



Escola Politécnica



PMR - Depto. de Enge. Mecatrônica

PMR 2490 - Laboratório

Prof. José Reinaldo Silva

PMR 3304 (2o. Semestre de 2016)

Teoria

Prof. Marcos Pereira Barretto (coordenador)

Laboratório PMR-2490 - T03

Prof. José Reinaldo Silva (T03, T04) & Prof. Marcos Sales G. Tsuzuki (T01, T02) A={T01, T03}, B={T02, T04}

Aula 1 (17/08): Introdução, Sistemas de Informação, **project statement**, Modelagem de Dados, Modelo E-R, **Para casa: proposição de um projeto (project statement).**

Aula 2 (31/08): DBML (1) Em aula: apresentação e discussão sobre eliciação e análise de requisitos, que devem ser modelados em UML. Tutorial - Bancos de Dados (DDL). **Para casa: preparar o documento de descrição dos requisitos. Usar o Astah. Caracterizar a lista de casos de uso do projeto, tendo pelo menos 4 casos de uso simples (ou 3 complexos) para cada aluno do grupo. O Gerente do Projeto pode ficar com 4 casos de uso simples.**

Aula 3(21/09) : DBML (2) Em aula: Apresentar e discutir a descrição dos casos de uso. Para casa: rever documento de casos de uso. Fazer modelo de dados do sistema.

Aula 4 (05/10): Design de Interfaces

Tutorial: HTML básico. Em aula: Apresentar e discutir o Modelo de Dados. **Para casa: Rever o modelo de dados. Preparação das interfaces em HTML.**

Aula 5 (26/10) : Tutorial : JSP (fazer conexão entre interface, processo e BD). Em aula: apresentação e discussão das interfaces. Para casa: Implementação das camadas de negócios e de dados.

Aula 6 (16/11): DevOps: desenvolvendo com GIT e Amazon. Em aula: Duvidas de codificação. Para casa: Implementação das camadas de negócios e dados.

Aula 7 (30/11): Em aula: Duvidas de codificação. Para casa: Implementação das camadas de negócios e dados.

Aula 8 (14/12?): Apresentação final do projeto para a banca. Entrega de documentação completa

PMR 3304 (2o. Semestre de 2016)

Esquema do curso é baseado em :

- Aulas teóricas
- Aulas práticas, tutoriais, exercícios de projeto;
- Projeto da turma (subdividido em subprojetos individuais)

A avaliação será baseada em :

Teoria  Provas, listas de exercícios, etc.

Laboratório  Exercícios e milestones, projeto final

PMR 3304 - Infra-estrutura

Para todas as turmas:

Sistema de acompanhamento online baseado no sistema Moodle-STOA da USP. Este sistema terá todo o conteúdo do curso, livros, notas de aula, transparências, e deve armazenar todos os exercícios e milestones, chats, foruns, etc.

<http://www>

Laboratório:

Sala com 25 computadores em rede que devem ser usados individualmente. Exercícios e tutoriais poderão ser enviados para casa via rede.

Cada aluno deve reproduzir o ambiente de trabalho (em casa, nos laboratórios de IC, no CCE, na sala pro-aluno, etc.) para ter uma alternativa para continuar o seu projeto e para realizar os exercícios.



O Uso dos computadores

Cada turma tem um acesso diferente nas máquinas do LAB:

Console console epusp	VM clicar em ReqVMw PMR2490	User: aluno pmr2490	MySQLuser aluno	Passwd pmr2490
------------------------------------	---	-------------------------------	---------------------------	--------------------------



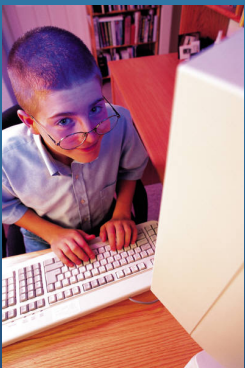
O que tem em cada máquina virtual?

- Astah Community (edição de documentação UML)
- MySQL 5.6 com Workbench 6.3
- Netbeans IDE 8.0.2

ATENÇÃO

Cada aluno é responsável pelos arquivos que deixar nos computadores.

- O staff não se responsabiliza por backups ou por ter os mesmos arquivos sempre na mesma máquina: portanto **SALVEM** seus arquivos ou mandem estes por e-mail para vocês mesmos, usem dropbox, google drive, etc.
- O mesmo se aplica à documentação de projeto feito pela turma.



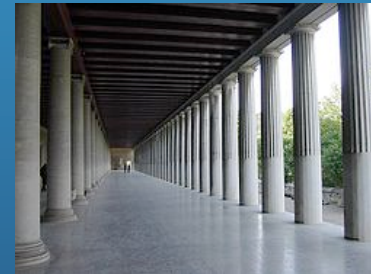
O que é o STOA-USP?

