



**Modelagem**

**Bancos de Dados**

**1000 Luiz Celso Gomes-Jr**

**gomesjr@dainf.ct.utfpr.edu.br**

[Bancos de Dados](http://Bancos de Dados)

- "Um banco de dados representa algum aspecto do mundo real, às vezes chamado de **mini-mundo** ou de **universo de discurso** (UoD – Universe of Discourse)." (Elmasri & Navathe, 2011)

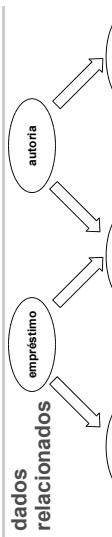
# Universo de Discurso ou Mini-mundo

- Recorte do mundo real a ser representado



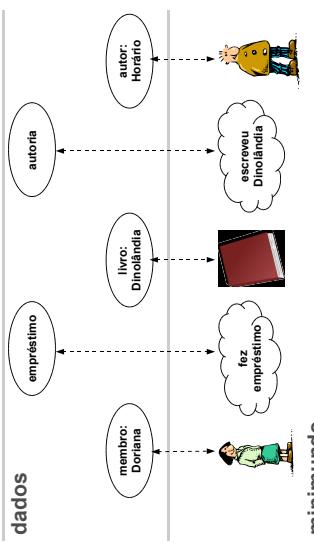
Banco de Dados

- Colección de documentos manuscritos



Daddos

- Estados registradas significando implícito

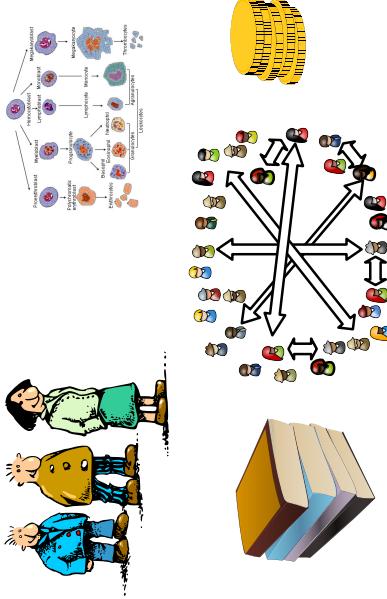


- # Problema x Abstração
- "Para resolver um problema é necessário escolher uma abstração da realidade" (Almeida, 2010)
- ## Abstração

### Abstração

- "processo mental que consiste em **escolher ou isolar um aspecto** determinado de um estado de coisas relativamente complexo, a fim de simplificar a sua avaliação, classificação ou para permitir a comunicação do mesmo" (Houaiss, 2006)
- Abstrações ajudam a gerenciar a complexidade do software (Shaw, 1984)

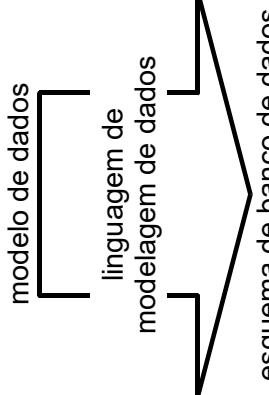
### Abstrações do Dia a Dia



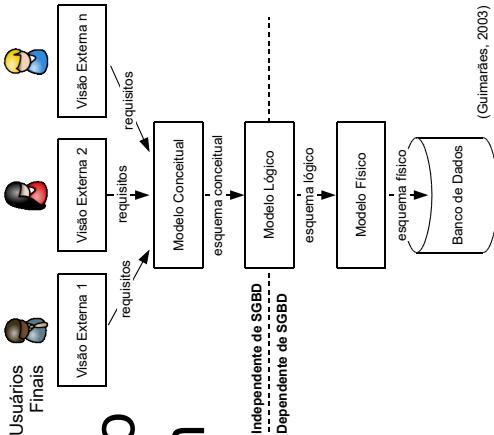
### Modelo de Dados

- Modelo de dados em Banco de dados:
  - "descrição formal da estrutura de um banco de dados"(Heuser, 2004)

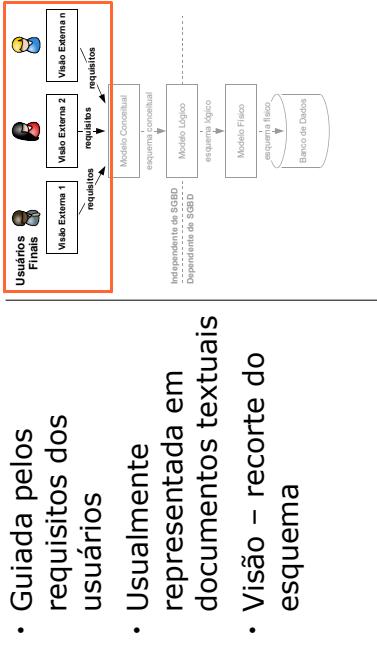
### Esquema de Banco de Dados



## Projeto de um BD



## Visão Externa

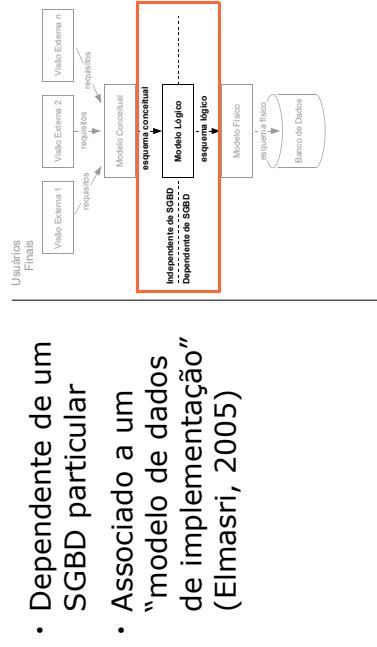


- Guiada pelos requisitos dos usuários
- Usualmente representada em documentos textuais
- Visão – recorte do esquema

## Modelo/Esquema Conceitual

- Descreve estrutura do Banco de Dados
  - entidades, tipos de dados, relações, restrições etc.
- Independente de implementação em SGBD
  - oculta detalhes de armazenamento físico

## Modelo/Esquema Lógico

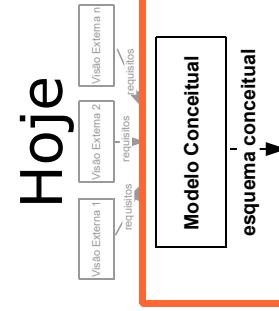
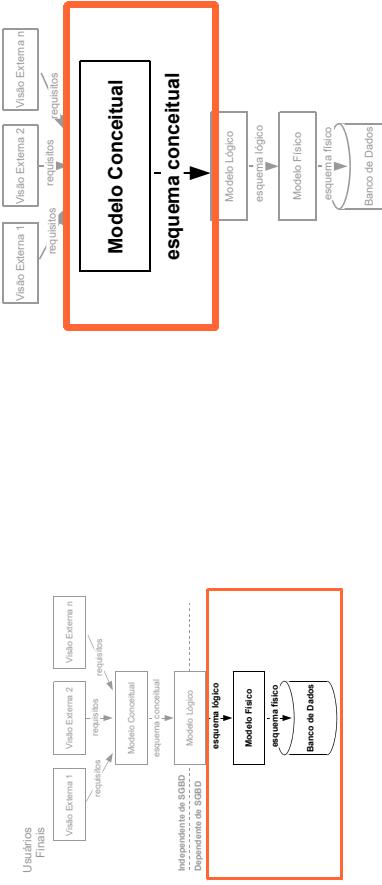


- Dependente de um SGBD particular
- Associado a um "modelo de dados de implementação" (Elmasri, 2005)

## Modelo/Esquema Físico

- Descreve a estrutura de armazenamento físico

## Hoje

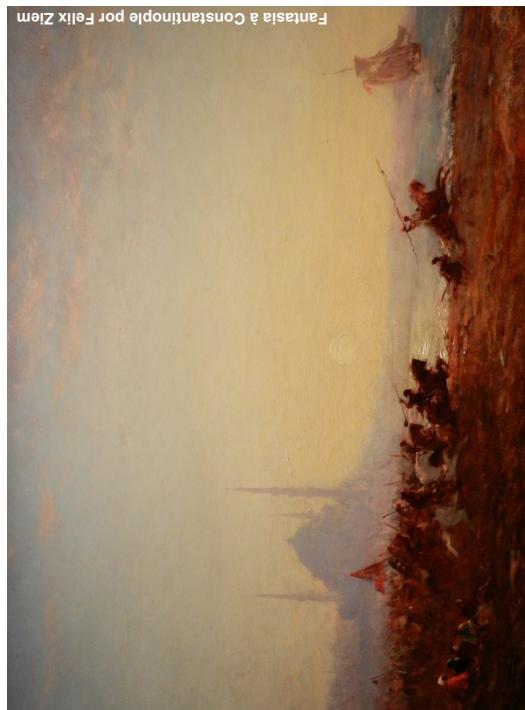


## Modelo Conceitual

- esquema conceitual

# Como modelamos o mundo

## Intuitivo Entidades / Objetos



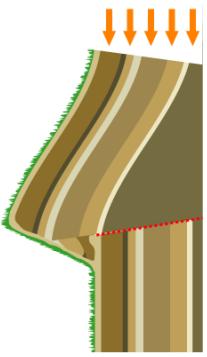
## Objetos

- Montanha



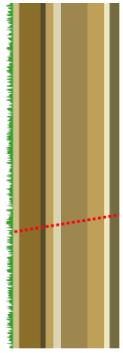
# Objetos

- Montanha



# Objetos

- Montanha



## Objetos - Ferroviária



## Noção de Objeto

- Psicologia do desenvolvimento:

- Quando crianças representam objetos como entidades permanentes?
- Que persistem:
  - Através do tempo e espaço
    - À oclusão

(Santos & Hood, 2009)

## Noção de Objetos

- Objetos permanecem?

- "Of course, the concept of object permanence itself is really a misnomer, as all objects comprise energy in continuous states of change." (Santos & Hood, 2009)

## Formal Entidades / Objetos

# Modelo Entidade-Relacionamento

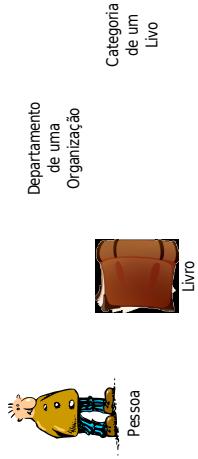
## Modelo Entidade-Relacionamento (ER)

- Padrão para modelagem conceitual
- Criada por Peter Chen em 1976

### Entidade

- Objeto do universo de discurso
- Identificável distintamente
- Existência independente

### Entidade



# Modelo Orientado a Objetos

- SIMULA 67
  - Primeira Linguagem Orientada a Objetos
- Smalltalk
  - Projeto Dynabook
    - "Este 'Dynabook' foi baseado na visão de computadores pessoais baratos do tamanho de um caderno, tanto para adultos quanto crianças, com a capacidade de lidar com todas as suas respectivas necessidades de informação". [KRE98]

## Modelo Orientado a Objetos (OO)

## ● Objeto

- Objeto do universo de discurso
- Identificável distintamente
- Existência independente

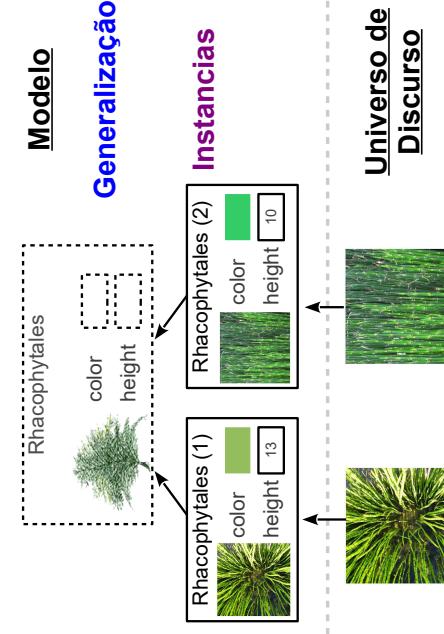


- Objetos são caracterizados por:
- identidade;
  - atributos;
  - comportamento.

## Exemplo de Objeto

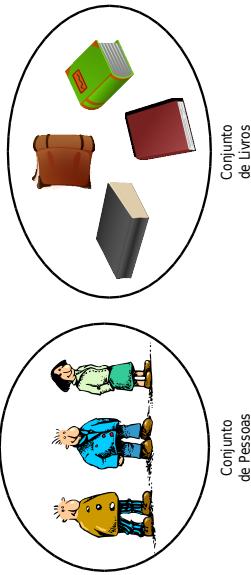
Objeto Esfera
Esf era Vermelha
Atributos (nome, valor)
(peso, 200 g)
(raio, 60 cm)
(elasticidade, alta)
(cor, vermelha)
Comportamento
aumentar, diminuir, se mover

## Estereótipos / Classes



## Tipo Entidade

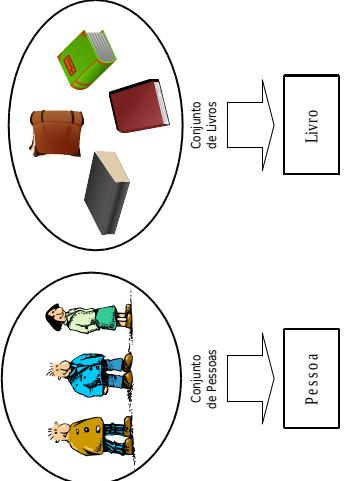
- Tipo Entidade ou Conjunto de Entidades
  - conjunto não disjunto
  - entidades similares – mesmos atributos



## ER: Tipo Entidade

## Tipo Entidade

- Representação:



## Classe

"Numa série ou num conjunto, grupo ou divisão que apresenta características ou atributos semelhantes." (Ferreira, 1989)

- Classificação de Carl Linné



## Classe

- Quando realizamos uma classificação de objetos, identificamos o seu comportamento e as características que eles possuem em comum.

- Classes definem:

- Atributos que irão descrever o objeto;
- Métodos que definem o comportamento dos mesmos.



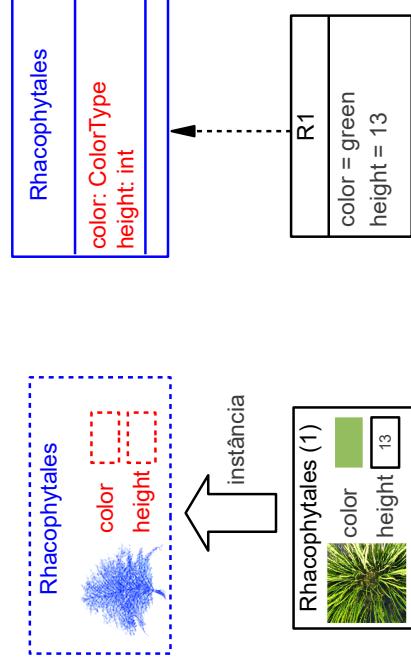
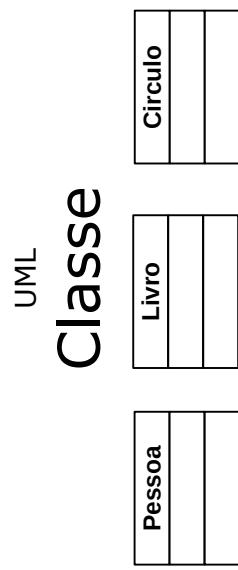
## Objetos e Classes

- Os objetos são organizados/divididos em grupos chamados classes.
- Objetos da mesma classe têm:
  - o mesmo conjunto de atributos (os valores dos atributos podem ser diferentes);
  - o mesmo conjunto de métodos.

## UML Unified Modeling Language

- <http://www.uml.org/>
- Desenvolvida entre 1994-96
- Criadores
  - Grady Booch, Ivar Jacobson and James Rumbaugh na Rational Software
  - Padrão OMG em 1997
  - OMG - Object Management Group
  - <http://omg.org/>

(Wikipedia, 2015)



## Exemplo de Classe

Esfera

Classe Esfera
Atributos (nome, tipo)
(peso, real)
(raio, real)
(elasticidade, string)
(cor, color)

### Comportamento

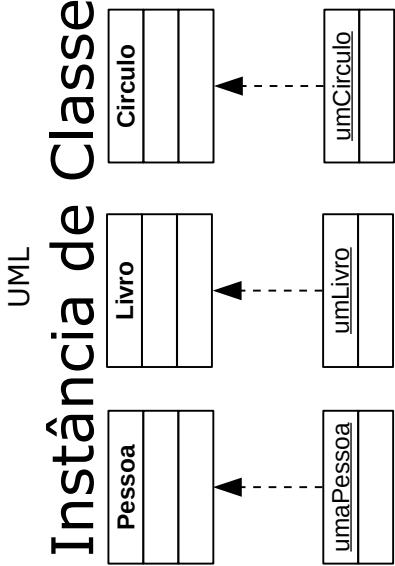
aumentar, diminuir, se mover

## Exemplo de Objeto

Esfera Vermelha

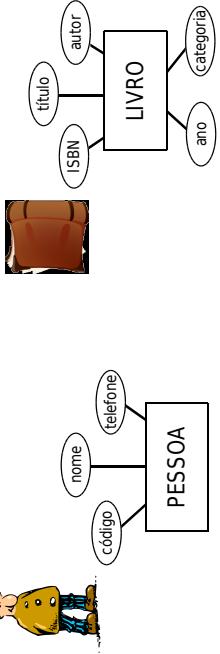
Objeto Esfera
Atributos (nome, valor)
(peso, 200 g)
(raio, 60 cm)
(elasticidade, alta)
(cor, vermelha)

## Instância de Classe



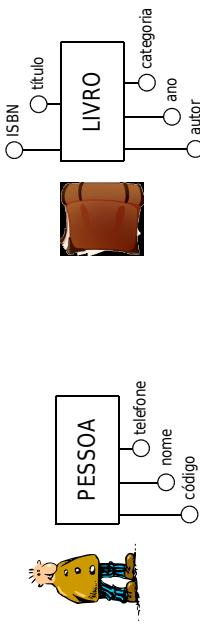
## Atributos

- Cada instância de entidade ou relacionamento tem atributos que a descrevem



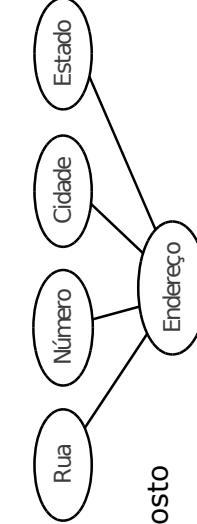
## Atributos

Representação Alternativa



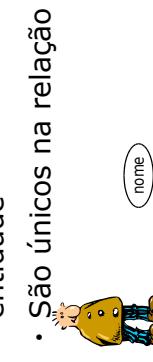
## Tipos de Atributo

- Simples (atômico)
- Multivalorado
- Composto



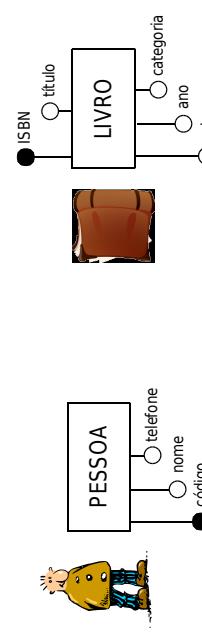
## Atributos-Chave

- Servem para distinguir ocorrências da entidade



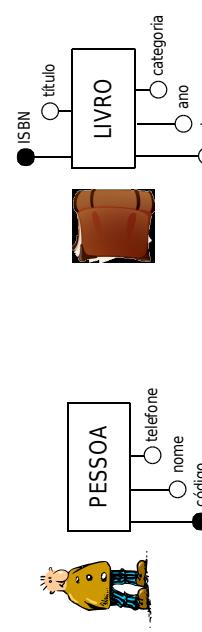
## Atributos-Chave

Representação Alternativa

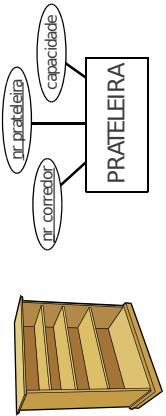


## Atributos-Chave

Representação Alternativa



## Atributo-chave Composto



Pessoa
codigo: String
nome: String
telefone: int

Circulo
centroX: int
centroY: int
raio: int

Livro
isbn: String
titulo: String
autor: String
ano: int
categoria: String

(Heuser, 2004)

## UML Métodos (operações)

Uma indústria farmacêutica quer desenvolver um banco de dados para registrar os medicamentos que ela produz, bem como os vírus tratados por estes medicamentos.

Elabore uma modelo conceitual para este banco de dados conforme o detalhamento a seguir:  
• Devem ser armazenados os nomes científicos e populares dos vírus bem como os períodos de incubação.

- Para medicamentos, o devem ser armazenados o nome de venda e o composto ativo.
- Em princípio considere que não há relação entre vírus e medicamentos.

Circulo
centroX: int
centroY: int
raio: int
area(): double
setRaio(novo: int)

## Exercício (parte 1)

Uma indústria farmacêutica quer desenvolver um banco de dados para registrar os medicamentos que ela produz, bem como os vírus tratados por estes medicamentos.

Elabore uma modelo conceitual para este banco de dados conforme o detalhamento a seguir:  
• Devem ser armazenados os nomes científicos e populares dos vírus bem como os períodos de incubação.

- Para medicamentos, o devem ser armazenados o nome de venda e o composto ativo.
- Em princípio considere que não há relação entre vírus e medicamentos.

## Relacionamento

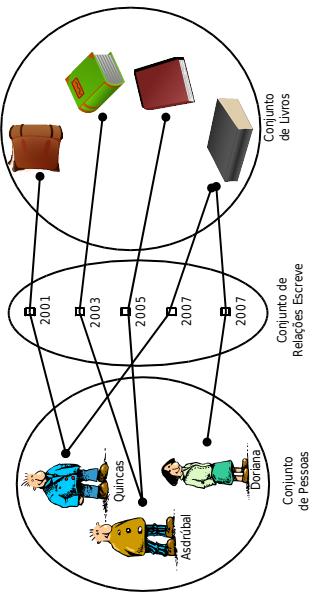
## ER: Relacionamento

## ER: Relacionamento

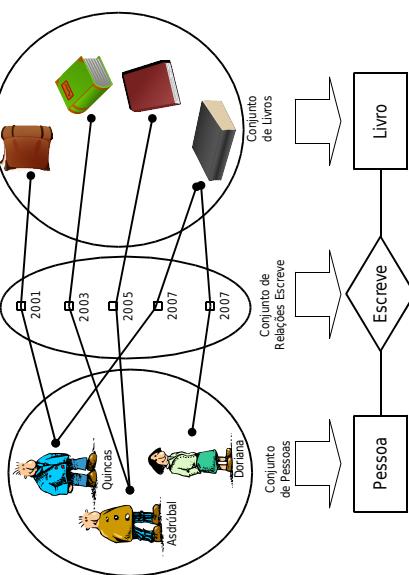
- Associação entre entidades
- Atributo de uma entidade que se refere a outra



## Conjunto de Relacionamentos



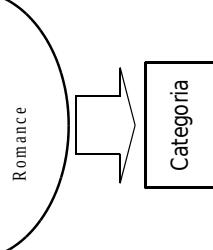
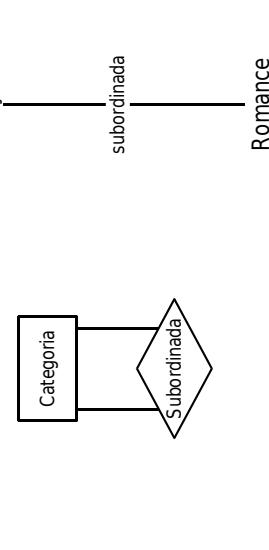
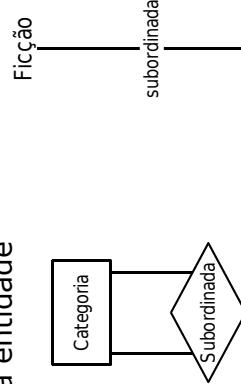
## Conjunto de Relacionamentos



## ER: Relacionamento Exemplo Pertence

### ER: Auto-Relacionamento

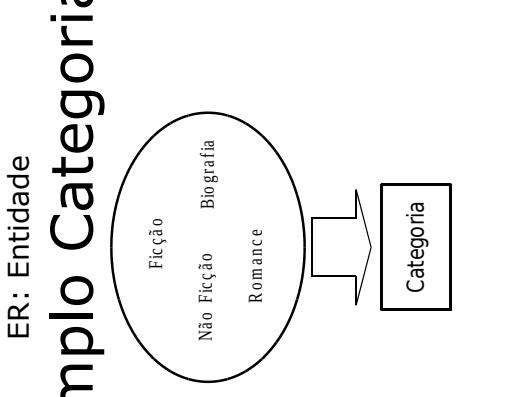
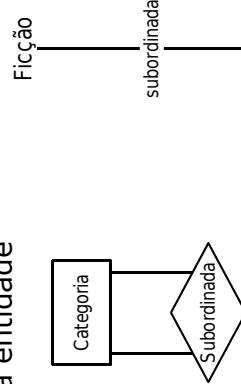
- Relacionamento entre ocorrências da mesma entidade



## ER: Entidade Exemplo Categoria

### ER: Auto-Relacionamento

- Relacionamento entre ocorrências da mesma entidade

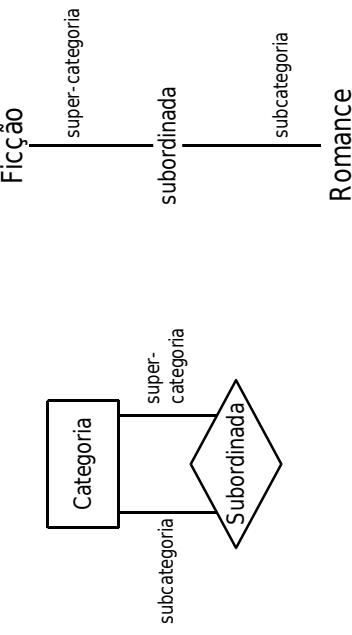


## ER: Papéis

- Função que instância de entidade cumpre dentro de instância de relacionamento



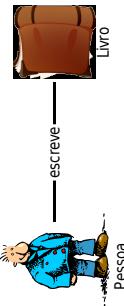
## ER: Papéis



## OO: Relacionamento

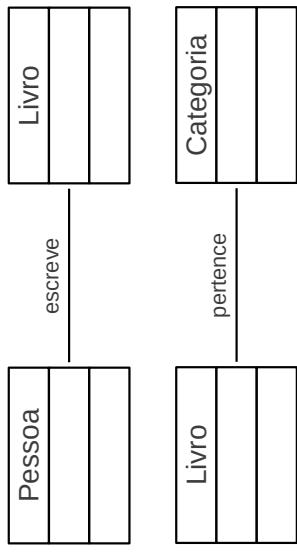
- Associação entre objetos
- Atributo de um objeto que se refere a outro
  - Atributo definido na classe

## OO: Relacionamento



## UML: Relacionamento

### Relacionamento Direcionado



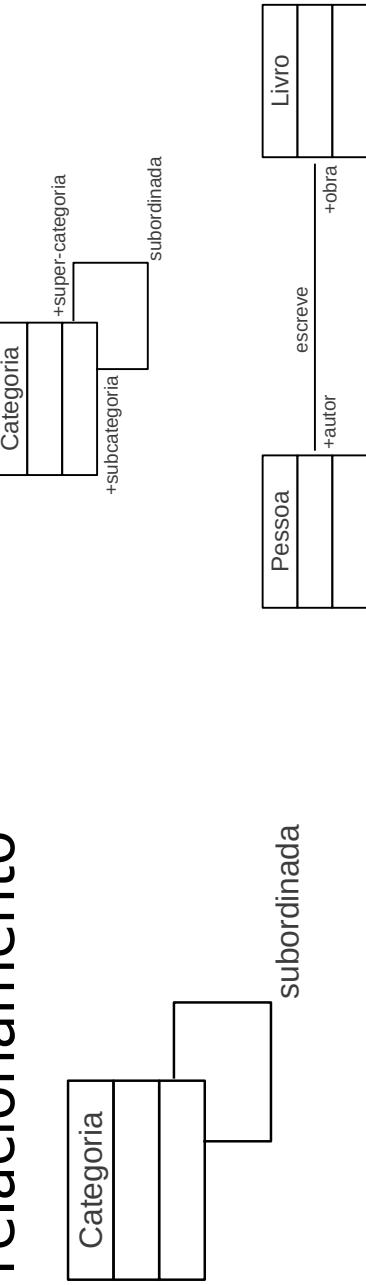
## UML: Relacionamento

### Direcionado



## UML: Auto-relacionamento

### UML: Papéis



## ER: Grau de Relacionamento

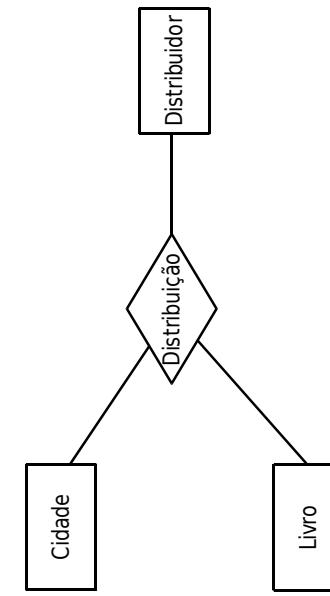
### Relacionamento Binário

- Número de entidades que participam do relacionamento



## Grau de Relacionamento Ternário

### Relacionamento no Ternário



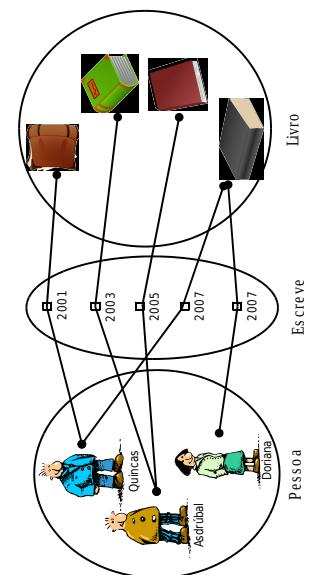
- Restrições que limitam a possibilidade de combinações de entidades em relacionamentos
- Cardinalidade:
  - Máxima
  - Mínima

## Razão de Cardinalidade

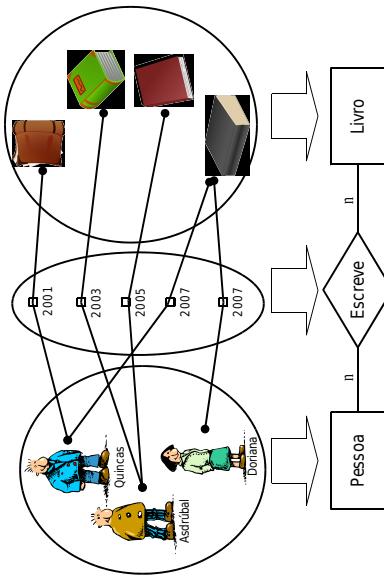
- É expressa a razão (ou proporção) de participação em um relacionamento.
- Transcrição gráfica das proporções: 1:1, 1:N, N:1 e N:N

## Razão de Cardinalidade

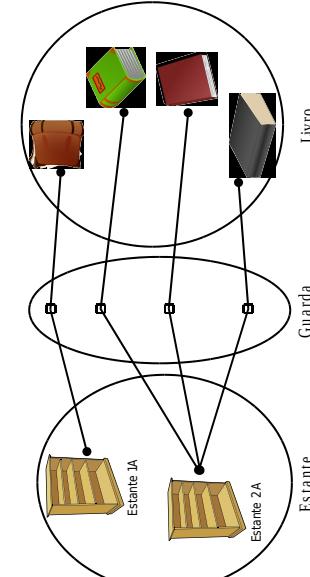
### Relacionamento n:n



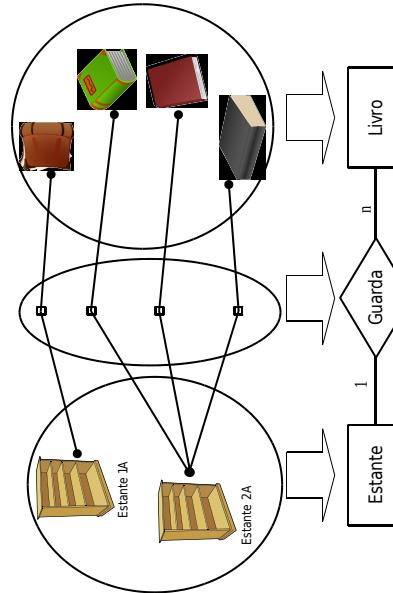
### Relacionamento n:n



### Relacionamento 1:n

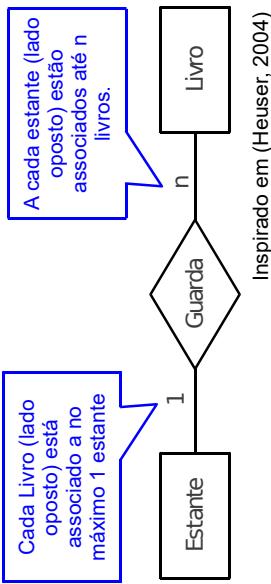


### Relacionamento 1:n



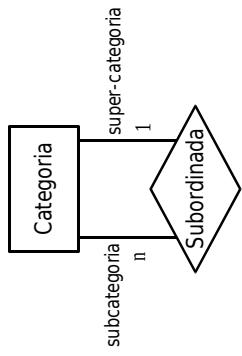
## Notação de Cardinalidade

- A notação com apenas um valor de cada lado representa a razão (ou proporção) na participação. Abaixo, proporção 1:N.



