

Objetivos:

- I. Introdução sobre o modelo conceitual;
- II. Entidades;
- III. Atributos;
- IV. Atributos chaves;
- V. Relacionamentos;
- VI. MER – Modelo Entidade Relacionamento;
- VII. DER – Diagrama Entidade Relacionamento.

I. Introdução sobre o modelo conceitual

É a primeira fase da modelagem, onde representamos o mundo real por meio de uma visão simplificada dos dados e seus relacionamentos. Assim poderemos determinar quais informações serão armazenadas no BD. Neste nível o projeto é independente de SGBD.

Sua principal finalidade é capturar os requisitos de informação e regras de negócio sob o ponto de vista do negócio. O objetivo é criar um modelo de forma gráfica, sendo este chamado de Diagrama Entidade Relacionamento (DER), que identificará todas as entidades e relacionamentos de uma forma global.

Neste nível, detalhes da implementação não aparecem, porém o modelo é suficientemente detalhado a ponto de ser possível descrever os dados requeridos, seus relacionamentos e regras de consistência.

II. Entidades

São objetos ou partes envolvidas em um domínio, podem ser classificados como **físicas** ou **lógicas**, de acordo com a sua existência no mundo real:

- ✓ Entidades físicas: são aquelas realmente tangíveis, existentes e visíveis no mundo real, tais como um cliente, produto, aluno e disciplina;
- ✓ Entidades lógicas: são aquelas que existem em decorrência da interação com entidades físicas, ou seja, elas só fazem sentido dentro de um certo domínio de negócio, mas que no mundo real não são objetos físicos (que ocupam lugar no espaço). São exemplos disso uma venda, o modelo de um produto e a matrícula em uma disciplina.

Observações:

- ✓ As entidades são nomeadas com substantivos concretos ou abstratos que representem de forma clara sua função dentro do domínio. Por exemplo, cliente, produto, venda, aluno, curso, disciplina e matrícula;
- ✓ Não usamos plural no nome de uma entidade, por exemplos, alunos e disciplinas.

Podemos classificar as entidades segundo o motivo de sua existência:

- ✓ Entidades fortes: são aquelas cuja existência independe de outras entidades, ou seja, por si só elas já possuem total sentido de existir. Em um sistema de vendas, a entidade produto independe de quaisquer outras para existir;
- ✓ Entidades fracas: ao contrário das entidades fortes, as fracas são aquelas que dependem de outras entidades para existirem, pois individualmente elas não fazem sentido. Mantendo o exemplo, a entidade venda depende da entidade cliente, pois uma venda sem cliente não tem sentido.

III. Atributos

São as características que descrevem cada entidade dentro do domínio. Como exemplo, a entidade aluno possui RA, nome, endereço, gênero, data de nascimento, telefone e e-mail. Durante a análise de requisitos, são identificados os atributos relevantes de cada entidade no contexto da aplicação, de forma a manter o modelo o mais simples possível e, conseqüentemente, armazenar apenas as informações que serão úteis futuramente.

Não devem ser modeladas informações irrelevantes no contexto da aplicação, como exemplo, uma pessoa possui os atributos altura e peso, mas eles não são relevantes para o sistema acadêmico de uma faculdade.

Os atributos podem ser classificados quanto à sua estrutura:

- ✓ Atributo simples: um único atributo define uma característica da entidade. Por exemplo: nome, gênero e e-mail;
- ✓ Atributo composto: para definir uma informação da entidade, são usados vários atributos. Por exemplo, o endereço pode ser composto por logradouro, número, bairro etc.

Os atributos podem ser classificados quanto à quantidade:

- ✓ Atributo monovalorado: possui um único valor para cada entidade. Por exemplo, nome e idade;
- ✓ Atributo multivalorado: possui múltiplos valores para cada entidade. Por exemplo, um aluno pode ter vários telefones e e-mails.

Os atributos podem ser classificados quanto à existência:

- ✓ Atributo armazenado: o valor do atributo será armazenado na memória do computador. Por exemplo: nome e data de nascimento;
- ✓ Atributo derivado: o valor do atributo será calculado, isto é, ele não deve ser armazenado. Por exemplo, o atributo idade pode ser calculado usando o atributo data de nascimento.

Os atributos podem ser classificados quanto à sua função:

- ✓ Atributos descritivos: representam características intrínsecas de uma entidade, por exemplo, nome, gênero e data de nascimento;
- ✓ Atributos nominativos: além de serem também descritivos, estes têm a função de definir e identificar um objeto. Por exemplo, nome, CPF e RA;

- ✓ Atributos referenciais: representam a ligação de uma entidade com outra em um relacionamento. Por exemplo, a entidade matrícula possui o RA do aluno, mas esse RA precisa estar vinculado com o atributo RA da entidade aluno.

IV. Atributos chaves

- Os atributos nominativos são chamados de **Chave Primária** (Primary Key): alguns atributos representam valores únicos que identificam a entidade dentro do domínio e não podem se repetir. Por exemplo, o CPF de um cliente e o RA de um aluno não podem se repetir;
- Os atributos referenciais são chamados de **Chave Estrangeira** (Foreign Key): são os atributos que estão ligados à chave primária de outra entidade. Como exemplo, a entidade aluno tem como chave primária seu RA, assim, a matrícula do aluno na disciplina possui o RA como chave estrangeira na entidade matrícula.

