

BANCO DE DADOS I

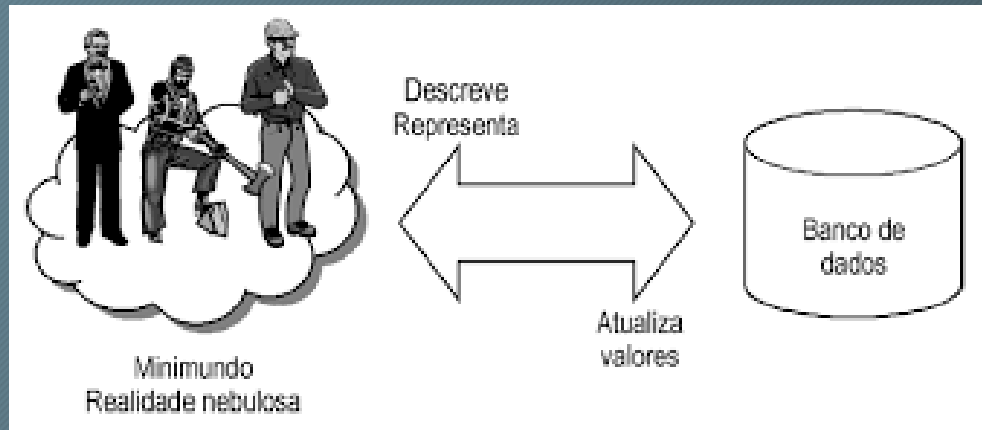
Projeto Lógico

Desenvolvimento:

- Há três tipos de modelos de dados: o conceitual, o lógico e o físico. Uma estrutura de dados com qualidade é imprescindível para garantir a legitimidade do banco de dados facilitando a manutenção do sistema de aplicação.
- Dado é a parte mais importante de um sistema de aplicação.
- Uma estrutura de dados acurado, permite ao analista de sistema da aplicação desenhar os processos, a interface de usuário, relatórios e análise estatística sempre que precisar.

Desenvolvimento: Passo a passo

- Na fase inicial do processo, o mundo real (ou mini mundo) deve ser entendido e seus objetos conceituais identificados.
- A este entendimento e identificação chamamos abstração de dados e o modelo produzido após esta fase chamamos modelo conceitual.

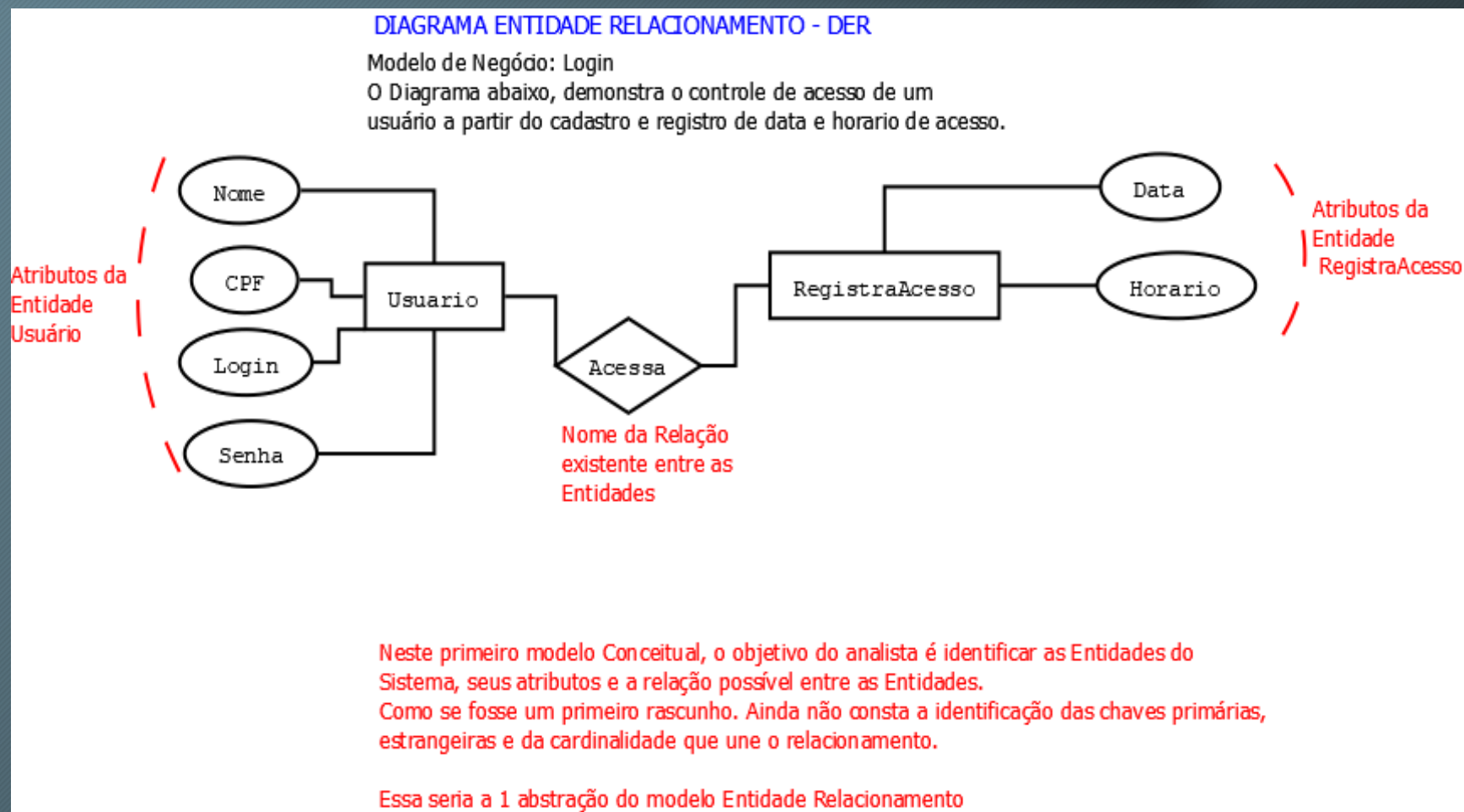


Desenvolvimento: Passo a passo - Modelo Conceitual

- Após a compreensão da realidade do negócio, o analista esboça o modelo conceitual com a identificação das Entidades, Atributos e possíveis relações
- Com este primeiro modelo, pode-se ilustrar uma primeira visão do Diagrama Entidade Relacionamento, ainda sem definir o grau de cardinalidade nas relações.

Desenvolvimento: Passo a passo - Modelo Conceitual

- A imagem ilustra a relação de duas entidades para controlar o acesso por meio de login a um sistema.



O Diagrama ao lado foi desenvolvido com a ferramenta Case DIA, utilizando a denotação de PETER CHEN.

Desenvolvimento:

Passo a passo - Início do modelo Lógico - Identificação das Chaves primárias

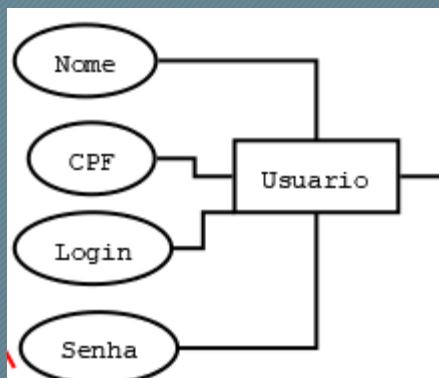
- Como já visto anteriormente, chave primária refere-se aos conjuntos de um ou mais campos, cujos valores, nunca se repetem na mesma tabela e, desta forma, podem ser usadas como um índice de referência para criar relacionamentos com as demais tabela do banco de dados, portanto, uma chave primária nunca pode ter valor nulo, nem repetição.