Stoa - USP

As Stoa eram espaços públicos onde as pessoas expunham quadros, obras de artesanato, vendiam produtos, víveres, declamavam, etc. Eram comuns na Grecia em vários períodos, incluindo o Helenístico.

O projeto Stoa-USP é uma stoa virtual onde a comunidade USP pode debater, postar opiniões, criar blogs, compartilhar notícias, grupos de discussão e também disciplinas, estas através do sistema Moodle de educação a distância. Vários cursos de graduação e pós-graduação se encontram no Stoa-USP incluindo PMR 5237.

Neste caso o sistema está acoplado diretamente com o sistema Janus e deve usar como e-mail o endereço oficial no sistema USP. Os alunos ouvintes devem se registrar no sistema STOA e serão incluídos depois no sistema.



Stoa de Atallos, Grecia, 159-138 AC, reformado recentemente

STOA-USP

Disciplinas

apoio às disciplinas da USP

USP

Usuário ou Número USP

Senha

Acessar

Cadastre-se

Esqueceu a senha?

Contato

Sobre

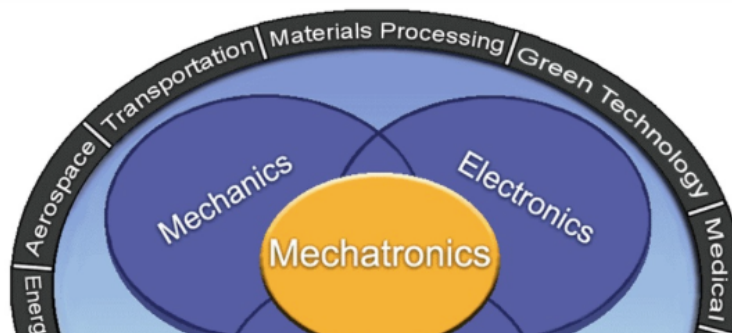
Suporte

[Início](#) > [Ambientes](#) > [EP](#) > [PMR](#) > [PMR2490-2016203](#)