V. Relacionamentos

Os relacionamentos são classificados de acordo com a quantidade de objetos (entidades) envolvidos em cada lado do relacionamento:

- ✓ Relacionamento **1..1** (um para um): cada uma das duas entidades envolvidas referenciam obrigatoriamente apenas uma unidade da outra. Por exemplo, em um BD de currículos, cada usuário cadastrado pode possuir apenas um currículo na base, ao mesmo tempo em que cada currículo só pertence a um único usuário cadastrado;
- ✓ Relacionamento **1..n** ou **1..*** (um para muitos): uma das entidades envolvidas pode referenciar várias unidades da outra, porém, do outro lado cada uma das várias unidades referenciadas só pode estar ligada a uma única unidade da outra entidade. Por exemplo, em um sistema de plano de saúde, um usuário pode ter vários dependentes, mas cada dependente só pode estar ligado a um usuário principal. Note que temos apenas duas entidades envolvidas: usuário e dependente. O que muda é a quantidade de unidades/exemplares envolvidas de cada lado;
- ✓ Relacionamento **n..n** ou ***..*** (muitos para muitos): neste tipo de relacionamento cada entidade, de ambos os lados, podem referenciar múltiplas unidades da outra. Por exemplo, em um sistema de biblioteca, um livro pode ser escrito por vários autores, ao mesmo tempo que um autor pode escrever vários livros. Assim, um objeto do tipo autor pode referenciar múltiplos objetos do tipo livro, e vice-versa.

Os relacionamentos em geral são nomeados com verbos ou expressões que representam a forma como as entidades interagem, ou a ação que uma exerce sobre a outra. Essa nomenclatura pode variar de acordo com a direção em que se lê o relacionamento. Por exemplo: um autor escreve vários livros, enquanto um livro é escrito por vários autores.

VI. MER – Modelo Entidade Relacionamento

O MER é um modelo baseado na percepção do mundo real como um conjunto de objetos chamados entidades e pelo conjunto de relacionamentos entre esses objetos. Então o 1º passo é identificar as entidades de um sistema a ser modelado. Tomaremos como exemplo um sistema para gerenciar o acadêmico de uma faculdade.

Uma entidade é algo do mundo real que possui uma existência independente. Nesse sistema podemos identificar as seguintes entidades: aluno, professor, secretária acadêmica, atendente, curso, disciplina e matrícula.

Cada entidade tem propriedades particulares, chamadas atributos. O 2º passo é listar os atributos de cada entidade:

- Aluno: RA, CPF, nome, endereço, telefone, e-mail, gênero, data de nascimento e senha;
- Professor: CPF, nome, endereço, telefone, e-mail, gênero, data de nascimento, titulação e senha;
- Secretária acadêmica: CPF, nome, endereço, telefone, e-mail, gênero, data de nascimento e senha;
- Atendente: CPF, nome, endereço, telefone, e-mail, gênero, data de nascimento e senha;
- Curso: nome e área;
- Disciplina: nome, carga horária e curso (percebe-se que existe um relacionamento entre as entidades Disciplina e Curso);
- Matrícula: data, semestre letivo, nota, falta e disciplina (percebe-se que existe um relacionamento entre as entidades Matrícula e Disciplina).

Cada entidade deve possuir conjuntos de valores de atributos distintos. Por exemplo: as entidades **c1** e **c2** possuem valores de atributos distintos, mas ambas são entidades do tipo **Curso**:

c1 { Nome = Análise e Desenvolvimento de Sistemas
Área = Ciência da computação

c2 { Nome = Gestão de RH
Área = Administração de empresas

Um **conjunto de entidades** (c1, c2, ...) agrupa entidades do mesmo tipo, aqui o tipo é **Curso**

Um tipo de entidade - por exemplo, **Curso** - deve possuir um conjunto de atributos cujos valores são distintos para cada entidade individual. Este conjunto de atributos é chamado **atributo chave** e seus valores podem ser utilizados para identificar cada entidade de forma única:

c1 { Nome = Análise e Desenvolvimento de Sistemas
Área = Ciência da computação

c2 { Nome = Gestão de RH
Área = Administração de empresas

Neste exemplo, o atributo **Nome** sozinho não pode ser utilizado como atributo chave, pois ele sozinho não é suficiente para identificar uma entidade **Curso**. Neste caso seria necessário utilizar uma **chave composta** pelos atributos **Nome + Área**

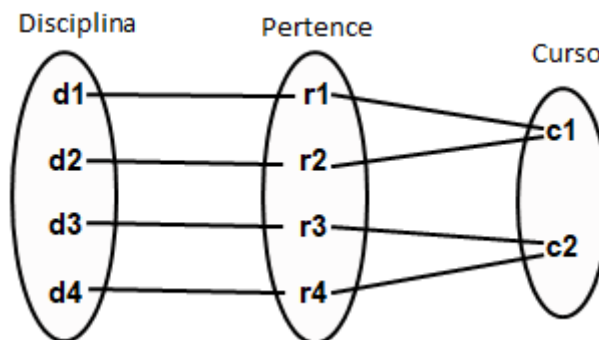
Mas se o tipo de entidade **Curso** possuir um atributo **Código**, neste caso bastaria o atributo **Código** como atributo chave:

- c1 {
Código = 1
Nome = Análise e Desenvolvimento de Sistemas
Área = Ciência da computação
- c2 {
Código = 2
Nome = Gestão de RH
Área = Administração de empresas

Para distinguir as entidades c1 e c2 basta olhar o atributo chave, neste exemplo, é o atributo **Código**

Não existe uma regra para definir os atributos chaves, porém recomenda-se que não sejam compostos e que o atributo chave tenha valores do conjunto dos números inteiros para tornar as operações mais simples no computador. Portanto, o atributo **Código** é uma chave primária melhor que a chave composta Nome + Área.