Desenvolvimento:

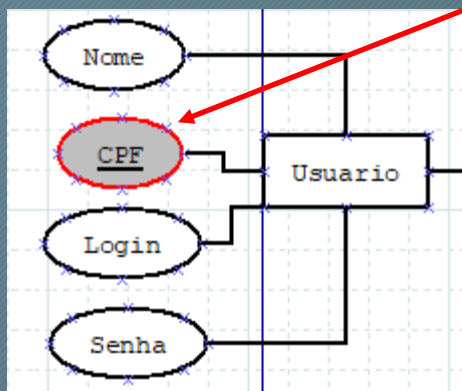
Passo a passo - Início do modelo Lógico - Identificação das Chaves primárias

- Considerando nosso exemplo, para cada campo deve ser aplicada a REGRA DE IDENTIDADE, ou seja, verificar se o campo:
 - É de preenchimento obrigatório (não pode conter valores nulos)
 - Não permite a repetição de dados para registros diferentes (não pode repetir valores).
- Quando mais de um campo na Entidade atender a REGRA DE IDENTIDADE, deve-se priorizar o atributo que seja menor e composto somente por números inteiros para ser definido como CHAVE PRIMÁRIA.
- Dependendo da necessidade do modelo de Negócio, ou se não houver campos que atendam a regra de Identidade, pode-se criar um ATRIBUTO para receber a chave primária.
 - Denominamos esse atributo de: ATRIBUTO ARTIFICIAL, não é nativo da Entidade, foi definido pelo analista por uma necessidade do modelo. Exemplo: CodProduto, RA, IdVeiculo, etc.

Desenvolvimento: Passo a passo - Identificação das Chaves primárias



Chave primária

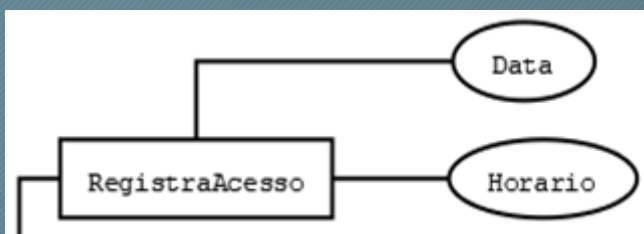


Para cada campo verificamos se o atributo pode receber valores repetidos ou valores nulos.

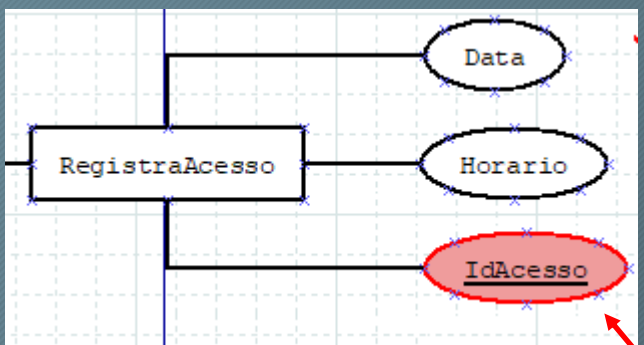
Entre os campos aptos para receberem a chave primária, deve-se optar pelo campo menor e formado por somente números inteiros.

USUARIO				
	Nome	CPF	Login	Senha
PODE REPETIR	SIM	NÃO	NÃO	SIM
PODE SER NULO	NÃO	NÃO	NÃO	NÃO
As colunas CPF e Login atendem a regra de Identidade e são campos aptos a receberem a chave primária				

Desenvolvimento: Passo a passo - Identificação das Chaves primárias



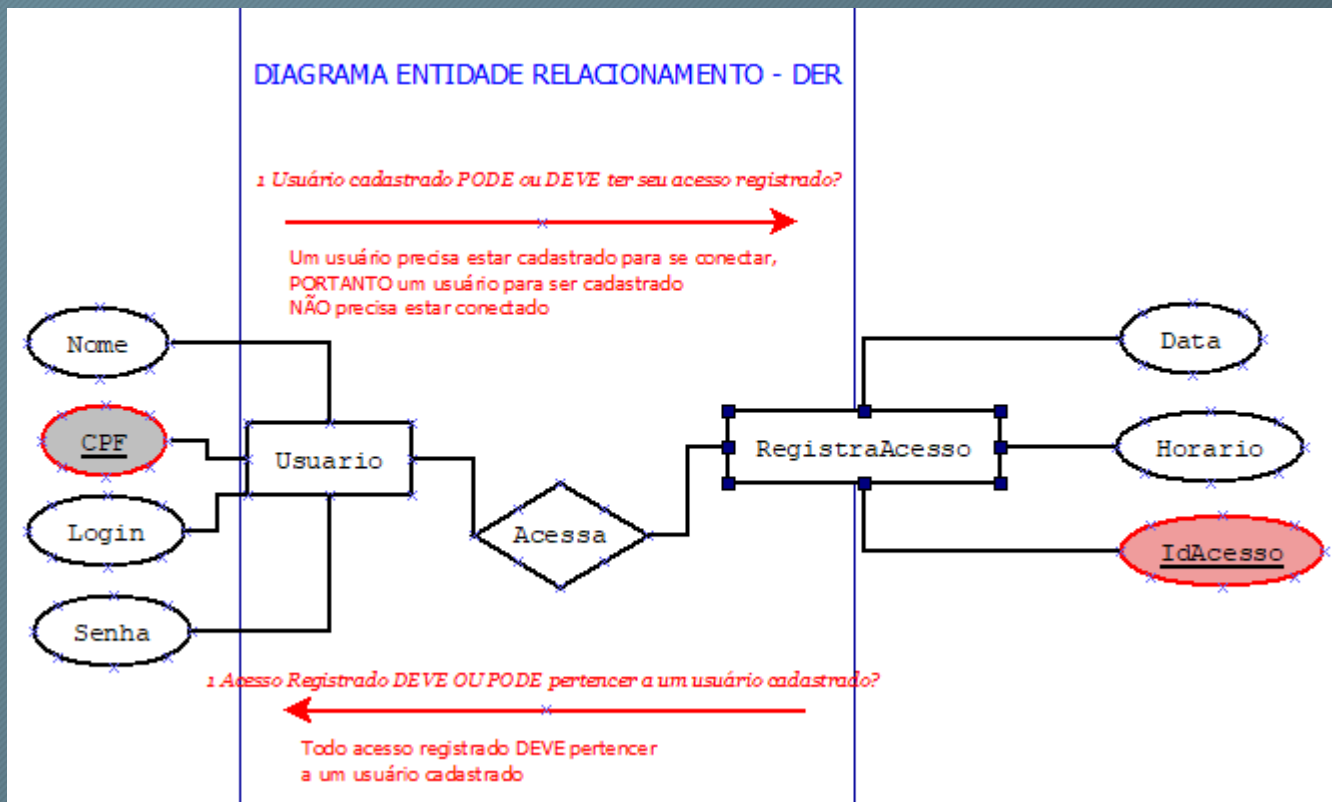
No caso da Entidade RegistraAcesso nenhum dos campos atendem a regra de Identidade, neste caso orienta-se criar Um atributo artificial para ser a chave primária.