Laboratório de Sistemas de Informação



Laboratory of Information Systems

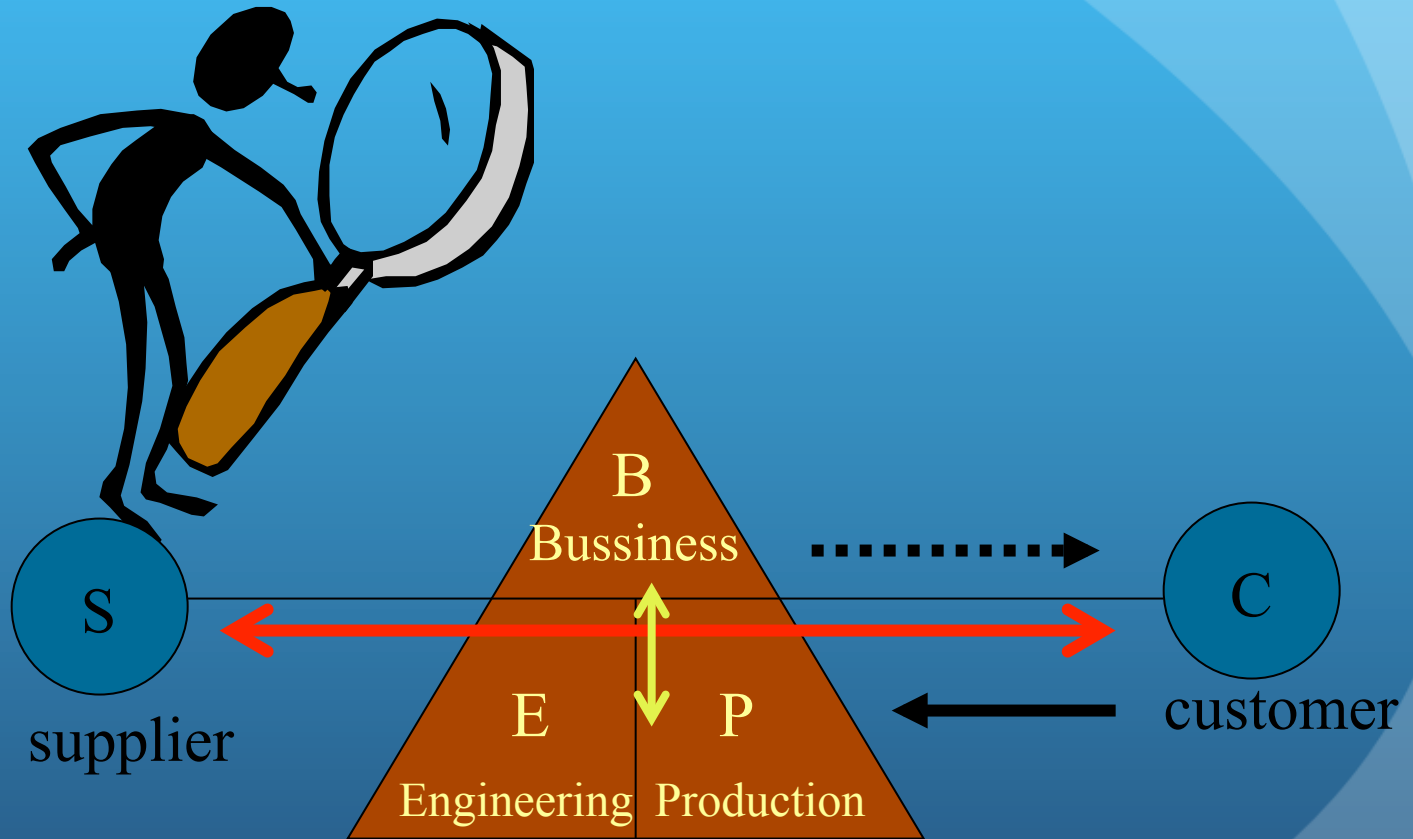


Aula 1

Sistemas de Informação: Entidades,
objetos, dados, informação

Modelagem conceitual

Aula 1- Introdução



Sistemas de Informação (SI)

Os sistemas de informação são uma maneira digitalizada de implementar “multiple views” ou “viewpoints” para sistemas de automação ou para sistemas de negócio em geral (automatizados ou não).



SI's e Automação



Portanto um sistema de informação é um artefato digital e interativo que permite que se automatize um dado sistema (ou sub-sistema) disponibilize interfaces diferentes, com níveis diferentes de informação, para grupos de usuários, gerentes, supervisores, mantenedores, desenvolvedores, etc. encapsulando as funcionalidades e protegendo as informações do sistema original.



Os componentes do SI

Informação é uma coleção de fatos organizados de modo a assumir um valor adicional que transcende o valor individual destes fatos.

Dados são fatos elementares, como o número USP, o CPF, o número horas trabalhadas, o tempo de um processo, o “part number” de um dado componente ou recurso, etc.

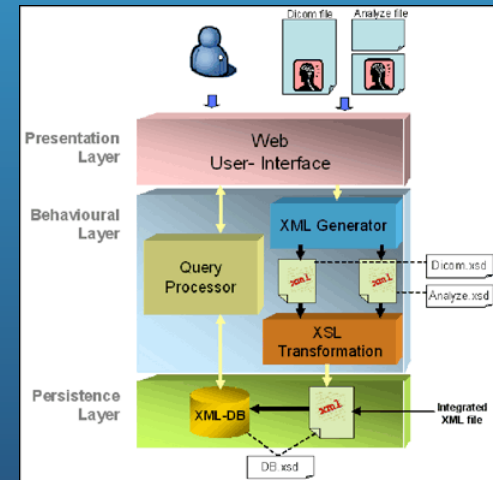
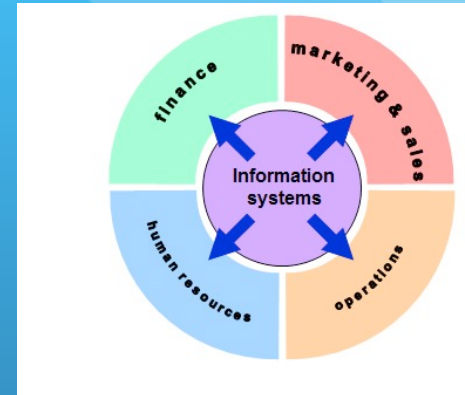
Arquitetura de um SI

Os constituintes elementares de um SI são:

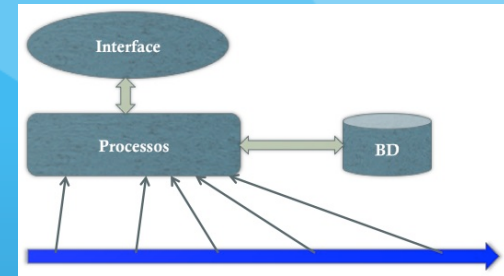
Uma interface (inteligente)

Um conjunto de processos

Um banco de dados (BD)



Dados e entidades



Portanto os **dados** são os elementos fundamentais sobre os quais gostaríamos de montar uma estrutura de elementos de informação mais complexos que satisfaçam e representem plenamente os componentes do sistema.

Mas, mesmo na arquitetura simples da transparência anterior podemos ver que queremos enxergar estes dados **como fatos associados a elementos invariantes da arquitetura**, isto é, elementos com vida própria (ou com um ciclo de vida estável) e perfeitamente identificáveis em qualquer estado do sistema.

Modelo Entidade-Relação (ER)

O Modelo ER surgiu como uma disciplina e uma representação da estrutura de dados que permite ter uma visão global estruturada dos dados, ao mesmo tempo que nos restringimos aos fatos mais elementares sobre os quais a informação será construída.