## Relacionamento 1:n

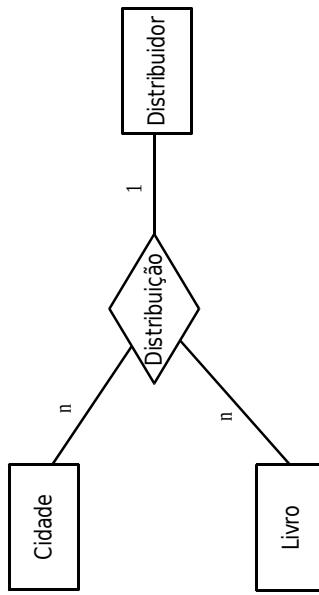
- A notação com apenas um valor de cada lado representa a razão (ou proporção) na participação. Abaixo, proporção 1:N.



## Relacionamento 1:1



## Cardinalidade em Relacionamento Ternário



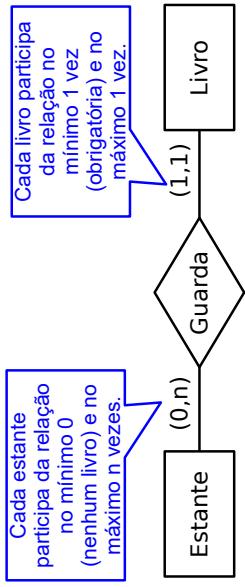
## Restrição de Participação na Relação

- Notação alternativa à razão de cardinalidade.
- Indica restrição mínima e máxima (min, max) de participação de cada entidade na relação.
- É indicado no lado correspondente à entidade (oposto do anterior).

## Restrição de Participação na Relação (Cardinalidade)

## Restrição de Participação na Relação

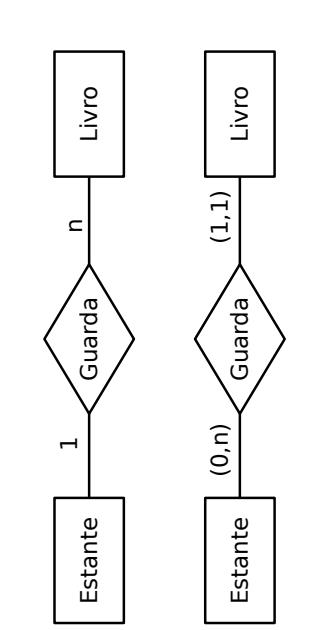
- É indicado no lado correspondente à entidade (oposto do anterior).



## Restrição de Participação na Relação



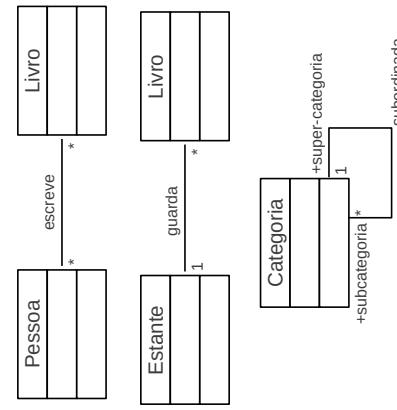
## Restrição de Participação na Relação



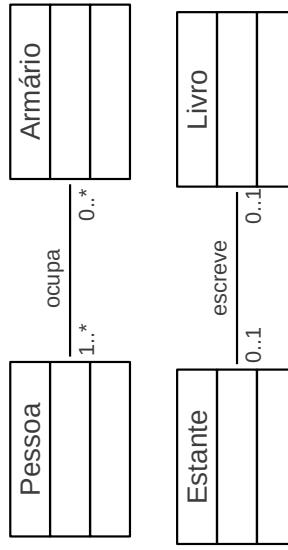
## Restrições de Participação

- Relacionadas à cardinalidade mínima:
  - Participação Total (obrigatória)  $\Rightarrow$  mínima 1
  - Participação Parcial (opcional)  $\Rightarrow$  mínima 0

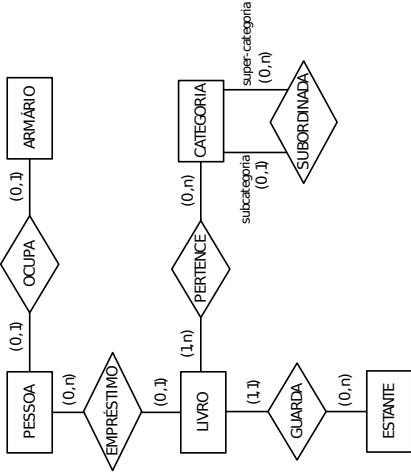
## UML: Cardinalidade Máxima



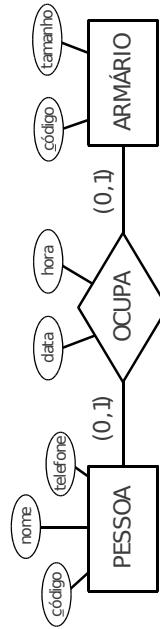
## UML: Cardinalidade Mínima



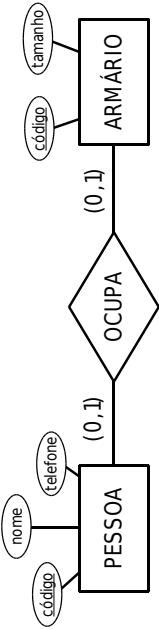
## Exemplo Diagrama ER



## Atributos no Relacionamento



## Atributos no Relacionamento



## Exercício

Uma indústria farmacêutica quer desenvolver um banco de dados para registrar os medicamentos que ela produz, bem como os vírus tratados por estes medicamentos.

Elabore uma modelo conceitual para este banco de dados conforme o detalhamento a seguir:

- Devem ser armazenados os nomes científicos e populares dos vírus bem como os períodos de incubação.
- Para medicamentos, o devem ser armazenados o nome de venda e o composto ativo.

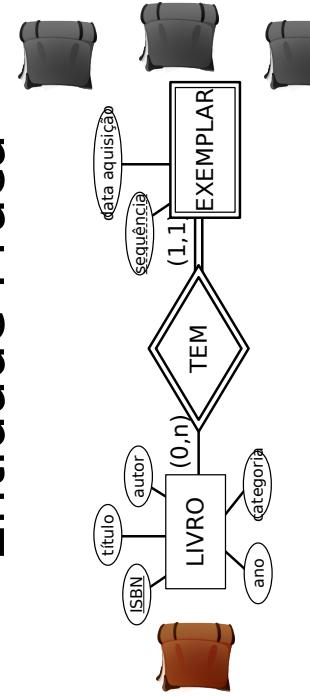
## Exercício

parte 2

a) Considere que um dado medicamento pode tratar vários vírus e um vírus pode ser tratado por vários medicamentos.

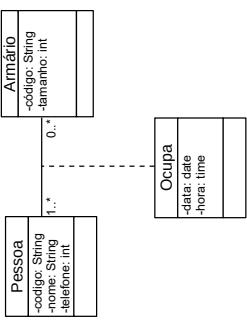
b) Medicamentos são fabricados por empresas que possuem nome e CNPJ. Uma empresa pode fabricar vários medicamentos, mas um medicamento é fabricado por uma única empresa.

## Exercício



## Classe de Associação

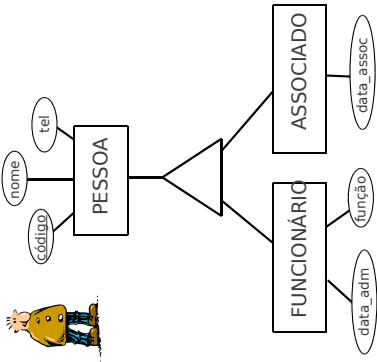
## EER – ER Estendido



- ER original não suporta generalização/especialização
- ER Estendido (EER) – acrescenta estes recursos

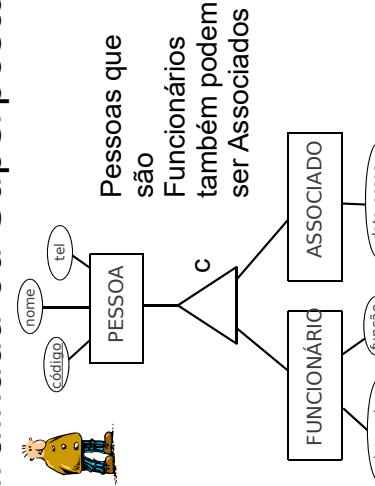
## EER: Especialização/Gener alização

### Generalização / Especialização



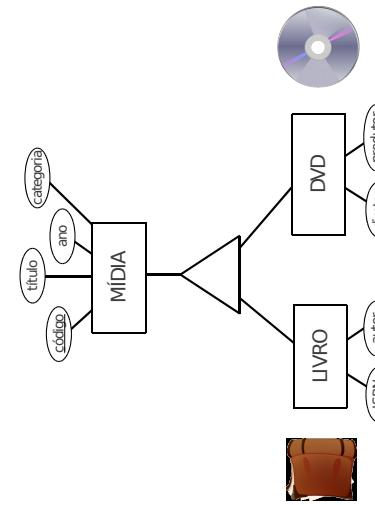
### Generalização / Especialização

### Compartilhada ou Superposta



### Generalização / Especialização

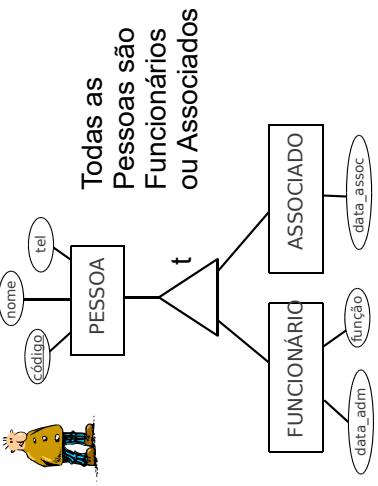
### Generalizada / Especializada



## Generalização / Especialização Exclusiva ou Disjunta

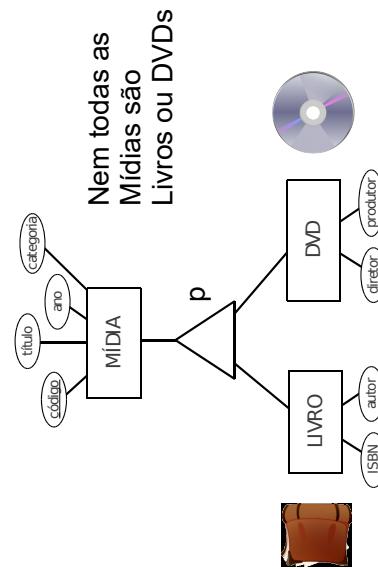
Generalização / Especialização

### Total



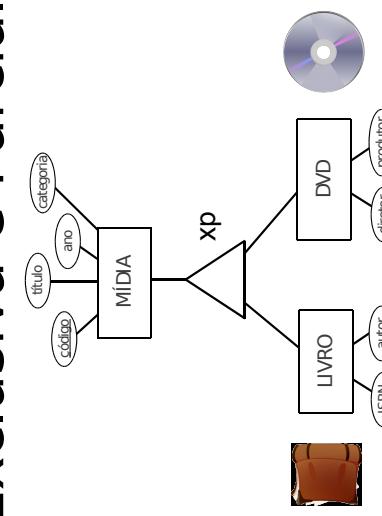
## Generalização / Especialização Parcial

### Compartilhada e Total



## Generalização / Especialização Exclusiva e Parcial

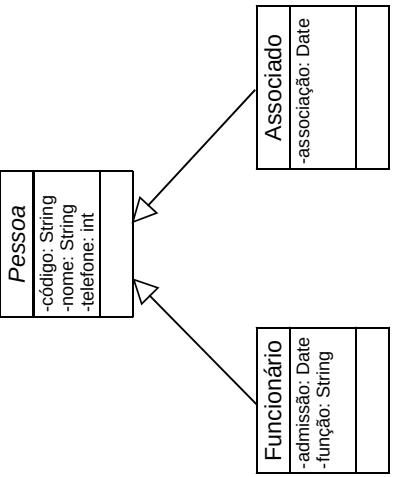
### Tipos



	Total (t)	Parcial (p)
Exclusiva (x)	xt	xp
Compartilhada (c)	ct	cp

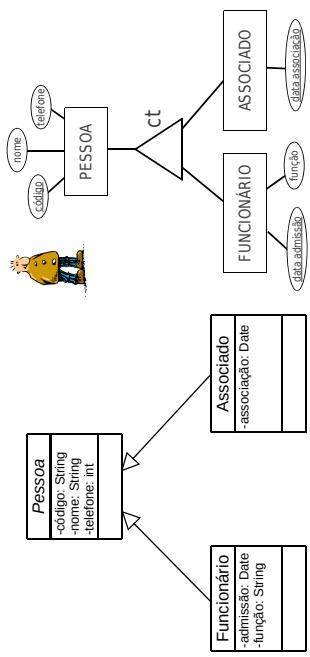
(Heuser, 2004)

## UML: Herança

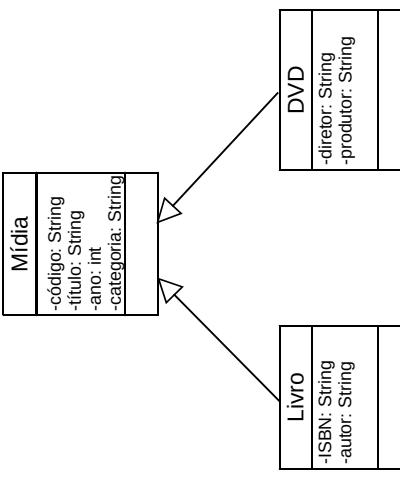


## OO: Herança

Especialização total x Classe abstrata

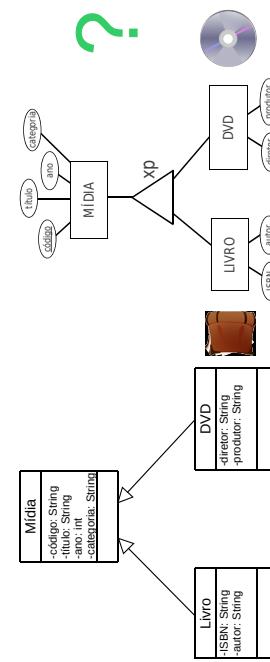


## UML: Herança

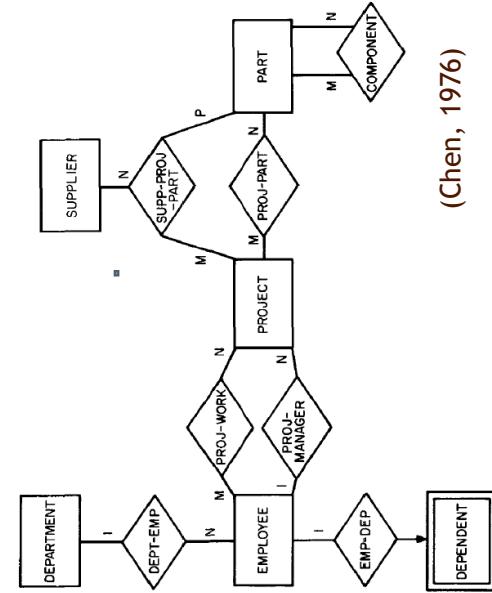


Especialização total x classe abstrata

Especialização parcial x Classe



?



(Chen, 1976)

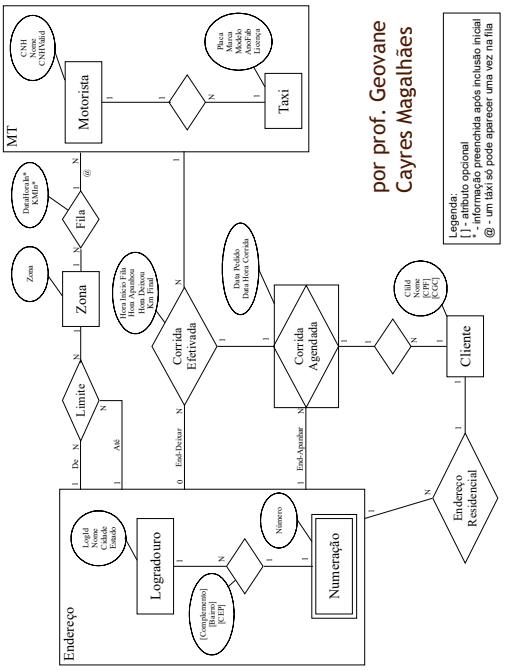
# Exercício

parte 3

- Vírus podem ser classificados em diversas categorias, mas considere apenas Vírus com DNA, Vírus com RNA e Retrovírus. Retrovírus são tratados com coquetéis de medicamentos. Um coquetel é composto por vários medicamentos, cada um em uma concentração específica. Um coquetel tem uma dosagem específica para o tratamento de um dado Retrovírus.

## Caso dos Taxis

- Exemplo criado por prof. Geovane Cayres Magalhães
  - <http://www.ic.unicamp.br/~geovane/mo410-091/caso.html>



## Referências

- Chen, Peter Pin-Shan (1976) **The entity-relationship model – toward a unified view of data**. ACM Trans. Database Systems, ACM, 1, 9-36.
- Elmasri, Ramez; Navathe, Shamkant B. (2005) **Sistemas de Bancos de Dados**. Addison-Wesley, 4<sup>a</sup> edição em português.
- Guimaraes, Célio (2003) **Fundamentos de Bancos de Dados: Modelagem, Projeto e Linguagem SQL**. Editora UNICAMP, 1<sup>a</sup> edição.
- Heuser, Carlos Alberto (2004) **Projeto de Banco de Dados**. Editora Sagra Luzzato, 5<sup>a</sup> edição.

## Referências

- Ramakrishnan, Raghu; Gehrke, Johannes (2003) **Database Management Systems**. McGraw-Hill, 3<sup>rd</sup> edition.

## Referências

- Almeida, Charles Ornelas ; Guerra, Israel; Ziviani, Nívio (2010) **Projeto de Algoritmos** (transparências aula).
- Bloch, Paul (2007) **Introduction to Psychology – transcrição das aulas (aula 17)**. Yale University.
- Ferreira, Aurélio B. H. (1989) **Minidicionário da Língua Portuguesa**. Rio de Janeiro, Editora Nova Fronteira.
- Houaiss, Instituto Antônio. **Dicionário Houaiss da língua portuguesa** (2006) Editora Objetiva, Março.
- IBM - International Business Machines Corporation. **IBM Smalltalk Tutorial** [Online] <http://www.wi2.uni-erlangen.de/sw/smalltalk/>
- Liskov, Barbara; Zilles, Stephen. **Programming with abstract data types** (1974) ACM SIGPLAN Notices, 9 (4) p. 50.

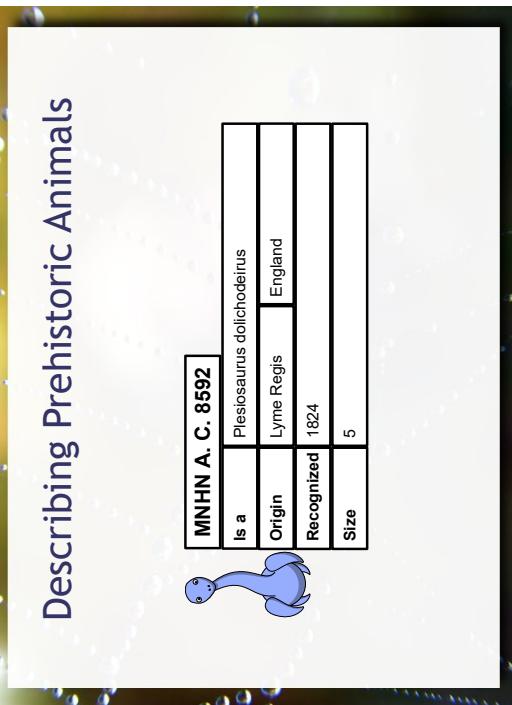
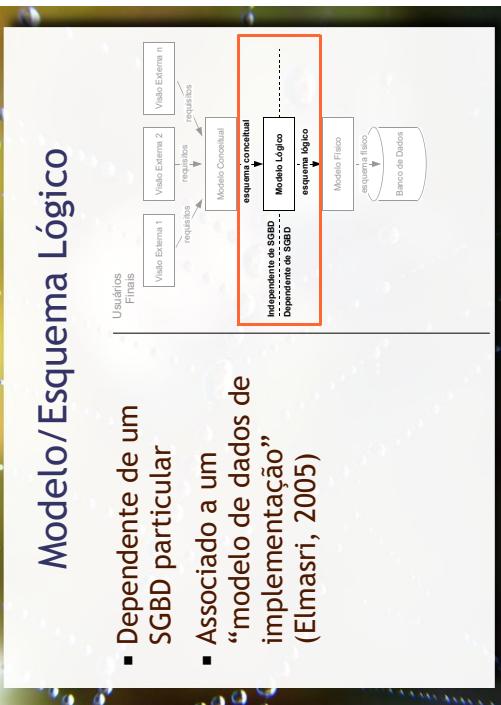
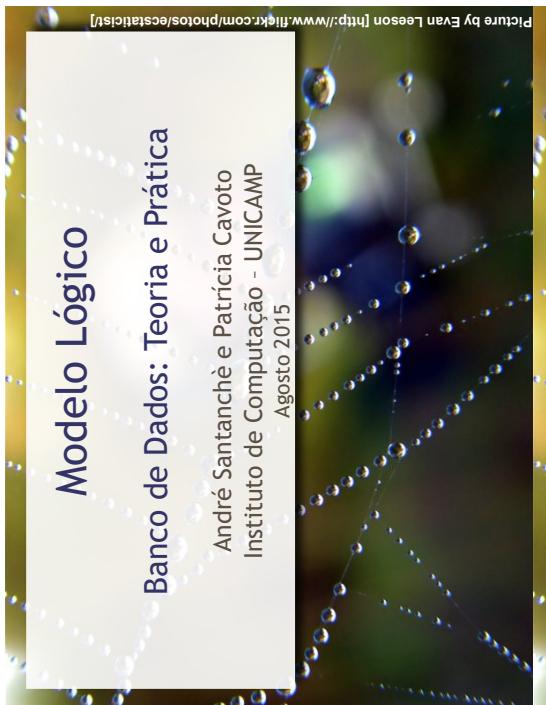
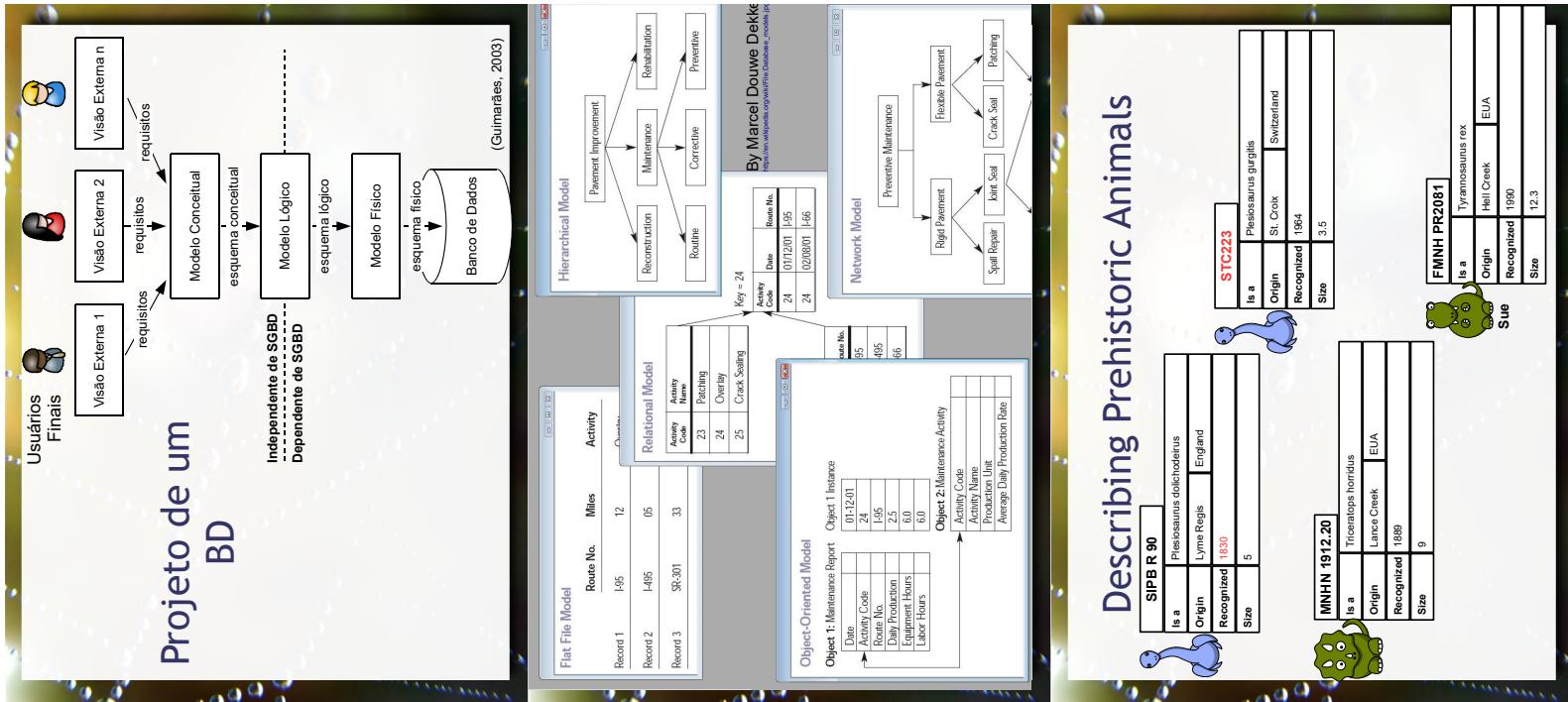
# Referências

## Bibliográficas

- Meyer, Bertrand (1997) **Object-oriented Software Construction – Second Edition**. USA, Prentice-Hall, Inc.
- Miller, Robert (2004) **6.831 User Interface Design and Implementation (lecture notes)**. MIT OpenCourseware.
- Rocha, Heloisa Vieira da; Baranauškas, Maria Cecilia Calani (2003) **Design e Avaliação de Interfaces Humanocomputador**. NIED/UNICAMP.
- Santos, L. R., & Hood, B. M. (2009). **Object representation as a central issue in cognitive science**. In The Origins of Object Knowledge: The Yale Symposium on the Origins of Object & Number Representation. Oxford: Oxford University Press.
- Shaw, M. **Abstraction Techniques in Modern Programming Languages** (1984) IEEE Software, 1, 4, 10-26.
- Bloom, Paul (2007) **Introduction to Psychology** – transcrição das aulas (aula 17). Yale University.
- Chen, Peter Pin-Shan (1976) **The entity-relationship model – toward a unified view of data**. ACM Trans. Database Systems, ACM, 1, 9-36.
- Dijkstra, E. W. (1986) **On a cultural gap**. The Mathematical Intelligencer. vol. 8, no. 1, pp. 48-52.
- Elmasri, Ramez; Navathe, Shamkant B. (2005) **Sistemas de Bancos de Dados**. Addison-Wesley, 4a. edição em português.
- Elmasri, Ramez; Navathe, Shamkant B. (2011) **Sistemas de Bancos de Dados**. Pearson, 6a. edição em português.
- Guimarães, Célio (2003) **Fundamentos de Bancos de Dados: Modelagem, Projeto e Linguagem SQL**. Editora INTICAMP 1a. edição.

