

Modelo de Dados Hierárquicos

Primeiro modelo de dados a ser reconhecido. Usa uma estrutura de árvores onde cada registo é considerado uma coleção de campos ou atributos.

Modelo de Dados Relacional

Sucessor do modelo Hierárquico. Baseia-se no conceito de Entidades e Relacionamentos.

Melhorias nos SGBD's

Os Sistemas Gerenciadores de Banco de Dados começam a ser melhorados devido a grande aceitação dos usuários.

Modelo de Dados NoSQL

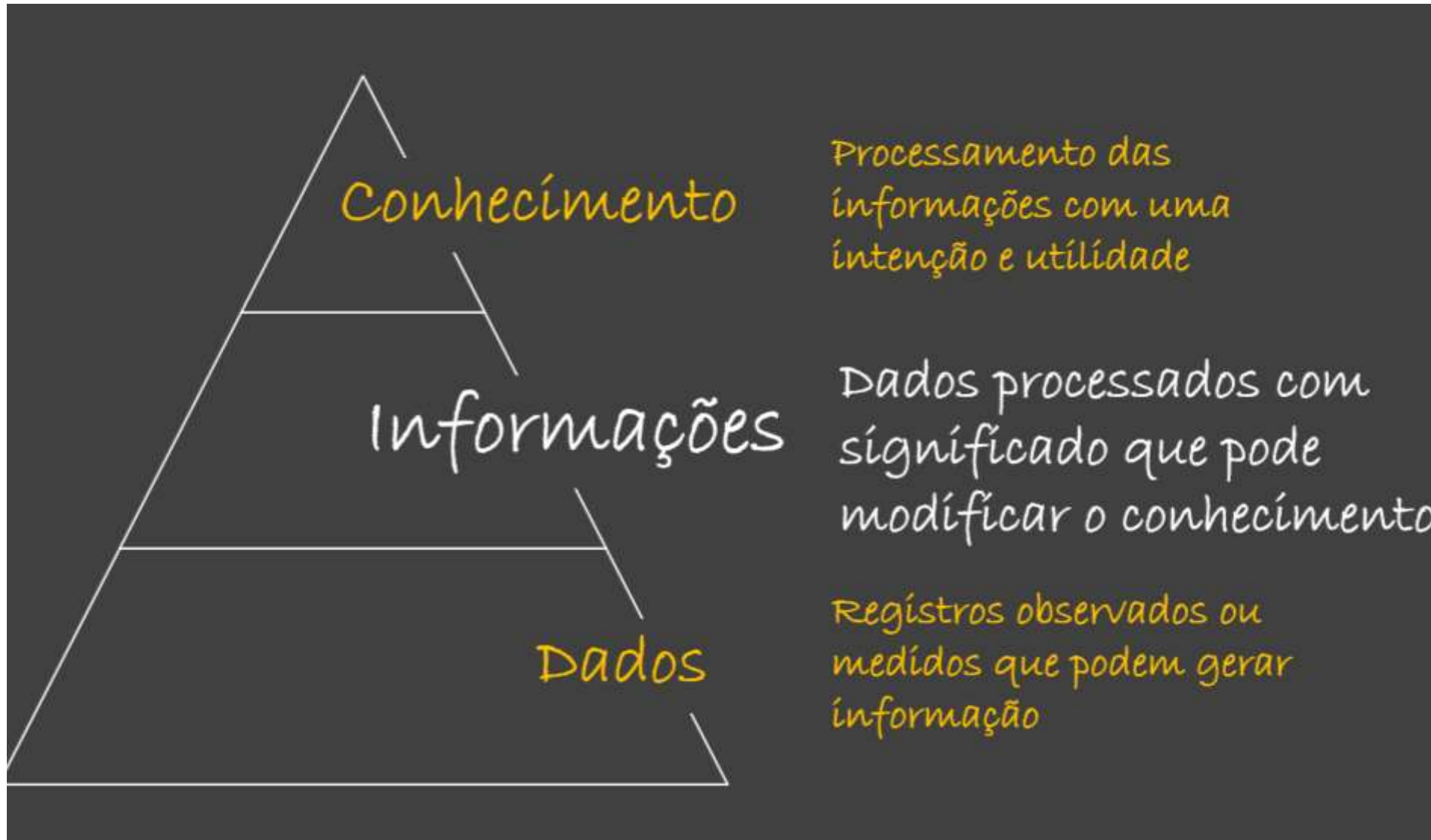
Surgem as primeiras alternativas aos modelos relacionais baseados em documentos, chave-valor ou famílias de colunas.

Modelo de Dados NoSQL

As bases de dados NoSQL começam a ser reconhecidas devido ao alto poder de performance e escalabilidade.



Dado -> Informação -> Conhecimento



Exemplo

3 – Ele vem fazendo ótimo trabalho = promoção.

2 – “João fez um ótimo trabalho nesse mês.

1 – “João”.



Banco de Dados

Um banco de dados é uma coleção organizada de dados que podem ser facilmente acessados, gerenciados e atualizados. Eles são utilizados para armazenar informações de forma estruturada, permitindo consultas eficientes e manipulação dos dados.