Historicamente o modelo ER foi criado por Peter Chen e apareceu pela primeira vez em 1976 no artigo “Entity-Relationship Model: Towards a Unified View of Data”.

As entidades

Uma entidade é uma representação esquemática e abstrata de um agente do sistema de informação.

É importante notar que este agente existe seja no nível mais baixo, isto é, de dados, participa dos processos, e se manifesta (recebendo ou fornecendo informação) na interface. Neste sentido dizemos que este agente é invariante, existe em qualquer nível, ainda que possa aparecer com uma “forma” diferente, e salvo em casos especiais, não deve ser removido do BD.

Grupo-entidade

Existe ainda o grupo-entidade, isto é, um agente genérico (arquetipo) que denota um grupo de entidades elementares com as mesmas características.

Exemplo: suponha que queremos construir o sistema de informação da USP. Quais seriam as entidades e os grupos-entidade?

Os atributos

Os atributos são propriedades estruturais das entidades (que as caracterizam) e que podem ser associadas a um valor fixo ou a um conjunto de valores.

Uma das entidades do SI da USP seria o grupo-entidade aluno. Quais seriam os seus atributos?

Classificação dos atributos

- **Atributo Monovalorado** : assume um único valor para cada elemento do conjunto- entidade. Ex. : Sobrenome, no_de_créditos, etc.
- **Atributo Composto** : formado por um ou mais atributos monovalorados. Ex. : Endereço (Rua, numero, Bairro, CEP, Cidade, Estado)
- **Atributo Multivalorado** : uma única entidade tem diversos valores para este atributo (seu “nome- é sempre representado no plural)Ex. : Dependentes, Pre-requisitos, etc.
- **Atributo Determinante** : identifica cada entidade de um conjunto- entidade (também conhecido com atributo chave). Ex. : número_USP, código_da_disciplina, etc.

Domínios

Define-se como o domínio de um Atributo ao conjunto de valores que este atributo pode assumir. Ex. : $\text{tipo_aluno} := \{\text{graduação, pós-graduação, extensão}\}$, ou $\text{idade} := \{0 \dots 150\}$.

Portanto uma entidade elementar, isto é, caracterizada por um conjunto de atributos monovalorados, tem o seu grupo-entidade plenamente definido pelo produto cartesiano dos respectivos domínios.

O Diagrama ER

O Diagrama ER é uma representação esquemática para as entidades e relacionamentos composta pelos seguintes elementos::

Retângulos : representam conjuntos- entidade



Elipses : representam atributos



Losangos : representam relacionamentos

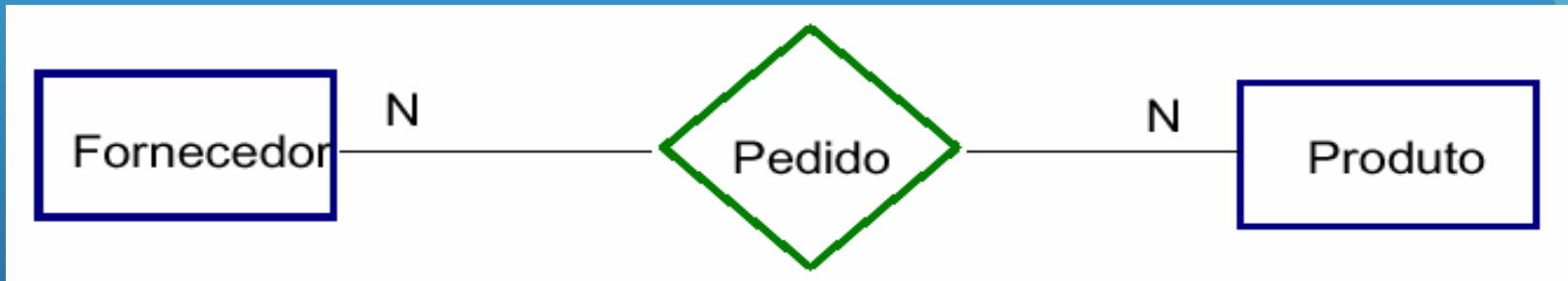


Linhas : ligam atributos a conjuntos; entidade e conjuntos; entidade a conjuntos ou relacionamentos.



Exemplo

Vamos tomar como exemplo uma firma que tem como um dos seus processos de negócio a aquisição de produtos ou insumos. O diagrama ER abaixo representa a relação entre fornecedor e produto que constam no BD como um pedido.