Atributo Artificial, criado para ser chave primária

REGISTRAACESSO		
	DATA	HORARIO
PODE REPETIR	SIM	SIM
PODE SER NULO	NAO	NAO

Desenvolvimento: Passo a passo - Modelo Lógico - Condicionalidade para o Relacionamento

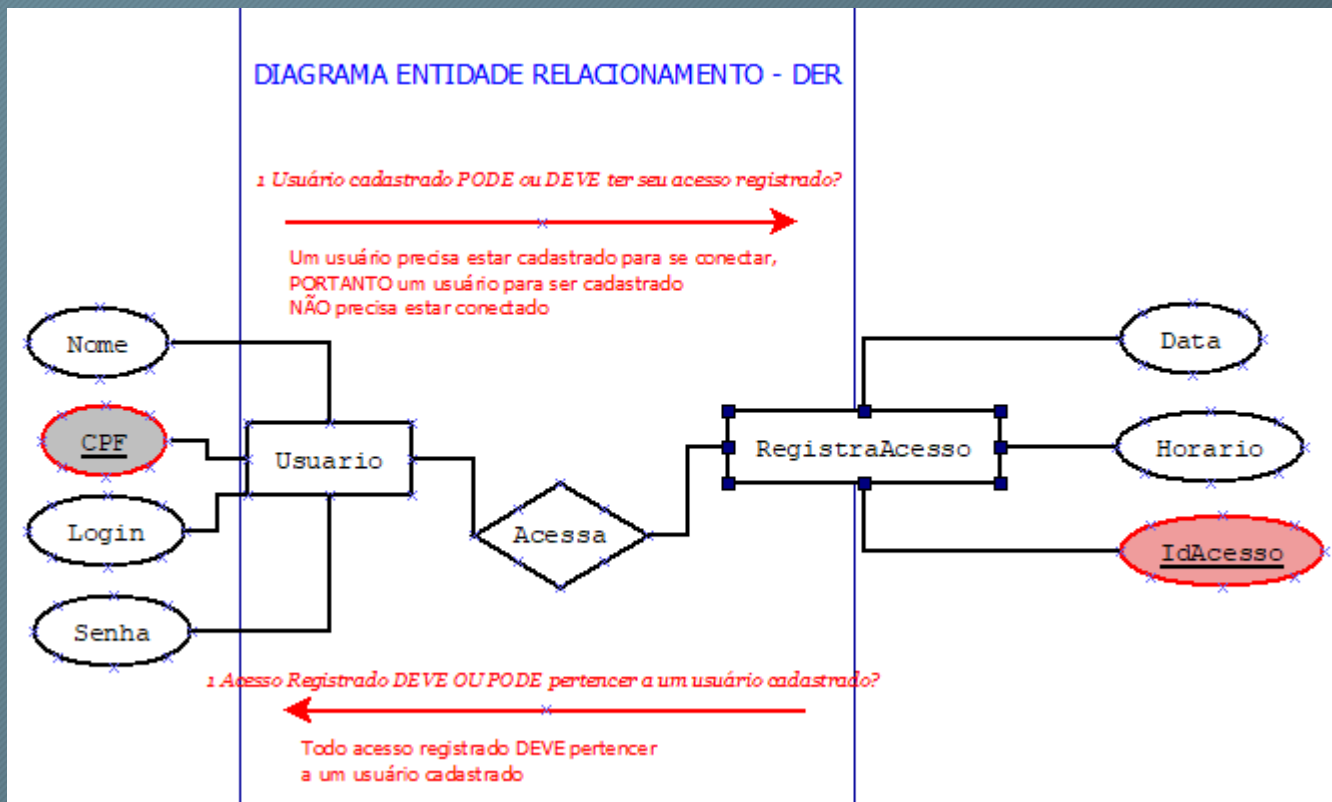


A CONDICIONALIDADE é importante para demonstrar ao desenvolvedor a prioridade de implementação das Entidades.

1 Usuário cadastrado PODE estar logado ou não, nesse caso a condição para o relacionamento é PARCIAL.

Um usuário precisa primeiro ser cadastrado no sistema, para depois se logar.

Desenvolvimento: Passo a passo - Modelo Lógico - Condicionalidade para o Relacionamento



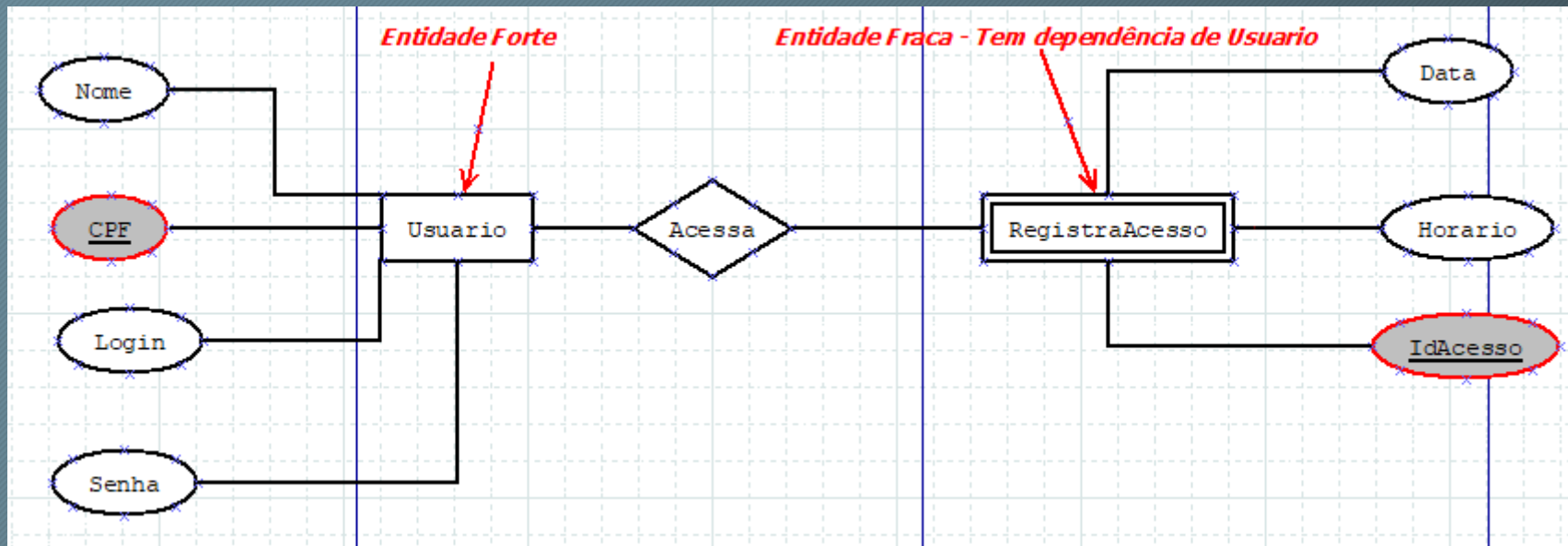
A condição É sempre verificada no DOIS sentidos.