## Referências e Agradecimentos

- Diversos slides baseados no curso de BD do Prof. André Santanchè (UNICAMP)
  - Site: <http://www.ic.unicamp.br/~santanche>
  - Canal YouTube:  
<https://www.youtube.com/santanche>



## Modelo Relacional

**Cliente (C)**

CId	Nome	CPF
1532	Asdrubal	448.754.253-65
1755	Dorian	567.387.387-44
1780	Quinca	546.373.762-02


**Táxi (TX)**

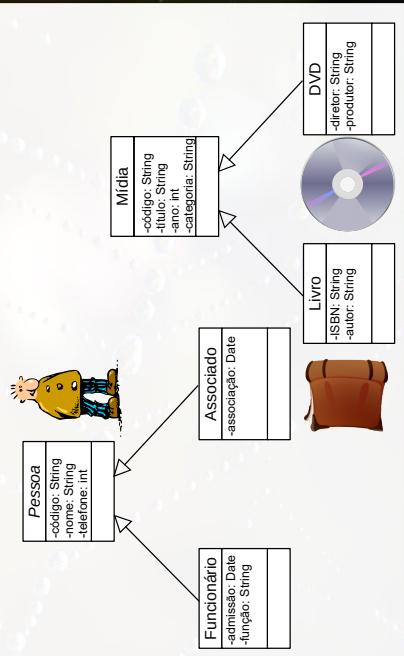
Placa	Marca	Modelo	AnoFab
DAE6534	Ford	Fiesta	1999
DKL4598	Volkswagen	Gol	2001
DKL7878	Ford	Fiesta	2001
JDM8776	Volkswagen	Santana	2002
JJM3692	Chevrolet	Corsa	1999

**Corrida (R1)**

CId	Placa	Data Pedido
1755	DAE6534	15/02/2003
1982	JDM8776	18/02/2003

Modelo original  
por prof. Geovane  
Cayres Magalhães

## Modelo Orientado a Objetos



## Modelo Relacional

CId	Nome	CPF
1532	Asdrubal	448.754.253-65
1755	Dorian	567.387.387-44
1780	Quinca	546.373.762-02


**Táxi (TX)**

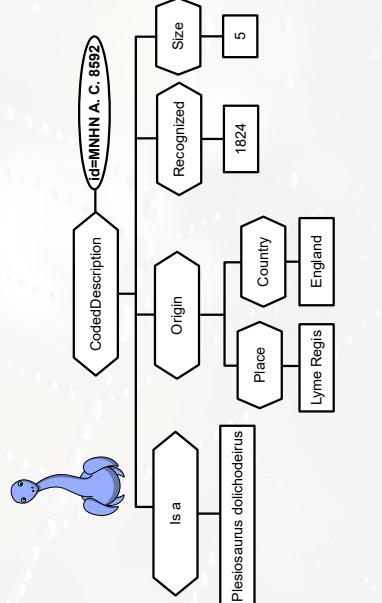
Placa	Marca	Modelo	AnoFab
DAE6534	Ford	Fiesta	1999
DKL4598	Volkswagen	Gol	2001
DKL7878	Ford	Fiesta	2001
JDM8776	Volkswagen	Santana	2002
JJM3692	Chevrolet	Corsa	1999

**Corrida (R1)**

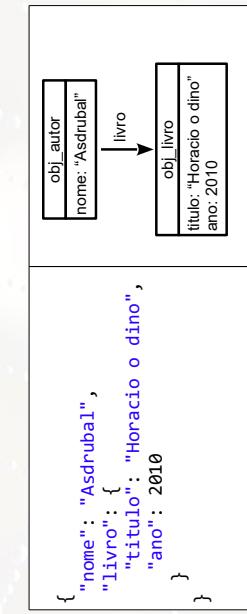
CId	Placa	Data Pedido
1755	DAE6534	15/02/2003
1982	JDM8776	18/02/2003

Modelo original  
por prof. Geovane  
Cayres Magalhães

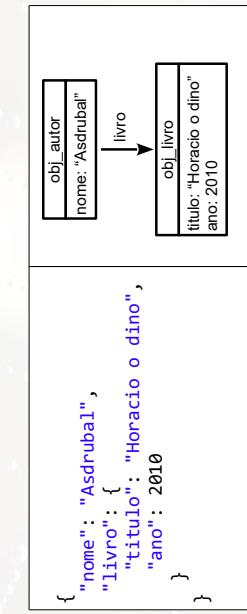
## Modelo Hierárquico (Documentos/XML)



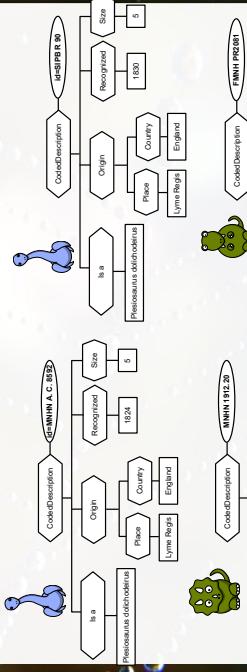
## Modelo Hierárquico (Documentos/XML)



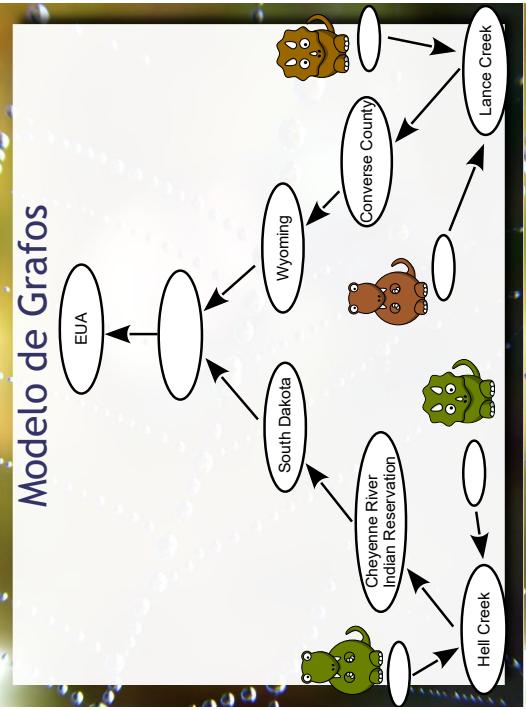
## Modelo Hierárquico (JSON - Documentos + OO)



## Modelo Hierárquico (Documentos/XML)

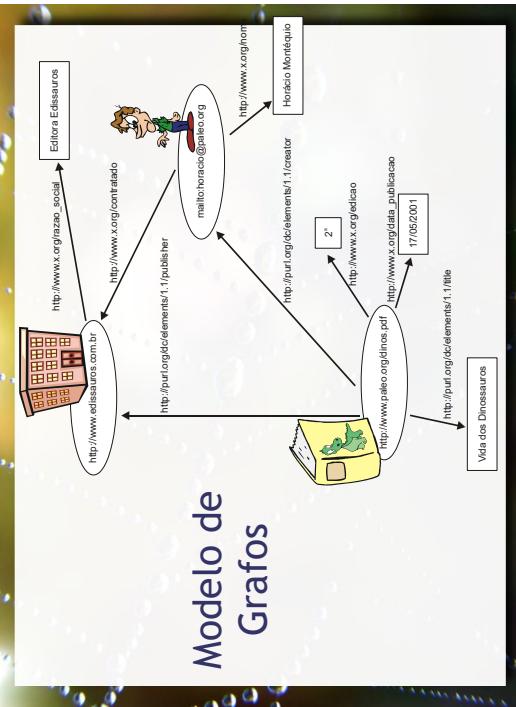
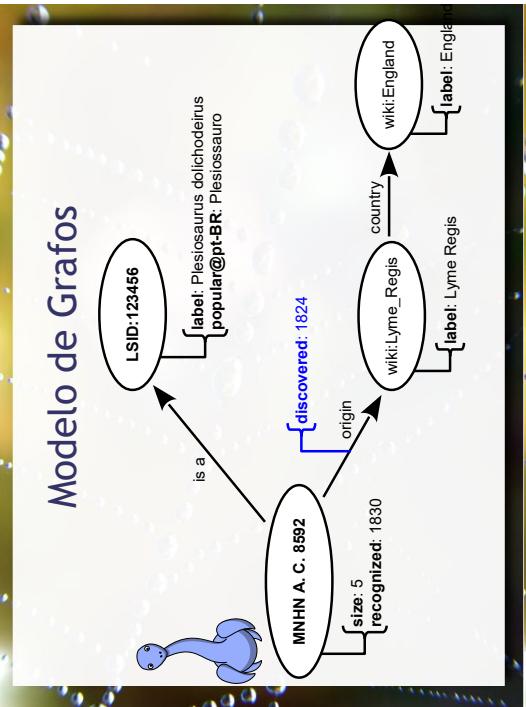


## Modelo Hierárquico (Documentos/XML)



### Agradecimentos

- Luiz Celso Gomes Jr (professor desta disciplina em 2014) pela contribuição na disciplina e nos slides.



### Referências

- Heuser, Carlos Alberto (2004) **Projeto de Banco de Dados**. Editora Sagra Luzzato, 5ª edição.
- Ramakrishnan, Raghunath; Gehrke, Johannes (2003) **Database Management Systems**. McGraw-Hill, 3rd edition.

## License

- These slides are shared under a Creative Commons License.  
Under the following conditions: Attribution, Noncommercial  
and Share Alike.
- See further details about this Creative Commons license at:  
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

André Santanchè

<http://www.íc.unicamp.br/~Santanche>

# O Modelo Relacional

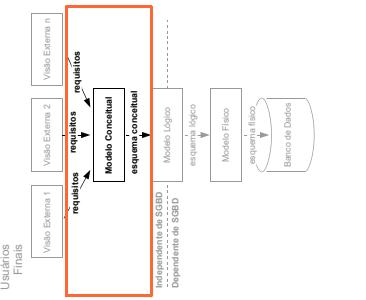
## Agenda

- Recapitulação
- Modelo Relacional
- Restrições de Integridade
- Exercícios
- Além do modelo relacional

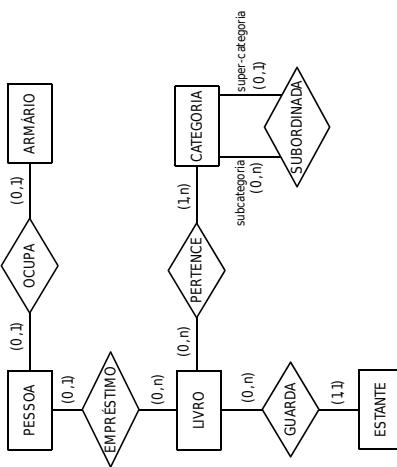
01010110 Banco de Dados  
1010001010111000111110101010101000  
10010010101101100110011001100110101  
01010001010111011000111010101010101  
1101111001111111010101010001011111  
1010111011100011101011001101111111  
101110101010101010111000010101010111  
0110101111001111111010101010101000  
0101 Luiz Celso Gomes-Jr. 01010110011011  
gomesjir@da.inf.ct.utfpr.edu.br 10101010101  
1000101011100111000111000111000111

## Modelo/Esquema Conceitual

- Descreve estrutura do Banco de Dados
  - entidades, tipos de dados, relações, restrições etc.
- Independente de implementação em SGBD
  - oculta detalhes de armazenamento físico

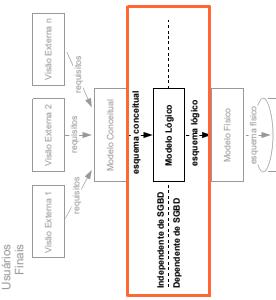


## Modelo ER



## Modelo/Esquema Lógico

- Dependente de um SGBD particular
- Associado a um "modelo de dados de implementação" (Elmasri, 2005)



## Modelo Relacional

- Proposto por E. F. Codd em 1970 no artigo:
  - "A Relational Model for Large Shared Data Banks"
- Resposta a um cenário complexo de incompatibilidade de modelos e sistemas
  - Independência da estrutura interna
    - "Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed [...]" (Codd, 1970)

# Modelo Relacional

## Larry Ellison, ORACLE

- Modelo mais amplamente utilizado por SGBDs
- Maiores empresas de informática oferecem soluções: IBM, Microsoft, Oracle, SAP
- Grandes projetos Open Source: MySQL, PostgreSQL, SQLite
- De celulares à data centers
- Mercado de U\$24bi (2012)



## Relação

- Banco de Dados Relacional contém um conjunto de **Relações**

- **Relação** – informalmente pode ser visualizada como uma tabela

Livro	relação	ISBN	Título	Categoria	Autor	Ano
		9880471444	Vidas Secas	Romance	Graciliano Ramos	1938
		988047850X	Agosto	Romance	Rubem Fonseca	1990
		0554253216	Micrografia	Ciências	Robert Hooke	1665
		0195087445	Divina Comédia	Poesia	Dante Alighieri	1308
		0559274289	Le Operæ	Ciências	Galileu Galilei	1611
		0451526929	Hamlet	Drama	William Shakespeare	1599
		1603033785	Othello	Drama	William Shakespeare	1565

## Relação (Tabela)

- Relação = conjunto não ordenado de tuplas (linhas)
- Não existem tuplas duplicadas

Livro	relação	ISBN	Título	Categoria	Autor	Ano
		9880471444	Vidas Secas	Romance	Graciliano Ramos	1938
		988047850X	Agosto	Romance	Rubem Fonseca	1990
		0554253216	Micrografia	Ciências	Robert Hooke	1665
		0195087445	Divina Comédia	Poesia	Dante Alighieri	1308
		0559274289	Le Operæ	Ciências	Galileu Galilei	1611
		0451526929	Hamlet	Drama	William Shakespeare	1599
		1603033785	Othello	Drama	William Shakespeare	1565

## Relação (Tabela)

- Tupla = conjunto ordenado de atributos
- Valores de atributos são atômicos e monovalorados

Livro	relação	ISBN	Título	Categoria	Autor	Ano
		9880471444	Vidas Secas	Romance	Graciliano Ramos	1938
		988047850X	Agosto	Romance	Rubem Fonseca	1990
		0554253216	Micrografia	Ciências	Robert Hooke	1665
		0195087445	Divina Comédia	Poesia	Dante Alighieri	1308
		0559274289	Le Operæ	Ciências	Galileu Galilei	1611
		0451526929	Hamlet	Drama	William Shakespeare	1599
		1603033785	Othello	Drama	William Shakespeare	1565

Exercício de aquecimento						
▪ Estime os dados abaixo pensando em uma grande empresa como a Petrobras:						
a) nº de tabelas de uma aplicação típica (média)						
b) nº de tabelas de uma aplicação grande e complexa						
c) nº total de tabelas distintas, considerando todas as aplicações						
d) média de nº de atributos por tabela						
e) número de atributos das "maiores" tabelas						
f) nº de DBAs envolvidos						
g) nº de administradores de dados envolvidos						

Livro	atributos	ISBN	Título	Categoria	Autor	Ano
		9880471444	Vidas Secas	Romance	Graciliano Ramos	1938
		988047850X	Agosto	Romance	Rubem Fonseca	1990
		0554253216	Micrografia	Ciências	Robert Hooke	1665
		0195087445	Divina Comédia	Poesia	Dante Alighieri	1308
		0559274289	Le Operæ	Ciências	Galileu Galilei	1611
		0451526929	Hamlet	Drama	William Shakespeare	1599
		1603033785	Othello	Drama	William Shakespeare	1565

## Exercício de aquecimento

- Estime os dados abaixo pensando em uma grande empresa como a Petrobras:
  - nº de tabelas de uma aplicação típica (média): **30 tabelas**
  - nº de tabelas de uma aplicação grande e complexa: **500 tabelas (quinhentas!)**
  - total de tabelas distintas, considerando todas as aplicações: **10.000 tabelas**
  - média de nº de atributos por tabela: **10 atributos**
  - número de atributos das "maiores" tabelas: **80 atributos**

## Exercício de aquecimento

- Estime os dados abaixo pensando em uma grande empresa como a Petrobras:
  - nº de DBAs envolvidos: cerca de 15 (parte física, replicação, backup, controle de acesso, performance, servidores - não contando as pessoas de infraestrutura que mantêm os servidores em si)
  - nº de administradores de dados envolvidos: cerca de 40 (envolvidos na elaboração e manutenção dos modelos e esquemas, manutenção de metadados, elaboração de scripts, procedures, views; dominam os assuntos / negócio)

## Relação (Tabela)

- Nome dos atributos – título das colunas

Livro

ISBN	Titulo	Categoria	Autor	Ano	nome dos atributos
9580471444	Vidas Secas	Romance	Graciliano Ramos	1938	
958047950X	Agosto	Romance	Rubem Fonseca	1990	
0554253216	Micrographia	Ciências	Robert Hooke	1665	
0198017445	Divina Comédia	Poesia	Dante Alighieri	1308	
0559274289	Le Operae	Ciências	Galileo Galilei	1611	
0451526929	Hamlet	Drama	William Shakespeare	1599	
1603033785	Othello	Drama	William Shakespeare	1565	

## Esquema

Livro	isbn	título	categoria	autor	ano
	9580471444	Vidas Secas	Romance	Graciliano Ramos	1938
	958047950X	Agosto	Romance	Rubem Fonseca	1990
	0554253216	Micrographia	Ciências	Robert Hooke	1665
	0198017445	Divina Comédia	Poesia	Dante Alighieri	1308
	0559274289	Le Operae	Ciências	Galileo Galilei	1611
	0451526929	Hamlet	Drama	William Shakespeare	1599
	1603033785	Othello	Drama	William Shakespeare	1565

## Esquema

- Esquema de uma Relação:
  - Denotado por  $R(A_1, A_2, \dots, A_n)$
  - $R \Leftrightarrow$  nome da relação
  - $A_1, A_2, \dots, A_n \Leftrightarrow$  atributos da relação
- Exemplo:
  - LIVRO (ISBN, Título, Categoria, Autor, Ano)
  - Nome da relação: LIVRO
  - Atributos: ISBN, Título, Categoria, Autor, Ano
- Nome do Atributo
  - Indica o significado dos valores do atributo
  - Designa o papel realizado por um domínio na relação

## Nome do Atributo

## Domínio do Atributo

- Determina os valores possíveis para um atributo
- Conjunto de valores atômicos
- Representação:  $D(A_x)$  – domínio do atributo  $A_x$
- Um domínio tem uma definição lógica
  - $D(\text{ISBN})$ : identificador de até 13 dígitos
  - Um domínio está associado a um tipo de dados
    - $D(\text{Título})$ : string de até 100 caracteres
    - $D(\text{Ano})$ : inteiro de 4 dígitos

## Relação ou Estado da Relação

- Uma relação  $r$  de um esquema  $R(A_1, A_2, \dots, A_n)$ 
  - conjunto de tuplas  $r = \{t_1, t_2, \dots, t_n\}$  em que
    - $t_i$  lista ordenada de valores  $t = <V_1, V_2, \dots, V_n>$  em que
      - $V_i$  é um elemento de  $\text{dom}(A_i)$  ou um valor nulo
      - $t[A]$  ou  $t.A_i \rightarrow$  valor do atributo  $i$
      - $t[i] \rightarrow$  notação alternativa (posicional) (Elmasri, 2010)

## Valor Nulo

- Tuplas podem conter o valor nulo (NULL) em atributos
- Nulo indica valor:
  - desconhecido
  - não disponível
  - indefinido (não se aplica à tupla)

(Elmasri, 2010)

## Esquema + Domínios

- Notação Usual:
  - nome da relação e atributos + tipos
  - Exemplo:
    - LIVRO (ISBN: string, Título: string, Categoria: string, Autor: string, Ano: integer)

## Tupla

- Esquema relação
  - LIVRO (ISBN, Título, Categoria, Autor, Ano)
- Tupla
  - $t = <9580471444, Vidas Secas, Romance, Graciliano Ramos, 1938>$
  - $t[\text{Categoria}] = <\text{Romance}>$
  - $t[\text{Ano}] = <1938>$
  - $t[2] = <\text{Vidas Secas}>$

## Relação (matemática)

- Construído a partir do conceito de **conjuntos matemáticos**
- Fundamentação matemática é importante para definição de linguagens de consulta e para otimização de processamento

## Relação (matemática)

- Considerando os conjuntos  $S_1, S_2, \dots, S_n$  (não necessariamente distintos)

• R é uma relação destes n conjuntos se:

- for uma relação de n tuplas em que:

- primeiro elemento for de  $S_1$
  - segundo elemento for de  $S_2$

...

- R é um subconjunto do produto

cartesiano

$S_1 \times S_2 \times \dots \times S_n$  (Codd, 1970)

## Sumário dos Conceitos

Termo Informal	Termo Formal
Tabela	Relação
Cabeçalho da Coluna	Attributo
Todos os valores possíveis de uma Coluna	Dominio
Linha	Tupla
Definição da Tabela	Esquema da Relação
Tabela populada	Estado da Relação

(Elmasri, 2007)

## Relação (matemática)

- Uma relação é um conjunto, portanto não admite valores repetidos de tuplas

## Restrições de Integridade

- Devem ser verdadeiras para cada instância do banco de dados
- Restrições:
  - de domínio
  - de chave
  - de integridade de entidades
  - de integridade referencial

## Restrições de Domínio

- Valores dos atributos devem ser atômicos
- Valor do atributo:
  - tem que ser do domínio do atributo
  - pode ser nulo (se permitido pelo atributo)

## Restrições de Chave

- Chave: identifica tuplas e é usada para estabelecer relacionamentos entre tabelas
- Superchave
  - conjunto de atributos de uma relação
  - não existem duas tuplas em uma relação com a mesma superchave
- Chave
  - Superchave que atende à “condição mínima”:
    - Se qualquer atributo for removido deixa de ser superchave

# Superchave & Chave

Exemplo

# Superchave & Chave

ISBN	Titulo	Autor	Ano	Categoria
9580471444	Vidas Secas	Graciliano Ramos	1938	Romance
958047950X	Agosto	Rubem Fonseca	1990	Romance
0554253216	Micrographia	Robert Hooke	1665	Ciências
0195087445	Divina Comédia	Dante Alighieri	1308	Poesia
0559274289	Le Opere	Galileu Galilei	1811	Ciências
0451526929	Hamlet	William Shakespeare	1599	Drama
1603033785	Othello	William Shakespeare	1565	Drama

Superchave?

# Superchave & Chave

Exemplo

# Superchave & Chave

ISBN	Titulo	Autor	Ano	Categoria
9580471444	Vidas Secas	Graciliano Ramos	1938	Romance
958047950X	Agosto	Rubem Fonseca	1990	Romance
0554253216	Micrographia	Robert Hooke	1665	Ciências
0195087445	Divina Comédia	Dante Alighieri	1308	Poesia
0559274289	Le Opere	Galileu Galilei	1811	Ciências
0451526929	Hamlet	William Shakespeare	1599	Drama
1603033785	Othello	William Shakespeare	1565	Drama

Superchave?

# Superchave & Chave

Exemplo

# Superchave & Chave

ISBN	Titulo	Autor	Ano	Categoria
9580471444	Vidas Secas	Graciliano Ramos	1938	Romance
958047950X	Agosto	Rubem Fonseca	1990	Romance
0554253216	Micrographia	Robert Hooke	1665	Ciências
0195087445	Divina Comédia	Dante Alighieri	1308	Poesia
0559274289	Le Opere	Galileu Galilei	1811	Ciências
0451526929	Hamlet	William Shakespeare	1599	Drama
1603033785	Othello	William Shakespeare	1565	Drama

Superchave?

Exemplo

# Superchave & Chave

ISBN	Titulo	Autor	Ano	Categoria
9580471444	Vidas Secas	Graciliano Ramos	1938	Romance
958047950X	Agosto	Rubem Fonseca	1990	Romance
0554253216	Micrographia	Robert Hooke	1665	Ciências
0195087445	Divina Comédia	Dante Alighieri	1308	Poesia
0559274289	Le Opere	Galileu Galilei	1811	Ciências
0451526929	Hamlet	William Shakespeare	1599	Drama
1603033785	Othello	William Shakespeare	1565	Drama

Superchave?

# Superchave & Chave

Exemplo

# Superchave & Chave

ISBN	Titulo	Autor	Ano	Categoria
9580471444	Vidas Secas	Graciliano Ramos	1938	Romance
958047950X	Agosto	Rubem Fonseca	1990	Romance
0554253216	Micrographia	Robert Hooke	1665	Ciências
0195087445	Divina Comédia	Dante Alighieri	1308	Poesia
0559274289	Le Opere	Galileu Galilei	1811	Ciências
0451526929	Hamlet	William Shakespeare	1599	Drama
1603033785	Othello	William Shakespeare	1565	Drama

Superchave?

# Superchave & Chave

Exemplo

# Superchave & Chave

ISBN	Titulo	Autor	Ano	Categoria
9580471444	Vidas Secas	Graciliano Ramos	1938	Romance
958047950X	Agosto	Rubem Fonseca	1990	Romance
0554253216	Micrographia	Robert Hooke	1665	Ciências
0195087445	Divina Comédia	Dante Alighieri	1308	Poesia
0559274289	Le Opere	Galileu Galilei	1811	Ciências
0451526929	Hamlet	William Shakespeare	1599	Drama
1603033785	Othello	William Shakespeare	1565	Drama

Superchave?

# Exemplo Superchave & Chave

## Exemplo Superchave & Chave

ISBN	Título	Autor	Ano	Categoria
9580471444	Vidas Secas	Graciliano Ramos	1938	Romance
958047950X	Agosto	Rubem Fonseca	1990	Romance
0554253216	Micrographia	Robert Hooke	1665	Ciências
0195087445	Divina Comédia	Dante Alighieri	1308	Poesia
0559274289	Le Opere	Galileu Galilei	1811	Ciências
0451526929	Hamlet	William Shakespeare	1599	Drama
1603033785	Othello	William Shakespeare	1565	Drama

Chave

# Exemplo Superchave & Chave

ISBN	Título	Autor	Ano	Categoria
9580471444	Vidas Secas	Graciliano Ramos	1938	Romance
958047950X	Agosto	Rubem Fonseca	1990	Romance
0554253216	Micrographia	Robert Hooke	1665	Ciências
0195087445	Divina Comédia	Dante Alighieri	1308	Poesia
0559274289	Le Opere	Galileu Galilei	1811	Ciências
0451526929	Hamlet	William Shakespeare	1599	Drama
1603033785	Othello	William Shakespeare	1565	Drama

Chave

# Chave Primária

- Chave cujos valores distinguem uma tupla das demais dentro de uma relação
- Identifica a tupla de forma única
- Usada como referência a partir de outra tupla
- Atributos da chave primária recebem sublinhado:

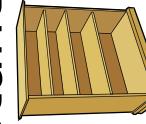
- LIVRO (ISBN, Título, Categoria, Autor, Ano)

# Chave Primária Livro



LIVRO (ISBN, Título, Autor, Ano, Categoria)

# Chave Primária Estante



Estante (Código, Tamanho)

ISBN	Título	Autor	Ano	Categoria
9580471444	Vidas Secas	Graciliano Ramos	1938	Romance
958047950X	Agosto	Rubem Fonseca	1990	Romance
0554253216	Micrographia	Robert Hooke	1665	Ciências

ESTANTE

Código	Tamanho
1A	simples
2A	dúplo
1B	simples
2B	dúplo

# Exercício 1

parte 1

- Liste as superchaves e chaves da seguinte relação:

col1	col2	col3
A	10	F
A	15	F
D	15	M
B	5	F
A	5	M
B	10	M

# Agradecimentos

- Diversos slides foram baseados no curso de BD do Prof. André Santanaché (UNICAMP)
  - Site: <http://www.ic.unicamp.br/~santanache>
  - Canal YouTube:  
<https://www.youtube.com/santanache>

# Exercício 1

parte 2

- As informações contidas na relação em questão são suficientes para determinar a chave primária? Justifique.

# Exercício 1

parte 3

- Uma relação sempre terá uma chave? Justifique.

# Integridade de Entidade

## BD Relacional

- Banco de Dados Relacional
  - conjunto de esquemas  $S = \{R_1, R_2, \dots, R_n\}$
  - conjunto de restrições de integridade RI
- Estado ou instância do Banco de Dados Relacional
  - conjunto de estados da relação DB =  $\{r_1, r_2, \dots, r_n\}$
  - $r_i$  é instância de  $R_i$
  - Estados de  $r_i$  satisfazem restrições de integridade (Elmasri, 2010)

# BD Relacional

## Banco de Dados Relacional

- Banco de Dados Relacional
  - conjunto de esquemas  $S = \{R_1, R_2, \dots, R_n\}$
  - conjunto de restrições de integridade RI
- Estado ou instância do Banco de Dados Relacional
  - conjunto de estados da relação DB =  $\{r_1, r_2, \dots, r_n\}$
  - $r_i$  é instância de  $R_i$
  - Estados de  $r_i$  satisfazem restrições de integridade (Elmasri, 2010)

## Chave Estrangeira

- Conjunto de campos em uma relação que é usado para fazer referência à chave primária da segunda relação
- Valor de cada chave estrangeira deve corresponder à chave primária existente da relação referenciada
- Funciona como um ‘ponteiro lógico’  
(Ramakrishman, 2003)

## Pessoa ocupa Armário

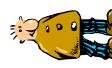


PESSOA (Código, Nome, Telefone)

ARMÁRIO (Código, Tamanho, Ocupante)

PESSOA		ARMÁRIO		
Código	Nome	Telefone	Código	Tamanho
1525	Astríbal	5432-1098	1A	simples
1637	Doriana	9876-5432	2A	dúplo
1701	Quinicas	8765-4321	1B	simples
2042	Melissa	7654-3210	2B	dúplo
2111	Horácio	6543-2109		

## Pessoa ocupa Armário



PESSOA (Código, Nome, Telefone)

ARMÁRIO (Código, Tamanho)

OCUPA (CodPessoa, CodArmário, Data, Hora)

PESSOA		OCUPA		
Código	Nome	Telefone	CodPessoa	CodArmário
1525	Astríbal	5432-1098	1637	1A
			2111	2B
1637	Doriana	9876-5432		
1701	Quinicas	8765-4321		
2042	Melissa	7654-3210		
2111	Horácio	6543-2109		



Corrida

Placa

Data Pedido

ClId	Placa	Data Pedido
1755	DAE6534	15/02/2003
1982	JDM8776	18/02/2003



Carro



Motorista

## Exemplo: Táxis

### Táxi

Placa	Marca	Modelo	AnoFab
DAE6534	Ford	Fiesta	1999
DKL4598	Volkswagen	Gol	2001
DKL7878	Ford	Fiesta	2001
JDM8776	Volkswagen	Santana	2002
JM3692	Chevrolet	Corsa	1999



## Exemplo: Táxis

## Exemplo: Táxis

Corrida		
CId	Placa	Data Pedido
1755	DAE6534	15/02/2003
1982	JDM8776	18/02/2003

- Chaves Estrangeiras
- Tupla deve referenciar tupla existente, utilizando chaves válidas
- SGDB deve garantir a consistência das referências
- SGDB deve garantir a integridade dos dados na inserção, exclusão e alteração de dados

## Integridade Referencial

## Exemplo: Táxis

Corrida

CId	Placa	Data Pedido
1755	DAE6534	15/02/2003
1982	JDM8776	18/02/2003

## Exercícios (2)

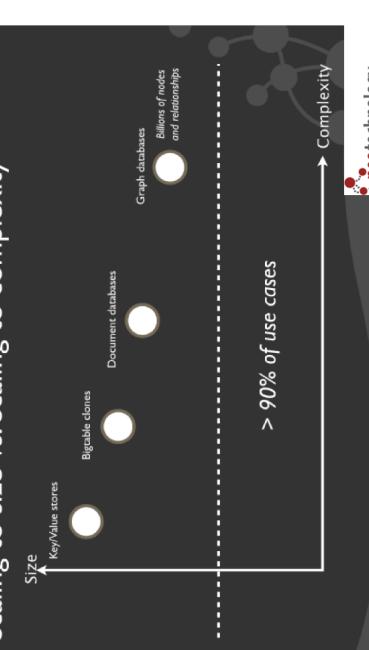
- Inserção de Corrida: CId e Placa devem existir
- Exclusão de Táxi ou Cliente: não é permitida se existirem corridas que fazem referência

• Faça o modelo relacional do seguinte diagrama:



- Desenvolva versões com uma e duas tabelas
- Adicione atributos importantes e defina chaves primárias
- Mostre dados de exemplo

## Próximas aulas



## Próximas aulas

## Conclusão

- Modelo Relacional dominou a área de banco de dados por décadas
- Importante em diversas aplicações
- Restrições de integridade garantem coesão dos dados
- Algumas aplicações demandam outros modelos

## Perguntas?

## Referências

- Codd, Edgar Frank (1970) **A relational model of data for large shared data banks**. Communications ACM 13(6), 377-387.
- Elmasri, Ramez; Navathe, Shamkant B. (2010) **Sistemas de Banco de Dados**. Pearson, 6<sup>a</sup> edição em português.
- Ramakrishnan, Raghuram; Gehrke, Johannes (2003) **Database Management Systems**. McGraw-Hill, 3<sup>rd</sup> edition.

# Modelagem

## Respostas dos Exercícios

### Exercício

Uma indústria farmacêutica quer desenvolver um banco de dados para registrar os medicamentos que ela produz, bem como os vírus tratados por estes medicamentos.