Um tipo de relacionamento **R** entre **n** entidades é um conjunto de associações entre entidades dos tipos participantes. Tomando como exemplo, o caso a seguir, cada instância de relacionamento **ri** em **Pertence** é uma associação de entidades, onde a associação inclui exatamente uma entidade de cada tipo de entidade **Disciplina** e **Curso** participante no tipo de relacionamento **Pertence**.

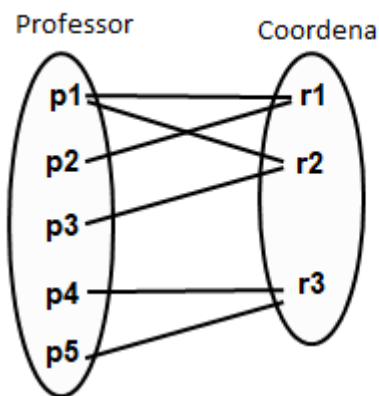


O objetivo dos relacionamentos é representar associações existentes no minimundo a ser modelado. Neste exemplo, o relacionamento **Pertence** identifica o curso de cada disciplina:

Disciplina	Pertence	Curso
d1 { Nome = Banco de Dados Carga horária = 80	r1 { Disciplina = d1 Curso = c1	c1 { Nome = Análise e Desenvolvimento de Sistemas Área = Ciência da computação
d2 { Nome = Algoritmos Carga horária = 80	r2 { Disciplina = d2 Curso = c1	c2 { Nome = Gestão de RH Área = Administração de empresas
d3 { Nome = Contabilidade Carga horária = 40	r3 { Disciplina = d3 Curso = c2	
d4 { Nome = Recrutamento Carga horária = 80	r4 { Disciplina = d4 Curso = c2	

Um mesmo **tipo de entidade** pode participar mais de uma vez em um tipo de relacionamento com diferentes papéis, estes tipos de relacionamentos são chamados de **recursivos**.

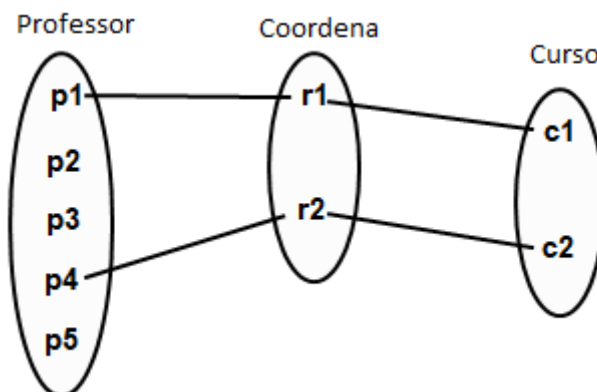
O tipo de relacionamento **Coordena** relaciona um **Professor** com o seu coordenador, onde ambas as entidades são membros do mesmo tipo de entidade **Professor**. Assim, o tipo de entidade **Professor** participa duas vezes: uma vez no papel de **professor** e outra no papel de **coordenador**.



Professor	
p1	{ CPF = 123.456.789-01 Nome = Ana Maria Endereço = R. Um, 11
p2	{ CPF = 098.765.432-10 Nome = Pedro Endereço = R. Dois, 22
p3	{ CPF = 765.432.098-55 Nome = Maria Endereço = R. Três, 33
p4	{ CPF = 008.432.765-22 Nome = Luiz Endereço = R. Quatro, 44
p5	{ CPF = 121.232.343-45 Nome = Carla Endereço = R. Cinco, 55

Pertence	
r1	{ Coordenador = p1 Professor = p2
r2	{ Coordenador = p1 Professor = p3
r3	{ Coordenador = p4 Professor = p5

Os **tipos de relacionamentos** sofrem certas restrições que limitam as possíveis combinações das entidades participantes. Estas restrições são derivadas de restrições impostas pelo estado destas entidades no minimundo. No exemplo a seguir temos a seguinte situação: 1 professor pode coordenar apenas 1 curso, enquanto 1 curso, pode ser coordenado por apenas 1 professor. A este tipo de restrição, nós chamamos de **cardinalidade**.

