453347	José da Costa Silva	M	03/05/1975	
212423	Maria Leonor Ribeiro Barbosa	Fem	12/07/2000	
...

✗ Integridade Referencial ?

✗ Integridade de Domínio

✗ Integridade de Entidade

✗ Integridade de Entidade

Consulta							
<u>nr_episodio</u>	<u>id_pac</u>	<u>id_med</u>	hora_ini	hora_fim	id_agenda	cod_proc	id_sec
12345678	212423	3456	2022-01-23 10:18:17	2022-01-23 10:38:27	123456789	P22	1212
14451643	453347	3224	2022-01-25 08:35:23	2022-01-25 09:00:12	223212434	P23	1598
14451643	212423	3371	2022-02-02 09:00:33	2022-02-02 09:15:20	345567811	NULL	1479
13415324	123456	3834	2022-02-04 12:34:11	2022-02-04 13:00:00	433212456	P22	1234
NULL	323431	3456	2022-02-12 11:20:23	2022-02-12 11:52:33	387612392	P24	1176
...

FASE 4: Modelação Lógica

➔ Verificar as restrições de integridade

Questão 3: Especifique quais as restrições de integridade necessárias, independentemente da forma como isso possa ser conseguido.

Próxima aula: Implementação Física