### Exercício

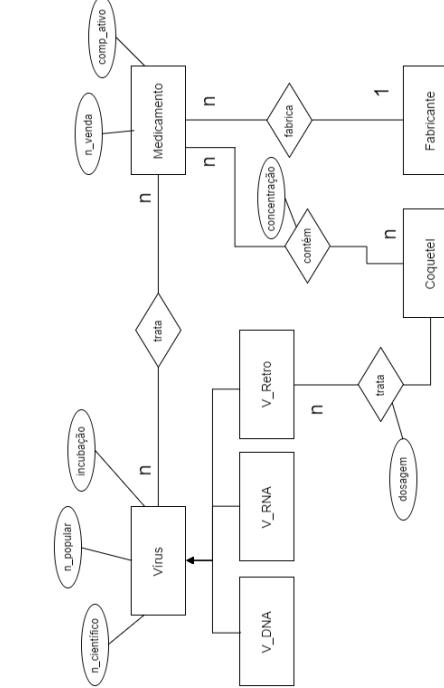
parte 2

- a) Considere que um dado medicamento pode tratar vários vírus e um vírus pode ser tratado por vários medicamentos.  
b) Medicamentos são fabricados por empresas que possuem nome e CNPJ. Uma empresa pode fabricar vários medicamentos, mas um medicamento é fabricado por uma única empresa.

### Exercício

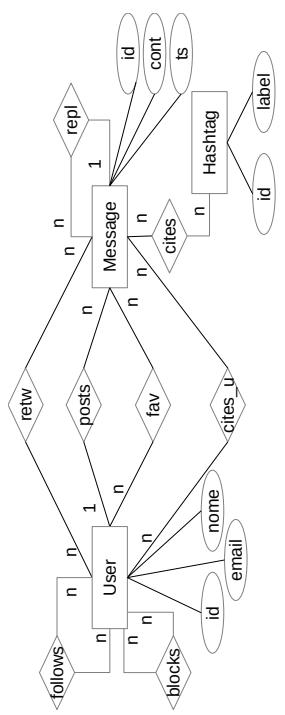
parte 3

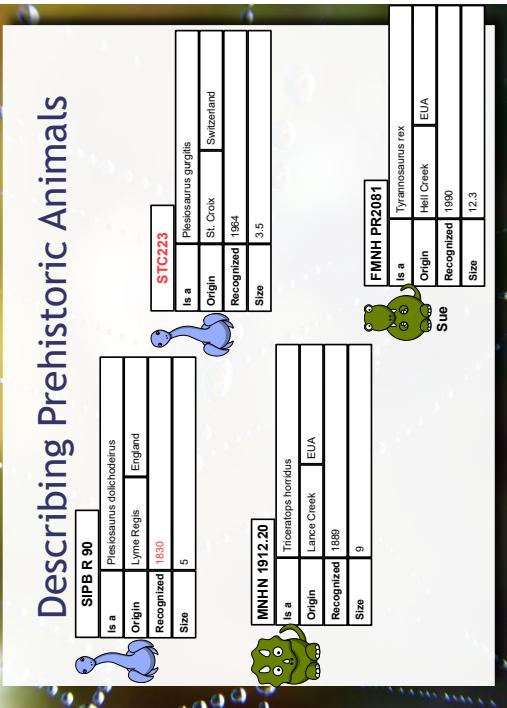
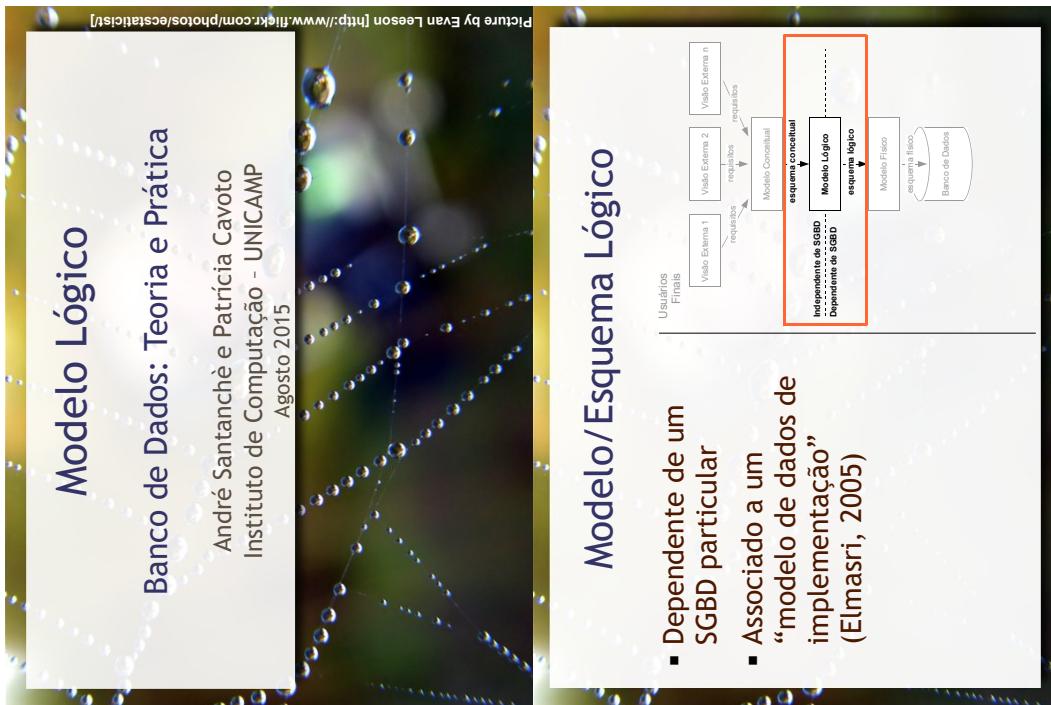
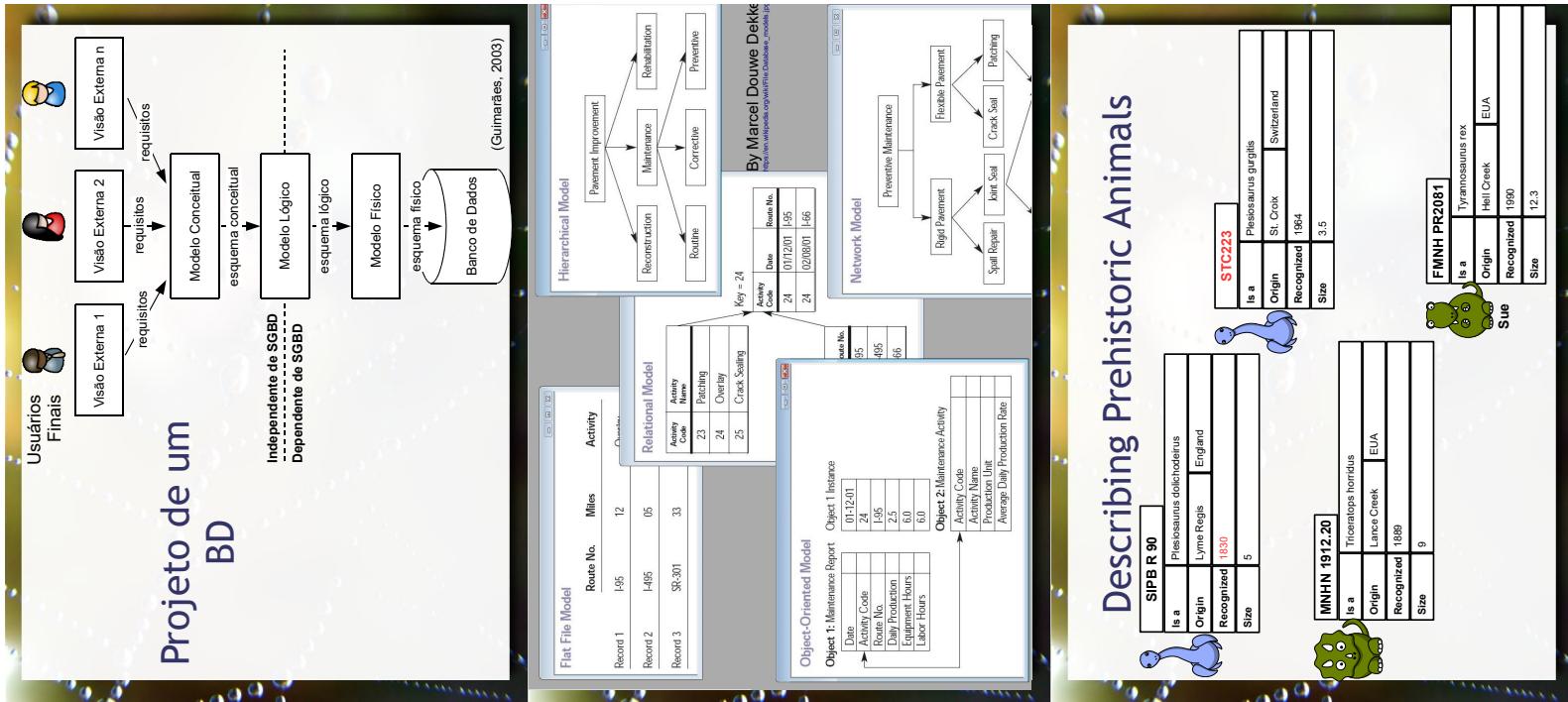
- Vírus podem ser classificados em diversas categorias, mas considere apenas Vírus com DNA, Vírus com RNA e Retrovírus. Retrovírus são tratados com coquetéis de medicamentos. Um coquetel é composto por vários medicamentos, cada um em uma concentração específica. Um coquetel tem uma dosagem específica para o tratamento de um dado Retrovírus.



### Exercício

Modele em um diagrama ER o banco de dados para o serviço de microbiologia e rede social Twitter. O serviço permite o cadastro de usuários que devem informar nome, e-mail, e um identificador único para a conta. Usuários podem seguir (follow) outros usuários. Usuários também podem bloquear outros usuários. Usuários postam mensagens (tweets). Mensagens possuem identificador, conteúdo (texto) e timestamp do momento de postagem (estilo Unix time, ou seja, representa o número total de segundos decorridos desde 0/0/1970). Mensagens podem mencionar outros usuários (conhecidos como @) e mencionar tópicos (conhecidos como hashtags ou #). Tópicos possuem id e rótulo (e.g., "banco de dados"). Usuários e tópicos são automaticamente extraídos do texto do post no momento de envio com base nas strings que começam por @ ou #. Por exemplo, a mensagem "Hoje é dia de prova de #banco\_de\_dados! @UFRJ @Computacao" referencia o tópico "banco\_de\_dados" e os usuários "UFRJ" e "Computacao". Mensagens podem ser repostadas (re-tweets), marcadas como favoritas, ou respondidas por qualquer usuário. Mensagens de resposta são novas mensagens que fazem referência à mensagem original. Mensagens repostadas são idênticas às originais, mas fazem referência à mensagem original. Mensagens repostadas são idênticas chaves primárias e cardinalidades. Minimize a redundância de informação.





## Describing Prehistoric Animals

## Modelo Relacional

**Cliente (C)**

CId	Nome	CPF
1532	Asdrubal	448.754.253-65
1755	Dorian	567.387.387-44
1780	Quinca	546.373.762-02


**Táxi (TX)**

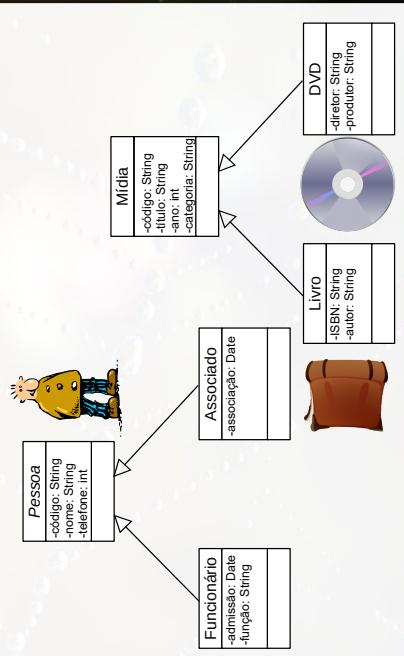
Placa	Marca	Modelo	AnoFab
DAE6534	Ford	Fiesta	1999
DKL4598	Volkswagen	Gol	2001
DKL7878	Ford	Fiesta	2001
JDM8776	Volkswagen	Santana	2002
JJM3692	Chevrolet	Corsa	1999

**Corrida (R1)**

CId	Placa	Data Pedido
1755	DAE6534	15/02/2003
1982	JDM8776	18/02/2003

Modelo original  
por prof. Geovane  
Cayres Magalhães

## Modelo Orientado a Objetos



## Modelo Relacional

CId	Nome	CPF
1532	Asdrubal	448.754.253-65
1755	Dorian	567.387.387-44
1780	Quinca	546.373.762-02


**Táxi (TX)**

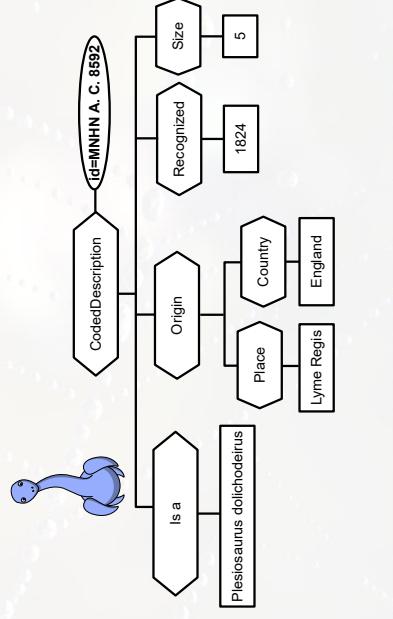
Placa	Marca	Modelo	AnoFab
DAE6534	Ford	Fiesta	1999
DKL4598	Volkswagen	Gol	2001
DKL7878	Ford	Fiesta	2001
JDM8776	Volkswagen	Santana	2002
JJM3692	Chevrolet	Corsa	1999

**Corrida (R1)**

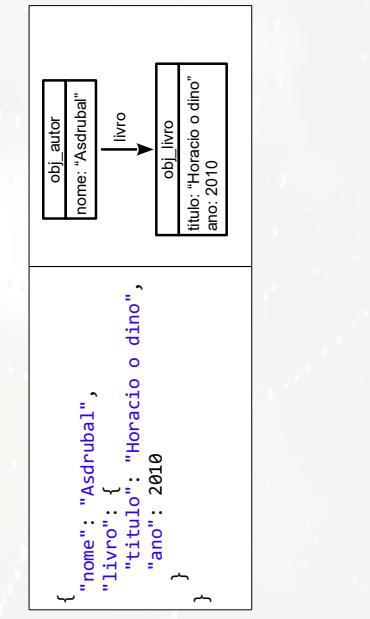
CId	Placa	Data Pedido
1755	DAE6534	15/02/2003
1982	JDM8776	18/02/2003

Modelo original  
por prof. Geovane  
Cayres Magalhães

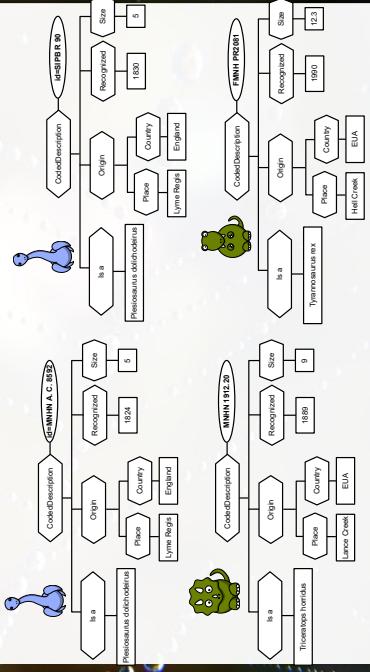
## Modelo Hierárquico (Documentos/XML)

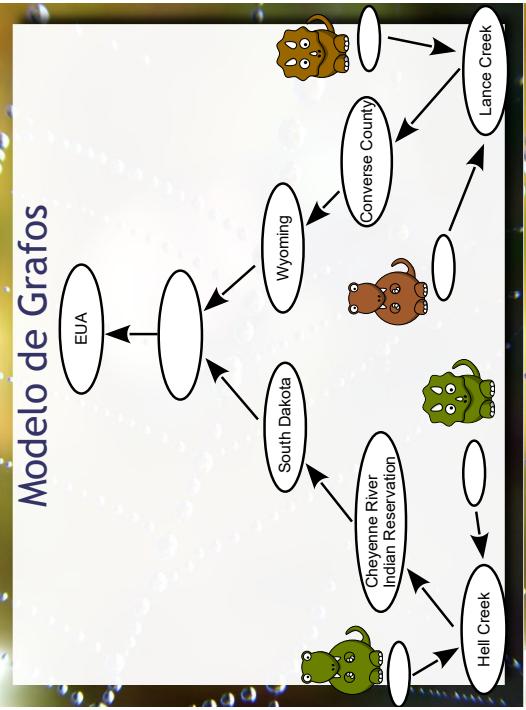


## Modelo Hierárquico (Documentos/XML)



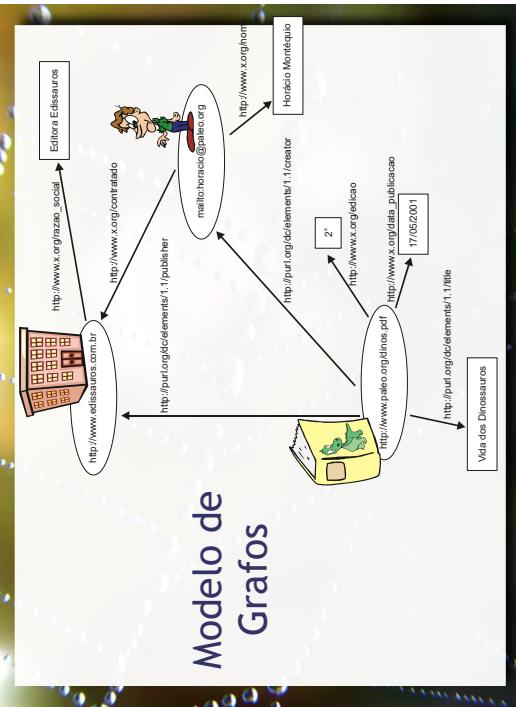
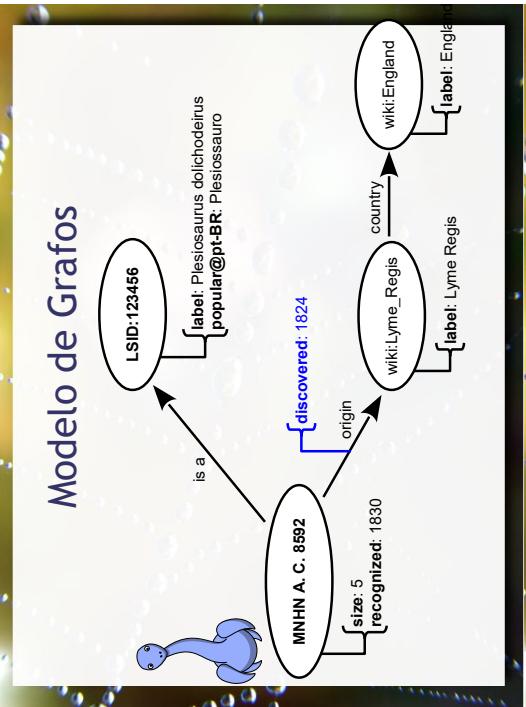
## Modelo Hierárquico (Documentos/XML)





### Agradecimentos

- Luiz Celso Gomes Jr (professor desta disciplina em 2014) pela contribuição na disciplina e nos slides.



### Referências

- Heuser, Carlos Alberto (2004) **Projeto de Banco de Dados**. Editora Sagra Luzzato, 5ª edição.
- Ramakrishnan, Raghunath; Gehrke, Johannes (2003) **Database Management Systems**. McGraw-Hill, 3rd edition.

## License

- These slides are shared under a Creative Commons License.  
Under the following conditions: Attribution, Noncommercial  
and Share Alike.
- See further details about this Creative Commons license at:  
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

André Santanchè

<http://www.íc.unicamp.br/~Santanche>

# O Modelo Relacional

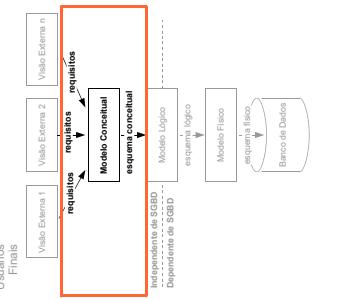
## Agenda

- Recapitulação
- Modelo Relacional
- Restrições de Integridade
- Exercícios
- Além do modelo relacional

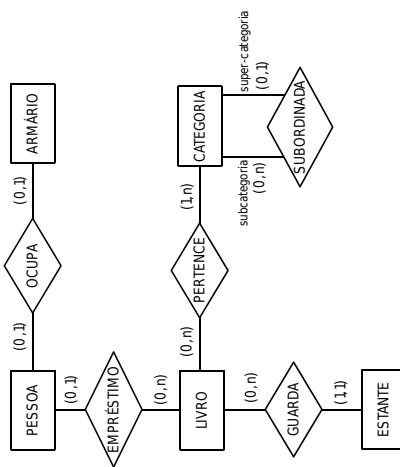
10010110 Banco de Dados 1010101000  
101000101011100011111010101010100  
1001001010110101111001111101010101  
01010001010111011000111010101010111  
1101111001111111010101010001011111  
10101110110001110101100011011111111  
10111010101010101110000101010101111  
0110101111001111111010101010101000  
0101 Luiz Celso Gomes-Jr 0101010011011  
gomesjir@da.inf.ct.utfpr.edu.br 101010101  
10001010111001110001110001110001111

## Modelo/Esquema Conceitual

- Descreve estrutura do Banco de Dados
  - entidades, tipos de dados, relações, restrições etc.
- Independente de implementação em SGBD
  - oculta detalhes de armazenamento físico

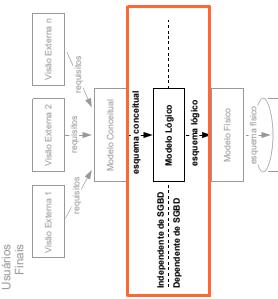


## Modelo ER



## Modelo/Esquema Lógico

- Dependente de um SGBD particular
- Associado a um “modelo de dados de implementação” (Elmasri, 2005)



## Modelo Relacional

- Proposto por E. F. Codd em 1970 no artigo:
  - “A Relational Model for Large Shared Data Banks”
- Resposta a um cenário complexo de incompatibilidade de modelos e sistemas
  - Independência da estrutura interna
    - “Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed [...]” (Codd, 1970)

# Modelo Relacional

## Larry Ellison, ORACLE

- Modelo mais amplamente utilizado por SGBDs
- Maiores empresas de informática oferecem soluções: IBM, Microsoft, Oracle, SAP
- Grandes projetos Open Source: MySQL, PostgreSQL, SQLite
- De celulares à data centers
- Mercado de U\$24bi (2012)



## Relação

- Banco de Dados Relacional contém um conjunto de **Relações**

- **Relação** – informalmente pode ser visualizada como uma tabela

Livro	relação	ISBN	Título	Categoria	Autor	Ano
		9880471444	Vidas Secas	Romance	Graciliano Ramos	1938
		988047850X	Agosto	Romance	Rubem Fonseca	1990
		0554253216	Micrografia	Ciências	Robert Hooke	1665
		0195087445	Divina Comédia	Poesia	Dante Alighieri	1308
		0559274289	Le Operæ	Ciências	Galileu Galilei	1611
		0451526929	Hamlet	Drama	William Shakespeare	1599
		1603033785	Othello	Drama	William Shakespeare	1565

## Relação (Tabela)

- Relação = conjunto não ordenado de tuplas (linhas)
- Não existem tuplas duplicadas

Livro	relação	ISBN	Título	Categoria	Autor	Ano
		9880471444	Vidas Secas	Romance	Graciliano Ramos	1938
		988047850X	Agosto	Romance	Rubem Fonseca	1990
		0554253216	Micrografia	Ciências	Robert Hooke	1665
		0195087445	Divina Comédia	Poesia	Dante Alighieri	1308
		0559274289	Le Operæ	Ciências	Galileu Galilei	1611
		0451526929	Hamlet	Drama	William Shakespeare	1599
		1603033785	Othello	Drama	William Shakespeare	1565

## Relação (Tabela)

- Tupla = conjunto ordenado de atributos
- Valores de atributos são atômicos e monovalorados

Livro	relação	ISBN	Título	Categoria	Autor	Ano
		9880471444	Vidas Secas	Romance	Graciliano Ramos	1938
		988047850X	Agosto	Romance	Rubem Fonseca	1990
		0554253216	Micrografia	Ciências	Robert Hooke	1665
		0195087445	Divina Comédia	Poesia	Dante Alighieri	1308
		0559274289	Le Operæ	Ciências	Galileu Galilei	1611
		0451526929	Hamlet	Drama	William Shakespeare	1599
		1603033785	Othello	Drama	William Shakespeare	1565

Exercício de aquecimento						
▪ Estime os dados abaixo pensando em uma grande empresa como a Petrobras:						
a) nº de tabelas de uma aplicação típica (média)						
b) nº de tabelas de uma aplicação grande e complexa						
c) nº total de tabelas distintas, considerando todas as aplicações						
d) média de nº de atributos por tabela						
e) número de atributos das "maiores" tabelas						
f) nº de DBAs envolvidos						
g) nº de administradores de dados envolvidos						

Livro	atributos	ISBN	Título	Categoria	Autor	Ano
		9880471444	Vidas Secas	Romance	Graciliano Ramos	1938
		988047850X	Agosto	Romance	Rubem Fonseca	1990
		0554253216	Micrografia	Ciências	Robert Hooke	1665
		0195087445	Divina Comédia	Poesia	Dante Alighieri	1308
		0559274289	Le Operæ	Ciências	Galileu Galilei	1611
		0451526929	Hamlet	Drama	William Shakespeare	1599
		1603033785	Othello	Drama	William Shakespeare	1565

## Exercício de aquecimento

- Estime os dados abaixo pensando em uma grande empresa como a Petrobras:
  - nº de tabelas de uma aplicação típica (média): **30 tabelas**
  - nº de tabelas de uma aplicação grande e complexa: **500 tabelas (quinhentas!)**
  - total de tabelas distintas, considerando todas as aplicações: **10.000 tabelas**
  - média de nº de atributos por tabela: **10 atributos**
  - número de atributos das "maiores" tabelas: **80 atributos**

## Exercício de aquecimento

- Estime os dados abaixo pensando em uma grande empresa como a Petrobras:
  - nº de DBAs envolvidos: cerca de 15 (parte física, replicação, backup, controle de acesso, performance, servidores - não contando as pessoas de infraestrutura que mantêm os servidores em si)
  - nº de administradores de dados envolvidos: cerca de 40 (envolvidos na elaboração e manutenção dos modelos e esquemas, manutenção de metadados, elaboração de scripts, procedures, views; dominam os assuntos / negócio)

## Relação (Tabela)

- Nome dos atributos – título das colunas

Livro

ISBN	Titulo	Categoria	Autor	Ano	nome dos atributos
9580471444	Vidas Secas	Romance	Graciliano Ramos	1938	
958047950X	Agosto	Romance	Rubem Fonseca	1990	
0554253216	Micrographia	Ciências	Robert Hooke	1665	
0198017445	Divina Comédia	Poesia	Dante Alighieri	1308	
0559274289	Le Operae	Ciências	Galileo Galilei	1611	
0451526929	Hamlet	Drama	William Shakespeare	1599	
1603033785	Othello	Drama	William Shakespeare	1565	

## Esquema

Livro	isbn	título	categoria	autor	ano
	9580471444	Vidas Secas	Romance	Graciliano Ramos	1938
	958047950X	Agosto	Romance	Rubem Fonseca	1990
	0554253216	Micrographia	Ciências	Robert Hooke	1665
	0198017445	Divina Comédia	Poesia	Dante Alighieri	1308
	0559274289	Le Operae	Ciências	Galileo Galilei	1611
	0451526929	Hamlet	Drama	William Shakespeare	1599
	1603033785	Othello	Drama	William Shakespeare	1565

## Esquema

- Esquema de uma Relação:
  - Denotado por  $R(A_1, A_2, \dots, A_n)$
  - $R \Leftrightarrow$  nome da relação
  - $A_1, A_2, \dots, A_n \Leftrightarrow$  atributos da relação
- Exemplo:
  - LIVRO (ISBN, Título, Categoria, Autor, Ano)
  - Nome da relação: LIVRO
  - Atributos: ISBN, Título, Categoria, Autor, Ano
- Nome do Atributo
  - Indica o significado dos valores do atributo
  - Designa o papel realizado por um domínio na relação

## Nome do Atributo

## Domínio do Atributo

- Determina os valores possíveis para um atributo
- Conjunto de valores atômicos
- Representação:  $D(A_x)$  – domínio do atributo  $A_x$
- Um domínio tem uma definição lógica
  - $D(\text{ISBN})$ : identificador de até 13 dígitos
  - Um domínio está associado a um tipo de dados
    - $D(\text{Título})$ : string de até 100 caracteres
    - $D(\text{Ano})$ : inteiro de 4 dígitos

## Relação ou Estado da Relação

- Uma relação  $r$  de um esquema  $R(A_1, A_2, \dots, A_n)$ 
  - conjunto de tuplas  $r = \{t_1, t_2, \dots, t_n\}$  em que
    - $t_i$  lista ordenada de valores  $t = <V_1, V_2, \dots, V_n>$  em que
      - $V_i$  é um elemento de  $\text{dom}(A_i)$  ou um valor nulo
      - $t[A]$  ou  $t.A_i \rightarrow$  valor do atributo  $i$
      - $t[i] \rightarrow$  notação alternativa (posicional) (Elmasri, 2010)

## Valor Nulo

- Tuplas podem conter o valor nulo (NULL) em atributos
- Nulo indica valor:
  - desconhecido
  - não disponível
  - indefinido (não se aplica à tupla)

(Elmasri, 2010)

## Esquema + Domínios

- Notação Usual:
  - nome da relação e atributos + tipos
  - Exemplo:
    - LIVRO (ISBN: string, Título: string, Categoria: string, Autor: string, Ano: integer)

## Tupla

- Esquema relação
  - LIVRO (ISBN, Título, Categoria, Autor, Ano)
- Tupla
  - $t = <9580471444, Vidas Secas, Romance, Graciliano Ramos, 1938>$
  - $t[\text{Categoria}] = <\text{Romance}>$
  - $t[\text{Ano}] = <1938>$
  - $t[2] = <\text{Vidas Secas}>$

## Relação (matemática)

- Construído a partir do conceito de **conjuntos matemáticos**
- Fundamentação matemática é importante para definição de linguagens de consulta e para otimização de processamento

## Relação (matemática)

- Considerando os conjuntos  $S_1, S_2, \dots, S_n$  (não necessariamente distintos)

• R é uma relação destes n conjuntos se:

- for uma relação de n tuplas em que:
  - primeiro elemento for de  $S_1$
  - segundo elemento for de  $S_2$
  - ...
- R é um subconjunto do produto cartesiano  $S_1 \times S_2 \times \dots \times S_n$  (Codd, 1970)

## Sumário dos Conceitos

Termo Informal	Termo Formal
Tabela	Relação
Cabeçalho da Coluna	Attributo
Todos os valores possíveis de uma Coluna	Dominio
Linha	Tupla
Definição da Tabela	Esquema da Relação
Tabela populada	Estado da Relação

(Elmasri, 2007)

## Relação (matemática)

- Uma relação é um conjunto, portanto não admite valores repetidos de tuplas

## Restrições de Integridade

- Devem ser verdadeiras para cada instância do banco de dados
- Restrições:
  - de domínio
  - de chave
  - de integridade de entidades
  - de integridade referencial

## Restrições de Domínio

- Valores dos atributos devem ser atômicos
- Valor do atributo:
  - tem que ser do domínio do atributo
  - pode ser nulo (se permitido pelo atributo)

## Restrições de Chave

- Chave: identifica tuplas e é usada para estabelecer relacionamentos entre tabelas
- Superchave
  - conjunto de atributos de uma relação
  - não existem duas tuplas em uma relação com a mesma superchave
- Chave
  - Superchave que atende à “condição mínima”:
    - Se qualquer atributo for removido deixa de ser superchave

# Superchave & Chave

Exemplo  
Superchave & Chave

ISBN	Titulo	Autor	Ano	Categoria
9580471444	Vidas Secas	Graciliano Ramos	1938	Romance
958047950X	Agosto	Rubem Fonseca	1990	Romance
0554253216	Micrographia	Robert Hooke	1665	Ciências
0195087445	Divina Comédia	Dante Alighieri	1308	Poesia
0559274289	Le Opere	Galileu Galilei	1811	Ciências
0451526929	Hamlet	William Shakespeare	1599	Drama
1603033785	Othello	William Shakespeare	1565	Drama

Superchave?