E-commerce

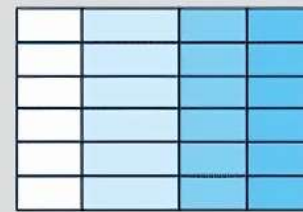
Estoque de uma farmácia

Reserva de voos

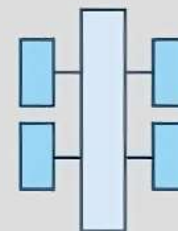


SQL

Relational

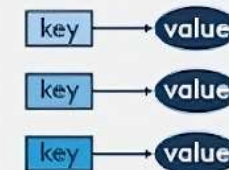


Analytical (OLAP)

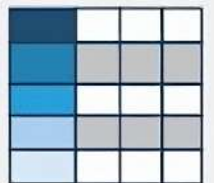


NoSQL

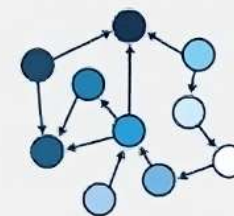
Key-Value



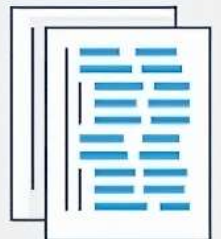
Column-Family



Graph



Document



Chaves e atributos

SGBD: Os Sistemas Gerenciadores de Banco de Dados (SGBDs) são softwares que permitem a criação, manipulação e administração de bancos de dados. Eles desempenham um papel crucial no gerenciamento eficiente dos dados, proporcionando um ambiente controlado e seguro para armazenar e acessar informações.



Podem ser computacionais ou não-computacionais (por exemplo: Registro em atas em um determinado formato para mantê-las de forma organizada e fácil de entender) ser gerenciados por um SGBD ou não (exemplo: pastas).



São computacionais gerenciados por um SGBD.

Organizados em tabelas, que são compostas por linhas e colunas e as tabelas podem se relacionar, sendo assim, dados mais detalhados podem ser gerados.



Um modelo de dados representa a forma como os dados se relacionam com o mundo real. Reflete como os dados devem ser estruturados e padronizados para melhor refletir os processos, definições, fluxos de trabalho e lógica da sua organização.

Um bom modelo de dados captura como a comunicação e o trabalho fluem naturalmente dentro da sua organização. Em contraste, um modelo de dados deficiente (ou inexistente) é aleatório, confuso e incoerente.



É a representação genérica de um componente do mundo real, sobre o qual desejamos armazenar informações, uma representação de quase todas as informações com várias propriedades que devem ser compreendidas pelo sistema de informação, qualquer coisa que produza ou consuma informações. Ex: pessoa, empresa, etc.

Entidade são coisas significativas sobre a qual a organização deseja guardar, ou seja, (coletar, manter e etc) dados podendo ser algo tangível ou intangível.



Tipos de Entidade

Coisas tangíveis: todos os elementos que tenham existência concreta.

Fisicamente existente. Ex: Produto, animal, carro.

Funções: todo o tipo de papel, atribuição, classificação, capacitação, ou outra característica que especifique atuação. Ex: Cliente, professor, departamento.

Eventos ou ocorrências: só conseguem ser percebidos ou caracterizados, enquanto uma certa ação se desenrola. Ex: Lançamento em conta corrente.



Atributos são propriedades que descrevem cada membro de um conjunto de entidades.

As entidades possuem valores similares para os atributos, mas cada entidade pode ter seu próprio valor.

*Ex: para a entidade Livro atributos: título, autor, editora, ano, ISBN, etc.;
para a entidade Autor atributos: nome, data de nascimento, nacionalidade, gênero, etc.*



Atributo Multivalorado

Um atributo multivalorado é um atributo de uma entidade em um banco de dados que pode conter múltiplos valores para uma única instância dessa entidade.

| PESSOA | | | | |
|--------|-------|------|--------------------|------------------|
| CPF | Nome | Sexo | Localização | Telefone |
| 111 | Ana | F | Rio de Janeiro, RJ | 999-444, 999-000 |
| 222 | Bruno | M | São Paulo, SP | 888-888, 444-333 |
| 333 | Carla | F | Belo Horizonte, MG | 555-777 |
| 444 | Diego | M | Vitória, ES | 999-999 |

Atributo Multivalorado



Atributos compostos são atributos que poderiam ser subdivididos em vários atributos:

| ID_Funcionario | Nome_Funcionario | Endereço |
|----------------|------------------|--|
| 1 | João | Av. Mem de Sá, 100, apto 101 – Centro – Rio de Janeiro - RJ |
| 2 | Katia | Av. Portugal, 324, casa 01 – Urca – Rio de Janeiro – RJ |
| 3 | Luis | Av. Vieira Souto, 1300, apto 802, Leblon – Rio de Janeiro – RJ |



Chave primária (Primary Key = PK)

A chave primária é a chave candidata escolhida que melhor identifica uma entidade dentro do seu conjunto de entidades.

É o atributo que distingue uma linha das demais dentro da tabela.