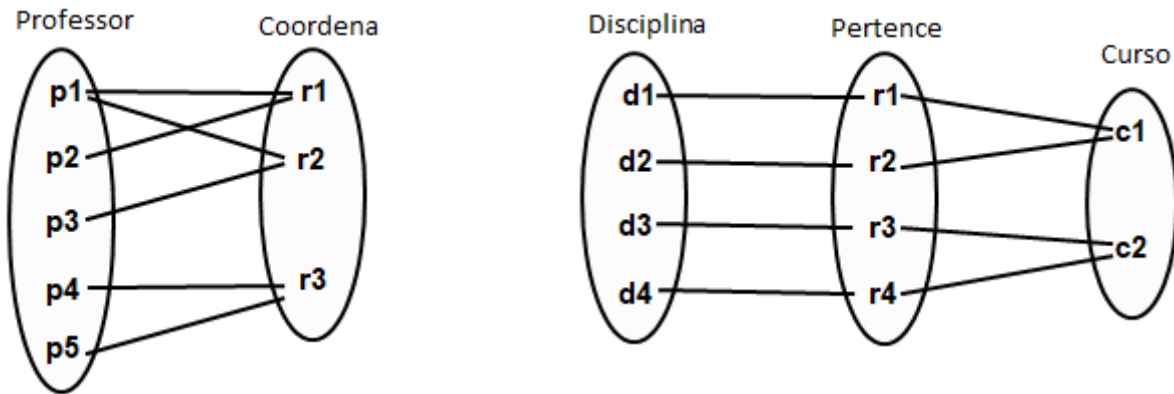


A cardinalidade indica o número de relacionamentos dos quais uma entidade pode participar. A cardinalidade pode ser: **1:1** (um para um), **1:N** (um para muitos) e **N:N** (muitos para muitos).

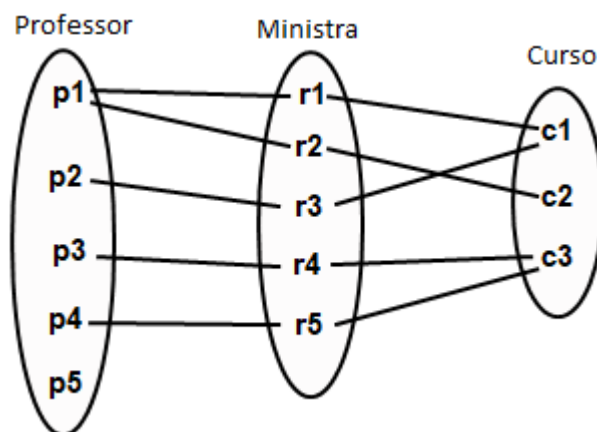
No exemplo anterior, a cardinalidade é **1:1**, pois cada entidade professor pode coordenar apenas 1 curso e 1 curso pode ser coordenado por apenas 1 professor.

Nos exemplos a seguir os relacionamentos **Coordena** e **Pertence** são **1:N**:

- 1 professor coordena **N** professores;
- 1 curso possui **N** disciplinas. Observe que lemos a cardinalidade do lado 1 primeiro.



Neste exemplo temos um relacionamento com cardinalidade **N:N**, pois 1 professor pode ministrar disciplinas em vários cursos, enquanto 1 curso pode ter vários professores ministrando disciplinas nele.



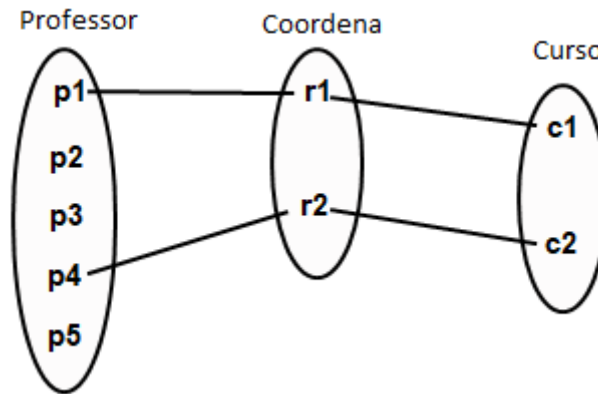
A **restrição de participação** especifica se a existência de uma entidade, depende de ela estar relacionada com outra entidade através de um relacionamento. Existem as restrições de participação **total** e **parcial**.

Se uma faculdade estabelece a regra de que todo professor deve ministrar aulas em um curso, então uma entidade Professor somente pode existir se ela participar do relacionamento Ministra, logo a entidade **p5** do exemplo anterior não deve existir.

A participação de **Professor** em **Ministra** é chamada **total**, o que significa que toda entidade Professor deve estar relacionada a uma entidade Curso via Ministra.

A restrição de participação total é algumas vezes chamada de dependência existencial.

No exemplo a seguir não é esperado que todo professor coordene um curso, assim a participação de Professor no tipo de relacionamento Coordena é **parcial**. Isso significa que algumas entidades, do conjunto de entidades Professor, poderão estar relacionadas a uma entidade Curso via Coordena, mas não necessariamente todas.



VII. DER – Diagrama Entidade Relacionamento

O MER e DER são modelos conceituais. Por possuir uma representação gráfica o DER é mais indicado na prática para construir o modelo conceitual. Em situações práticas, o diagrama é tido muitas vezes como sinônimo de modelo, uma vez que sem uma forma de visualizar as informações, o modelo pode ficar abstrato demais para auxiliar no desenvolvimento do sistema. Dessa forma, quando se está modelando um domínio, o mais comum é já criar sua representação gráfica.

O diagrama facilita a comunicação entre os integrantes da equipe, pois oferece uma linguagem comum utilizada para a equipe de desenvolvedores conversar com o cliente.