Para registrar um acesso obrigatoriamente o usuário DEVE estar cadastrado no sistema.

Portanto, para que a entidade RegistraAcesso receba um valor, PRIMEIRO o dado deve existir na Entidade Usuário.

Denomina-se CONDIÇÃO TOTAL para o relacionamento. A entidade RegistraAcesso depende da entidade Usuário.

Desenvolvimento: Passo a passo - Modelo Lógico - Condicionalidade para o Relacionamento



A Entidade que possui dependência da outra é denominada de:
ENTIDADE FRACA ou DEPENDENTE

A Entidade que possui prioridade de preenchimento ou NÃO DEPENDÊNCIA é chamada de ENTIDADE FORTE.

Desenvolvimento:

Passo a passo - Modelo Lógico - Cardinalidade para o Relacionamento

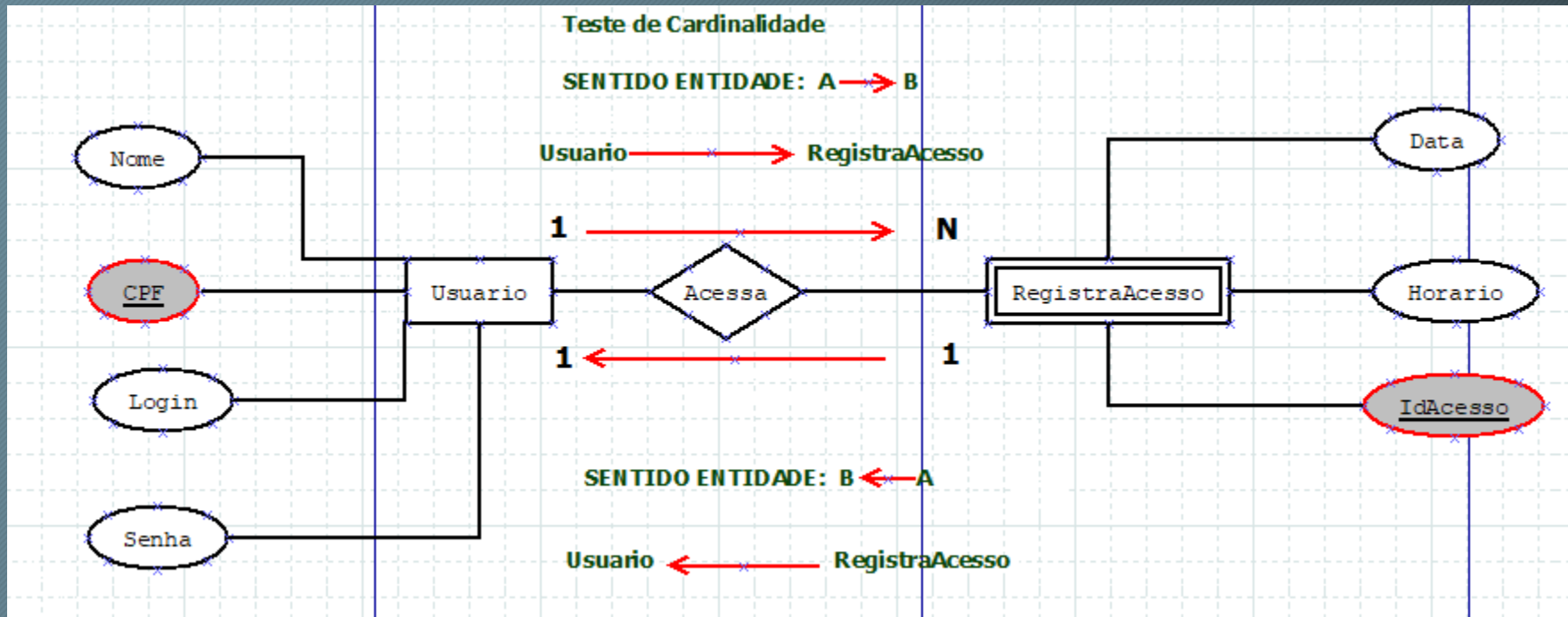
Como já visto anteriormente, a CARDINALIDADE, define o grau do Relacionamento.

Considerando sempre um registro, verificamos com quantos registros da outra entidade ele pode se relacionar.

Para calcular a cardinalidade DEVE-SE SEMPRE, analisar os relacionamento nos dois sentidos:

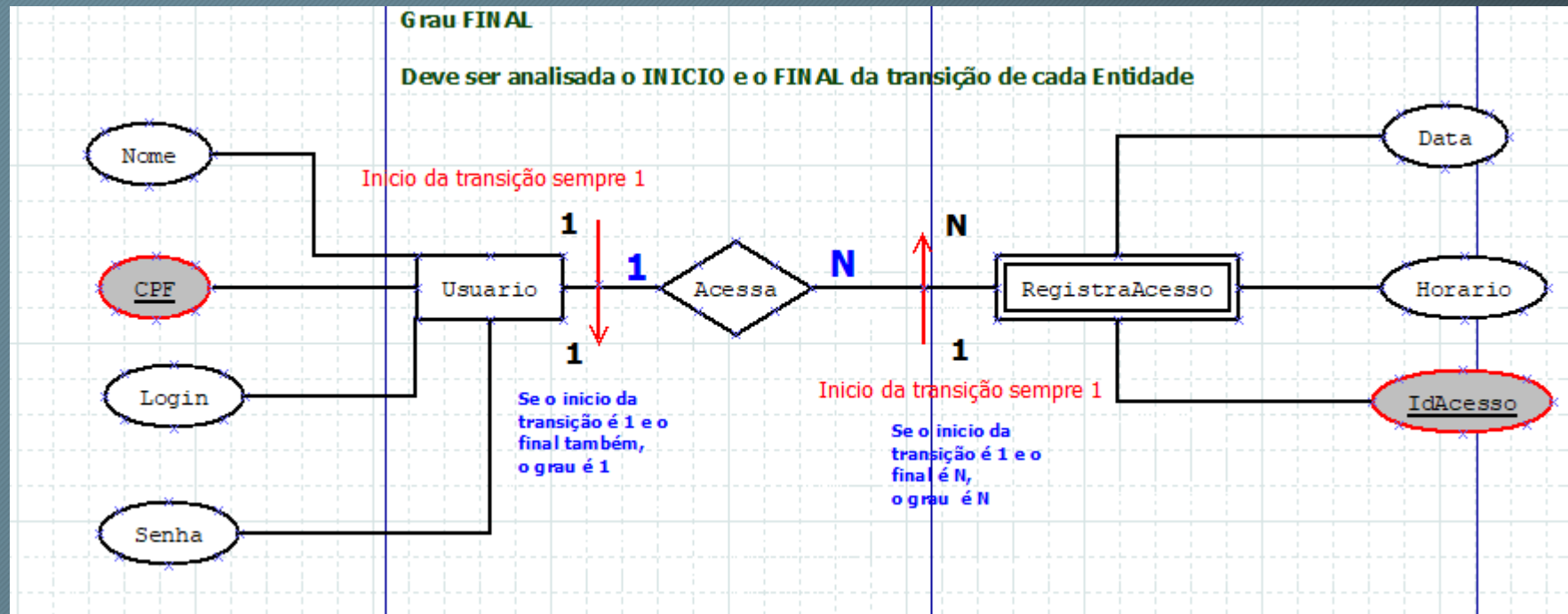


Desenvolvimento: Passo a passo - Modelo Lógico - Cardinalidade para o Relacionamento



LÊ - SE: Um **usuário** cadastrado PODE efetuar UM OU MAIS **acessos ao sistema** (A - B).
UM **registro de acesso** cadastrado DEVE pertencer a UM e SOMENTE UM **usuário** (B - A).