Ex: matrícula.

curso

| codigo | titulo |
|---------------|----------------|
| 1 | Banco de dados |
| 2 | Linux |



Tipos de chave primária

Uma chave primária natural é uma coluna ou um conjunto de colunas que já contém valores únicos de forma inerente, como números de identificação (CPF, por exemplo) ou endereços de email.

| CPF | NOME | ENDERECO | TELEFONE |
|----------------|---------|---------------|-----------|
| 000.000.000-00 | José | Rua ali 23 | 0000-0000 |
| 111.111.111-00 | José | Rua lá 30 | 1111-1111 |
| 222.222.222-00 | Roberto | Rua perto 1 | 2222-2222 |
| 333.333.333-00 | Olivia | Rua longe 999 | 3333-3333 |

Uma chave primária artificial, também conhecida como chave substituta ou surrogate key, é uma coluna adicionada especificamente para servir como identificador único, como um número sequencial ou um identificador gerado automaticamente pelo sistema, sem significado intrínseco no contexto dos dados.

TABELA CLIENTES

| ID_CLIENTE | NOME |
|------------|--------|
| 1 | DANIEL |
| 2 | JOAO |
| 3 | PAULO |

Surrogate key
autoincrement



Chave estrangeira (Foreign Key = FK)

Chave estrangeira, ou Foreign Key (FK) é a chave que permite a referência a registros oriundos de outras tabelas.

TABELA CLIENTES

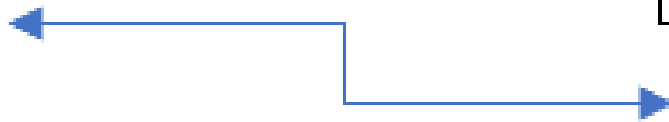
| ID_CLIENTE | NOME |
|------------|--------|
| 1 | DANIEL |
| 2 | JOAO |
| 3 | PAULO |

PK

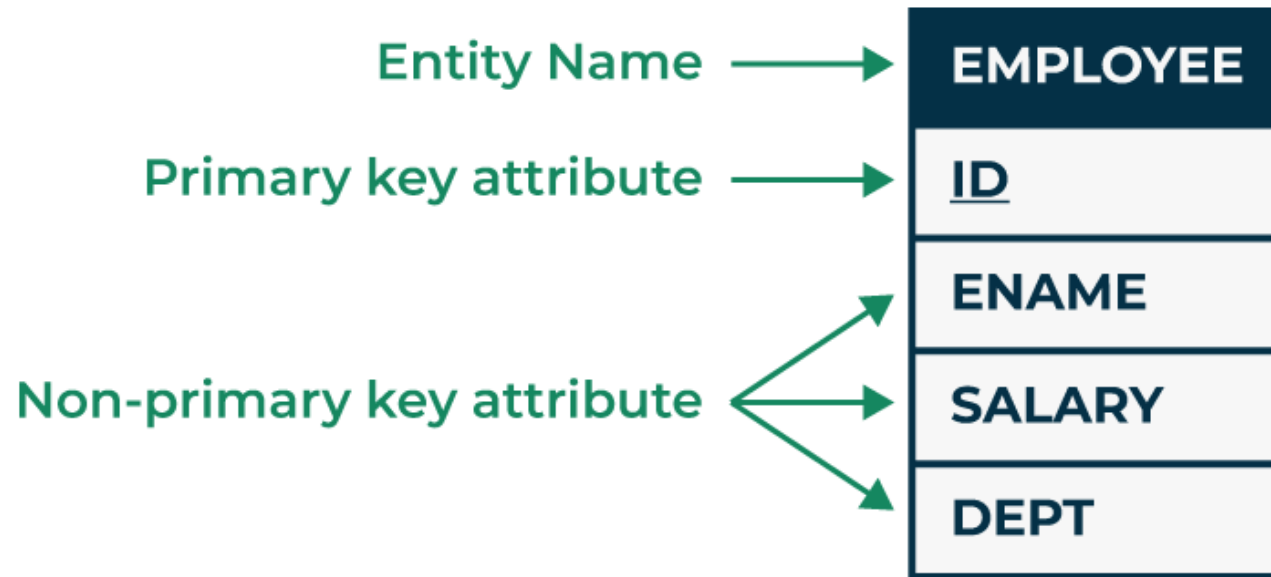
TABELA VENDAS

| IDCLIENTE | PRODUTO | QUANTIDADE |
|-----------|-----------|------------|
| 1 | AGUA | 1 |
| 1 | BISCOITO | 2 |
| 1 | CHOCOLATE | 1 |
| 2 | AGUA | 2 |