# Superchave & Chave

Exemplo  
Superchave & Chave

ISBN	Titulo	Autor	Ano	Categoria
9580471444	Vidas Secas	Graciliano Ramos	1938	Romance
958047950X	Agosto	Rubem Fonseca	1990	Romance
0554253216	Micrographia	Robert Hooke	1665	Ciências
0195087445	Divina Comédia	Dante Alighieri	1308	Poesia
0559274289	Le Opere	Galileu Galilei	1811	Ciências
0451526929	Hamlet	William Shakespeare	1599	Drama
1603033785	Othello	William Shakespeare	1565	Drama

Superchave?

# Superchave & Chave

Exemplo  
Superchave & Chave

ISBN	Titulo	Autor	Ano	Categoria
9580471444	Vidas Secas	Graciliano Ramos	1938	Romance
958047950X	Agosto	Rubem Fonseca	1990	Romance
0554253216	Micrographia	Robert Hooke	1665	Ciências
0195087445	Divina Comédia	Dante Alighieri	1308	Poesia
0559274289	Le Opere	Galileu Galilei	1811	Ciências
0451526929	Hamlet	William Shakespeare	1599	Drama
1603033785	Othello	William Shakespeare	1565	Drama

Superchave?

# Superchave & Chave

Exemplo  
Superchave & Chave

ISBN	Titulo	Autor	Ano	Categoria
9580471444	Vidas Secas	Graciliano Ramos	1938	Romance
958047950X	Agosto	Rubem Fonseca	1990	Romance
0554253216	Micrographia	Robert Hooke	1665	Ciências
0195087445	Divina Comédia	Dante Alighieri	1308	Poesia
0559274289	Le Opere	Galileu Galilei	1811	Ciências
0451526929	Hamlet	William Shakespeare	1599	Drama
1603033785	Othello	William Shakespeare	1565	Drama

Superchave?

# Superchave & Chave

Exemplo  
Superchave & Chave

ISBN	Titulo	Autor	Ano	Categoria
9580471444	Vidas Secas	Graciliano Ramos	1938	Romance
958047950X	Agosto	Rubem Fonseca	1990	Romance
0554253216	Micrographia	Robert Hooke	1665	Ciências
0195087445	Divina Comédia	Dante Alighieri	1308	Poesia
0559274289	Le Opere	Galileu Galilei	1811	Ciências
0451526929	Hamlet	William Shakespeare	1599	Drama
1603033785	Othello	William Shakespeare	1565	Drama

Superchave?

# Superchave & Chave

Exemplo  
Superchave & Chave

ISBN	Titulo	Autor	Ano	Categoria
9580471444	Vidas Secas	Graciliano Ramos	1938	Romance
958047950X	Agosto	Rubem Fonseca	1990	Romance
0554253216	Micrographia	Robert Hooke	1665	Ciências
0195087445	Divina Comédia	Dante Alighieri	1308	Poesia
0559274289	Le Opere	Galileu Galilei	1811	Ciências
0451526929	Hamlet	William Shakespeare	1599	Drama
1603033785	Othello	William Shakespeare	1565	Drama

Superchave?

# Exemplo Superchave & Chave

## Exemplo Superchave & Chave

ISBN	Título	Autor	Ano	Categoria
9580471444	Vidas Secas	Graciliano Ramos	1938	Romance
958047950X	Agosto	Rubem Fonseca	1990	Romance
0554253216	Micrographia	Robert Hooke	1665	Ciências
0195087445	Divina Comédia	Dante Alighieri	1308	Poesia
0559274289	Le Opere	Galileu Galilei	1811	Ciências
0451526929	Hamlet	William Shakespeare	1599	Drama
1603033785	Othello	William Shakespeare	1565	Drama

Chave

# Exemplo Superchave & Chave

ISBN	Título	Autor	Ano	Categoria
9580471444	Vidas Secas	Graciliano Ramos	1938	Romance
958047950X	Agosto	Rubem Fonseca	1990	Romance
0554253216	Micrographia	Robert Hooke	1665	Ciências
0195087445	Divina Comédia	Dante Alighieri	1308	Poesia
0559274289	Le Opere	Galileu Galilei	1811	Ciências
0451526929	Hamlet	William Shakespeare	1599	Drama
1603033785	Othello	William Shakespeare	1565	Drama

Chave

# Chave Primária

- Chave cujos valores distinguem uma tupla das demais dentro de uma relação
- Identifica a tupla de forma única
- Usada como referência a partir de outra tupla
- Atributos da chave primária recebem sublinhado:

- LIVRO (ISBN, Título, Categoria, Autor, Ano)

# Chave Primária Livro



LIVRO (ISBN, Título, Autor, Ano, Categoria)

# Chave Primária Estante



Estante (Código, Tamanho)

ISBN	Título	Autor	Ano	Categoria
9580471444	Vidas Secas	Graciliano Ramos	1938	Romance
958047950X	Agosto	Rubem Fonseca	1990	Romance
0554253216	Micrographia	Robert Hooke	1665	Ciências

ESTANTE	
Código	Tamanho
1A	simples
2A	dúplo
1B	simples
2B	dúplo

# Exercício 1

parte 1

- Liste as superchaves e chaves da seguinte relação:

col1	col2	col3
A	10	F
A	15	F
D	15	M
B	5	F
A	5	M
B	10	M

# Agradecimentos

- Diversos slides foram baseados no curso de BD do Prof. André Santanaché (UNICAMP)
  - Site: <http://www.ic.unicamp.br/~santanache>
  - Canal YouTube:  
<https://www.youtube.com/santanache>

# Exercício 1

parte 2

- As informações contidas na relação em questão são suficientes para determinar a chave primária? Justifique.

# Exercício 1

parte 3

- Uma relação sempre terá uma chave? Justifique.

# Integridade de Entidade

## BD Relacional

- Banco de Dados Relacional
  - conjunto de esquemas  $S = \{R_1, R_2, \dots, R_n\}$
  - conjunto de restrições de integridade RI
- Estado ou instância do Banco de Dados Relacional
  - conjunto de estados da relação DB =  $\{r_1, r_2, \dots, r_n\}$
  - $r_i$  é instância de  $R_i$
  - Estados de  $r_i$  satisfazem restrições de integridade (Elmasri, 2010)

# BD Relacional

## BD Relacional

- Banco de Dados Relacional
  - conjunto de esquemas  $S = \{R_1, R_2, \dots, R_n\}$
  - conjunto de restrições de integridade RI
- Estado ou instância do Banco de Dados Relacional
  - conjunto de estados da relação DB =  $\{r_1, r_2, \dots, r_n\}$
  - $r_i$  é instância de  $R_i$
  - Estados de  $r_i$  satisfazem restrições de integridade (Elmasri, 2010)

## Chave Estrangeira

- Conjunto de campos em uma relação que é usado para fazer referência à chave primária da segunda relação
- Valor de cada chave estrangeira deve corresponder à chave primária existente da relação referenciada
- Funciona como um ‘ponteiro lógico’  
(Ramakrishman, 2003)

## Pessoa ocupa Armário

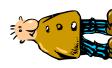


PESSOA (Código, Nome, Telefone)

ARMÁRIO (Código, Tamanho, Ocupante)

PESSOA		ARMÁRIO		
Código	Nome	Telefone	Código	Tamanho
1525	Astríbal	5432-1098	1A	simples
1637	Doriana	9876-5432	2A	dúplo
1701	Quinicas	8765-4321	1B	simples
2042	Melissa	7654-3210	2B	dúplo
2111	Horácio	6543-2109		

## Pessoa ocupa Armário



PESSOA (Código, Nome, Telefone)

ARMÁRIO (Código, Tamanho)

OCCUPA		ARMÁRIO		
Código	Nome	Telefone	CodPessoa	CodArmário
1525	Astríbal	5432-1098	1637	1A
1637	Doriana	9876-5432	2111	2B
1701	Quinicas	8765-4321		
2042	Melissa	7654-3210		
2111	Horácio	6543-2109		

## Exemplo: Táxis

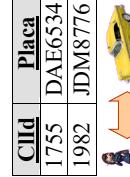
### Táxi

Placa	Marca	Modelo	AnoFab
DAE6534	Ford	Fiesta	1999
DKL4598	Volkswagen	Gol	2001
DKL7878	Ford	Fiesta	2001
JDM8776	Volkswagen	Santana	2002
JM3692	Chevrolet	Corsa	1999



Corrida
ClId
Placa
DataPedido

ClId	Placa	DataPedido
1755	DAE6534	15/02/2003
1982	JDM8776	18/02/2003



## Exemplo: Táxis

## Exemplo: Táxis

Corrida		
CId	Placa	Data Pedido
1755	DAE6534	15/02/2003
1982	JDM8776	18/02/2003

- Chaves Estrangeiras
- Tupla deve referenciar tupla existente, utilizando chaves válidas
- SGDB deve garantir a consistência das referências
- SGDB deve garantir a integridade dos dados na inserção, exclusão e alteração de dados

## Integridade Referencial

## Exemplo: Táxis

Corrida

CId	Placa	Data Pedido
1755	DAE6534	15/02/2003
1982	JDM8776	18/02/2003

## Exercícios (2)

- Inserção de Corrida: CId e Placa devem existir
- Exclusão de Táxi ou Cliente: não é permitida se existirem corridas que fazem referência

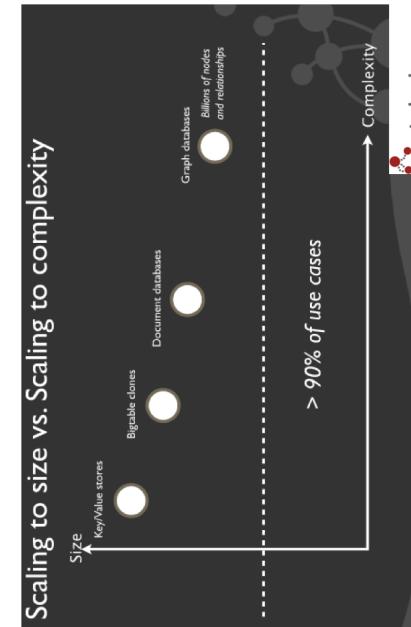
• Faça o modelo relacional do seguinte diagrama:



- Desenvolva versões com uma e duas tabelas
- Adicione atributos importantes e defina chaves primárias
- Mostre dados de exemplo

## Próximas aulas

- Mapeamento ER -> Relacional
- Modelagem/Normalização
- Consultas
- Otimização de consultas
- Controle de transação



## Próximas aulas

Scaling to size vs. Scaling to complexity

## Conclusão

- Modelo Relacional dominou a área de banco de dados por décadas
- Importante em diversas aplicações
- Restrições de integridade garantem coesão dos dados
- Algumas aplicações demandam outros modelos

## Perguntas?

## Referências

- Codd, Edgar Frank (1970) **A relational model of data for large shared data banks**. Communications ACM 13(6), 377-387.
- Elmasri, Ramez; Navathe, Shamkant B. (2010) **Sistemas de Banco de Dados**. Pearson, 6<sup>a</sup> edição em português.
- Ramakrishnan, Raghuram; Gehrke, Johannes (2003) **Database Management Systems**. McGraw-Hill, 3<sup>rd</sup> edition.

## Laudelino Cordeiro Bastos

### Análise e Projeto de Sistemas

#### Introdução

**Prof. Laudelino Cordeiro Bastos**

- ⌚ Formação Acadêmica
  - ▣ Doutor pela POLI-USP, na área de Sistemas de Informação Hospitalar (2002).
  - ▣ Mestre pelo CEFET-PR, na área de Processamento de Imagens (1995).
  - ▣ Engenheiro de Computação pela PUCPR (1991).

- ⌚ Atividades Profissionais
  - ▣ Telepar, Nutron, Equitel-Siemens (Engenheiro de Computação), PUCPR (coordenador do curso de Engenharia de Computação e do PPGTS) e UTFFPR (coordenador do curso de Engenharia de Computação, professor do PPGCA).
    - ▣ SBEB, SBIS, SBC.
    - ▣ 104ª turma na UTFFPR.

• Lattes: <http://lattes.cnpq.br/1231141260610815>

### Por que fazer Análise e Projeto de Sistemas?

#### Exemplo:

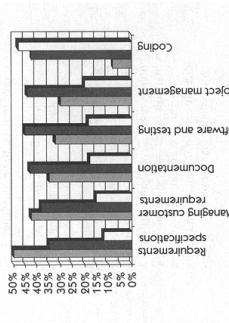
- ⌚ A SAP (Systems, Applications and Products in Data Processing) faturou 14,23 bilhões de euros em 2011.
- ▣ Seu sistema para gestão de negócios, tem mais de 400 milhões de linhas de código.
- ▣ É inviável desenvolver um sistema desse porte sem análise e projeto.

### Qual o maior sistema feito por vocês?

### Por que fazer Análise e Projeto de Sistemas?

- ⌚ Causas Principais de Sucesso ou Falha de Projetos:

- ▣ Um estudo do European Software Process Improvement Training Initiative (ESPTI), de 1995, identifica a importância relativa de vários tipos de problemas de software (LEFFINGWELL; WIDRIG, 2002). As 3800 respostas levaram ao seguinte:



### Análise e Projeto de Sistemas: Ementa

- ⌚ Introdução.
- ⌚ Teoria Geral de Sistemas.
- ⌚ Conceitos de Análise e Projeto de Sistemas.
- ⌚ Paradigmas de Análise e Projeto de Sistemas.
- ⌚ Ferramentas da Análise e Projeto de Sistemas.
- ⌚ Critérios em Projetos de Sistemas.
- ⌚ Estágios e Objetivos do Projeto.
- ⌚ Técnicas de documentação.

## Bibliografia

---

- ➊ PRESSMAN, Roger S. Engenharia de Software - Uma Abordagem Profissional, 7<sup>a</sup> edição, Porto Alegre: AMGH, 2011.
- ➋ SOMMERVILLE, Ian. Engenharia de Software, São Paulo: Pearson Addison Wesley, 9<sup>a</sup> edição. 2011.
- ➌ FAIRLEY, Richard E. Software Engineering Concepts. McGraw-Hill, 1985.
- ➍ SARSON, Trish; GANE, Chris. Análise Estruturada de Sistemas. Rio de Janeiro: LTC, 1984.
- ➎ YOURDON, Edward. Análise Estruturada Moderna. Rio de Janeiro. Campus. 1990.
- ➏ DAVIS, William S. Análise e Projeto de Sistemas - Uma Abordagem Estruturada. Rio de Janeiro. LTC. 1994.
- ➐ HEUSER, Carlos A. Projeto de Banco de Dados. Ed. Sagra Luzzatto, Porto Alegre, 2001, 4a. edição.
- ➑ SILBERSCHATZ, Abraham; KORTH, Henry F. e SUDARSHAN, S. Sistemas de Bancos de Dados. São Paulo: Makron Books, 1999. 3<sup>a</sup> edição.
- ➒ TEOREY, Toby; LIGHTSTONE, Sam; NADEAU, Tom. Projeto e Modelagem de Banco de Dados. Rio de Janeiro: Elsevier, 2007. 276p.
- ➓ BOOCHE, Grady.; RUMBAUGH, James.; JACOBSON, Ivar. UML: Guia do Usuário. Rio de Janeiro: Campus. 2006. 472 p.
- ➔ GHOFF, James R.; WEINBERG, Paul N.; EBRARY, Inc. SQL, the complete reference. 2<sup>nd</sup> ed Berkeley, Calif.: McGraw-Hill Professional, 2002. 1082 p.

## Bibliografia

---

## Método de avaliação

---

## Tópicos da Aula

---

- ➊ Software:
  - ▢ Definição.
  - ▢ Características.
- ➋ Análise e Projeto de Sistemas:
  - ▢ Definições.

## Lembrando: Dado, Informação e Conhecimento

---

Dado: 40



Informação: 40 °C

## Software

Conhecimento: O paciente está com febre

→  
Tomar uma decisão

## O Que é Software?

- ➊ Possíveis definições:
  - É um programa (instruções) de computador.
  - Aquilo que pode ser executado por um equipamento.
  - Um produto que consiste de um sistema de rotinas e funções.
- ➋ Pergunta: mas será que é apenas isso?

## Evolução do Software (1950-1965)

- ➊ O hardware sofreu mudanças contínuas nesse período.
- ➋ O software era uma tarefa secundária, para a qual havia poucos métodos sistemáticos.
- ➌ O hardware era de propósito geral.
- ➍ O software era específico para cada aplicação.
- ➎ Os programas eram geralmente executados em batch.
- ➏ Não havia documentação.
- ➐ A rotatividade dos empregos era baixa.

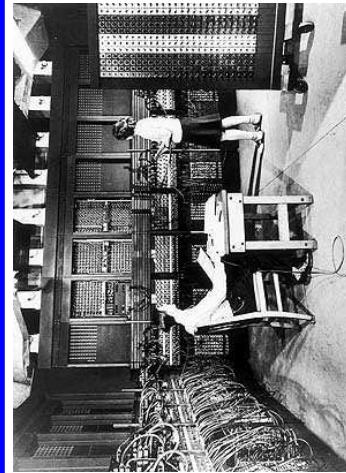
## Válvulas: ENIAC (Electronic Numerical Integrator and Computer)

- ➊ Em 1943 teve início o trabalho de construção do ENIAC, um equipamento com a capacidade de realizar até 5000 adições por segundo, sendo finalizado em 1946, tarde demais para ser usado durante a Segunda Guerra Mundial.
- ➋ Mauchly (professor) e Eckert (aluno de pós-graduação) da Universidade da Pensilvânia propuseram a sua criação.
- ➌ Seria primeiramente utilizado para fazer tabelas de trajetória para novas armas.
- ➍ Primeira tarefa: ajudar a verificar se a bomba H poderia ser construída.
- ➎ Utilizado até 1955 pelo Laboratório de Pesquisas Balísticas do exército dos EUA.

## ENIAC: Detalhes

- ➊ Decimal (não binário).
- ➋ 20 acumuladores de 10 dígitos.
- ➌ Programado manualmente por chaves.
- ➍ 18,000 válvulas.
- ➎ 1500 relés.
- ➏ 30 toneladas.
- ➐ 140 metros quadrados.
- ➑ Consumo de 140 kW de energia elétrica.
- ➒ Em 1952, mais de 19,000 válvulas haviam sido substituídas.

## ENIAC



## Transitores (1)

- ➊ Substituíram as válvulas.
- ➋ Menores.
- ➌ Mais baratos.
- ➍ Menor dissipação de calor.
- ➎ Dispositivos de estado sólido.
- ➏ Feitos de Silício (Areia).
- ➐ Inventados pela Bell Labs em 1947.

## Transistores (2)

---

- ⌚ Aparecimento de um software para controle do sistema.
- ⌚ Uso de linguagens de alto nível.
- ⌚ Aparecimento da Digital Equipment Corporation (DEC) em 1957 com o lançamento do PDP-1 (Programmed Data Processors) que inicia o fenômeno do mini-computador.

## Circuitos Integrados (CIs)

---

- ⌚ Cada transistor utilizado era um dispositivo individual que deveria ser soldado na placa de circuito impresso. Assim, conforme o número de circuitos aumentaram, o trabalho de montagem se tornou cada vez mais difícil.
- ⌚ A invenção dos circuitos integrados em 1958 iniciam a era da microeletrônica.

## Evolução do Software (1965-1975) (1)

---

- ⌚ Multiprogramação e sistemas multiusuários.
- ⌚ Técnicas interativas.
- ⌚ Sistemas de tempo real.
- ⌚ Primeira geração de SGBD.
- ⌚ Produtos de software feitos por Software Houses.
- ⌚ Cresce o número de sistemas baseado em computador.

## Evolução do Software (1965-1975) (2)

---

- ⌚ A necessidade de manutenção aumentou muito.
- ⌚ A natureza personalizada de muitos programas tornava a manutenção muito difícil, devido a falta de documentação e ao número reduzido de pessoas para tal tarefa.
- ⌚ Por volta de 1970, Tom DeMarco desenvolveu a Engenharia de Software baseada em modelos, técnica que possibilita aos usuários a visualização do funcionamento de um sistema futuro antes que o mesmo seja construído. Quebrou assim o paradigma do “começar já codificando”.

## CIs com Alta Densidade de Componentes

---

- ⌚ LSI (Large-Scale Integration) - até 10.000 componentes podem ser colocados em um único circuito integrado (após 1972).
- ⌚ VLSI (Very Large-Scale Integration) - mais de 10.000 componentes podem ser colocados em um único circuito integrado (após 1978).

## Fatos Históricos

---

- ⌚ Em 1971, a Intel lança o primeiro chip que contém todos os componentes de uma CPU, o 4004, iniciando a era dos microprocessadores.
- ⌚ Em 1972 a Intel lança o primeiro microprocessador de 8 bits, o Intel 8008.
- ⌚ Em 1974 é lançado o Intel 8080, o primeiro microprocessador de uso não específico, desenvolvido para ser usado em computadores de uso geral.
- ⌚ Em 1976 é lançado o Cray-1, primeiro supercomputador de uso comercial. Continha 200 mil circuitos integrados.

## Cray-1



## Evolução dos Microprocessadores Intel

- ⌚ Velocidade de clock: de 108 KHz (4004 de 1971) à 2,4 GHz e mais (Pentium IV).
- ⌚ Número de transistores: 2.300 no 4004; 29000 no 8086 em 1978; 134.000 no 80286 em 1982; 275.000 no 80386 em 1985; 1.200.000 no 80486 em 1989; 3.100.000 no Pentium em 1993; 42 milhões no Pentium 4 em 2001; 220 milhões no Itanium 2 em 2002; 500 milhões na versão "Madison" do Itanium 2 em 2002.
- ⌚ 2004: microprocessadores com mais de um núcleo. A justificativa é o custo extremamente alto para fazer o arrefecimento dos microprocessadores.
- ⌚ 2005: versão "Montecito" do Itanium 2 com dois núcleos e mais de 1 bilhão de transistores.

## Evolução dos Microprocessadores Intel

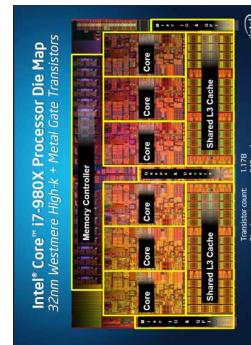
- ⌚ 2005: versão do Dual-Core Itanium 2 com dois núcleos e mais de 1 bilhão de transistores.
- 
- ⌚ 2006: microprocessadores com 4 núcleos. Quad-Core Intel Xeon com 820 milhões de transistores à 3,7 GHz.

## Evolução dos Microprocessadores Intel

- ⌚ 2007: Pesquisadores da Intel construíram um chip de 80 núcleos. Rodando a 3,16 GHz, o chip chega a 1,01 teraflops de computação.
- 
- ⌚ O chip tem 275 milímetros quadrados (o tamanho de uma unha) e utiliza apenas 62 watts de energia, menos que um chip para desktop.

## Evolução dos Microprocessadores Intel

- ⌚ 2008: i7 Roda a 3,33 GHz, com de 731 milhões a 1,17 bilhões de transistores.



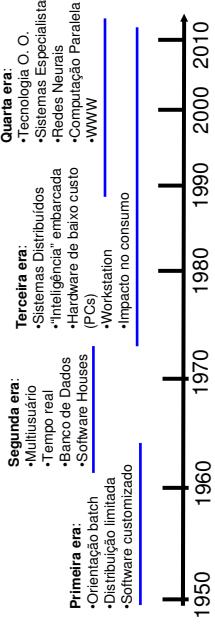
## Evolução do Software (1975 - até hoje)

- ⌚ Sistemas distribuídos.
- ⌚ Redes locais e globais.
- ⌚ Uso generalizado de microprocessadores e computadores pessoais.
- ⌚ Criação de produtos inteligentes, como automóveis, forno de microondas, etc.
- ⌚ Hardware de baixo custo gerando aumento no consumo de computadores.

## Evolução do Software (1985 - até hoje)

- ⌚ Tecnologias orientadas a objetos.
- ⌚ Sistemas especialistas, software de inteligência artificial utilizados na prática, multi-agentes.
- ⌚ Agentes móveis.
- ⌚ Redes neurais artificiais.
- ⌚ Algoritmos genéticos.
- ⌚ WWW.
- ⌚ Computação Paralela.

## Evolução do Software (1950 - até hoje)



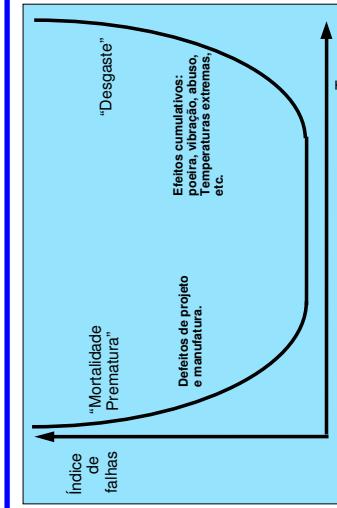
## Então, pelo que é formado o Software?

- É formado por:
  - ⌚ **Instruções** (programas de computador) que quando executadas produzem a função e o desempenho desejados.
  - ⌚ **Estruturas de Dados** que possibilitam que os programas manipulem adequadamente a informação.
  - ⌚ **Documentos** que descrevem a operação e o uso dos programas.

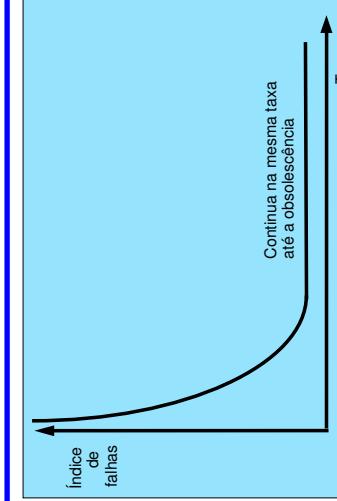
## Características do Software

- ⌚ Ele é desenvolvido ou projetado por engenharia, mas não é manufaturado no sentido clássico:
  - ⌚ Os custos do software estão concentrados no trabalho de engenharia. Isso significa que os projetos de software não podem ser geridos como se fossem projetos de manufatura.
  - ⌚ Não se "desgasta", mas se deteriora devido as modificações feitas durante sua vida útil (ver slides seguintes).
  - ⌚ A maioria é feita sob medida em vez de ser montada a partir de componentes existentes.

## Curva de Falhas de Hardware



## Curva Idealizada de Falhas de Software



## Curva Real de Falhas de Software, em função das mudanças de Requisitos

- Segundo Brooks (1995), ao se corrigir um defeito, há de 20% a 50% de chance de se introduzir um defeito no software.

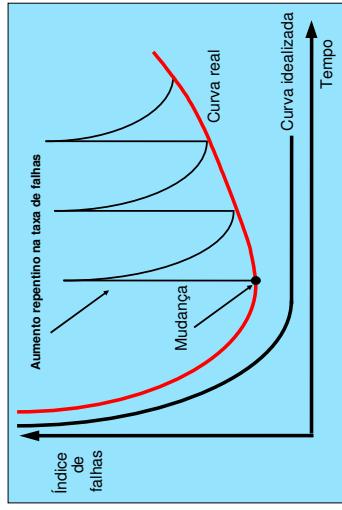
Brooks, Frederick. The Mythical Man Month. Addison-Wesley Publishing Company, 1995. 214 p.

## Comparação entre Hardware e Software

- Quando um componente de hardware se desgasta ele é substituído por uma “peça de reposição”.

- Não existe “peça de reposição” para software:
  - Toda falha indica um erro no projeto ou no processo de tradução para o código executável.
  - A manutenção de software é mais complexa do que a de hardware.

## Curva Real de Falhas de Software



## Definições

### O que é Análise?

- Derivado do grego analyein (desatar, soltar), significa: dissolução de um conjunto em suas partes.
- Análise é um estudo de um problema, que antecede uma ação para resolvê-lo. O seu propósito é modelar um sistema de forma que ele possa ser entendido.

### O que é Processo?

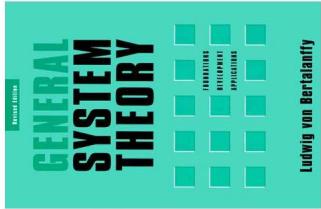
- Série de fenômenos sucessivos com relação de causa e efeito. Por exemplo, uma empresa é uma série de causas - matérias primas, recursos humanos, tecnologia - que geram um efeito, no caso, os produtos.

## O que é Sistema? Definições:

- ➊ Um grupo de itens que interagem entre si ou que sejam interdependentes, formando um todo unificado.
- ➋ Um conjunto organizado de doutrinas, idéias ou princípios, habitualmente previsto para explicar a organização ou funcionamento de um conjunto sistemático.
- ➌ Um procedimento organizado ou estabelecido.

## Teoria Geral de Sistemas

- ➊ A Teoria Geral de Sistemas teve como um dos precursores o biólogo austriaco Ludwig von Bertalanffy.
- ➋ Auxiliou a criação da Análise e Projeto Estruturado de Sistemas (Tom DeMarco e Edward Yourdon).
- ➌ BERTALANFFY, L. Von. General Theory of Systems: Foundations, Development, Applications, New York: George Braziller, 1968.



## Tipos de Sistemas (1)

- ➊ Naturais ou feitos pelo homem.
- ➋ Sistemas Naturais:
  - Sistemas Estelares (ver slides):
    - Galáxias, sistemas solares, etc.
  - Sistemas Geológicos:
    - Rios, cadeias de montanhas, etc.
  - Sistemas Moleculares:
    - Organizações complexas de átomos.

## Tipos de Sistemas (2)

- ➊ Sistemas feitos pelo homem:
  - Sistemas Sociais:
    - Organizações de leis, doutrinas, costumes, etc.
  - Sistemas de Transporte:
    - Redes rodoviárias, canais, linhas aéreas, petroleiros, e semelhantes , etc.
  - Sistemas de Comunicação:
    - Telefone, telex, sinais de fumaça, sinais manuais, etc.

## Tipos de Sistemas (3)

- ➊ Sistemas feitos pelo homem:
  - Sistemas de Manufatura:
    - Fábricas, linhas de montagem, etc.
  - Sistemas Financeiros:
    - Contabilidade, inventários, controle de estoque, etc.
  - Sistemas Automatizados:
    - Hardware de computadores, software, etc.

## Tipos de Sistemas (4)

- ➊ Concretos: compostos por objetos reais como equipamentos, instalações, hardware.
- ➋ Abstratos: compostos por conceitos, idéias como planos, teorias, software.