Desenvolvimento: Passo a passo - Modelo Lógico - Cardinalidade para o Relacionamento



- Para o lado A (Entidade USUARIO) o início e o final da transição é 1, então, a cardinalidade final é 1.
- O lado B (Entidade REGISTRAACESSO) a transição inicia também em 1, mas finaliza em N, sendo a cardinalidade final N.

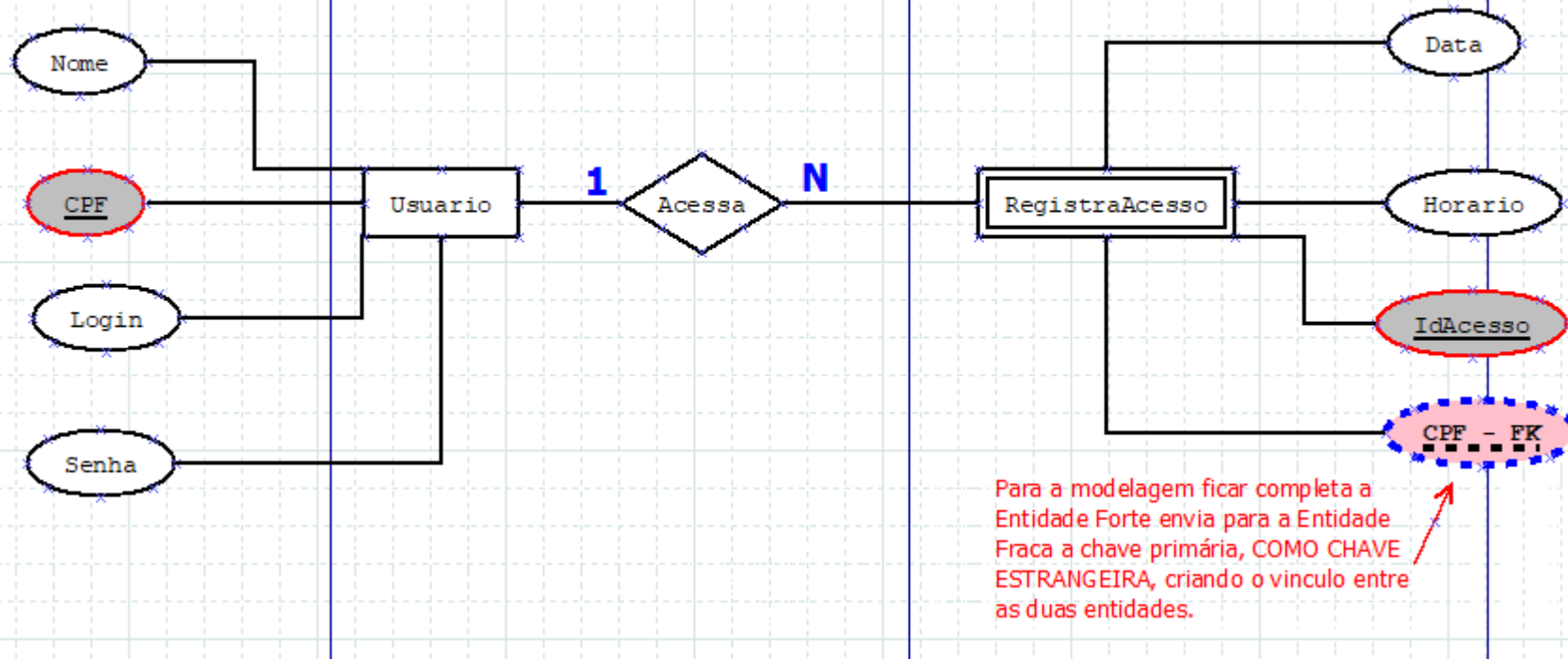
GRAU FINAL DO RELACIONAMENTO: 1:N

Usuario (Entidade Forte) 1

RegistraAcesso (Entidade Fraca) N

Desenvolvimento: Passo a passo - Modelo Lógico - Definir chave estrangeira

DIAGRAMA ENTIDADE RELACIONAMENTO - LOGIN



- Para finalizar o modelo Lógico, deve-se definir a chave estrangeira.
- Cria-se na Entidade Fraca (lado N) uma coluna com as mesmas propriedades da chave primária do lado 1.
- E a denomina de CHAVE ESTRANGEIRA (FOREIGN KEY - *FK), referenciando a Entidade que envia a Chave.