FK



Chaves e atributos



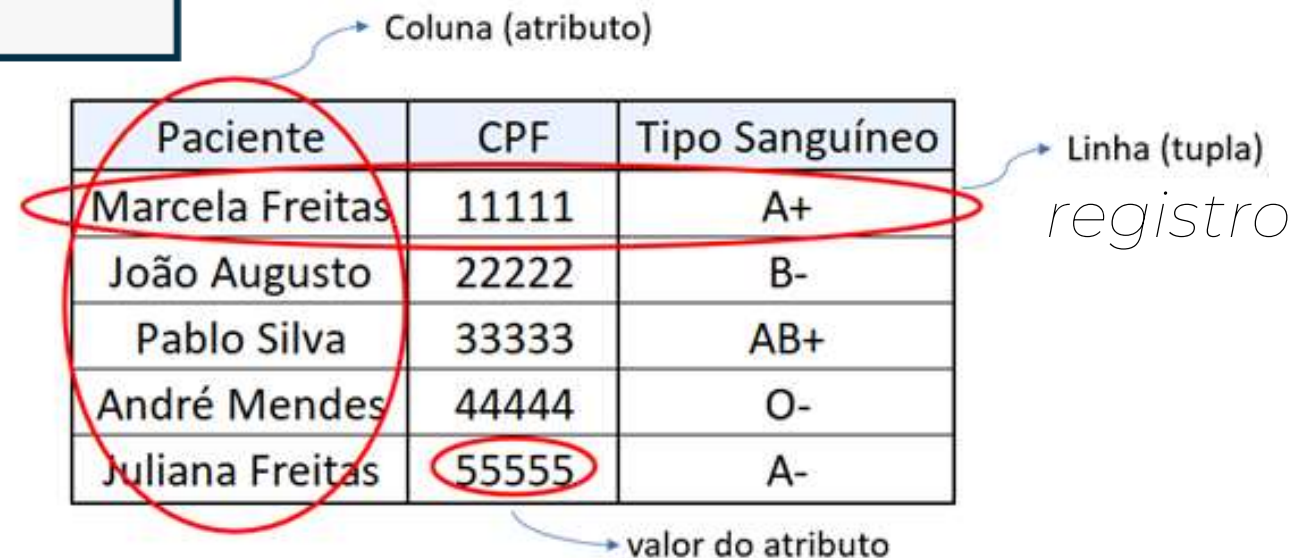
Entidade = Tabela

Coluna (atributo)

| Paciente | CPF | Tipo Sanguíneo |
|-----------------|-------|----------------|
| Marcela Freitas | 11111 | A+ |
| João Augusto | 22222 | B- |
| Pablo Silva | 33333 | AB+ |
| André Mendes | 44444 | O- |
| Juliana Freitas | 55555 | A- |

Linha (tupla)
registro

valor do atributo

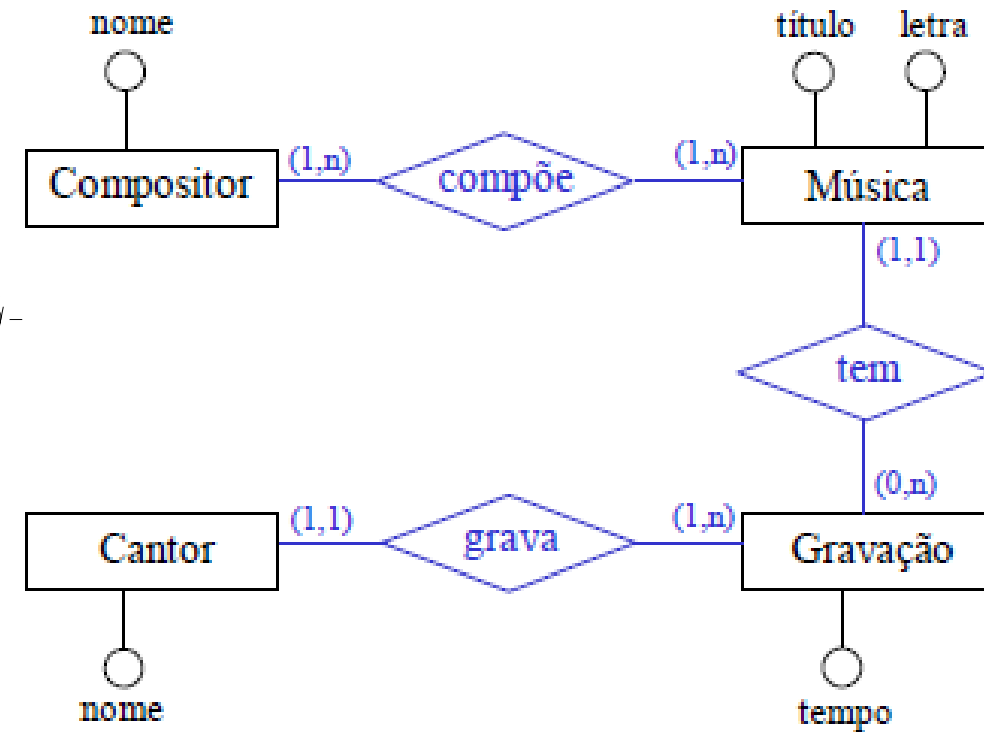


Modelo Conceitual

Contém lógica e regras de negócios e descreve os dados do sistema, como esquemas, tabelas e campos (nomes e tipos).

Ao criar um modelo conceitual, geralmente é útil visualizá-lo em um diagrama entidade-relacionamento (ER), que é uma ferramenta padrão para visualizar os relacionamentos entre várias entidades em seus dados (pedidos, clientes, produtos, etc.).