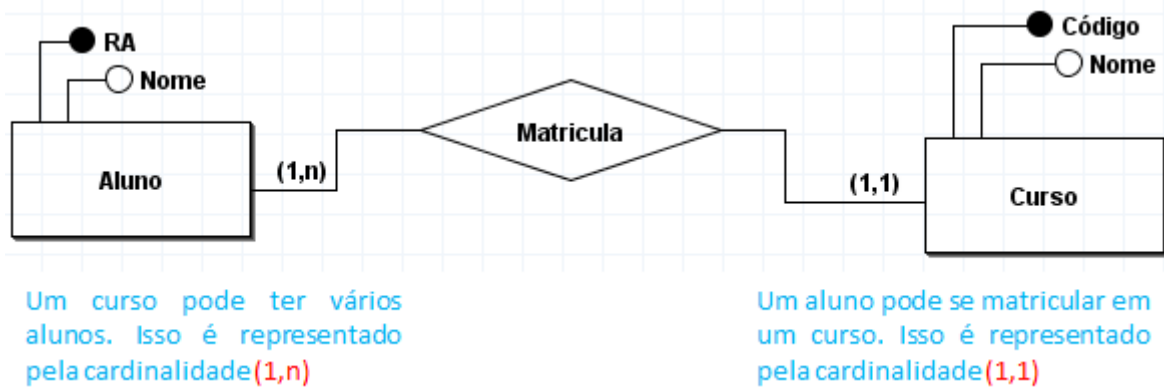
O DER é composto por um conjunto de objetos gráficos que visa representar todos os objetos do MER, tais como, **entidades**, **atributos**, **relacionamentos** etc. A ênfase do DER está na representação de **esquemas** ao invés de **instâncias**, como o MER.

Tipos de entidades são representados em um DER por meio de retângulos contendo o nome do tipo de entidade no seu interior e arestas ligando esses retângulos aos atributos. Em sua notação original, proposta por Peter Chen (idealizador do modelo e do diagrama), as entidades deveriam ser representadas por retângulos, seus atributos por elipses e os relacionamentos por losangos, ligados às entidades por linhas, contendo também sua cardinalidade (1..1, 1..n ou n..n). Porém, notações mais modernas abandonaram o uso de elipses para atributos e passaram a utilizar o formato mais utilizado na UML, em que os atributos já aparecem listados na própria entidade. Essa forma torna o diagrama mais limpo e fácil de ser lido.

Um DER possui conceitos que os usuários leigos são capazes de entender, ao mesmo tempo em que não está distante do modelo físico do BD.

Neste nível o projeto é independente de SGBD. Tomamos como exemplo a representação a seguir de uma matrícula do aluno em um curso usando a notação de DER.

Nesta representação 1 Aluno deve obrigatoriamente participar do relacionamento 1 vez (representado pela cardinalidade **1:1**) e 1 Curso precisa ter no mínimo uma matrícula e no máximo várias matrículas (representado pela cardinalidade **1:n**).

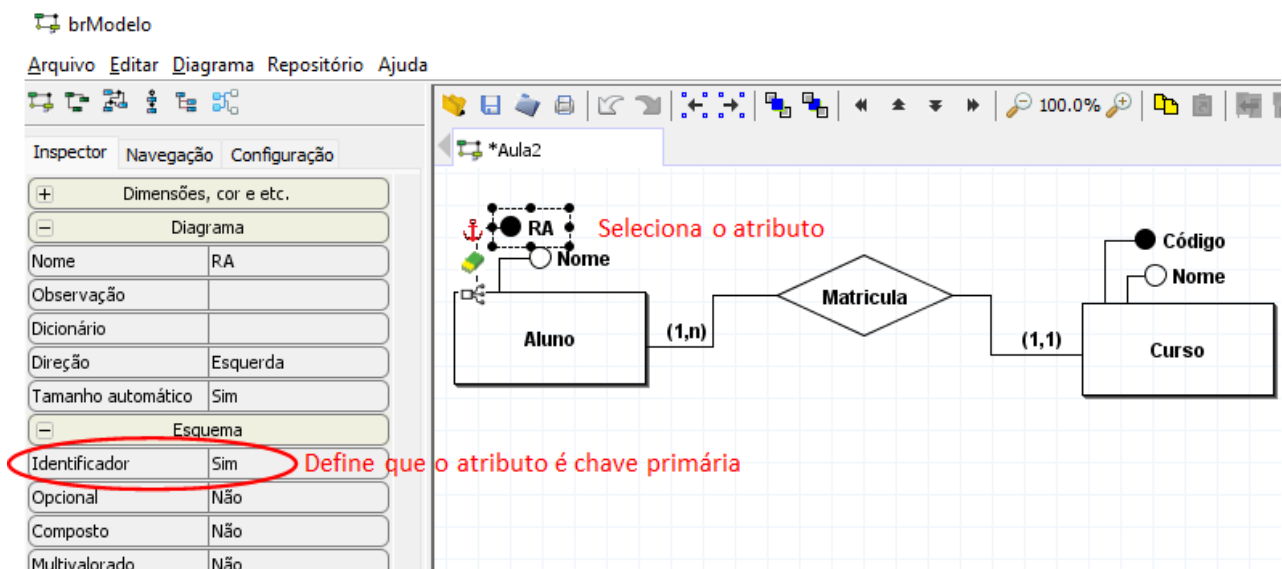


As cardinalidades no DER são representadas graficamente pelos números 0, 1 e n junto às arestas que relacionam os tipos de entidades.