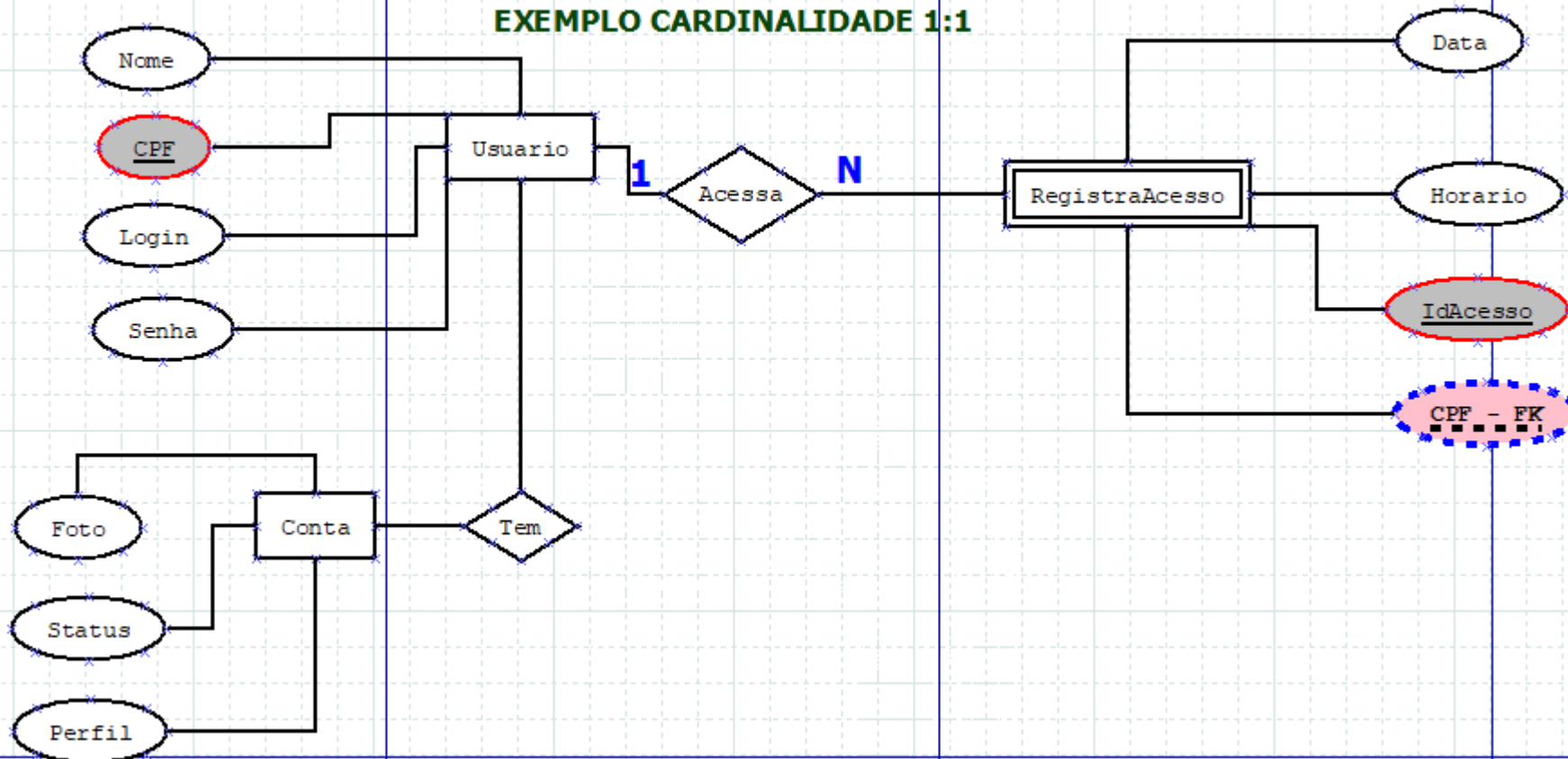
Passo a passo - Modelo Lógico

Exemplos:

Cardinalidade 1:1

DIAGRAMA ENTIDADE RELACIONAMENTO - LOGIN

EXEMPLO CARDINALIDADE 1:1

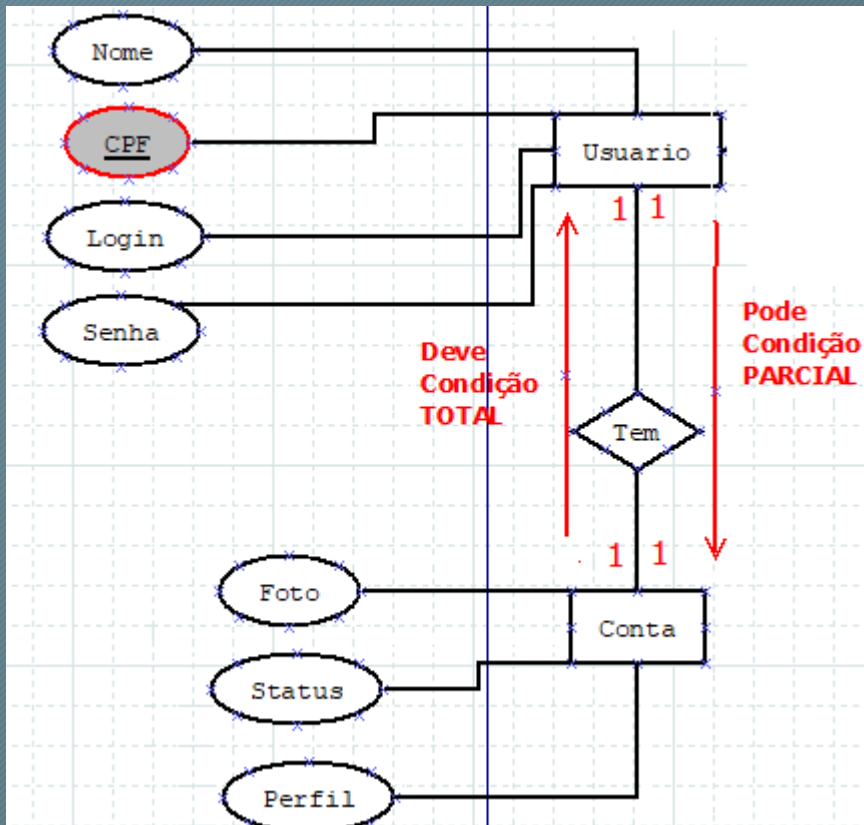


- Considerando ainda o modelo Login, criamos a entidade Conta
- Cada usuário cadastrado PODE TER UMA E SOMENTE UMA CONTA CADASTRADA NO SISTEMA .
- Uma conta existente DEVE PERTENCER A UM E SOMENTE UM usuário cadastrado.

Passo a passo - Modelo Lógico

Exemplos:

Cardinalidade 1:1

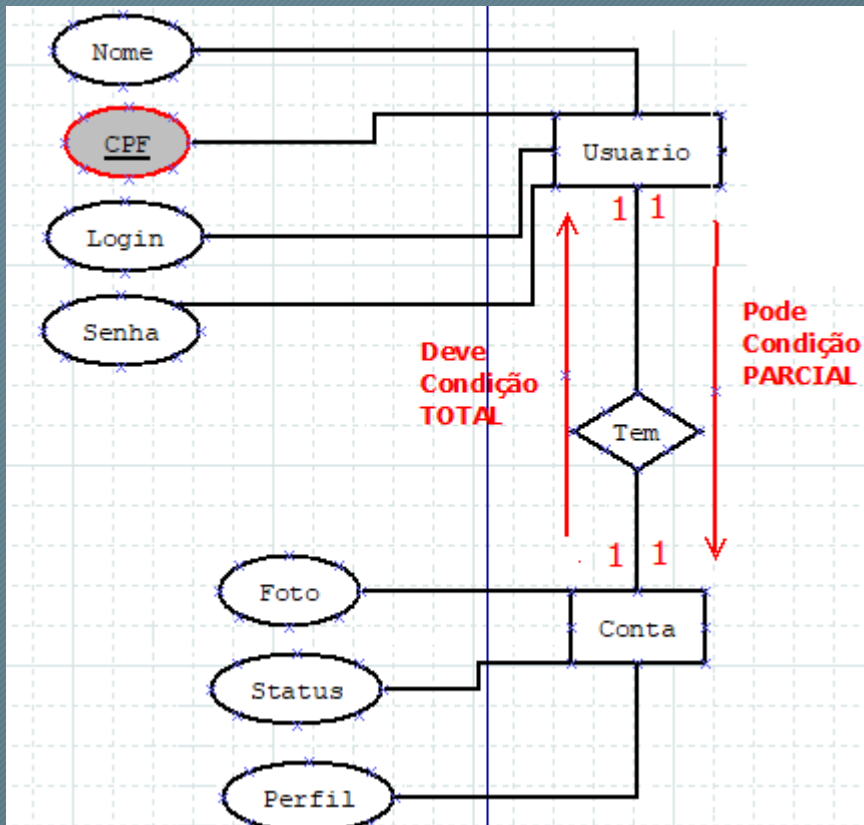


- Os casos de cardinalidade 1:1, se identificam a partir de casos de GENERALIZAÇÃO/ESPECIALIZAÇÃO.
- No nosso exemplo, os usuários são cadastrados e associados a uma conta, onde é definido status (ativo/inativo), perfil (visibilidade: Administrador, usuário ler /cadastrar e usuário que só pode ler).
- Primeiro DEVE-SE identificar a entidade Generalista, principal e a Entidade Especialista, criada a partir da Generalista.