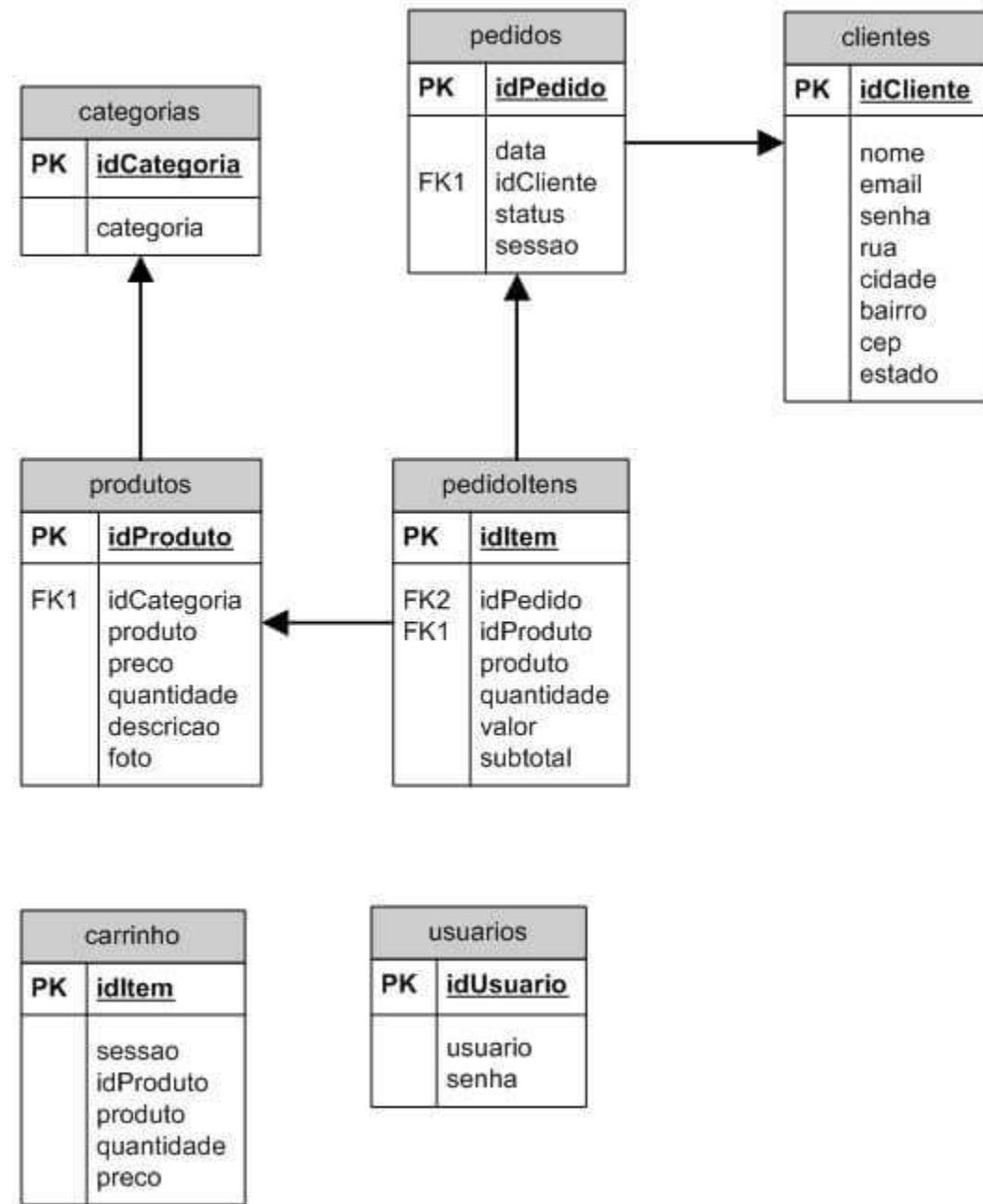
A visualização de relacionamentos de entidade é altamente recomendada para projetar um modelo de dados conceitual coerente.