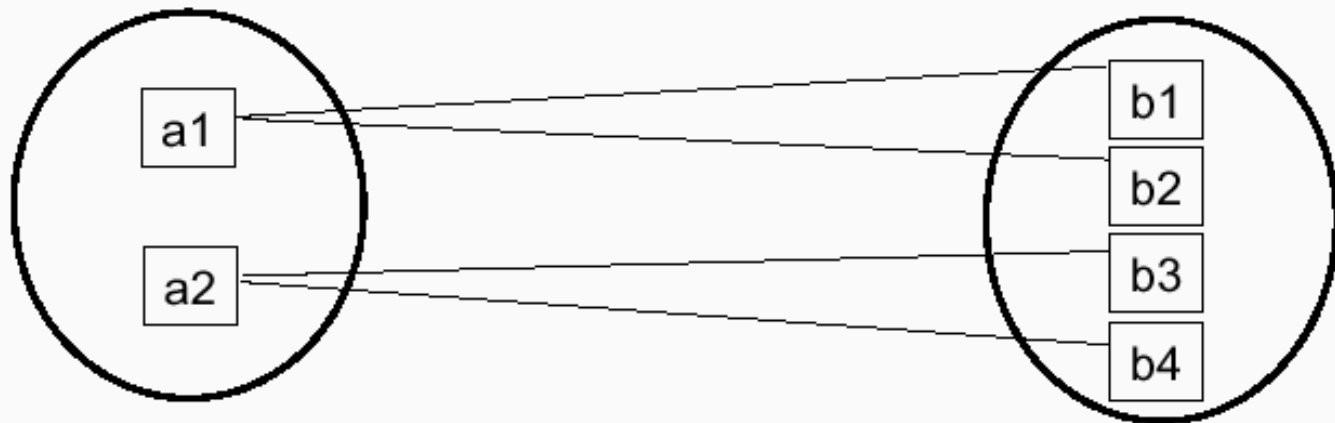
Propriedades dos relacionamentos

(CARDINALIDADE).

- a) **Um- para- um** (1 x 1): uma entidade em A está associada no máximo a uma entidade em B e uma entidade em B está associada no máximo a uma entidade em A.
- b) **Um- para- muitos** (1 x N): uma entidade em A está associada a qualquer número de entidades em B, enquanto uma entidade em B está associada no máximo a uma entidade em A.
- c) **Muitos- para- muitos** (M x N): Uma entidade em A está associada a qualquer número de entidades em B, e uma entidade em B está associada a qualquer número de entidades em A.