Passo a passo - Modelo Lógico

Exemplos:

Cardinalidade 1:1

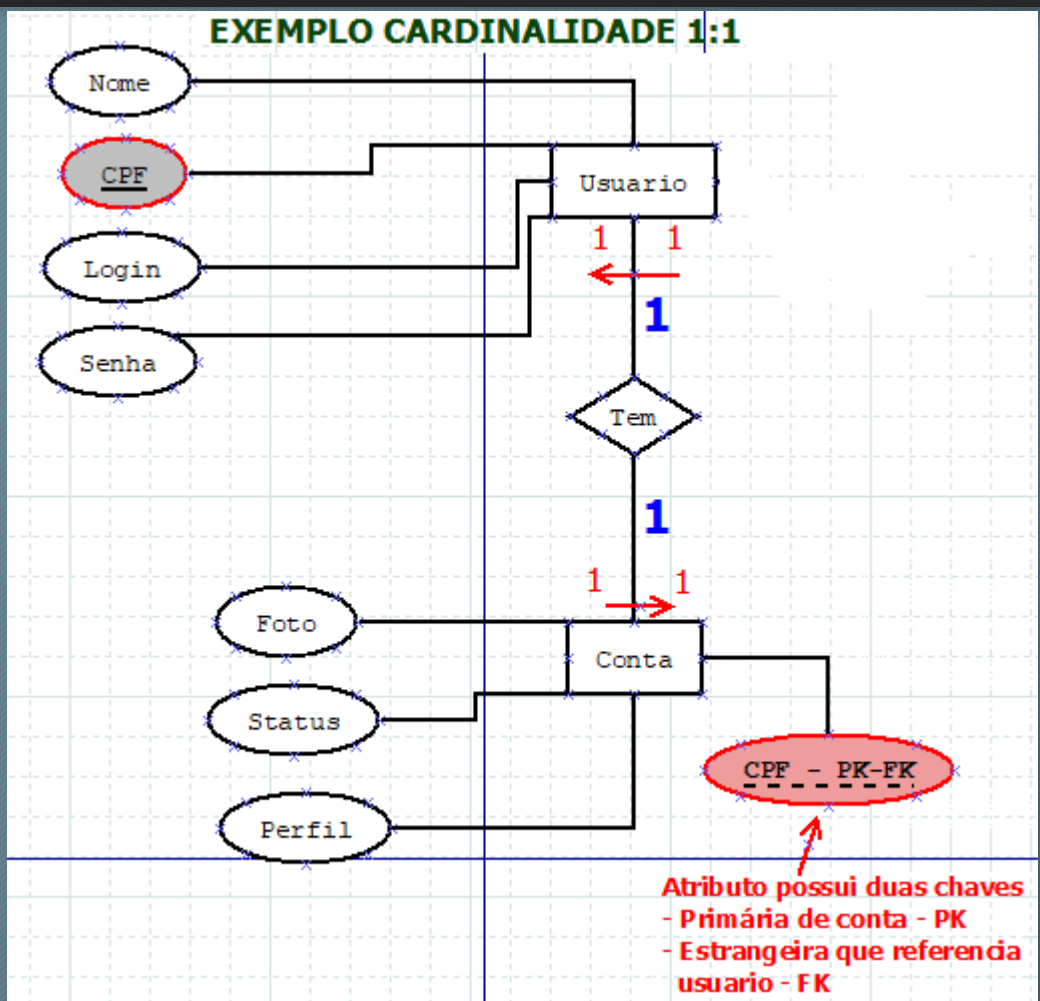


- A Entidade Generalista sempre é a que possui a Condição PARCIAL para existir, não tem dependência da entidade associada para receber dados.
 - Usuario é GENERALISTA, pois primeiro deve ser feito o cadastro.
- A Entidade Especialista tem sua origem a partir da existência da primeira, ou seja, condição TOTAL para existir.
 - Conta é ESPECIALISTA, pois a partir do cadastro do usuário é que se ASSOCIA uma conta para ele.

Passo a passo - Modelo Lógico

Exemplos:

Cardinalidade 1:1 - Chaves Primárias e Estrangeiras



- Para os casos de Generalização/Especialização, a transição inicial e final nos dois sentidos é 1.
- Gerando o grau final de relacionamento 1:1

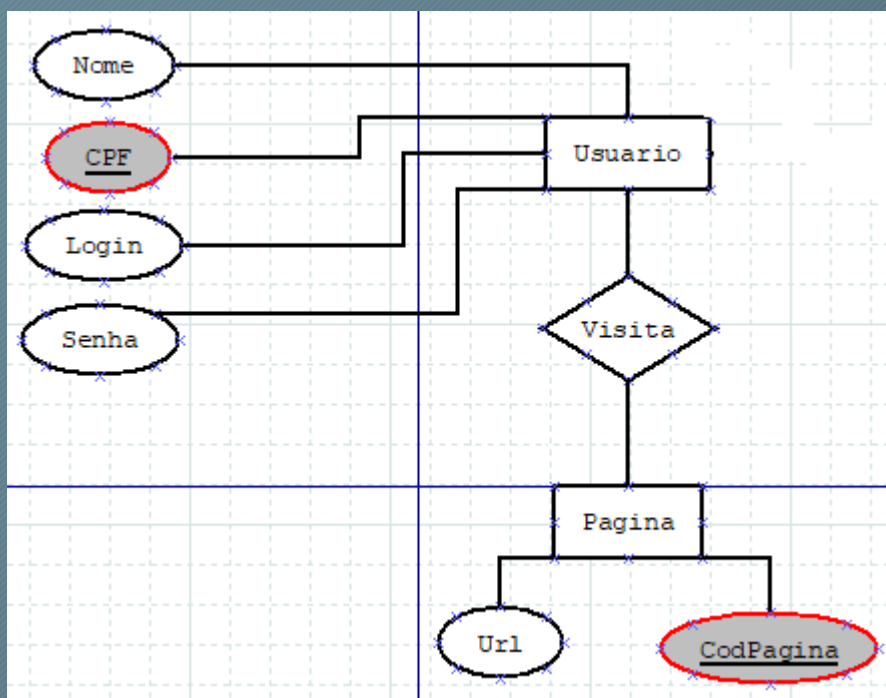
Sendo que:

- A Entidade Especialista deve receber um atributo com as mesmas propriedades da chave primária da Entidade Generalista.
- Esse atributo criado deve ser a chave primária da Entidade Especialista e receber a chave estrangeira referenciando a Entidade Generalista, ou seja, a coluna CPF na entidade conta possui duas chaves a primária e a estrangeira.

Passo a passo - Modelo Lógico

Exemplos:

Cardinalidade N : N - Chaves Primárias



- Analisando o modelo de Negócio Login, incluindo uma nova Entidade para rastrear o acesso do usuário, pode-se, por meio da Entidade Pagina, verificar as url's que o usuário logado visita.