Modelo Lógico

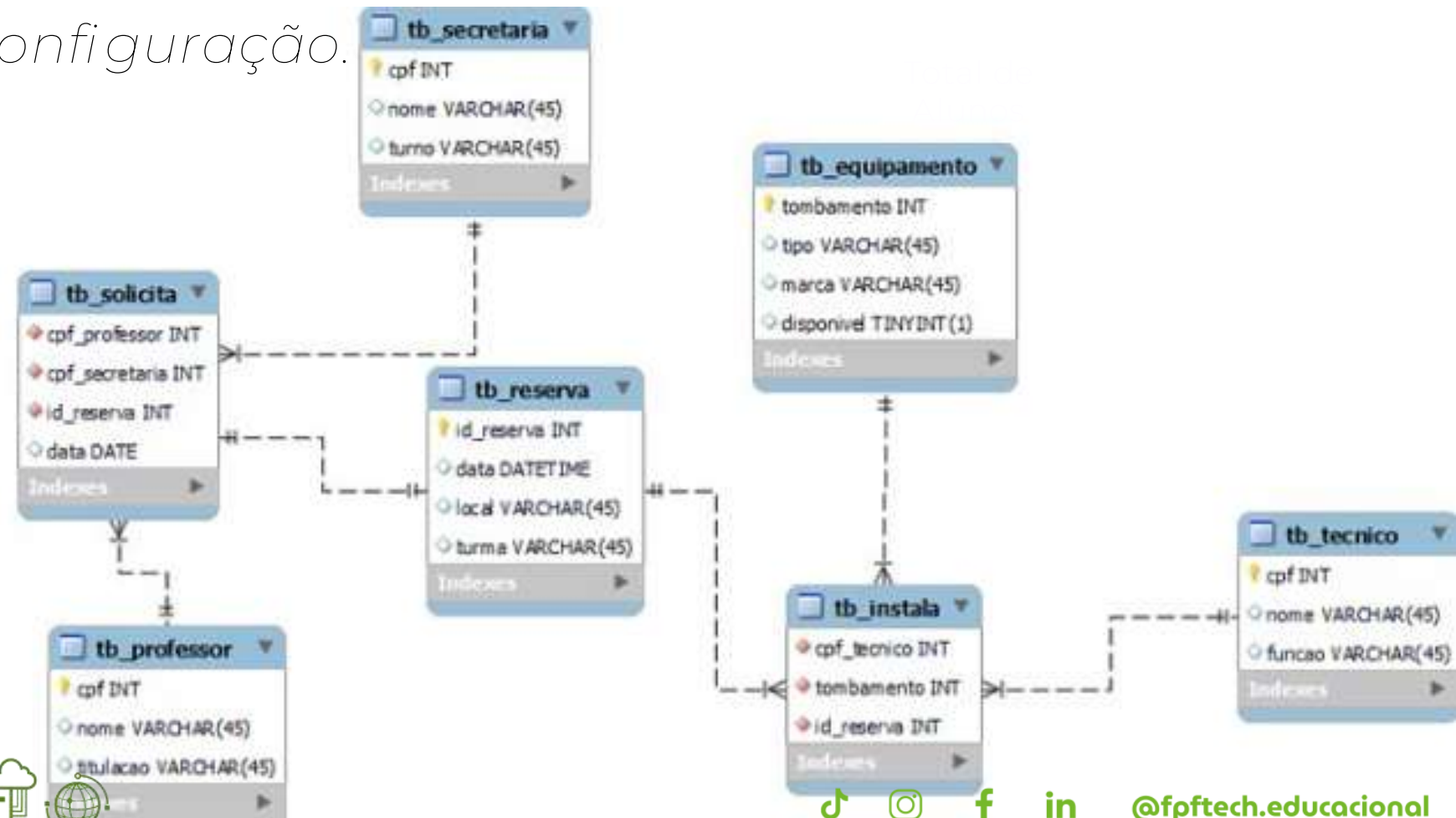
Detalha como o modelo conceitual será implementado na prática adicionando significativamente mais detalhes. Por exemplo, adicionaríamos informações sobre os tipos de ID do cliente, nomes de clientes e endereços personalizados.

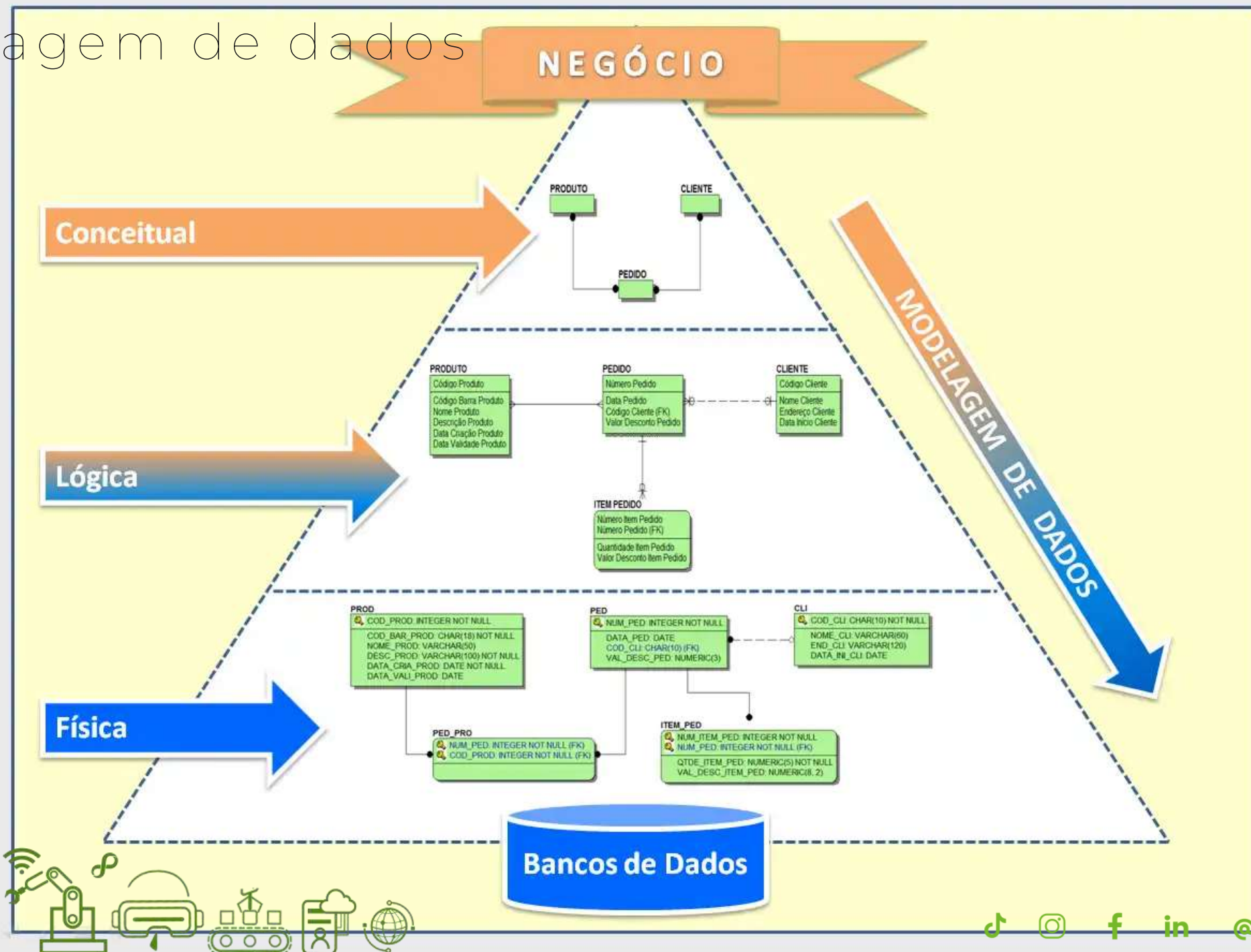
Além disso, mapeamos chaves primárias PKs e estrangeiras FKs.