## Tipos de Sistemas (5)

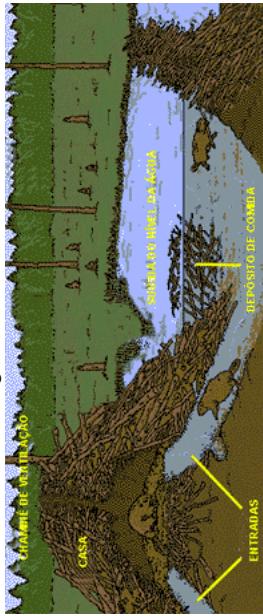
- ➊ Fechados: não recebem nenhuma influência do ambiente que os circunda. Não recebem nenhum recurso externo e nada produzem que seja enviado para fora. Exemplo: a Teoria da Relatividade de Einstein não será alterada em função do surgimento de novas teorias.
- ➋ Abertos: são sistemas que apresentam trocas com o ambiente ao redor, através de entradas e saídas. Exemplo: software.

## Tipos de Sistemas (6)

- ➊ Emergentes:
  - ▣ Surgem naturalmente. Exemplo: ecossistemas.
- ➋ Teleológicos:
  - ▣ São planejados, sendo dirigidos por objetivos. Exemplo: residências, fábricas, plantações, hardware, software.
  - ▣ Não são exclusividade dos seres humanos.

## Tipos de Sistemas (7)

- ➊ Sistema Teleológico. Casa do castor:



Disponível em: <http://www.saudeanimal.com.br/castor.htm>

## O que é Análise e Projeto de Sistemas?

- ➊ Representa o estudo detalhado de uma área de trabalho (processo), que antecede uma ação que, **quase sempre**, implica no desenvolvimento de um conjunto de softwares integrados (sistema) destinado à execução, controle e acompanhamento do processo.

## Participantes: Responsabilidades do Analista e do Usuário

- ➊ O usuário, que será o utilizador do sistema. Ele é o responsável pela decisão de integração do sistema dentro das operações da empresa, ou não. Somente ele pode aceitar ou não o sistema.
- ➋ O cliente, que normalmente é quem contrata os serviços da empresa desenvolvedora de software.
- ➌ O analista, que é o responsável por: estudos de viabilidade e alternativas, relações custo/benefício, especificações, prazos e testes de aceitação.

## **Problemas no Desenvolvimento de Sistemas**

---

- Produtividade.
- Demanda reprimida.
- Tempo necessário para desenvolvimento.
- Descontinuidade no desenvolvimento.
- Confabilidade.
- Manutenção.
- Eficiência.
- Portabilidade.
- Segurança.

**Obrigado.**

## Análise e Projeto de Sistemas

- Análise de Requisitos de um Projeto: Entrevistas, Questionários, Reuniões e Observação
- Estudo de Viabilidade

**Prof. Laudelino Cordeiro Bastos**

## **Análise de Requisitos de um Projeto**

- Análise de Requisitos de um Projeto: Entrevistas, Questionários, Reuniões e Observação
- Estudo de Viabilidade

### **Entrevista**

#### **Entrevista**

- ⌚ Definição: Conversa entre duas pessoas, provocada por uma delas, com objetivo definido.
- ⌚ Proporciona contato pessoal que faz com que o entrevistado se sinta parte influente do processo.

#### **Entrevista**

- ⌚ Objetivos:
  - Obtenção de informações registradas apenas na memória.
  - Obtenção de informações a respeito dos integrantes do processo.

### **Preparação para a Entrevista**

- ⌚ Procure conhecer o assunto através de:
  - Manuais.
  - Revistas.
  - Normas.
  - Legislação.
  - Pedidos de Proposta (RFP - Request for Proposal).
  - Outros sistemas semelhantes.
- ⌚ Caso tenha um acesso direto à organização:
  - Planos de negócio.
  - Fluxos de processos atuais.
  - Modelos de dados lógicos.
- ⌚ Identifique as pessoas que estarão envolvidas e as suas características:
  - Posição na hierarquia da empresa.
  - Poder de decisão.
  - Interesse no assunto em questão.
  - Participação em outras atividades.

### **Preparação para a Entrevista**

## Preparação para a Entrevista

---

### Preparação para a Entrevista

---

- ➊ Elabore uma agenda:
  - ❑ Marque o evento com antecedência.
  - ❑ Estabeleça a pauta do que vai ser discutido.
  - ❑ Estabeleça local, horário e duração.
  - ❑ Confirme a entrevista um dia antes.

## Durante a Entrevista

---

- ➋ Procure vestir-se de maneira adequada.
- ➋ Conquistar confiança do entrevistado.
- ➋ Evitar o desvio para outros assuntos.
- ➋ Escute mais, seja neutro em questões pessoais.
- ➋ Esclareça a pessoa sobre o assunto.

## Durante a Entrevista

---

- ➌ Elabore um roteiro:
  - ❑ Objetivos a atingir.
  - ❑ Principais tópicos a serem abordados.
  - ❑ Prioridade dos tópicos.
  - ❑ Nível de informação desejado.

## Após a Entrevista

---

- ➍ Revise a entrevista após seu término.
- ➍ Elabore uma ata dos assuntos discutidos e peça aprovação do cliente ou do usuário.

## Entrevista

---

- ➎ Limitações:
  - ❑ Despreparo do entrevistado.
  - ❑ Tendência em dar respostas agradáveis.
  - ❑ Palpites e respostas falsas.
  - ❑ Tomar notas durante entrevista.
  - ❑ Escolha da pessoa incorreta.
  - ❑ Ambiguidade: uma ambiguidade ocorre, em uma entrevista de levantamento de requisitos, quando um cliente articula um conjunto de informações e o significado atribuído pelo analista de requisitos para a articulação difere do significado pretendido pelo cliente.
  - ❑ Conhecimento tácito: em Engenharia de Requisitos, o conhecimento tácito é definido como o conhecimento que o cliente tem, mas que não passa para o analista de requisitos por algum motivo.

## **Entrevista: Classes de Conhecimento dos Requisitos (Ferrari, Spoletrini e Paola, 2016)**

### **Questionário**

- ⌚ Informação conhecida pelo analista e pelo cliente; uma informação relevante é passada com sucesso do cliente para o analista.
- ⌚ Informação conhecida pelo analista e não conhecida pelo cliente; uma informação relevante que não foi expressada pelo cliente, mas a qual o analista sabe ou suspeita que ele tem.
- ⌚ Informação não conhecida pelo analista e conhecida pelo cliente (conhecimento tácito); uma informação relevante que o cliente poderia expressar, mas não o faz, e o analista não sabe da sua existência.
- ⌚ Informação não conhecida pelo analista e não conhecida pelo cliente; uma informação relevante que o cliente e o analista não conhecem.

Ferrari, Alessio; Spoletrini, Paola; Galesi, Stefania. Ambiguity and tacit knowledge in requirements elicitation interviews. Requirements Engineering, 2016, pg. 335-355.

### **Preparação do Questionário**

- ⌚ Realize uma introdução esclarecendo os objetivos.
- ⌚ Faça instruções sobre a forma de preenchimento.
- ⌚ Estime a duração do preenchimento do questionário.
- ⌚ Estabeleça o prazo de devolução.
- ⌚ Contate um superior hierárquico quando necessário.

### **Questionário**

- ⌚ Definição: conjunto de perguntas com respostas objetivas ou graduais dispostas em sequência lógica e progressiva, aplicado a uma ou mais pessoas.
- ⌚ As perguntas devem ser claras:
  - ▢ A linguagem deve ser a mais clara possível.
  - ▢ As frases devem ser concisas.
  - ▢ Os termos devem possuir significado preciso e específico.
- ⌚ As perguntas devem ser simples e diretas:
  - ▢ Utilize sempre que possível perguntas com respostas fechadas.

### **Questionário**

- ⌚ Limitações:
  - ▢ Resistência a sua aceitação e preenchimento.
  - ▢ São frios e impessoais, impedem a troca de informações.
  - ▢ Tendência a registrar respostas distorcidas.
- ⌚ Tipos de perguntas:
  - ▢ Pergunta complementar: é aquela que através de uma sequência lógica se esgota um tema.
  - ▢ Pergunta encadeada: tem por finalidade testar a coerência das respostas fornecidas.
- ⌚ Realize um teste com os seus usuários antes de enviar o questionário.

### **Questionário**

## **Reunião**

---

### **Reunião**

---

● É o intercâmbio de idéias, sugestões e opiniões entre determinados indivíduos visando a aceitação de um ponto de vista por parte dos participantes.

- Objetivos:
  - Comunicação direta entre os participantes.
  - Interação do grupo em torno de um assunto.
  - Definição e esclarecimento de problemas.
  - Tomada de decisão.
  - Troca de experiências.

## **Preparação para a Reunião**

---

### **Ambiente físico:**

- Sala ampla, ventilada, bem iluminada.
- Mesa central.
- Cadeiras adequadas.
- Quadro para anotações.
- Flipchart e canetas.
- Material de consumo como lápis, papel, borracha.
- Aviso na porta: “Reunião”.
- Verificar se o ambiente está adequado para os equipamentos que serão utilizados.

## **Reunião**

---

- Abertura
  - Apresentação do condutor.
  - Apresentação do objetivo e agenda da reunião.
  - Apresentação dos participantes.
  - Determinar as regras da reunião.
  - Estabelecer os papéis.

## **Reunião**

---

### **Recomendações para a Reunião**

---

- Tópicos:
  - Exposição do problema.
  - Intercâmbio de experiências.
  - Levantamento de causas e consequências.
  - Levantamento de alternativas.
  - Avaliação das consequências de cada alternativa.
  - Escolha de alternativas.
  - Busca de consenso.
  - Definição dos responsáveis e dos prazos.
- Procure sempre o consenso do grupo.
- Permita que todos tenham oportunidade de expor suas idéias.
- Estabelecer pauta, antecipadamente.
- Cumprir o tempo determinado para reunião.
- Não defenda suas idéias ao extremo. Mudar de opinião se verificar um erro.

## **Limitações das Reuniões**

- ⦿ Deslocamento dos participantes.
- ⦿ Domínio da reunião por um único indivíduo.
- ⦿ Faltas e atrasos.
- ⦿ Comparecimento desnecessário.
- ⦿ Ambiente físico inadequado.
- ⦿ Não participação por falta de confiança.

## **Características do Condutor**

- ⦿ Deve ser um conciliador.
- ⦿ Deve abrir espaço para as pessoas colocarem o seu ponto de vista.
- ⦿ Deve administrar os conflitos.
- ⦿ Deve esclarecer as definições.
- ⦿ Deve controlar pauta e tempo.

## **Técnicas de Perguntas**

- ⦿ Pergunta dirigida: pergunta encaminhada para um determinado participante.
- ⦿ Geral: perguntas dirigidas a todos os participantes.
- ⦿ Redistribuídas: pergunta feita ao condutor que é retornada a um dos participantes ou ao grupo.
- ⦿ Reversa: pergunta que o condutor devolve a quem formulou.

## **Observação**

- ⦿ Definição: é uma técnica que utiliza os sentidos na obtenção de determinados aspectos da realidade.

## **Observação**

- ⦿ Recomendações:
  - ▢ Colocar-se no ambiente sem interferir.
  - ▢ Solicite que as pessoas expliquem sua rotina de trabalho.
  - ▢ Verifique se os manuais correspondem com o serviço realizado.
  - ▢ Faça a observação em um período que o serviço possa ser caracterizado como normal.
- ⦿ Objetivos:
  - ▢ Confirmar as informações obtidas das entrevistas, revisão da literatura, comparando-as com a realidade.
  - ▢ Levantar volumes (quantidade) das informações.
  - ▢ Identificar o fluxo de documentos ao longo das tarefas.

## **Observação**

### **Cuidados:**

- Empregados realizando tarefas que não dizem respeito a sua atividade.
- Confusão ou ruído além do normal.
- Chefes com mesa cheia de papéis.
- Pessoas perambulando pelo local de trabalho.
- Ausência de funcionários.

## **Conclusão**

- Além do domínio das técnicas de análise e projeto de sistemas, realizar análise de requisitos requer dedicação, paciência, facilidade para o diálogo e sensibilidade.

## **Estudo de Viabilidade: Definições**

- Estudos preliminares que investigam as necessidades de informação dos potenciais usuários e determinam os requisitos de recursos, relações custos/benefícios e viabilidade do projeto.
- São realizados por meio de métodos de coleta de informação:
  - Entrevistas e/ou questionários com funcionários, usuários, clientes, gerentes.
  - Observação pessoal, gravação em vídeo ou envolvimento nas atividades de trabalho dos usuários finais.
  - Exame de documentos, relatórios, manuais de procedimentos e outros registros.
  - Desenvolvimento, simulação e observação de um modelo das atividades de trabalho.

## **Estudo de Viabilidade**

## **Estudo de Viabilidade**

- A viabilidade pode ser avaliada em termos de 4 categorias:
  - Organizacional.
  - Econômica.
  - Técnica.
  - Operacional.
- Viabilidade pode ser avaliada em termos de 4 categorias:
  - Organizacional.
  - Econômica.
  - Técnica.
  - Operacional.

Eficácia com que o sistema proposto apóia os objetivos estratégicos da organização.

## **Estudo de Viabilidade**

• Viabilidade pode ser avaliada em termos de 4 categorias:

- Organizacional.
- Econômica.
- Técnica.
- Operacional.

- Economia de custos.
- Aumento da receita.
- Redução do investimento.
- Aumento dos lucros.

## **Estudo de Viabilidade**

• Viabilidade pode ser avaliada em termos de 4 categorias:

- Organizacional.
- Econômica.
- Técnica.
- Operacional.

- Benefícios:
  - Tangíveis: aumento das vendas; redução de custos operacionais; aumento da eficiência.
  - Intangíveis: disponibilidade de informações; posição competitiva; imagem da organização; melhor atendimento ao cliente.

## **Estudo de Viabilidade**

• Viabilidade pode ser avaliada em termos de 4 categorias:

- Organizacional.
- Econômica.
- Técnica.
- Operacional.

- Custos:
  - tangíveis: hardware, software, salários, etc.
  - intangíveis: satisfação do cliente e dos usuários finais.

## **Estudo de Viabilidade**

• Viabilidade pode ser avaliada em termos de 4 categorias:

- Organizacional.
- Econômica.
- Técnica.
- Operacional.

- Capacidade, confiabilidade e disponibilidade de hardware, de software e de rede.
- Capacidade do pessoal.

## **Estudo de Viabilidade**

• Viabilidade pode ser avaliada em termos de 4 categorias:

- Organizacional.
- Econômica.
- Técnica.
- Operacional.

- Aceitação dos usuários finais.
- Apoio administrativo.
- Requisitos dos clientes, fornecedores e governo.

## **Estudo de Viabilidade: Exemplos de Questões para a Elicitação**

• Questões Organizacionais:

- Quais as principais necessidades do cliente?
- Quais os principais benefícios que o projeto trará para o cliente?

• Questões Econômicas:

- Qual será aproximadamente o custo do projeto?
- Existe mercado real para o produto?

• Questões Técnicas

- Quais as ferramentas de software e hardware a serem utilizadas?
- O cliente dispõe de infra-estrutura suficiente para sustentar o novo sistema?

• Questões Operacionais

- Quais são os futuros usuários deste sistema?
- Os futuros usuários apóiam o desenvolvimento do sistema?

## PMBOK como Guia para Projetos

● PMBOK - Project Management Body of Knowledge ou Universo de Conhecimento em Gerência de Projetos. Esta denominação representa o somatório de conhecimentos dentro da profissão de gerência de projetos, sendo identificado como guia para os profissionais da área.

## PMBOK como Guia para Projetos

● Este guia é de autoria do Standards Committee (Comitê de Padronização) do Project Management Institute – PMI e procura contemplar os principais aspectos que podem ser abordados no gerenciamento de um projeto genérico. O “PMBOK Guide” é mundialmente reconhecido, além de ser aceito, desde 1999, como padrão de gerenciamento de projetos pelo ANSI – American National Standards Institute.

● Ver a revista *Project Management Journal* no Periódicos CAPES.

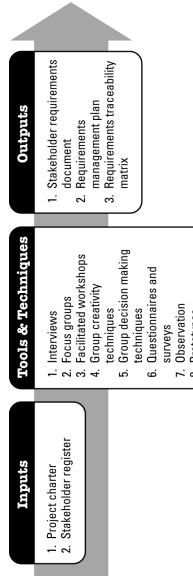
## PMBOK como Guia para Projetos

● Pelo modelo do PMBOK a estrutura do gerenciamento do projeto é decomposta em áreas de conhecimento. ● Baseado na concepção utilizada pelo PMBOK, o gerenciamento de projetos é considerado como um conjunto de processos interligados que possuem nove áreas que representam o somatório de conhecimento da gerência de projetos.

● Para o PMI, o universo de conhecimento necessário para gerenciar projetos consiste nestas áreas que englobam as técnicas para tornar o processo de gerenciamento eficaz.

## Visão Geral

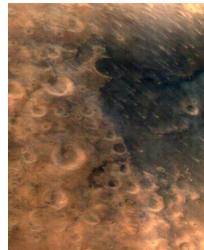
● Collect Requirements: Inputs, Tools & Techniques, and Outputs



## Vídeo: Dois engenheiros e vários milhões

● Mário Sérgio Cortella: Mestre e Doutor em Educação pela PUC de São Paulo.

## Sonda Indiana Mangalyaan



Com um orçamento de 74 milhões de dólares, a missão indiana custou apenas uma pequena parte do valor estimado da sonda MAVEN (Mars Atmosphere and Volatile Evolution), da NASA (US\$ 671 milhões), alcançando a órbita marciana em 28 de setembro de 2014.

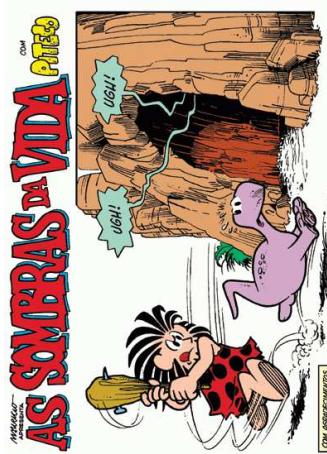
A missão espacial foi concebida segundo o "jugaad" princípio tipicamente indiano, que consiste em encontrar a solução ao mesmo tempo mais engenhosa e menos onerosa possível.

Os engenheiros da ISRO (Indian Space Research Organisation) tiveram a ideia de que o foguete indiano desse voltas ao redor da Terra durante um mês a fim de ganhar velocidade para escapar da força gravitacional da Terra.

## Nem tudo é o que parece

Platão, Alegoria da Caverna  
Turma da Mônica:  
<http://www.monica.com.br/comics/piteco/welcome.htm>

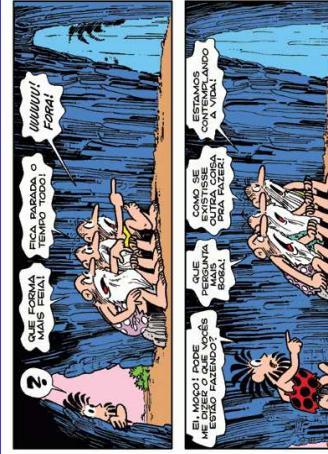
### Platão, Alegoria da Caverna



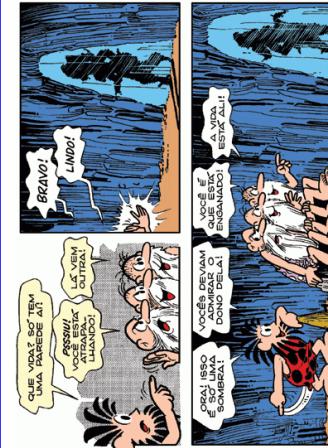
### Platão, Alegoria da Caverna



### Platão, Alegoria da Caverna



### Platão, Alegoria da Caverna



### Platão, Alegoria da Caverna



### Platão, Alegoria da Caverna



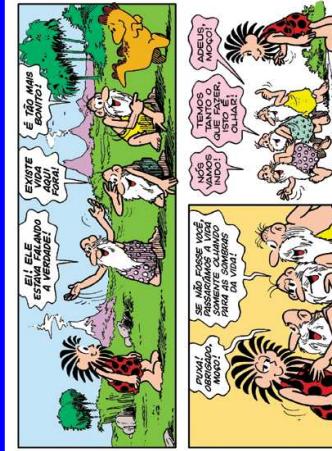
### Platão, Alegoria da Caverna



### Platão, Alegoria da Caverna



### Platão, Alegoria da Caverna



### Platão, Alegoria da Caverna



## **Sobre o trabalho**

---

- Não podem ser realizados projetos relativos a:
  - Videoladoras e similares.
  - A linguagem deve ser orientada a objetos, C++ ou Java, e suas linguagens derivadas.
  - Não pode ser utilizada a linguagem PHP.
  - Equipes de 3 ou 4 alunos. A formação e dissolução das equipes é de responsabilidade dos alunos.

**Obrigado.**

## Abrangência da Engenharia de Software

---

### Análise e Projeto de Sistemas

---

### O Processo de Software

---

**Prof. Laudelino Cordeiro Bastos**

- ⦿ A Engenharia de Software, que teve origem na Engenharia de Sistemas e de Hardware, abrange um conjunto de três elementos fundamentais: **Métodos, Ferramentas e Procedimentos** (Processos).
- ⦿ Tal conjunto tem como objetivos:
  - ⦿ Possibilitar ao gerente o controle do processo de desenvolvimento.
  - ⦿ Oferecer ao profissional uma base para a construção de software de alta qualidade.

## Métodos da Engenharia de Software

---

- ⦿ Os **métodos** proporcionam os detalhes de “como fazer” para construir o software. Eles envolvem um amplo conjunto de tarefas:
  - ⦿ Planejamento e estimativa de projeto.
  - ⦿ Análise de requisitos de software.
  - ⦿ Projeto da estrutura de dados.
  - ⦿ Arquitetura do programa e algoritmos de processamento.
  - ⦿ Codificação.
  - ⦿ Testes.
  - ⦿ Manutenção.

## Ferramentas da Engenharia de Software

---

- ⦿ As **ferramentas** fornecem suporte automatizado ou semi-automatizado aos métodos:
  - ⦿ Atualmente existem ferramentas para sustentar cada um dos métodos.
  - ⦿ Quando as ferramentas são integradas é estabelecido um sistema de suporte ao desenvolvimento de software chamado Computer Aided Software Engineering (CASE).

## Procedimentos da Engenharia de Software

---

- ⦿ Os **procedimentos** constituem o elo de ligação entre os métodos e ferramentas. Eles definem:
  - ⦿ A seqüência em que os métodos serão aplicados.
  - ⦿ Os produtos que se exige que sejam entregues (documentos, relatórios, formulários, diagramas, código).
  - ⦿ Os controles que ajudam a assegurar a qualidade e coordenar as alterações.
  - ⦿ Os marcos de referência que possibilitam administrar o progresso do software.

## Paradigmas da Engenharia de Software

---

- ⦿ Os paradigmas da Engenharia de Software compreendem um conjunto de etapas que envolve métodos, ferramentas e procedimentos.
  - ⦿ Essas etapas são chamadas de **Ciclos de Vida** ou **Modelos de Processo de Software**.
  - ⦿ Os ciclos de vida são uma estratégia de desenvolvimento.

## **Escolha de um Modelo de Processo**

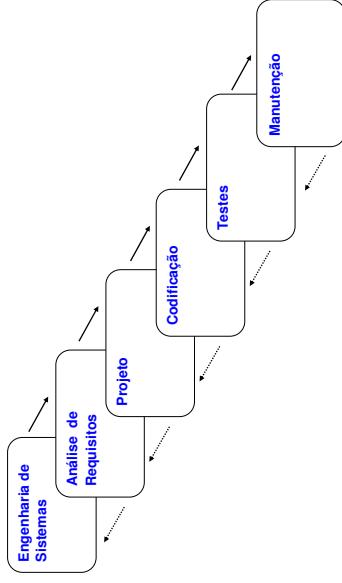
- ⦿ Qual processo de desenvolvimento será adotado? Para responder essa pergunta, é necessário analisar:
  - ▢ Verificação da adequação do modelo de processo à aplicação.
  - ▢ Métodos e ferramentas a serem utilizados.
  - ▢ Controles e produtos que precisam ser entregues.
  - ▢ Características dos processos: visibilidade, clareza, apoio de ferramentas, produtividade, qualidade, etc.

## **Modelo de Ciclo de Vida Clássico (Cascatas)**

- ⦿ Modelo mais antigo da engenharia de software.
- ⦿ Modelado em função do ciclo da engenharia convencional.
- ⦿ Requer uma abordagem sistemática e sequencial para o desenvolvimento de software.
- ⦿ Cada atividade é uma fase em separado. A passagem entre fases é formal.

## **Modelo de Ciclo de Vida Clássico (Cascatas)**

- ⦿ Cada fase envolve a elaboração de um ou mais documentos, que devem ser aprovados antes de se iniciar a fase seguinte. Assim, uma fase só deve ser iniciada após a conclusão daquela que a precede.
- ⦿ Na prática, como essas fases se sobrepõem de alguma forma, geralmente se permite um retorno à fase anterior para a correção dos erros encontrados.



## **Problemas com o Ciclo de Vida Básico**

- ⦿ Os projetos reais raramente seguem o fluxo sequencial que o modelo propõe.
- ⦿ É difícil estabelecer explicitamente, logo no início, todos os requisitos. No começo dos projetos sempre existe uma incerteza natural.
- ⦿ O cliente deve ter paciência. Uma versão executável do software só fica disponível numa etapa avançada do desenvolvimento.

## **Vantagens do Ciclo de Vida Clássico**

- ⦿ A visibilidade do processo permite um fácil entendimento do mesmo.
- ⦿ Embora o Ciclo de Vida Clássico tenha fragilidades, ele é significativamente melhor do que uma abordagem casual no desenvolvimento de software.

**Prototipação (1)**

- Processo que possibilita que o desenvolvedor crie um modelo do software que deve ser construído.
  - Idealmente, o modelo (protótipo) serve como um mecanismo para identificar os requisitos de software:
    - Protótipo do sistema que retrata a interação com o usuário.
    - Protótipo que implemente algumas funções exigidas.

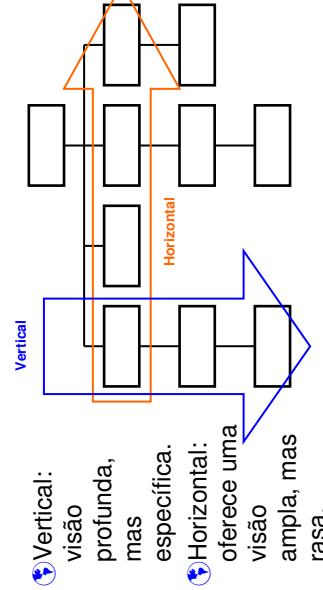
**Prototipação (2)**

- ❑ A Prototipação pode ser classificada em:
    - ❑ Prototipação de baixa fidelidade, quando os protótipos são desenhados na forma de esboços e sem muitos detalhes. Os protótipos criados nesse tipo de prototipação são também chamados de mockups ou wireframes.
    - ❑ Prototipação de alta fidelidade, quando os protótipos apresentam um nível alto de detalhes do ponto de vista da aparência. São construídos utilizando a própria ferramenta de desenvolvimento de software.

## **Prototipação (3): Exemplo de Mockup**

Cadastrando dados da aula																																																	
     	     																																																
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;">Nome:</td> <td style="width: 15%;"><input type="text" value="Participante 1"/></td> <td style="width: 15%;">Nome Completo:</td> <td style="width: 15%;"><input type="text" value="Participante 1"/></td> <td style="width: 15%;">Data de Nascimento:</td> <td style="width: 15%;"><input type="text" value="00/00/0000"/></td> </tr> <tr> <td colspan="2">Competência Básica</td> <td colspan="2">Intermediária</td> <td colspan="2">Avançada</td> </tr> <tr> <td><input type="checkbox"/> Java</td> <td><input checked="" type="checkbox"/> C</td> <td><input type="checkbox"/> PHP</td> <td><input type="checkbox"/> Python</td> <td><input type="checkbox"/> React.js</td> <td><input type="checkbox"/> React Native</td> </tr> <tr> <td><input type="checkbox"/> SQL</td> <td><input type="checkbox"/> MySQL</td> <td><input type="checkbox"/> SQL Server</td> <td><input type="checkbox"/> PostgreSQL</td> <td><input type="checkbox"/> MongoDB</td> <td><input type="checkbox"/> Redis</td> </tr> <tr> <td colspan="2"><input type="checkbox"/> C++</td> <td colspan="2"><input type="checkbox"/> JavaScript</td> <td colspan="2"><input type="checkbox"/> Angular.js</td> </tr> <tr> <td colspan="6" style="text-align: center;">Crash trabalho, função que desempenha:</td> </tr> <tr> <td colspan="6" style="text-align: center;"> <input type="checkbox"/> Servo <input type="checkbox"/> Osciloscópio <input type="checkbox"/> Feminino         </td> </tr> <tr> <td colspan="6" style="text-align: right; padding-top: 10px;"> <input type="button" value="Sair"/> <input type="button" value="Cancelar"/> <input type="button" value="Enviar"/> </td> </tr> </table>		Nome:	<input type="text" value="Participante 1"/>	Nome Completo:	<input type="text" value="Participante 1"/>	Data de Nascimento:	<input type="text" value="00/00/0000"/>	Competência Básica		Intermediária		Avançada		<input type="checkbox"/> Java	<input checked="" type="checkbox"/> C	<input type="checkbox"/> PHP	<input type="checkbox"/> Python	<input type="checkbox"/> React.js	<input type="checkbox"/> React Native	<input type="checkbox"/> SQL	<input type="checkbox"/> MySQL	<input type="checkbox"/> SQL Server	<input type="checkbox"/> PostgreSQL	<input type="checkbox"/> MongoDB	<input type="checkbox"/> Redis	<input type="checkbox"/> C++		<input type="checkbox"/> JavaScript		<input type="checkbox"/> Angular.js		Crash trabalho, função que desempenha:						<input type="checkbox"/> Servo <input type="checkbox"/> Osciloscópio <input type="checkbox"/> Feminino						<input type="button" value="Sair"/> <input type="button" value="Cancelar"/> <input type="button" value="Enviar"/>					
Nome:	<input type="text" value="Participante 1"/>	Nome Completo:	<input type="text" value="Participante 1"/>	Data de Nascimento:	<input type="text" value="00/00/0000"/>																																												
Competência Básica		Intermediária		Avançada																																													
<input type="checkbox"/> Java	<input checked="" type="checkbox"/> C	<input type="checkbox"/> PHP	<input type="checkbox"/> Python	<input type="checkbox"/> React.js	<input type="checkbox"/> React Native																																												
<input type="checkbox"/> SQL	<input type="checkbox"/> MySQL	<input type="checkbox"/> SQL Server	<input type="checkbox"/> PostgreSQL	<input type="checkbox"/> MongoDB	<input type="checkbox"/> Redis																																												
<input type="checkbox"/> C++		<input type="checkbox"/> JavaScript		<input type="checkbox"/> Angular.js																																													
Crash trabalho, função que desempenha:																																																	
<input type="checkbox"/> Servo <input type="checkbox"/> Osciloscópio <input type="checkbox"/> Feminino																																																	
<input type="button" value="Sair"/> <input type="button" value="Cancelar"/> <input type="button" value="Enviar"/>																																																	

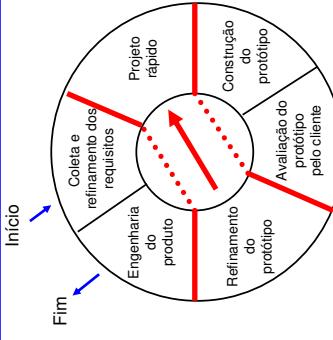
## **Prototipação (4): Vertical e Horizontal**



## **Prototipação (5)**

- A prototipação é apropriada quando:
    - O cliente definiu um conjunto de objetivos gerais para o software, mas não identificou requisitos de entrada, processamento e saída com detalhes.
    - O desenvolvedor não tem certeza da eficiência de um algoritmo ou da forma da interação homem/máquina

Prototipação (3)



## Atividades na Prototipação (1)

- ➊ Coleta e Refinamento dos Requisitos:  
desenvolvedor e cliente definem os objetivos gerais do software, identificam quais requisitos são conhecidos e as áreas que necessitam de definições adicionais.
- ➋ Projeto Rápido: representação dos aspectos do software que são visíveis ao usuário, como as abordagens de entrada e os formatos de saída.