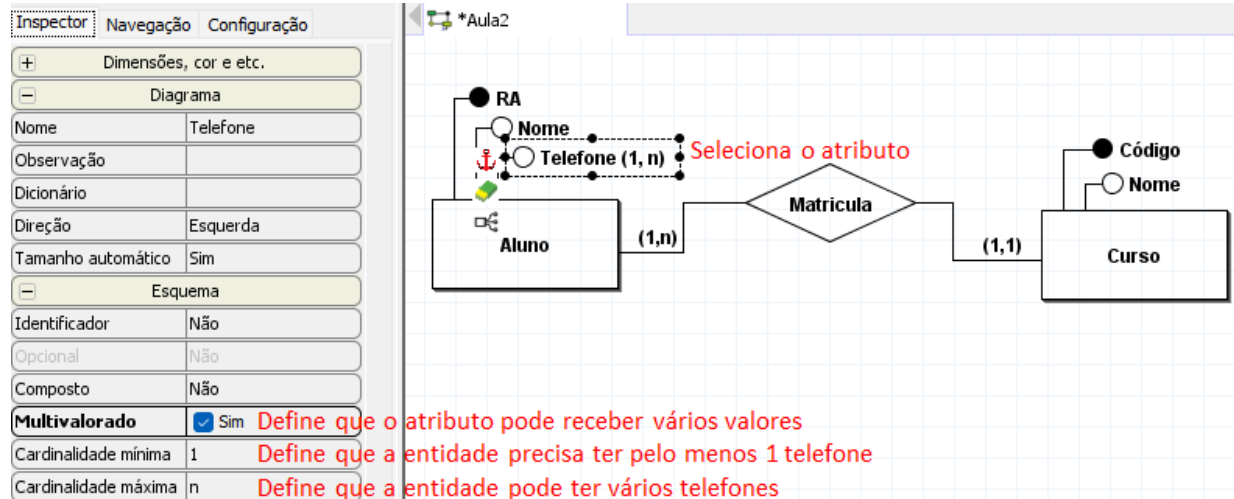
O DER foi feito usando a ferramenta brModelo (<http://www.sis4.com/brModelo/download.html>). Faça o download do arquivo brModelo.zip e descompacte no seu computador. Copie somente a pasta **dist** para o local de sua preferência. Para executar o software é necessário dar um duplo clique sobre o arquivo **brModelo.jar**.

No DER os atributos recebem propriedades:

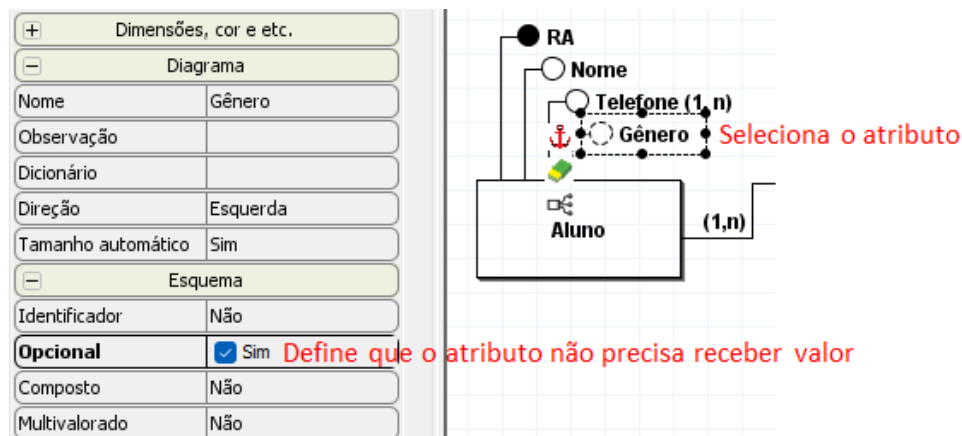
- Atributos chaves: são os identificadores únicos da entidade. Na representação do brModelo os atributos chaves são preenchidos. No exemplo a seguir os atributos **RA** e **Código** são chaves primárias;