Modelo Físico

Implementado em um sistema de banco de dados. Adicionaríamos bancos de dados, esquemas e tabelas específicos ao nosso modelo lógico, incluindo detalhes de configuração.





Cardinalidade

A relação entre as tabelas, especificamente ao número de registros em uma tabela que podem estar relacionados a um único registro em outra tabela.

Existem três tipos principais de cardinalidade:

Um-para-um (1:1)

Um-para-muitos (1:n)

Muitos-para-muitos (n:n)



Cardinalidade 1:1

Cardinalidade 1 para 1

Cada instância de uma entidade se relaciona com uma e somente uma instância de outra entidade.

Pegamos o máximo dos 2.

Cardinalidade 1:1

Se Lê: Cardinalidade um pra um

Exemplo: Empresa



Cardinalidade 1:n

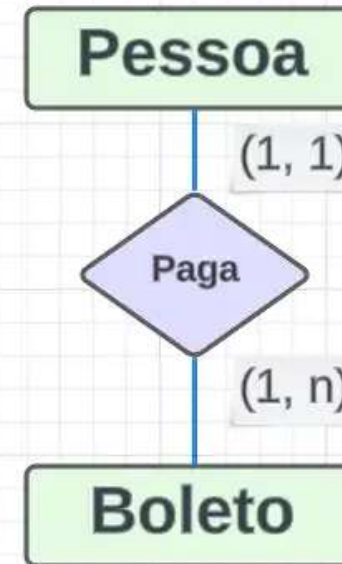
Cardinalidade 1 para n

No relacionamento um-para-muitos significa que cada instância de uma entidade pode se relacionar com várias instâncias de outra entidade, mas não o contrário.

Cardinalidade 1:n

Se Lê: Cardinalidade um pra Muitos

Exemplo: Boletos



Um Boleto só pode ser pago por no mínimo 1 pessoa e no máximo 1 pessoa

Uma pessoa paga no mínimo 1 boleto e no máximo muitos boletos

Legenda

(1, n) n = muitos

No Mínimo

No Máximo

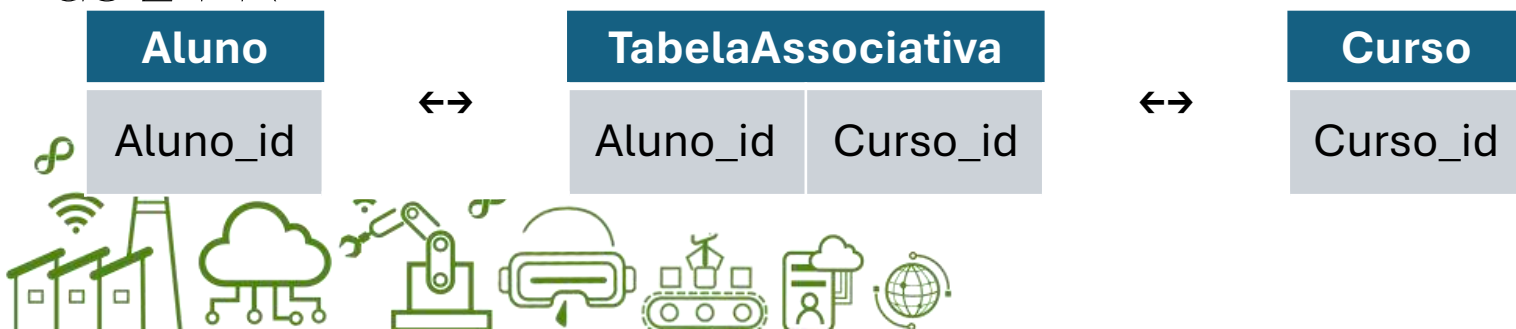


Cardinalidade n:n

Cardinalidade n para n

No relacionamento muitos-para-muitos significa que várias instâncias de uma entidade podem se relacionar com várias instâncias de outra entidade

cria-se uma tabela associativa, associando as 2 FK



Cardinalidade n:m

Se Lê: Cardinalidade muitos pra vários

Exemplo: Cursos DIO

Aluno DIO

(1, n)