## Atividades na Prototipação (2)

- ➊ Construção do Protótipo: a implementação do projeto rápido serve como o “primeiro sistema”, recomendado que se deixe de lado futuramente.
- ➋ Avaliação do Protótipo: cliente e desenvolvedor avaliam o protótipo.

## Atividades na Prototipação (3)

- ➊ Refinamento do Protótipo: cliente e desenvolvedor refinam os requisitos do software a ser desenvolvido. Ocorre nesse ponto um processo de interação até que as necessidades do cliente sejam satisfeitas e o desenvolvedor compreenda o que precisa ser feito.
- ➋ Engenharia do Produto: identificados os requisitos, o protótipo deve ser descartado e a versão de produção deve ser construída, considerando os critérios de qualidade.

## Problemas com a Prototipação

- ➊ O cliente não sabe que o software, que ele vê, não considerou, durante o desenvolvimento, a qualidade global e a manutenibilidade a longo prazo:
  - Ele não aceita bem a ideia que a versão final do software será construída e “força” a utilização do protótipo como produto final.
- ➋ O desenvolvedor freqüentemente faz uma implementação comprometida (utilizando o que está disponível) com o objetivo de produzir rapidamente um protótipo:
  - Depois de um tempo, ele se familiariza com essas escolhas, e esquece que elas não são apropriadas para o produto final.

## Vantagens da Prototipação

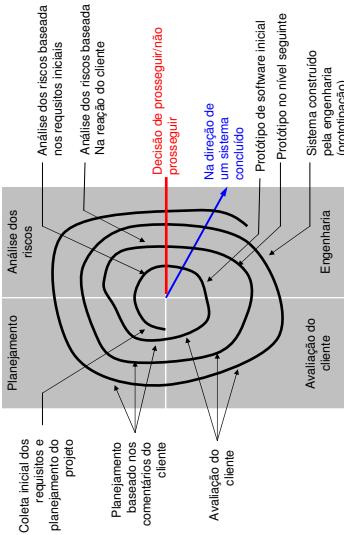
- ➊ Ainda que possam ocorrer problemas, a prototipação é um ciclo de vida eficiente:
  - A chave é definir as regras do jogo logo no começo.
  - O cliente e o desenvolvedor devem ambos concordar que o protótipo seja construído para servir como um mecanismo a fim de definir os requisitos.

## Ciclo de Vida em Espiral (1)

- ➊ Engloba as melhores características do ciclo de vida Clássico como o da Prototipação, adicionando um novo elemento: a Análise dos Riscos.
- ➋ Segue a abordagem de passos sistemáticos do Ciclo de Vida Clássico incorporando-os numa Estrutura Iterativa que reflete mais realisticamente o mundo real.
- ➌ Usa a Prototipação, em qualquer etapa da evolução do produto, como mecanismo de redução de riscos.

## Ciclo de Vida em Espiral (2)

### Atividades do Ciclo de Vida em Espiral



### Comentários sobre o Ciclo de Vida Espiral

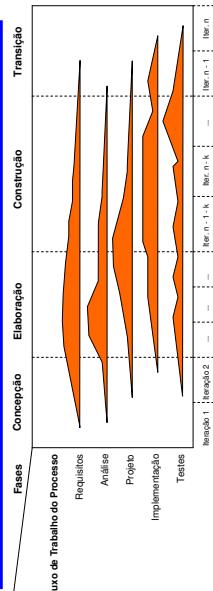
- É uma abordagem realística para o desenvolvimento de software em grande escala.
- Usa uma abordagem que capacita o desenvolvedor e o cliente a entenderem e a reagirem aos riscos em cada etapa evolutiva.
- Pode ser difícil convencer os clientes que uma abordagem "evolutiva" é controlável.
- Exige considerável experiência na determinação de riscos e depende dessa experiência para ter sucesso.
- A cada iteração ao redor da espiral, versões progressivamente mais completas do software são construídas.

## Processo Unificado

- Planejamento: determinação dos objetivos, alternativas e restrições.
- Análise de Risco: análise das alternativas, identificação e resolução dos riscos.
- Construção: desenvolvimento do produto no nível seguinte.
- Avaliação do Cliente: avaliação do produto e planejamento das novas fases.

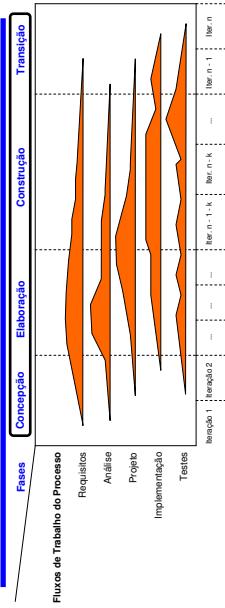
## Processo Unificado

### Fases do Processo Unificado



- O Processo Unificado considera quatro fases no desenvolvimento de software:
  - Iniciação ou Concepção (Inception): ênfase no escopo do sistema.
  - Elaboração (Elaboration): ênfase na arquitetura.
  - Construção (Construction): ênfase no desenvolvimento.
  - Transição (Transition): ênfase na implantação.
- Cada fase é composta pela iteração dos seguintes Fluxos de Trabalho do Processo (Disciplinas):
  - Requisitos.
  - Análise.
  - Projeto.
  - Implementação.
  - Testes.

## Processo Unificado - Fases



## Elaboração - Fases do Processo Unificado

- Os objetivos desta fase são analisar o domínio do problema, estabelecer a arquitetura do mesmo, refinar o plano do projeto e identificar seus maiores riscos.
- Em termos do processo de desenvolvimento, o foco são as atividades de análise e projeto.
- Decisões sobre a arquitetura tem que ser feitas com uma compreensão de todo o sistema: seu escopo, suas principais funcionalidades e seus requisitos não funcionais, tais como os requisitos de desempenho.

## Concepção - Fases do Processo Unificado

- Durante a fase de concepção, o foco está na comunicação com o cliente para a identificação de requisitos e nas atividades de planejamento, a fim de delimitar o escopo do projeto.
- Para conseguir isso, devem ser identificadas todas as entidades externas com as quais o sistema irá interagir (atores), definindo a natureza dessa interação em um alto nível, identificando todos os casos de uso e o plano do projeto.
- Protótipos podem ser construídos para apoiar a comunicação com o cliente.

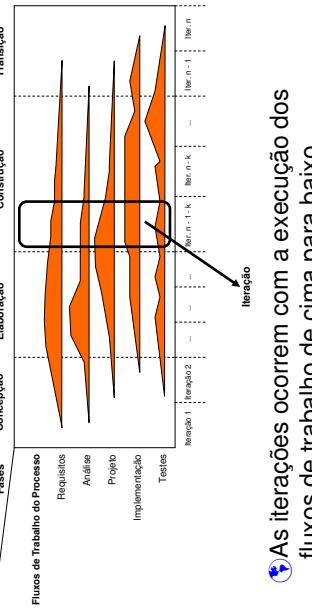
## Construção - Fases do Processo Unificado

- Na fase de construção, todos os demais componentes e as características da aplicação são desenvolvidos e integrados ao produto, sendo que todas as características são exaustivamente testadas.
- A fase de construção é, em certo sentido, um processo de fabricação onde a ênfase é colocada sobre o gerenciamento de recursos e controle de operações para otimizar custos, programações e qualidade.
- O gerenciamento do processo executa uma transição a partir do desenvolvimento da propriedade intelectual durante as fases de concepção e elaboração, para o desenvolvimento do software e da documentação a serem entregues durante a construção e a transição.

## Transição - Fases do Processo Unificado

- O propósito desta fase é fazer a transição do sistema do ambiente de desenvolvimento para o ambiente de produção, ou seja para os usuários.
- Uma vez que o produto tenha sido entregue para o usuário final, surgem normalmente questões como o desenvolvimento de novas versões, a correção de alguns problemas ou mesmo desenvolver os características que foram adiadas.

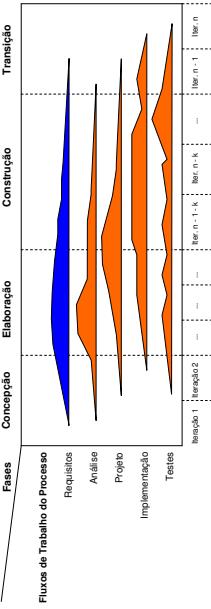
## Processo Unificado



- As interações ocorrem com a execução dos fluxos de trabalho de cima para baixo.

## Requisitos (Processo Unificado)

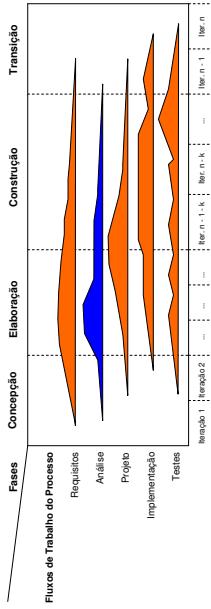
### Requisitos



- O objetivo do fluxo de trabalho de Requisitos é descrever o que o sistema deve fazer. Para conseguir isso, deve-se obter, organizar, documentar as funcionalidades e restrições exigidas.
- A modelagem de Casos de Uso representa os Requisitos Funcionais.
- Um Caso de Uso especifica uma sequência de ações, incluindo caminhos alternativos, que o sistema realiza e que levam a um resultado observável produzido por um determinado ator.

## Análise (Processo Unificado)

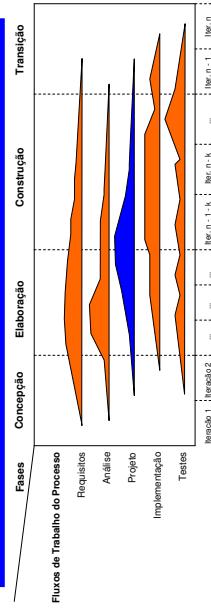
### Análise



- Neste fluxo de trabalho são analisados os requisitos e é montado o modelo de classes e de objetos, com foco nas classes de negócio, mais o dicionário de informações.
- O Modelo de Caso de Uso é a entrada para o Modelo de Análise.
- O Modelo de Análise é uma especificação detalhada dos requisitos, sendo usado pelos desenvolvedores para uma compreensão mais precisa dos Casos de Uso, definindo-os como uma colaboração entre tipos conceituais de objetos.

## Projeto (Processo Unificado)

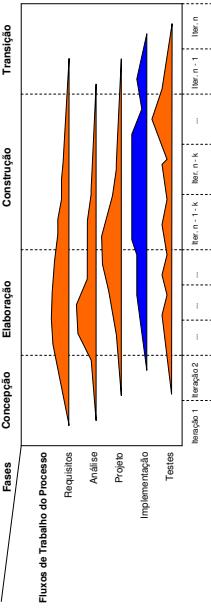
### Projeto



- O Modelo de Projeto é uma descrição da implementação do sistema, a partir do Modelo de Análise, e precisa se adequar ao ambiente de implementação (tecnologia de distribuição de objetos, ambiente gráfico, bancos de dados, reuso de sistemas legados, bibliotecas).
- São incluídos os Diagramas de Sequência, de Estado e de Atividades.

## Implementação (Processo Unificado)

### Implementação

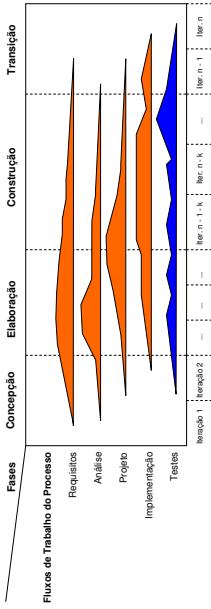


Os objetivos da implementação são:

- Definir a organização do código em termos de subsistemas de implementação, organizados em camadas.
- Implementar classes e objetos em termos de componentes (arquivos-fonte, códigos binários, executáveis e outros).
- Testar os componentes desenvolvidos como unidades.
- Integrar os resultados produzidos por implementadores individuais, ou por equipes, em um sistema executável.

## Testes (Processo Unificado)

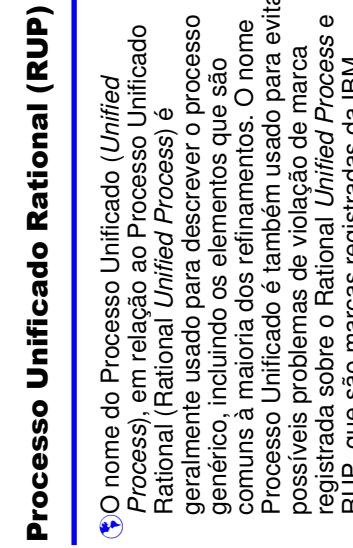
### Testes



Os objetivos da atividade de Testes são:

- Verificar a interação entre os objetos.
- Verificar a integração adequada de todos os componentes do software.
- Verificar se todos os requisitos foram corretamente implementados.
- Identificar e garantir que os defeitos sejam encontrados e corrigidos antes da implantação do software.

## Processo Unificado Rational (RUP)



O nome do Processo Unificado (*Unified Process*), em relação ao Processo Unificado Rational (*Rational Unified Process*) é geralmente usado para descrever o processo genérico, incluindo os elementos que são comuns à maioria dos refinamentos. O nome Processo Unificado é também usado para evitar possíveis problemas de violação de marca registrada sobre o Rational *Unified Process* e RUP, que são marcas registradas da IBM.

Obrigado.

## Tópicos

### Análise e Projeto de Sistemas

• Requisitos.

• Atores.

• Casos de Uso.

• *Include, Extend e Generalização.*

• Cenário Descritivo ou Especificação de Casos de Uso.

### **Requisitos** **Diagramas de Casos de Uso** **(Use Case Diagram)**

## Prof. Laudelino Cordeiro Bastos

### O que é um Requisito?

• Segundo o Dicionário Aurélio, um requisito é:

▫ Uma condição necessária para a obtenção de certo objetivo.

• No dicionário IEEE 610-1990 Standard Computer Dictionary:

▫ Os requisitos são uma condição ou capacidade de que um usuário necessita para resolver um problema ou atingir um objetivo; ou

▫ uma condição ou capacidade que precisa ser atingida por um sistema para satisfazer um contrato, norma, especificação ou algum outro documento.

### Elicitação de Requisitos

- Objetiva descobrir o domínio da aplicação, os serviços que devem ser fornecidos, bem como as restrições.
- Deve envolver cliente, usuários finais, gerentes, pessoal envolvido na manutenção, especialistas no domínio, entre outros. Ou seja, as Partes Envolvidas ou Stakeholders.

### Tipos de Requisitos

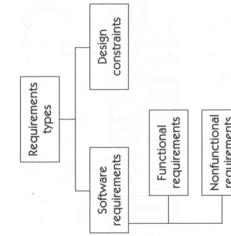
• Segundo Leffingwell e Widrig (2002), os requisitos podem ser divididos em:

▫ Requisitos de Software:

▫ Funcionais.

▫ Não funcionais.

▫ Restrições de Projeto.



### Tipos de Requisitos: Requisitos de Software

• Requisitos Funcionais:

▫ Descrevem o comportamento do sistema, suas ações para cada entrada, ou seja, é **aquilo que tem que ser feito** pelo sistema.

▫ Esses requisitos são usualmente orientados a ações: "quando o usuário executa X, o sistema faz Y".

▫ A maioria dos requisitos funcionais pode ser escrito na forma de uma simples frase declarativa ou na forma de um **Caso de Uso**.

## Tipos de Requisitos: Requisitos de Software

### Requisitos Não Funcionais:

- ❑ São aqueles que expressam **como deve ser feito** o sistema. São muito importantes, pois definem se o sistema será eficiente para a tarefa que se propõe a fazer ou não. Um sistema ineficiente certamente não será utilizado.
- ❑ Neles também são apresentados restrições e especificações de uso para os requisitos funcionais.

## Tipos de Requisitos: Requisitos de Projeto

### Restrições de Projeto:

- ❑ São aqueles requisitos que **impõe limitações** sobre o projeto do sistema, ou sobre os processos que são utilizados para construir um sistema. Por exemplo:
  - ❑ Sempre que possível, as opções de projeto devem ser deixadas livres para os desenvolvedores, ao invés de serem indicadas como requisitos, a fim de que sejam escolhidas as melhores possibilidades técnicas e econômicas.
- ❑ Desenvolver um projeto utilizando Oracle Database Express Edition é uma restrição de projeto.

## Tipos de Requisitos: Classificação dos Requisitos Não Funcionais

### Classificação dos Requisitos Não Funcionais:

❑ Grady (1992) classifica os Requisitos Não Funcionais em:

- ❑ Usabilidade (Usability).
- ❑ Confiabilidade (Reliability).
- ❑ Desempenho (Performance).
- ❑ Suportability.

## Tipos de Requisitos: Classificação dos Requisitos Não Funcionais

### Usabilidade:

❑ O esforço para aprender, operar, preparar a entrada e interpretar a saída de um sistema.

### Confiabilidade:

- ❑ Quanto que se pode esperar que um programa execute a função pretendida com a precisão exigida.
- ❑ A Confiabilidade pode ainda ser dividida em:
  - ❑ Disponibilidade (Availability); por exemplo, 99% do tempo em funcionamento.
  - ❑ Tempo médio entre falhas (Mean time between failures - MTBF).
  - ❑ Tempo médio para reparação (Mean time to repair - MTTR); por exemplo, o tempo médio para reparação de uma falha no sistema é de 5 minutos.
  - ❑ Acurácia (Accuracy); A precisão das computações e do controle de um sistema.
  - ❑ Taxa de Defeitos (Defect Rate); a quantidade de erros de um sistema, por exemplo, quantidade de erros por milhares de linhas de código
  - ❑ Erros por tipo (Bugs per type); geralmente caracterizados por erros pequenos, significantes ou críticos.

## Tipos de Requisitos: Classificação dos Requisitos Não Funcionais

### Desempenho:

- ❑ Qual o desempenho esperado em relação à carga que o sistema recebe.

### Supportability (Manutenabilidade e Flexibilidade):

Indica o nível de facilidade com que um sistema pode ser modificado para acomodar melhoramentos e reparos.

### Exemplos

#### Exemplos de Requisitos Funcionais:

- ❑ RF01: O médico pode inserir um paciente.
  - ❑ RF02: O usuário pode pesquisar todo ou um subconjunto do banco de dados.
- ❑ Exemplos de Requisitos Não Funcionais:
    - ❑ RNF01: O médico pode inserir um paciente que já esteja cadastrado.
    - ❑ RNF02: As consultas ao banco de dados não devem ultrapassar 3 segundos.

- ❑ Exemplo de Restrições de Projeto:
  - ❑ RP01 : Desenvolver o projeto utilizando o Sistema Gerenciador de Banco de Dados MySQL.

Grady, R. Practical Software Metrics for Project Management and Process Improvement. Englewood Cliffs, NJ: Prentice-Hall, 1992

## Diagrama de Casos de Uso

- ➊ O Diagrama de Casos de Uso é utilizado nas atividades de negócios e representação dos requisitos da engenharia de software.
- ➋ Descreve o que o novo sistema deverá fazer ou o que um sistema existente faz. É utilizado para agrupar os requisitos e descrever as funcionalidades do sistema, representando também os papéis de negócio que interagem com as funcionalidades deste sistema.

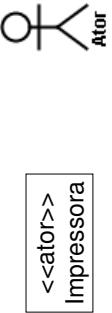
## Diagrama de Casos de Uso

- ➊ Define inicialmente **o que o sistema faz, não pensando em como irá fazê-lo**. Portanto não se preocupa com questões de implementação, mas sim com as necessidades do negócio.

## Diagrama de Casos de Uso

### Atores

- ➊ São agentes que interagem com o sistema.
- ➋ Os atores são entidades externas (já não utilizado na análise estruturada essencial) que tem interesse em interagir com o sistema.
- ➌ Representam papéis no negócio.
- ➍ Um ator pode ser representado através de retângulos, sempre indicando que se trata de um ator, ou através de ícones humanos:



## Diagrama de Casos de Uso

### Atores

- ➊ Um ator pode ser:
  - ➌ Tipo de usuário (ex: Gerente, Motorista, Profissional de Saúde, entre outros).
  - ➍ Hardware (ex: eletrocardiógrafo, controlador de temperatura, dispositivo GPS, entre outros).
  - ➎ Software (ex: sistema de estoque, folha de pagamento, prontuário eletrônico do paciente, banco de dados, entre outros).

## Diagrama de Casos de Uso

### Há 3 tipos de Atores:

- ➊ **Ator Principal:** tem objetivos de usuários satisfeitos por uso do sistema a ser desenvolvido.
  - ➌ Por que identificar? Para encontrar os objetivos do usuário, que guiam os casos de uso.
- ➋ **Ator de Suporte:** fornece um serviço (como, por exemplo, informações) para o sistema.
  - ➌ Por que identificar? Para esclarecer interfaces externas e protocolos de comunicação.
- ➌ **Ator de Bastidor:** tem interesse no comportamento do caso de uso, mas não é um ator principal ou de suporte (como, por exemplo, um órgão governamental).
  - ➌ Por que identificar? Para garantir que todos os interesses necessários estejam identificados e satisfeitos. Interesses deste tipo de ator são sutis ou de fácil esquecimento, a menos que sejam explicitamente nomeados.

## Diagrama de Casos de Uso

### Caso de Uso

- ➊ Caso de Uso (Use Case)
  - ➌ São funcionalidades requeridas do sistema. Cada Caso de Uso representa uma funcionalidade completa, conforme percebida pelo ator. O caso de uso não deve ser confundido com os conceitos de módulo ou função.



## Diagrama de Casos de Uso

Responder às seguintes perguntas pode auxiliar a encontrar casos de uso:

- Quais funções o ator necessita do sistema?
- O que o ator precisa fazer?
- O ator precisa criar, apagar, ler, armazenar ou modificar alguma informação no sistema?
- O ator precisa ser notificado de eventos do sistema?
- O ator precisa notificar o sistema sobre algum evento?
- O trabalho diário do ator poderia ser simplificado ou tornado mais eficiente através de novas funcionalidades do sistema?
- Quais entradas e saídas o sistema necessita?
- Quais os principais problemas com o método ou sistema atual?

## Relacionamentos entre Atores e Casos de Uso

Os relacionamentos em um Diagrama de Casos de Uso podem envolver:

- Dois atores.
- Um ator e um caso de uso.
- Dois casos de uso.

## Relacionamentos entre Atores

Como os atores são entidades externas, da mesma forma que as entidades externas na Análise Estruturada Essencial, as relações entre eles não fazem parte do sistema.

Mesmo assim, é possível incluí-las nos diagramas de casos de uso, normalmente para auxiliar na representação do modelo de negócios da empresa.



O relacionamento de generalização ou especialização indica que um ator é um caso específico de outro ator.

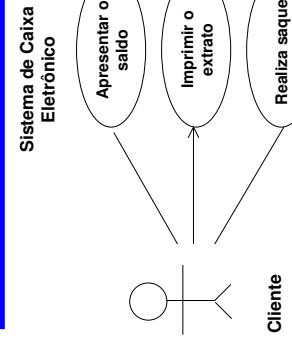


## Relacionamentos entre Atores e Casos de Uso

O relacionamento entre um ator e um caso de uso expressa uma comunicação entre ambos.

Os atores se comunicam com o sistema sempre através dos casos de uso.

As setas podem indicar a ativação do caso de uso ou a indicação do sentido do fluxo de dados nas comunicações.



## Relacionamentos entre Atores e Casos de Uso (Diagrama)

## Relacionamentos entre Casos de Uso

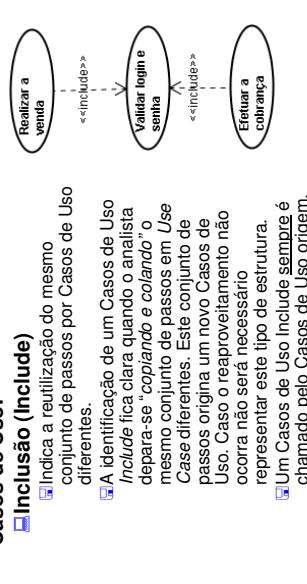
### Diagrama de Casos de Uso: Include

- As relações entre caso de uso e caso de uso normalmente não são de comunicação. Os casos de uso são aplicações completas de um sistema, se comunicando normalmente através do banco de dados do sistema.
- Os relacionamentos entre casos de uso são:
  - Inclusão (include).
  - Extensão (extend).
  - Generalização.

#### Tipos de Relacionamentos entre Casos de Uso:

##### ■ Inclusão (include)

- Indica a reutilização do mesmo conjunto de passos por Casos de Uso diferentes.
  - A identificação de um Caso de Uso *Include* liga clara quando o analista depara-se “copiando e colando” o mesmo conjunto de passos em Use Cases diferentes. Este conjunto de passos origina um novo Casos de Uso. Caso o reaproveitamento não ocorra não será necessário representar este tipo de estrutura.
  - Um Casos de Uso *Include* sempre é chamado pelo Casos de Uso origem.

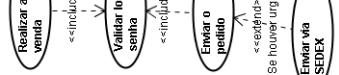


### Diagrama de Casos de Uso: Extend

#### Tipos de Estruturação entre Casos de Uso (cont.)

##### ■ Extensão (Extend):

- Indica uma extensão de uma funcionalidade, ou seja, caso identifique algum conjunto de passos que são realizados somente sob determinada condição, pode-se criar um outro Caso de Uso com uma relação de extensão.
  - Diferentemente da relação *include*, o Caso de Uso de extensão nem sempre será executado. O Caso de Uso estendido pode acessar e modificar propriedades de um Caso de Uso base, processo que não é realizado um uma relação *Include*.
  - No exemplo, envia via SEDEX apenas se houver urgência.

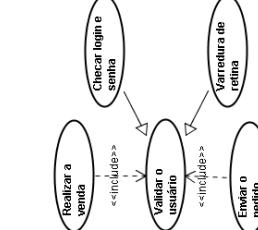


### Diagrama de Casos de Uso: Generalização

#### Tipos de Estruturação entre Casos de Uso (cont.)

##### ■ Generalização:

- Indica um relacionamento entre um Caso de Uso geral e outro mais específico.
  - Por outro lado, o caso de uso geral é uma generalização ou abstração dos casos de uso mais específicos.



## Especificação de Casos de Uso (Cenário descritivo)

### Especificação de Casos de Uso (Cenário descritivo)

- Provê uma melhor compreensão e detalhamento dos Casos de Uso modelados.

#### Estrutura e composição:

##### ■ Nome:

##### ■ Descruição:

- Descrição sucinta do Use Case. Ex: Realiza a emissão de Nota Fiscal (NF) e, após isso, será solicitada a 'baixa' no estoque dos produtos relacionados.

#### Estrutura e composição (cont.):

##### ■ Pré-condições:

- Mencionar pré-condições para a execução do Caso de Uso. A pré-condição determina qual deve ser o estado do sistema no inicio do use case. Pode não se aplicar em alguns use cases (é opcional), e devem ser utilizadas apenas quando for embutir clareza ao use case. Ex: Cliente e Transportador as devem estar cadastradas e Produto deve existir em estoque.

##### ■ Pós-condições:

- Listar as pós-condições após a realização do Caso de Uso. A pós-condição determina qual deve ser o estado do sistema no final do use case. A pós-condição deve ser verdadeira não importa qual ramo ou alternativa é seguida pelo use case. Pode não se aplicar em alguns use cases (é opcional), e devem ser utilizadas apenas quando for embutir clareza ao use case. Ex: Produto atualizado em estoque (caso a exceção exemplificada anteriormente não existisse).

## Especificação de Casos de Uso (Cenário descritivo)

## Especificação de Casos de Uso (Cenário descritivo)

### • Estrutura e composição (cont.):

#### ■ Passos do Caso de Uso - Fluxo Básico:

□ Lista de passos a serem seguidos pelo caso de uso sob a visão do usuário. Estas tarefas consideram situações de perfeição, onde a execução de uma está condicionada ao término de outra(s).

#### ■ Passos do Caso de Uso - Fluxos Alternativos:

□ Seqüências alternativas de eventos. Podem representar alguma variação em relação ao fluxo básico de eventos ou uma seqüência de eventos diferente, ou seja, um outro cenário que esteja relacionado aos objetivos e funcionalidades oferecidas pelo caso de uso. Uso de Help, limpeza de campos,...

## Cenário dos Casos de Uso

• Normalmente descrito utilizando-se um editor de texto.

### • Partes constituintes do Cenário dos Casos de Uso:

#### ■ Diagrama de Caso de Uso: nome do diagrama.

□ Descrição: o que o diagrama de casos de uso faz.

□ Pré-condições: condições necessárias para o diagrama poder ser executado.

□ Pos-condições: resultados da execução do diagrama de casos de uso.

□ Fluxo Básico: tarefas que consideram situações de perfeição.

□ Fluxo Alternativo: variações em relação ao fluxo básico de eventos.

□ Regras de Negócio: condições ou restrições sobre os processos de negócio.

□ Fluxo de Exceção: executado nas situações nas quais o fluxo não pode ser concluído.

## Especificação de Casos de Uso (Cenário descritivo)

### • Estrutura e composição (cont.):

#### ■ Passos do Caso de Uso - Fluxos de Exceção:

□ Determinam as ações que devem ser tomadas em situações nas quais o fluxo não possa ser concluído, devido a alguma regra de negócio ter sido quebrada, tanto no fluxo básico como nos fluxos alternativos.

#### ■ Regras de Negócio:

□ Condições ou restrições sobre os processos de negócio, ou seja, sobre a forma como o negócio é executado. As Regras de Negócio devem ser checadas nos fluxo de eventos do caso de uso.

## Estudo de Caso: Controle de Leitos de UTI

## Declaração dos Objetivos do Sistema

### • Objetivo geral:

□ Controlar a utilização dos leitos de uma UTI.

### • Objetivos específicos:

□ Controlar o serviço de reservas de leitos de UTI.

□ Controlar o serviço de liberação de leitos.

□ Controlar a limpeza dos leitos.

□ Emitir relatórios sobre a reserva de leitos.

## Requisitos Identificados

• Quando o Médico Assistente necessita de uma vaga de UTI, solicita a reserva de vaga.

□ O Médico Assistente pode cancelar a solicitação da reserva de vaga.

□ O Médico Intensivista recebe dados da nova Solicitação de Reserva de Vaga (SRV) para análise.

□ Quando o Médico Intensivista analisa a SRV, informa se a solicitação foi aceita ou não.

## Requisitos Identificados

- Profissional de Saúde faz o seu atendimento ao paciente e registra esses dados.
- Profissional de Saúde solicita a emissão de relatórios referente às internações na UTI.
- O Médico Intensivista comunica a alta do paciente.
- A Equipe de Higienização recebe dados da liberação do leito, através da alta do paciente.
- Quando ocorre o final da higienização do leito a mesma deve ser indicada.