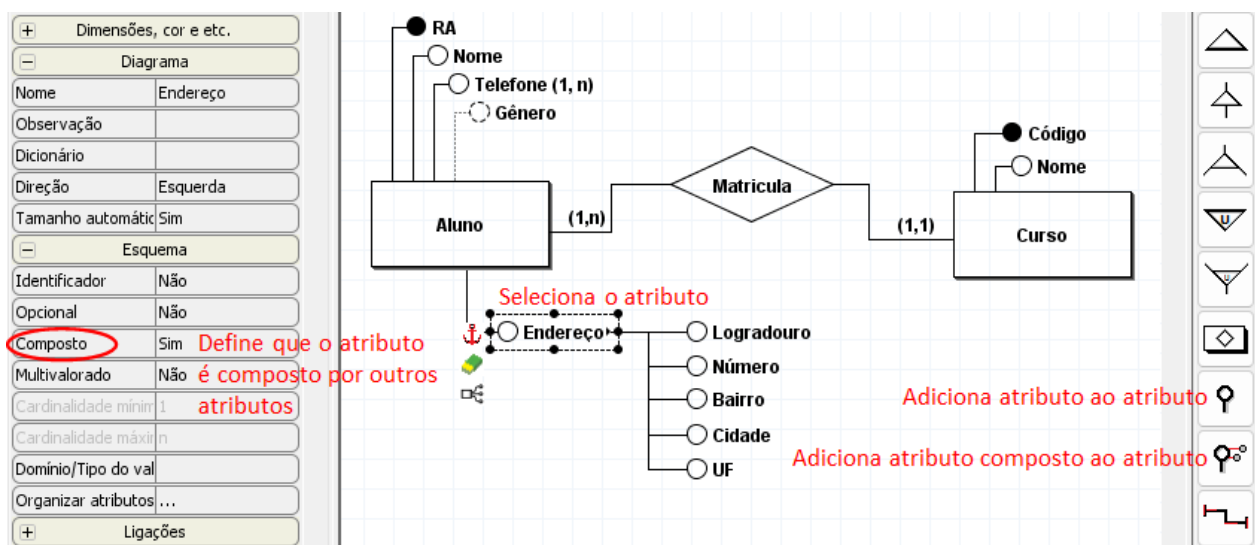
- Atributo multivalorado: um aluno pode ter vários telefones. Na representação do brModelo os atributos multivalorados são representados pela cardinalidade do atributo. No exemplo a seguir o atributo Telefone pode receber vários valores, sendo que o mínimo é 1 telefone;



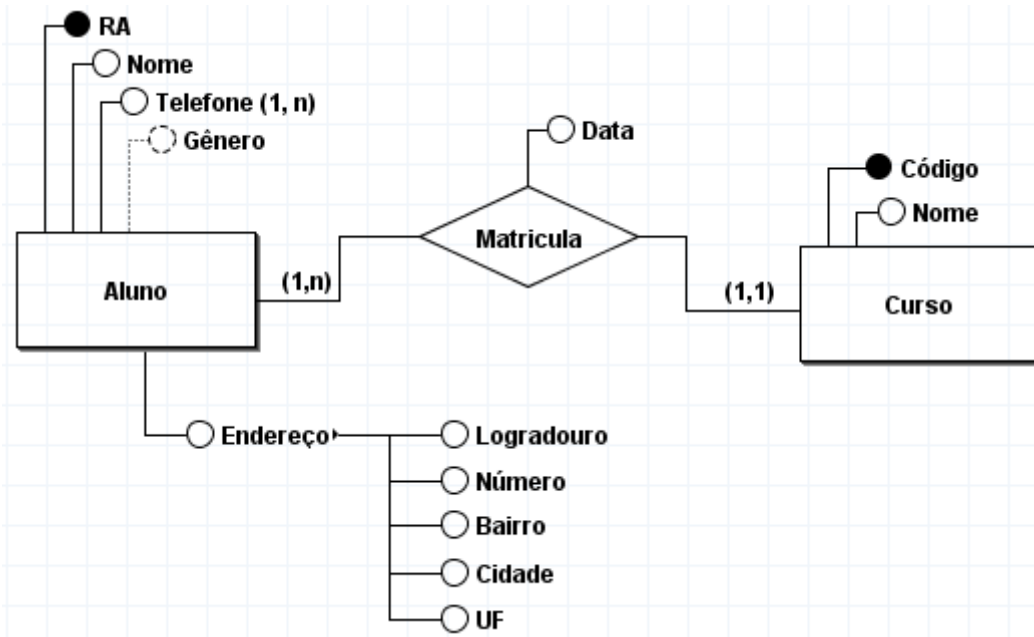
- Atributo opcional: é aquele que aceita valores nulos, isto é, não tem a exigência de receber valor. No brModelo o atributo opcional é representado por uma bola tracejada;



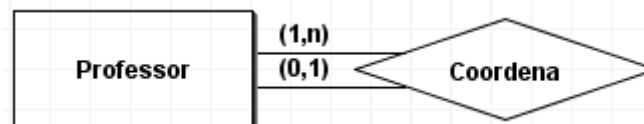
- Atributo composto: é aquele que é formado por outros atributos. No exemplo a seguir o Endereço é formado pelos atributos Logradouro, Número, Bairro, Cidade e UF. No BD não existirá o atributo Endereço, somente os atributos Logradouro, Número, Bairro, Cidade e UF;



Os relacionamentos podem ter atributos. No exemplo a seguir o atributo **Data** da matrícula está associado ao relacionamento.



O autorelacionamento ocorre quando entidades de um tipo de relacionamento se relacionam com entidades do mesmo tipo. No exemplo a seguir um Professor pode coordenar vários professores. O número zero significa que nem todos os professores são coordenadores.



Exercícios

Exercício 1: O que é um relacionamento recursivo?

Exercício 2: Qual é a diferença entre as restrições de participação total e parcial no tipo de relacionamento?