(1, m)

Bootcamp

Um Bootcamp pode ser cursado por no mínimo 1 aluno DIO e no máximo muitos alunos DIO

Um aluno DIO pode cursar no mínimo 1 bootcamp e no máximo vários bootcamps

Legenda

(1, n)

n = muitos

No Mínimo

No Máximo

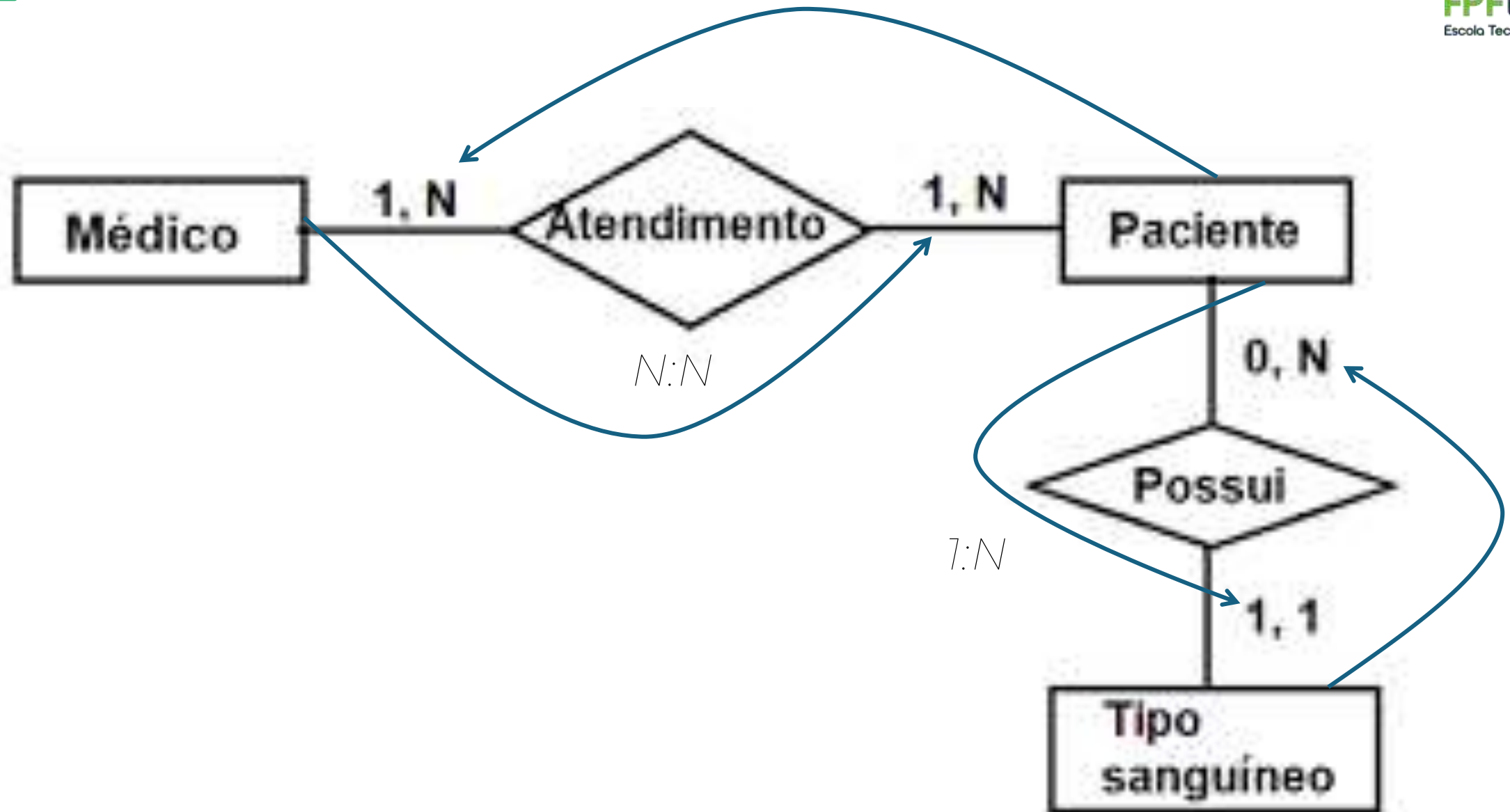
(1, m)

m = vários

No Mínimo

No Máximo

Exemplo



Cardinalidade Exercício

1 – Livro e Autor = Livro de autor único

2 – Pessoa e Passaporte

3 – Produtos e Pedidos

4 – Estudantes e Cursos

5 – Cliente e Pedido

6 – Professores, Turmas, Alunos

7 – Pessoa e Habilitação

8 – Funcionário e Departamento



Obrigado!



    @fpftech.educacional