Sendo que:

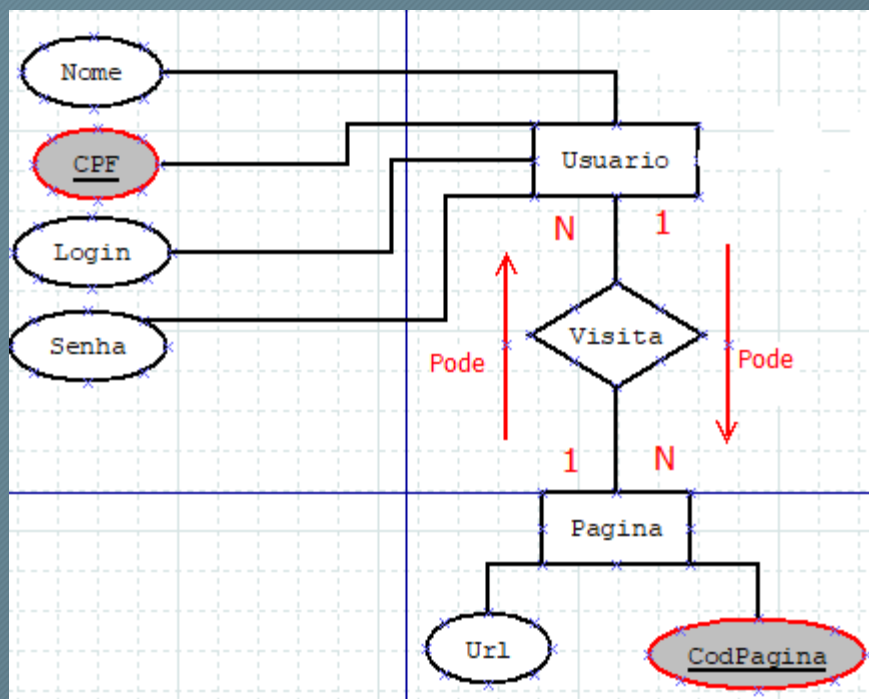
- 1º - a Entidade Página deve ter uma chave primária, URL não pode repetir e não pode ser nulo, mas é muito longo para ser uma chave.

O atributo CodPagina, foi criado artificialmente para ser chave primária.

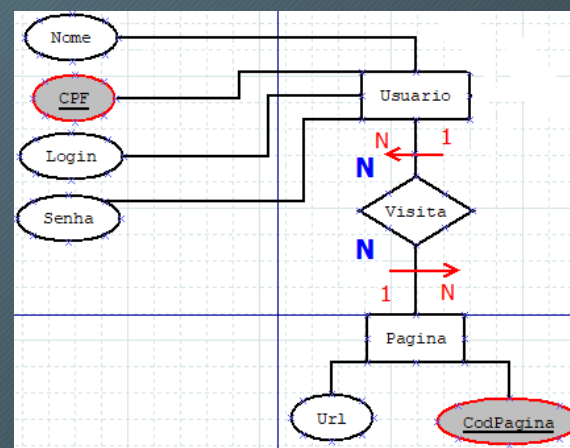
Passo a passo - Modelo Lógico

Exemplos:

Cardinalidade N : N - Relacionamento Parcial



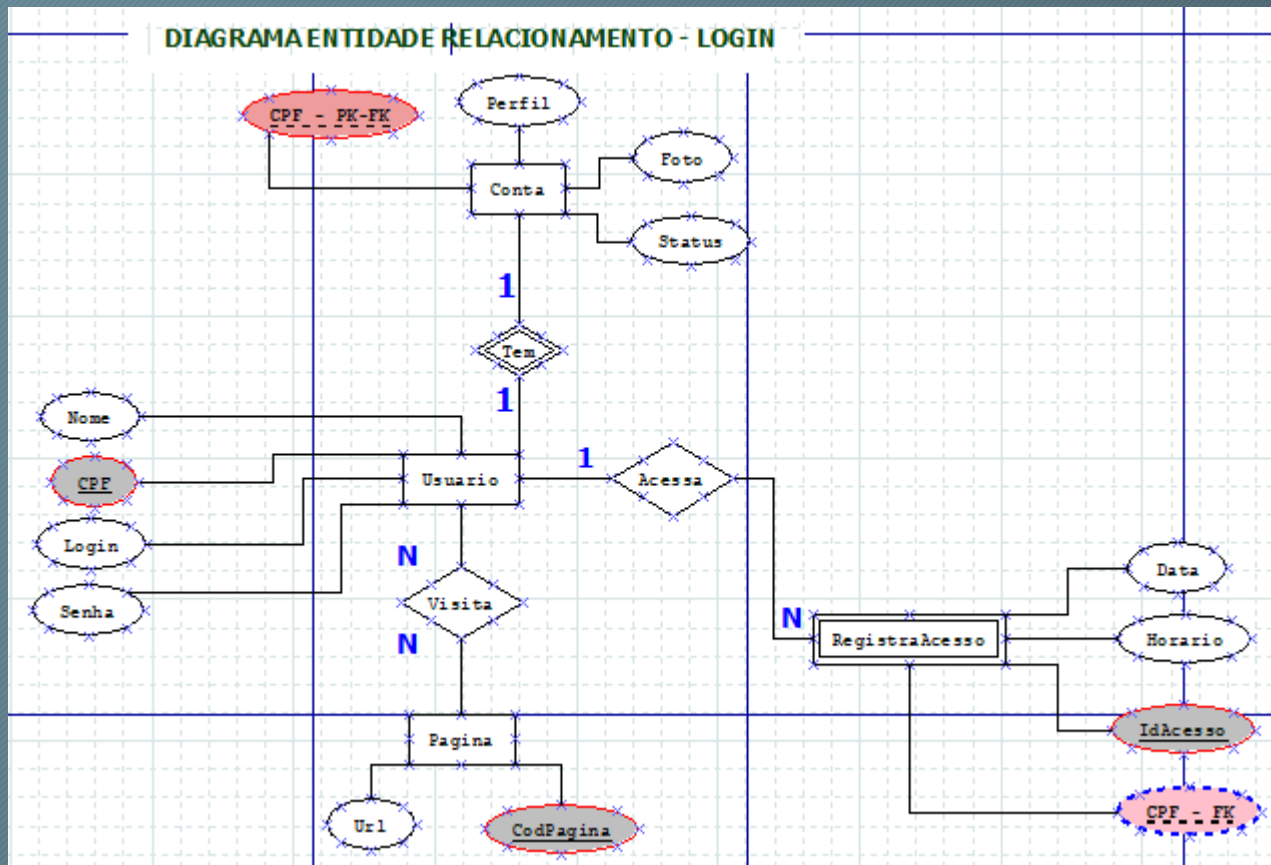
- Relacionamento:
- Um usuário pode acessar UMA OU MAIS PAGINAS dos sistema
- Uma pagina pode ser acessada por UM OU MAIS USUARIOS.



Se o início da transição for 1 e o final N, nos dois sentidos a cardinalidade final do relacionamento será do tipo N:N.

Modelo de Negócio LOGIN

Diagrama Entidade Relacionamento DER - FINAL



Com base no passo a passo do modelo Lógico, o DER final apresenta o resultado da figura ao lado.

Neste material foi discutido o passo a passo para montar um Diagrama Entidade Relacionamento, desde a elucidação da realidade nebulosa, a definição das chaves.

Na próxima aula discutiremos a criação das chaves estrangeiras e dicionário de dados para os atributos.