Exercício 3: Um administrador de salão de eventos gostaria de ter um sistema para registrar as pessoas que participam de cada evento. Fazer o Modelo Conceitual considerando os seguintes requisitos:

- O sistema deverá manter um cadastro dos clientes responsáveis pelos eventos com os seguintes dados: CPF, nome do cliente, telefones e e-mails (opcional);
- O sistema deverá manter um cadastro dos eventos com os seguintes dados: data, horário de início, horário de fim, cliente responsável pelo evento e pessoas convidadas para o evento. Cada evento possui somente um cliente responsável, mas um cliente pode ser responsável por vários eventos;

- O sistema deverá manter um cadastro de convidados: CPF, nome, telefone (opcional) e e-mail (opcional). Um convidado pode participar de vários eventos.
- O sistema deverá manter o registro dos convidados que compareceram ao evento.

Exercício 4: Um programador gostaria de desenvolver um sistema de agenda. Fazer o Modelo Conceitual considerando os seguintes requisitos:

- O sistema deverá manter um cadastro de contatos com os seguintes dados: nome, telefones, e-mails (opcional) e endereços (opcional);
- O endereço deverá ser formado somente pelo nome, tal como, Fatec Jacareí, Fatec SJC, Chácara São João, Apartamento da Vila Maria etc. Não será necessário armazenar o logradouro, bairro, cidade etc. do endereço;
- O mesmo endereço pode estar associado à vários contatos;
- Para cada endereço do contato pode ser possível cadastrar o horário que ele se encontra, sendo possível estipular somente uma das seguintes opções: manhã, tarde, noite e finais de semana.

Exercício 5: O patrimônio de uma empresa gostaria de criar um sistema para manter o registro dos equipamentos. Fazer o Modelo Conceitual considerando os seguintes requisitos:

- O sistema deverá manter um cadastro de prédios com o nome do prédio, por exemplo, Pavilhão João Garcia, Centro de Convenções Ana Albuquerque etc.;
- O sistema deverá manter um cadastro de salas formado pelo nome da sala e o prédio. Exemplos de nomes de salas seriam Auditório, Secretaria etc.;
- O sistema deverá manter um cadastro de equipamentos formado pelo nome do equipamento. Exemplos de equipamentos seriam computador, impressora, scanner etc.;
- O sistema deverá manter um cadastro dos itens patrimoniados formado pelo número de patrimônio, tipo de equipamento e período que o equipamento esteve em cada sala.

Exercício 6: O administrador da secretaria de agricultura do município deseja criar um sistema de cadastro de propriedades rurais. Fazer o Modelo Conceitual considerando os seguintes requisitos:

- O sistema deverá manter um cadastro de propriedades com o nome da fazenda e área;
- O sistema deverá manter um cadastro de donos de fazendas com o nome do fazendeiro e telefone;
- Uma fazenda pode ter vários donos e um fazendeiro pode ter várias fazendas;
- O sistema deverá manter no cadastro as atividades agropecuárias da fazenda. Exemplos de atividades são: pecuária de corte, pecuária leiteira, plantação de soja, plantação de milho etc. Uma fazenda poderá ter várias atividades agropecuárias.

Exercício 7: O administrador de redes de uma empresa deseja criar um sistema de registros de acessos dos usuários aos computadores. Fazer o Modelo Conceitual considerando os seguintes requisitos:

- O sistema deverá manter um cadastro de computadores com o nome da máquina;
- O sistema deverá manter um cadastro de usuários com os seguintes dados: nome, e-mail e senha;
- O usuário precisará efetuar login na máquina usando e-mail e senha, ao final o usuário deverá efetuar o logout. O sistema deverá manter um cadastro de máquinas acessadas por cada usuário incluindo a data e horário de início e fim do acesso;
- O sistema deverá manter um cadastro dos endereços (URL) acessados por cada máquina e o horário de acesso.

Exercício 8: A concessionária de uma rodovia deseja criar um sistema de registros de passagens de veículos nas suas praças de pedágio. Fazer o Modelo Conceitual considerando os seguintes requisitos:

- O sistema deverá manter um cadastro de praças de pedágio com o nome da praça e o sentido (norte, sul, leste, oeste etc.);
- O sistema deverá manter um cadastro dos guichês de cada praça com o número do guichê;
- Cada praça de pedágio possui uma tabela de valores para as categorias de veículos (motocicleta, automóvel, caminhão etc.);
- O sistema deverá manter um registro dos veículos que passaram em cada guichê contendo a placa do automóvel, data e horário e valor pago.

Exercício 9: O dono de um supermercado deseja criar um sistema para auxiliar na gestão do estoque. Fazer o Modelo Conceitual considerando os seguintes requisitos:

- O sistema deverá manter um cadastro de fornecedores com nome, telefone e e-mail;
- O sistema deverá manter um cadastro de produtos no estoque com nome, valor de custo, valor de venda e quantidade;
- O sistema deverá manter um cadastro de clientes com o nome;
- O sistema deverá manter um cadastro de vendas com data e horário, cliente (opcional) e produtos.

Exercício 10: O dono de um estacionamento deseja criar um sistema para auxiliar na gestão das vagas. Fazer o Modelo Conceitual considerando os seguintes requisitos:

- O sistema deverá manter um cadastro das vagas disponíveis. Cada vaga é identificada por um nome, tal como, 1A, 3D etc.;
- O sistema deverá manter um cadastro dos veículos que usaram o estacionamento com os dados de placa, marca e modelo;
- O sistema deverá manter um cadastro das vagas ocupadas por cada veículo com o horário de início e fim e o valor pago.