Exemplos

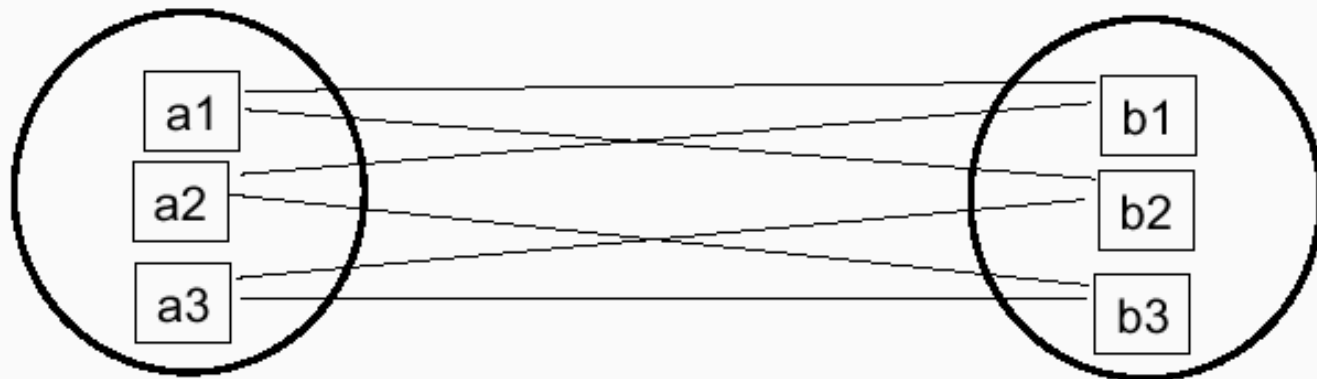




Conjunto-Entidade A

Conjunto-Entidade B



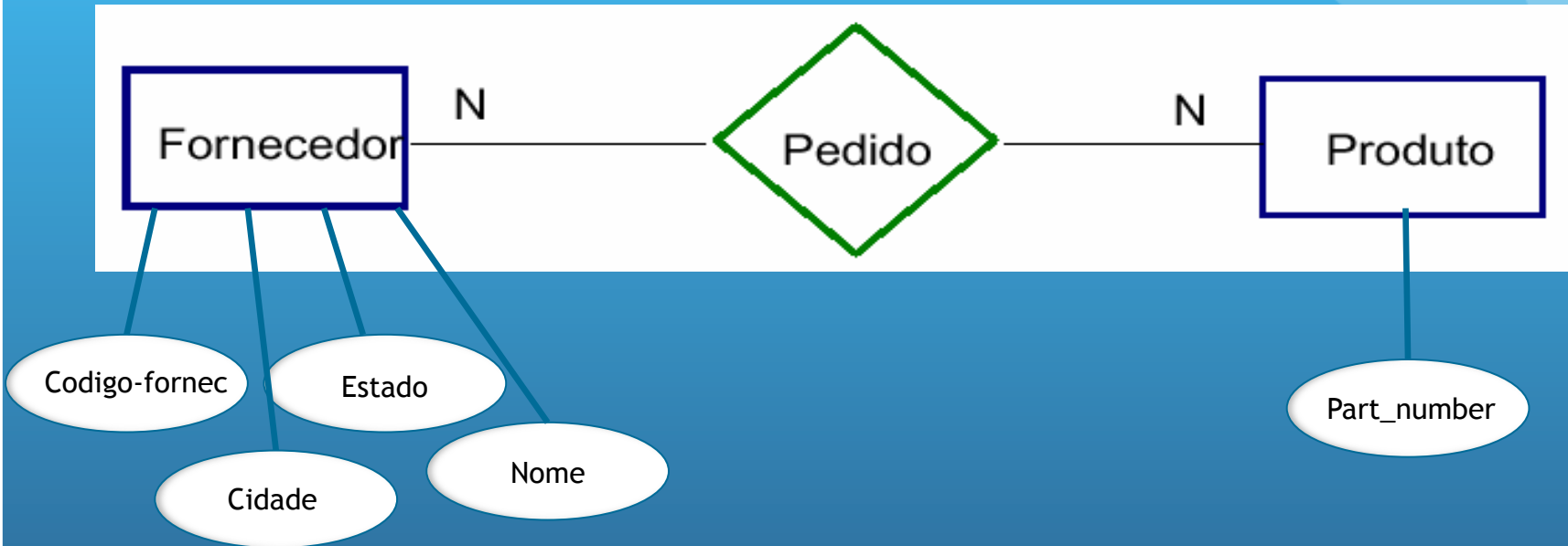


Conjunto-Entidade A

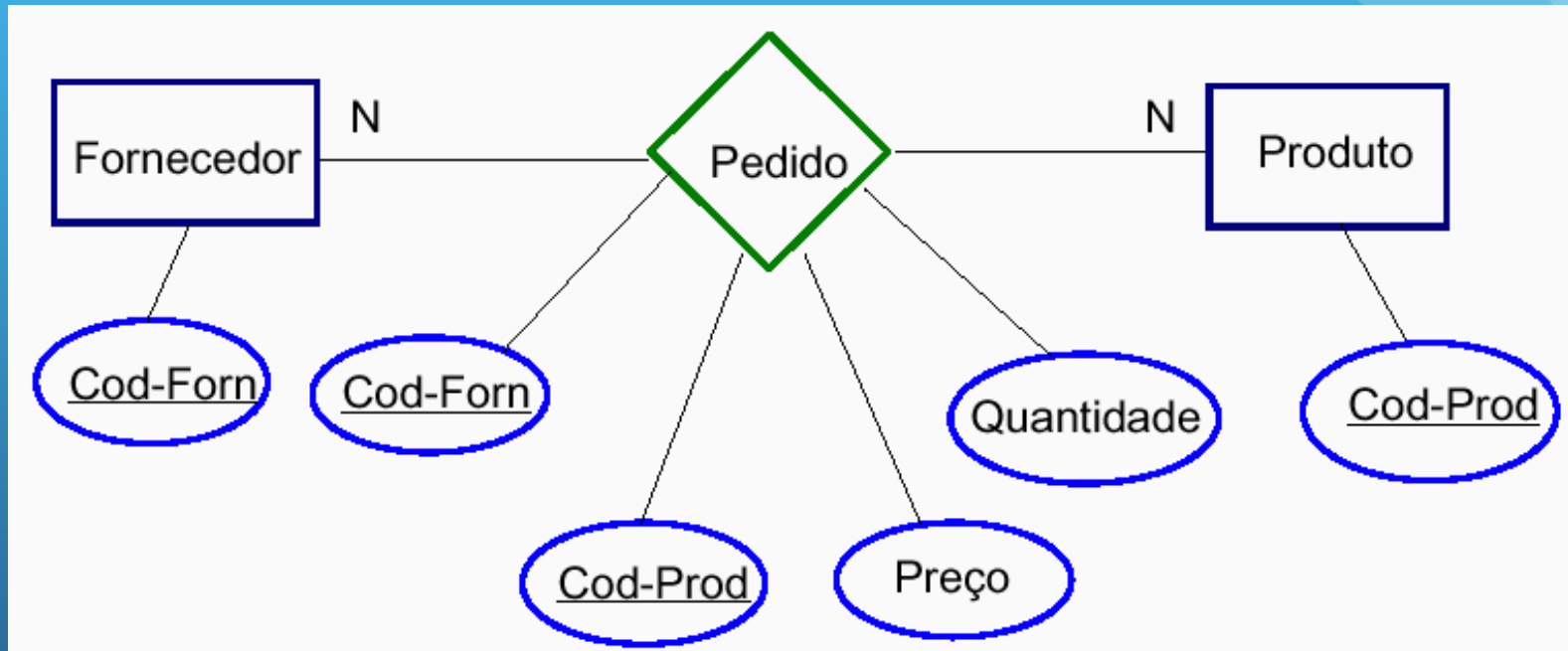
Conjunto-Entidade B



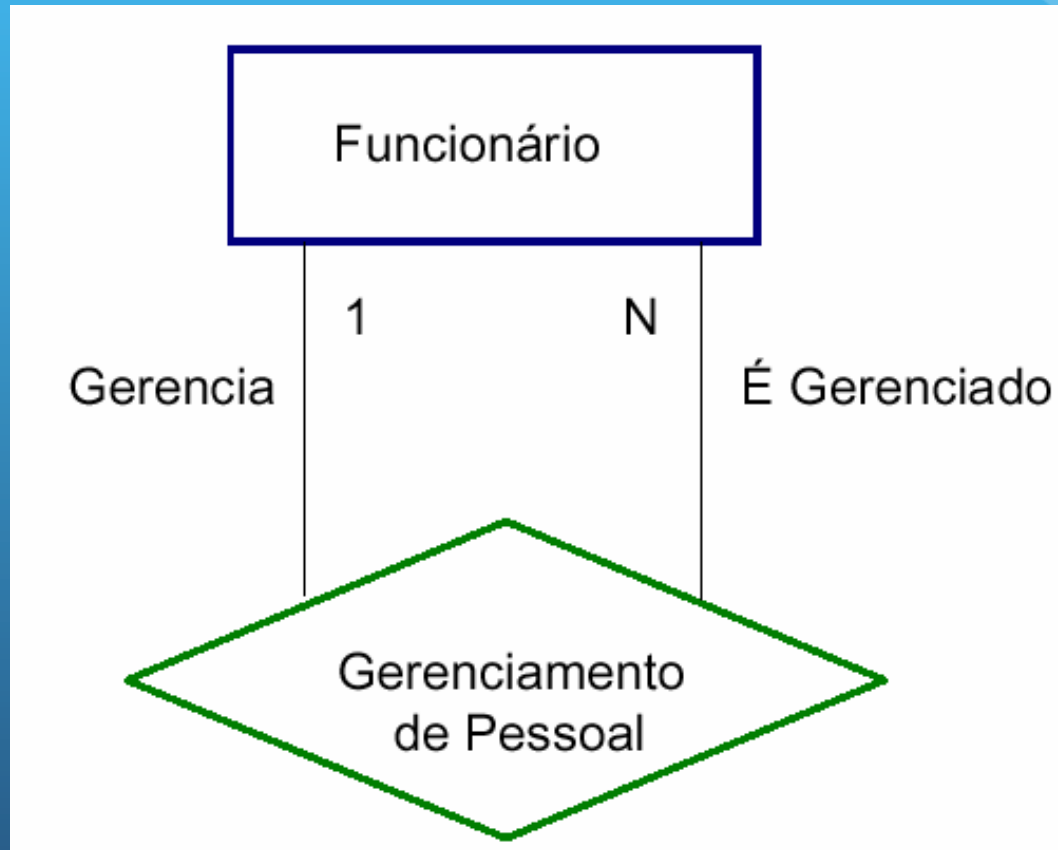
Inserindo atributos



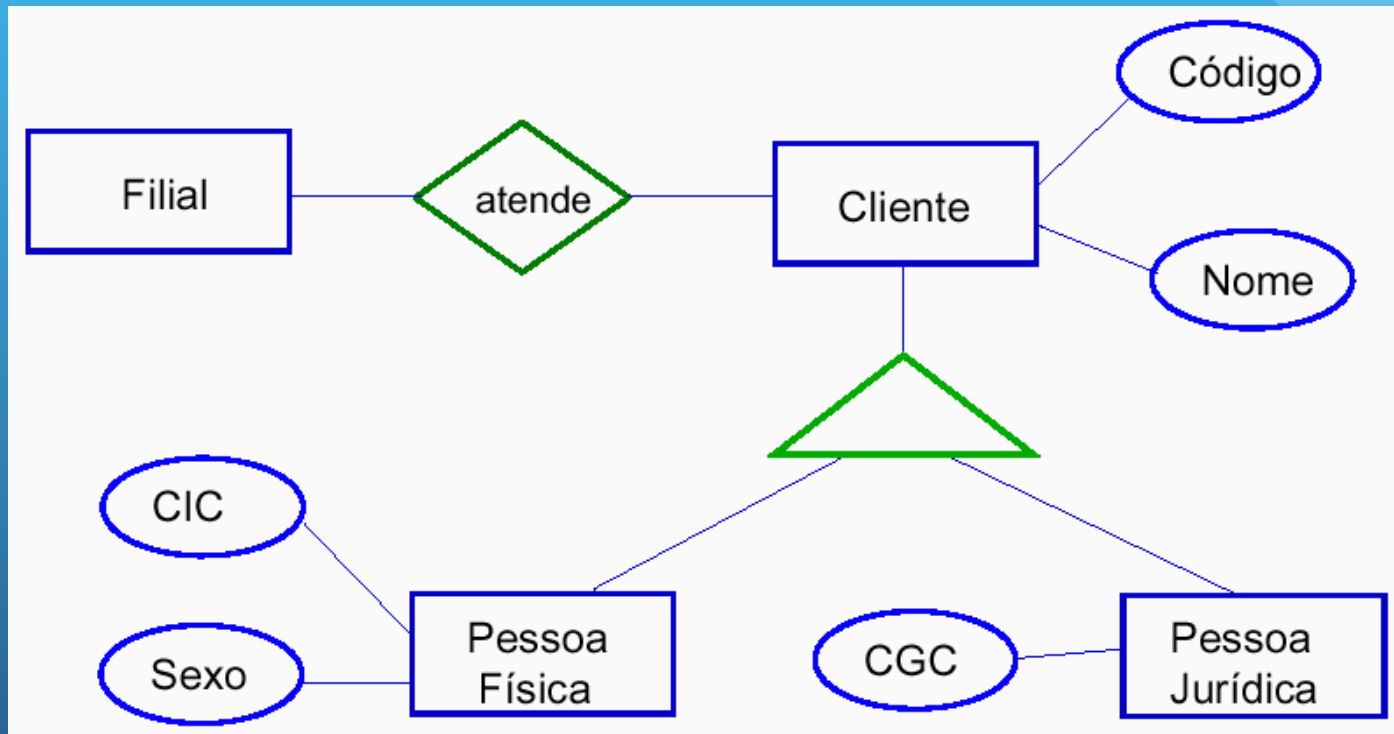
Atributos de relacionamentos



Auto-relacionamento



Uso do grupo-entidade



Entidade fraca

DEPENDÊNCIA EXISTENCIAL E ENTIDADES FRACAS (MER EXTENDIDO)

Dependência existencial ocorre quando a existência de uma determinada entidade está condicionada à existência de uma outra entidade a ela relacionada.



Uma **entidade fraca** não possui sequer identidade própria, sendo sua chave primária composta pela chave estrangeira proveniente da entidade dona concatenada a um identificador de si própria (que pode repetir para diferentes instâncias da entidade dona).



Projeto Final - para a próxima aula

A classe toda será transformada em duas “equipes de projeto”. A estrutura da equipe de projeto será composta por um “project manager” nas sub-equipes: a de documentação (3 participantes), a equipe de teste (2 participantes). Os demais serão “programadores”.

A equipe toda deve achar uma proposta de sistema de informação e trazer esta proposta na próxima aula de modo que tenha pelo menos QUATRO “use-cases” simples (complexos) para cada aluno. O “manager” pode ter TRÊS use-cases simples.

Na aula que vem cada equipe deve apresentar um documento com a descrição do projeto da turma, o project statement, e a composição da equipe como os nomes de todo e a função(ões). O mesmo documento deve ter a atribuição de cada um (manager, documentação, teste, prog.).

Exercícios: (turmas 3 e 4)

Crie o MER de um sistema de BD para ser utilizado pelos departamentos da universidade. Um departamento é responsável por no mínimo uma disciplina. Uma disciplina por sua vez pode ter de zero à várias turmas que estão sendo oferecidas. Cada turma por sua vez possui no mínimo um aluno inscrito, e não pode existir uma turma que não esteja relacionada à uma disciplina. Um aluno cadastrado no sistema pode estar ou não matriculado em alguma disciplina, e pode também estar matriculado em várias disciplinas.

Um departamento deve ter em seu registro a sigla, o nome e o nome do chefe. Cada disciplina deve ter um nome, um código, a descrição da ementa, e a bibliografia. Cada Turma deve ter um registro do seu turno (manhã, tarde ou noite) e professor que ministra. Cada aluno deve ter em seus registros nome, RG, CPF, N de carteirinha, endereço e telefone

Documentação

Cada turma deve ter um único projeto. Portanto deve definir este projeto através de um documento contendo: título do projeto, escopo do projeto, identificação de beneficiários (comerciais ou institucionais, chamados *stakeholders*, e de usuários finais, deixando claro qual é o benefício em cada caso), e por *um parágrafo que identifica a funcionalidade global do projeto* .

Este parágrafo é o que chamamos de ***project statement***. A capacidade de síntese é fundamental nesta etapa do projeto.

Os “use-cases” refletem os requisitos (geralmente funcionais) do projeto que são baseados na interação entre agentes (usuários) e o sistema de informação.



The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

Obrigado