## Atores Identificados

- Médico Assistente.
  - Médico Intensivista.
  - Equipe de Higienização.
  - Profissional de Saúde.
- 
- 
- 
- 

## Requisitos Funcionais Identificados

- Solicitação de reserva de vaga de UTI pelo Médico Assistente. Representada pelos requisitos funcionais **"Médico Assistente pode solicitar reserva de leito - FR01"** e **"Médico Assistente pode Registrar reserva de leito - FR02"**.
- Cancelamento da solicitação da reserva de vaga pelo Médico Assistente. Representado pelo requisito funcional **"Médico Assistente pode solicitar o cancelamento da reserva - FR03"**.
- Efetivação da Solicitação de Reserva de Vaga, informando se a solicitação foi aceita ou não. Representada pelo requisito funcional **"Médico Intensivista pode efetivar a reserva de leito - FR04"**.

## Requisitos Funcionais Identificados

- Registro do atendimento ao paciente pelo Profissional de Saúde. Representado pelo requisito funcional **"Profissional de Saúde Registra o atendimento - FR05"**.
- Emissão de relatórios, referentes às internações na UTI, pelo Profissional de Saúde. Representada pelo caso de uso **"Profissional de Saúde solicita a emissão de relatórios - FR06"**.
- Comunicação da alta do paciente pelo Médico Intensivista, informação para a Equipe de Higienização sobre os dados da liberação do leito, indicação do final da higienização pela Equipe de Higienização. Representação pelo caso de uso **"Médico Intensivista pode conceder alta ao paciente - FR07"**.

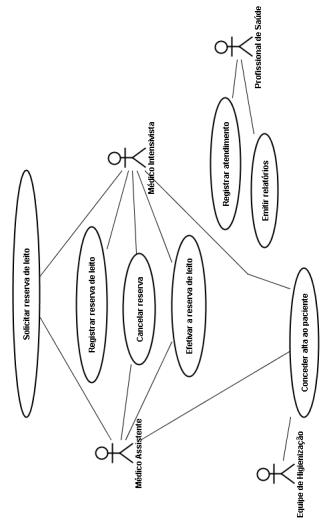
## Casos de Uso Identificados

- Solicitação de reserva de vaga de UTI pelo Médico Assistente. Representada pelos casos de uso **"Solicitar reserva de leito - UC01"** e **"Registrar reserva de leito - UC02"**.
- Cancelamento da solicitação da reserva de vaga pelo Médico Assistente. Representado pelo caso de uso **"Cancelar reserva - UC03"**.
- Efetivação da Solicitação de Reserva de Vaga, informando se a solicitação foi aceita ou não. Representada pelo caso de uso **"Efetivar a reserva de leito - UC04"**.

## Casos de Uso Identificados

- Registro do atendimento ao paciente pelo Profissional de Saúde. Representado pelo caso de uso **"Registrar atendimento - UC05"**.
- Emissão de relatórios, referentes às internações na UTI, pelo Profissional de Saúde. Representada pelo caso de uso **"Emitir relatórios - UC06"**.
- Comunicação da alta do paciente pelo Médico Intensivista, informação para a Equipe de Higienização sobre os dados da liberação do leito, indicação do final da higienização pela Equipe de Higienização. Representação pelo caso de uso **"Conceder alta ao paciente - UC07"**.

## **Diagrama de Casos de Uso Visão Geral - Controle Leitos de UTI**



## **Especificação dos casos de uso: Solicitar reserva de leito**

<b>Nome</b>	UC001 - Solicitar reserva de leito
<b>Atores</b>	Autor Principal: Médico Assistente. Aktor de Suporte: Médico Intensivista.
<b>Descrição</b>	Caso de uso executado quando o médico assistente necessita realizar uma solicitação de reserva de leito.
<b>Pré-condições</b>	- Paciente deve estar cadastrado. - Médico Assistente tem que estar autorizado.
<b>Pos-condições</b>	Reserva de leito solicitada.

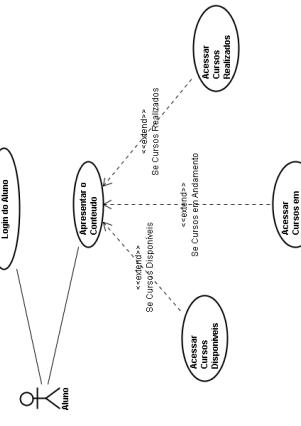


	<b>Fluxo Básico</b>	<b>Ações dos atores</b>	<b>Ações do sistema</b>
1.	Médico Assistente seleciona a opção de solicitação da reserva de leito.		2.- Exibe a tela de solicitação da reserva de leito com as seguintes informações: nome do paciente, nome de nascimento, sexo, diagnóstico, equipamentos necessários, urgência.
3.	O Médico Assistente preenche as informações.		
4.	O Médico Assistente confirma as informações.		
			5 - Verifica as informações conforme as Regras de Negócio RN001 e RN002. Caso haja uma regra de negócios do atendida, vai para 5... 6 - Grava todas as informações. 7 - Registra a data da solicitação. 8 - Envia a solicitação ao Médico Intensivista.
			<b>Regras de Negócio</b>
			[RN001] O campo "Nome do paciente" é obrigatório. [RN002] A data de nascimento deve ter o formato dd/mm/aaaa.

## **Especificação dos casos de uso: Solicitar reserva de leito**

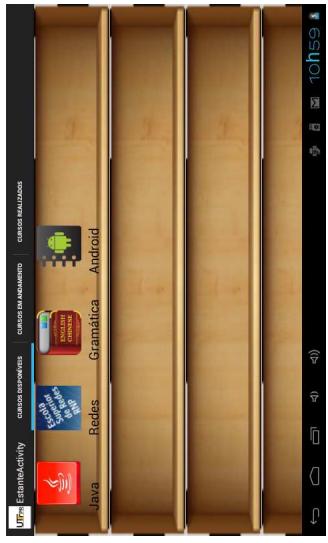
Fluxo Alternativo 1		Fluxo Alternativo 2		Fluxo de Execução	
Ações dos atores	Ações do sistema	Ações dos atores	Ações do sistema	Ações dos atores	Ações do sistema
1 - A qualquer momento o usuário seleciona cancelar.	2 - Desconsidera informações digitadas e encerra o caso de uso.	1 - A qualquer momento o usuário seleciona limpar.	2 - Limpas todas as informações digitadas.		5.a. Caso RN003 não seja atendido, apresenta a mensagem 'Nome deve ser obrigatório' e volta para o Fluxo Básico 3. Caso RN002 não seja atendido, apresenta a mensagem 'Data de nascimento deve ter o formato dd/mm/aaaa' e volta para o Fluxo Básico 3.

## **Diagrama de Casos de Uso Visão Geral - EaD Remota Tablet**



## **Estudo de Caso: EaD Remota Tablet**

## Especificação dos casos de uso: Apresentar o Conteúdo



## Especificação dos casos de uso: Apresentar o Conteúdo

Name	UC02: Apresentar o Conteúdo
Atores	Alunos.
Descrição	Descreve as etapas percorridas por um aluno para utilizar o sistema.
Pré-condições	Estar autenticado no sistema.
Pós-condições	Ter feito uma opção da apresentação do conteúdo.

## Especificação dos casos de uso: Apresentar o Conteúdo

Fluxo Básico	Ações do sistema	Ações do sistema
Acões dos atores	1 - O sistema apresenta a interface representando uma estante com as opções: a) Cursos Disponíveis; b) Cursos em Andamento; e c) Cursos Realizados.	Regras de Negócio
2 - O usuário escolhe uma das opções apresentadas.		Não há neste caso de uso.
	3 - Caso seja escolhida a opção Cursos Disponíveis, vai para 4. Caso seja escolhida a opção Cursos em Andamento, vai para 6. Caso seja escolhida a opção Cursos Realizados, vai para 8.	Fluxo Alternativo 1
	4 - O sistema apresenta os ícones dos cursos disponíveis na estante virtual.	Ações do atores
	5 - O usuário escolhe um dos cursos disponíveis apresentados.	Fluxo Alternativo 1
	6 - O sistema apresenta os ícones dos cursos em andamento na estante virtual.	Ações do sistema
	7 - O usuário escolhe um dos cursos em andamento apresentados.	2 a. Foi escolhida a opção voltar e o sistema retorna ao caso de uso "Log do Aluno".
	8 - O sistema apresenta os ícones dos cursos realizados na estante virtual.	5 a. Foi escolhida a opção voltar e o sistema retorna à ação do sistema número 1.
	9 - O usuário escolhe um dos cursos realizados apresentados.	7 a. Foi escolhida a opção voltar e o sistema retorna à ação do sistema número 1.
		9 a. Foi escolhida a opção voltar e o sistema retorna à ação do sistema número 1.

## Especificação dos casos de uso: Apresentar o Conteúdo

Name	Regras de Negócio
Atores	Não há neste caso de uso.
Descrição	
Pré-condições	
Pós-condições	

## Especificação dos casos de uso: Apresentar o Conteúdo

Fluxo Básico	Ações do sistema	Ações do sistema
Acões dos atores	1- Caso o usuário escolha o ícone de curso disponível, o sistema executa o caso de uso "Acessar Cursos em Andamento", a partir do curso disponivel escolhido.	Fluxo Alternativo 2 - Acessar Cursos Disponíveis
	1- Caso o usuário escolha um dos cursos disponiveis apresentados, o sistema executa o caso de uso "Acessar Cursos em Andamento", a partir do curso disponivel escolhido.	Ações do sistema
	1- Caso o usuário escolha o ícone de curso realizado, o sistema executa o caso de uso "Acessar Cursos Realizados", a partir do curso realizado escolhido.	Fluxo Alternativo 3 - Acessar Cursos em Andamento
	1- Caso o usuário escolha um dos cursos realizados apresentados, o sistema executa o caso de uso "Acessar Cursos Realizados", a partir do curso realizado escolhido.	Ações do sistema
		Fluxo de Execução
		Não há neste caso de uso.

## Resumo para Elaborar o Diagrama de Casos de Uso de um Sistema

- Identificar os Requisitos Funcionais do sistema.
- Relacionar os Requisitos Funcionais com Casos de Uso do Sistema.
- Identificar os Atores do Sistema.
- Identificar as Associações entre Atores e Casos de Uso.
- Elaborar o Diagrama de Casos de Uso.
- Descrever os cenários dos Casos de Uso
- Verificar relacionamentos entre casos de uso: inclusão, extensão e generalização.

Obrigado.

## Classes

### Análise e Projeto de Sistemas

#### Diagrama de Classes

#### Dicionário de Informações

**Prof. Laudelino Cordeiro Bastos**

## Classes

- Para a construção de classes:
  - Identificar os substantivos.
  - Verificar palavras importantes no contexto.
  - Não incluir classes desnecessárias:
    - Redundantes, vagas, irrelevantes.
    - Atributos, operações ou métodos.
    - Papéis.
    - Construções de implementação.

## Classes

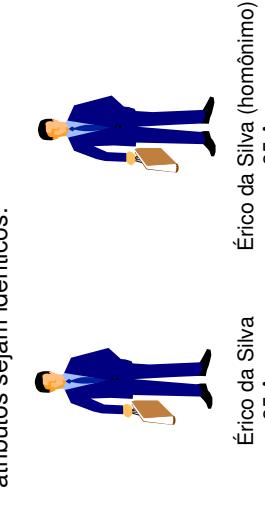
- A classe é a descrição de um tipo de objeto.
- A classe descreve as propriedades (conjunto de atributos) e comportamento (conjunto de operações) de um tipo de objeto.
- É o agrupamento de objetos com a mesma estrutura de dados (atributos) e comportamento (operações), segundo Rumbaugh.
- Cada objeto é uma *instância* de uma classe:
  - *Instanciar* uma classe é criar um novo objeto da classe.

## Classe e Objeto

Pessoa	Erico. Pessoa
nome	Erico da Silva
enderecoResidencial	endereçoResidencial: R. Sete, 54
telefoneResidencial	telefoneResidencial: 333334444
celular	celular: 999998888

Classe

Objeto	Erico da Silva
	Erico da Silva (homônimo) 35 Anos



## Atributos

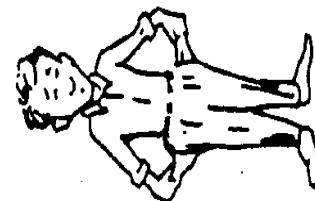
### Atributos

- ➊ Descrevem as características presentes nos objetos.
- ➋ O valor de todos os atributos de um objeto em um dado momento definem o seu estado.
- ➌ Cada instância poderá ter valores diferentes para os mesmos atributos.
- ➍ Somente os atributos que são de interesse do sistema devem ser descritos na classe.



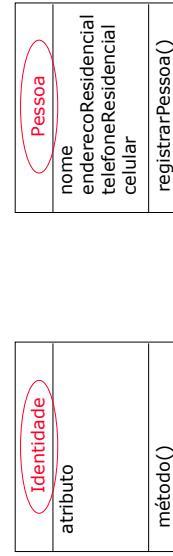
## Atributos

- ➊ Empregado:
- ➋ Nome.
- ⌚ Salário.
- ⌚ Departamento.
- ⌚ Cargo.



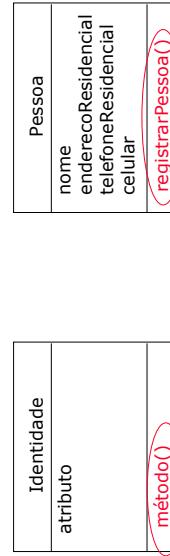
## Diagrama de Classes

- ➊ Identidade (nome da classe):
  - ⌚ Cada classe possui um nome ou identidade que a diferencia das demais classes.



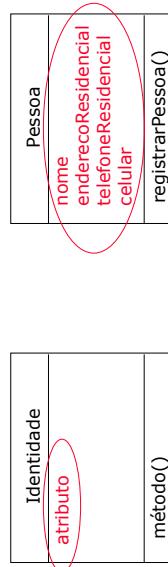
## Diagrama de Classes

- ➊ Método:
  - ⌚ Representa uma "habilidade". Definir um método em uma classe implica em que todos os objetos daquela classe são capazes de realizar aquela função.



## Diagrama de Classes

- ➊ Atributo:
  - ⌚ Representa uma informação que é necessária a determinada classe para que consiga realizar suas tarefas, ou seja, executar seus métodos.
- ⌚ Importante: não confundir Classes com Entidades de Dados ou Tabelas, onde se identificam atributos de chave primária (PK) ou chave estrangeira (FK). Estes tipos de atributos não são utilizados na Orientação a Objetos.



## Associações

### Associações: Exemplo

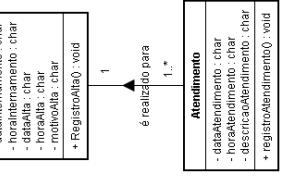
Denotam relacionamento semântico entre duas classes.

São semelhantes aos relacionamentos utilizados no modelo relacional (banco de dados).

Inclui modelagem de cardinalidades (é uma forma de medir a quantidade de elementos de um conjunto).

As associações possuem identificação que explicam a natureza do relacionamento.

Associação em UML:



- Relacionamento no DER:
  - Necessidade de criação de chave primária. A chave primária é um atributo cujo valor distingue cada instância em uma tabela.

## Multiplicidade ou Cardinalidade

Descrevem o número de objetos associado a outro objeto.

Notação utilizada:

- 0..1 (zero a um)
  - 1..1 (um e somente um)
  - 0..\* ou \* (zero a muitos, ou simplesmente muitos)
  - 1..\* (um a muitos)
  - 2 (somente dois)
  - 2..5 (dois a cinco)
  - 1,4,6,12 (um, quatro, seis ou doze)

## Agregação

Trata-se de um caso especial de associação.

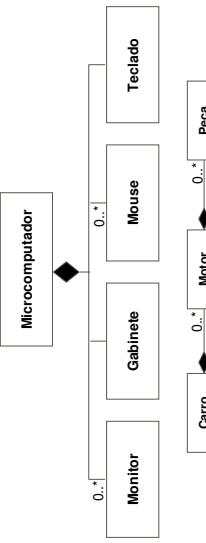
A hierarquia de agregação permite representar no Modelo de Objetos as estruturas "parte-todo" ou "parte-de".

Um objeto (o agregado) contém, como atributos seus, outros objetos, de sua ou de outras classes. Não faz sentido agrregar objetos que não compõe ou montam o outro. Ex.: um endereço não compõe uma pessoa.



## Composição

É uma forma especial de Associação.  
Em uma composição, um objeto poderá ser uma parte de somente uma composição em determinado momento. Não faz sentido agrregar objetos que não compõe ou montam o outro. Ex.: um endereço não compõe uma pessoa.



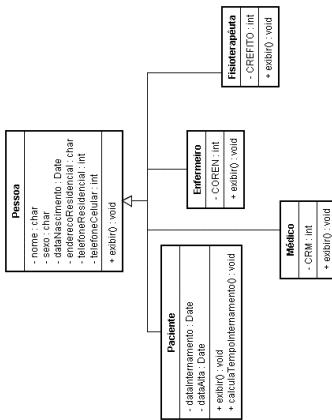
## Herança

É o compartilhamento de **atributos** e **operações** entre as classes baseado em uma relação hierárquica (Rumbaugh).

Qualquer mecanismo que permite um objeto **incluir** toda ou parte da definição de outro objeto como parte de sua própria definição (Yourdon).

## Herança

## Herança



## Herança: nomenclatura

- ➊ Super-Classe e Sub-Classe.
- ➋ Classe Pai e Classe Filho.
- ➌ Classe Base e Classe Derivada.
- ➍ Generalização e Especialização.
- ➎ Um dos objetivos do uso da herança:
  - ➏ Reuso de código.

## Herança

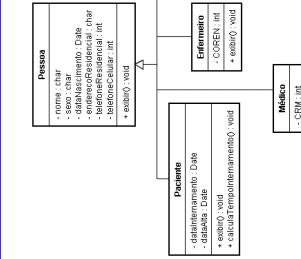
- ➊ Uma classe pode ser definida abrangentemente, e então ser refinada em sucessivas sub-classes específicas.
- ➋ Cada subclasse incorpora, ou herda todas as propriedades de sua superclasse e adiciona suas propriedades específicas.
- ➌ As propriedades da superclasse não precisam ser repetidas na subclasse.

Relação      ou      é-um      or      is-a

## Herança

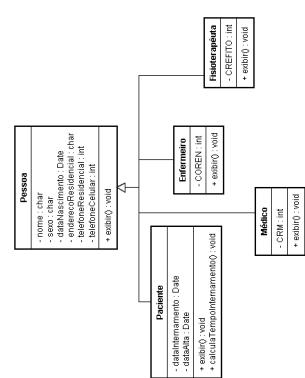
## Generalização

- ➊ Reutilização de atributos e métodos.
- ➋ Organização em camadas mais abstratas ou mais específicas.
- ➌ Herança simples: Uma classe base.
- ➍ Herança múltipla: mais de uma classe base.

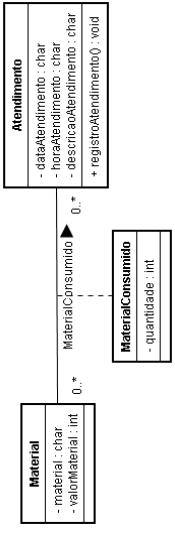


## Especialização - Especificação

### Classe de Associação



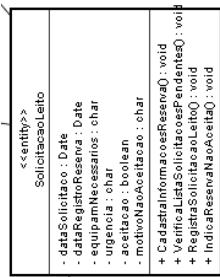
💡 Uma classe de associação é um elemento de modelagem que tem propriedades de associação e de classe, podendo ser vista tanto como uma associação que tem propriedades de classe ou como uma classe que tem propriedades de associação. Surge apenas a partir de uma associação com multiplicidade "muitos para muitos" e se houverem atributos a serem considerados.



### Diagrama de Classes: inclusão de estereótipos

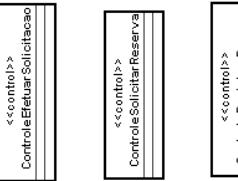
💡 Esterótipos indicam o tipo de Classes:

▣ entidade (entity): identifica as classes cujo papel principal é armazenar dados. Contém dados recebidos ou gerados pelo sistema.



### Diagrama de Classes: inclusão de estereótipos

▣ Controle (control): identifica as classes cujo papel é executar processos. Normalmente apresentam o fluxo de execução de casos de uso, podendo comandar outras classes na execução de processos. Interpretam os eventos ocorridos com os objetos *boundary*, repassando-os para os objetos *entity*.



### Diagrama de Classes: inclusão de interface

▣ Fronteira (boundary): o papel principal dessas é realizar a interface com os atores. Contém o protocolo de comunicação com monitores, impressoras, placas de rede, entre outros.



### Diagrama de Classes: visibilidade para os atributos

▣ Visibilidade. Indica se o atributo pode ser acessado de fora da classe:  
▣ +: visibilidade pública. O atributo é visível fora da classe.  
▣ -: visibilidade privada. Apenas métodos da classe podem acessar o atributo.  
▣ #: visibilidade protegida. Métodos membros de classes derivadas podem acessar o atributo.

## **Diagrama de Classes: visibilidade para os métodos**

- ⌚ Visibilidade. Indica se o método pode ser chamado de fora da classe:
  - ▣+: visibilidade pública. O método é visível fora da classe. Funções internas à classe ou externas a ela pode acessar o método.
  - ▣-: visibilidade privada. Apenas funções membro da classe podem chamar o método.
  - ▣#: visibilidade protegida. Funções membros de classes derivadas podem chamar o método.

## **Diagrama de Classes: complementação da teoria**

- ⌚ O Modelo de Classes é um modelo que poderá ser criado através do levantamento de requisitos ou surgirá como consequência das fontes de informação.
- ▣ Trata-se de um diagrama (documento) único, que permite visualizar todos os objetos de entidade da análise, bem como os relacionamentos entre eles.
- ⌚ O modelo é criado consolidando-se os objetos e todas as descrições de casos de uso em um único diagrama, identificando as estruturas hierárquicas, herança, agregação e associações.

## **Diagrama de Classes: complementação da teoria**

- ⌚ Recomendações para identificação de Classes:
  - ▣ Identificar os substantivos através de documentos referentes ao negócio (ou seja, o problema a ser resolvido).
  - ▣ Buscar classes já identificadas em projetos existentes.

## **Diagrama de Classes: complementação da teoria**

- ⌚ Perguntas que podem ser feitas durante a investigação para a criação das classes:
  - ▣ Existem informações que devem ser registradas ou transformadas? Se houver qualquer informação que deve ser armazenada, transformada, analisada ou manipulada de qualquer forma, então existe pelo menos uma classe, responsável por esses atributos.
  - ▣ Existem sistemas externos? Se sim, normalmente são de interesse no modelo. Sistemas externos devem ser vistos como classes, com o qual o nosso sistema interage.

## **Diagrama de Classes: complementação da teoria**

- ⌚ Perguntas que podem ser feitas durante a investigação para a criação das classes (continuação):
  - ▣ Existem representações organizacionais? Outras áreas de negócio, departamentos, filiais, etc, que sejam relevantes ao sistema em análise normalmente são representados como classes, especialmente em modelos de negócio.
  - ▣ Quals papéis os atores representam no ambiente de negócio? Estes papéis podem ser vistos como classes. Exemplo: usuário, operador do sistema, cliente, etc.

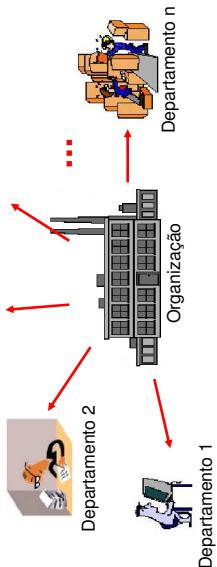
## **Diagrama de Classes: complementação da teoria**

- ⌚ Perguntas que podem ser feitas durante a investigação para a criação das classes (continuação):
  - ▣ Existem dispositivos que serão manipulados pelo sistema? Quaisquer dispositivos técnicos conectados ao sistema são candidatos a classes.
  - ▣ Existem padrões, bibliotecas, componentes ou outros itens? Se existem (de projetos anteriores, parceiros ou fornecedores), elas normalmente contêm candidatos a classes.

## Informações

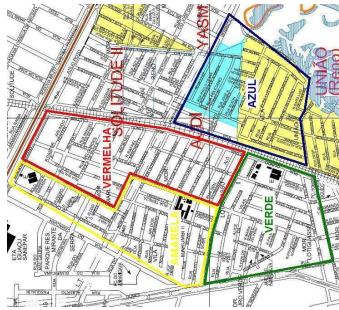
### Quais Informações são Necessárias? (1)

- ⌚ Fator fundamental no desenvolvimento de sistemas de informação.



### Quais Informações são Necessárias? (2)

- ⌚ Há um obstáculo geográfico entre os clientes e a empresa?
- ⌚ Esse obstáculo geográfico impede a movimentação dos clientes para a empresa?



### Quais Informações são Necessárias? (3)

- ⌚ Informações importantes sobre a empresa:
  - Funcionários.
  - Departamentos.
  - Clientes.
  - Produtos Vendidos.
  - Etc.
- ⌚ Existe a necessidade de organizar as informações para permitir a tomada de decisões corretas.

## Dicionário de Informações

- ⌚ O Dicionário de Informações é uma listagem organizada de todos os elementos pertinentes a um sistema, com definições precisas e rigorosas para que o usuário e o desenvolvedor possam conhecer todas as classes.

## Dicionário de Informações: Formato

- ⌚ Alfanumérico: X.
- ⌚ Alfabetico: A.
- ⌚ Numérico: 9.
- ⌚ Hora: 99:99.
- ⌚ Data: 99/99/9999.

## Definição de Iterações

- ➊ São válidas as seguintes opções:

- (1) a = 1{b}
- (2) a = {b}10
- (3) a = 1{b}10
- (4) a = {b}

No exemplo (1), só há limitação do número mínimo de ocorrências; no exemplo (2), há limitação só do máximo; no exemplo (3), há limitação do mínimo e do máximo; no exemplo (4), não há definição de mínimo ou máximo.

## Domínios de Valores e Significados dos Valores (1)

- ➋ Existem dois tipos de elementos de dados:

➌ Aquelas que, para todas as finalidades, pode assumir qualquer valor de um domínio como: quantia em reais de zero à R\$ 999.999,99 até o centavo mais próximo ou uma temperatura de 0°C à 300°C. Este tipo de elemento de dado é chamado **contínuo**, pois há um intervalo onde os valores são válidos. Neste caso, todos os dados estão relacionados a um significado.

➍ Aquelas que somente aceitam certos valores, como número de departamento, que pode ser 36, 08, 29 ou 71 e mais nenhum outro valor. Este tipo de elemento de dado é chamado **discreto**, pois há valores determinados, é necessário colocar o valor e o significado. Um outro exemplo deste segundo tipo é estado civil, que pode ser solteiro, casado, viúvo, ou divorciado. Geralmente os valores são código, tendo algum significado.

## Dicionário de Informações: Resumo

- ➋ Representam as informações a respeitos dos atributos (metadados).
- ➌ Todas as classes são descritas, com exceção das abstratas.
- ➍ Não existe um padrão de representação formal.

Classe: descrição				
Atributo	Descrição	Tamanho	Tipo	Domínio
Nome do atributo	Descrição do atributo	tamanho	Alfabético Alfanumérico Numérico Data	Contínuo ou Discreto (colocar os valores)

## Representação de Seleções

- ➊ A representação de seleção de opções é indicada pelos colchetes, separadas por barras.

- ➋ Exemplo:

sex = [Masculino | Feminino]

tipoCliente = [Governo | Indústria | Universidade | Outro].

## Domínios de Valores e Significados dos Valores (2)

Número de Departamento (Discreto)	
Valor	Significado
36	Vendas
08	Contas
29	Depósito
71	Divulgação
Estado Civil (Discreto)	
Valor	Significado
C	Casado
S	Solteiro
D	Divorciado
V	Viúvo
Temperatura (Continuo)	
Valor	
0°C a 300°C	

## Estudo de Caso: Controle de Leitos de UTI

## Declaração dos Objetivos do Sistema

### Atores Identificados

#### Objetivo geral:

- ❑ Controlar a utilização dos leitos de uma UTI.

#### Objetivos específicos:

- ❑ Controlar o serviço de reservas de leitos de UTI.
- ❑ Controlar o serviço de liberação de leitos.
- ❑ Controlar a limpeza dos leitos.
- ❑ Emitir relatórios sobre a reserva de leitos.



### Requisitos Funcionais Identificados

● Solicitação de reserva de vaga de UTI pelo Médico Assistente. Representada pelos requisitos funcionais "Médico Assistente pode solicitar reserva de leito - FR01" e "Médico Assistente pode Registrar reserva de leito - FR02".

● Cancelamento da solicitação da reserva de vaga pelo Médico Assistente. Representado pelo requisito funcional "Médico Assistente pode solicitar o cancelamento da reserva - FR03".

● Efetivação da Solicitação de Reserva de Vaga, informando se a solicitação foi aceita ou não. Representada pelo requisito funcional "Médico Intensivista pode efetivar a reserva de leito - FR04".

### Requisitos Funcionais Identificados

● Registro do atendimento ao paciente pelo Profissional de Saúde. Representado pelo requisito funcional "Profissional de Saúde Registra o atendimento - FR05".

● Emissão de relatórios, referentes às internações na UTI, pelo Profissional de Saúde. Representada pelo caso de uso "Profissional de Saúde solicita a emissão de relatórios - FR06".

● Comunicação da alta do paciente pelo Médico Intensivista, informação para a Equipe de Higienização sobre os dados da liberação do leito, indicação do final da higienização pela Equipe de Higienização. Representação pelo caso de uso "Médico Intensivista pode conceder alta ao paciente - FR07".

### Casos de Uso Identificados

● Solicitação de reserva de vaga de UTI pelo Médico Assistente. Representada pelos casos de uso "Solicitar reserva de leito - UC01" e "Registrar reserva de leito - UC02".

● Cancelamento da solicitação da reserva de vaga pelo Médico Assistente. Representado pelo caso de uso "Cancelar reserva - UC03".

● Efetivação da Solicitação de Reserva de Vaga, informando se a solicitação foi aceita ou não. Representada pelo caso de uso "Efetivar a reserva de leito - UC04".

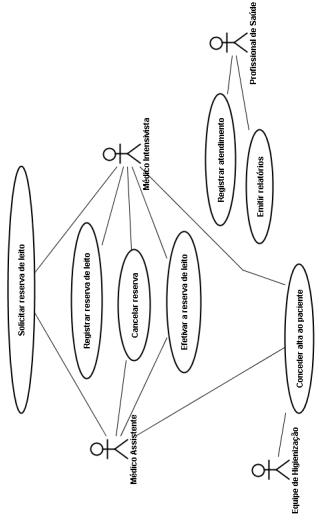
### Casos de Uso Identificados

● Registro do atendimento ao paciente pelo Profissional de Saúde. Representado pelo caso de uso "Registrar atendimento - UC05".

● Emissão de relatórios, referentes às internações na UTI, pelo Profissional de Saúde. Representada pelo caso de uso "Emitir relatórios - UC06".

● Comunicação da alta do paciente pelo Médico Intensivista, informação para a Equipe de Higienização sobre os dados da liberação do leito, indicação do final da higienização pela Equipe de Higienização. Representação pelo caso de uso "Conceder alta ao paciente - UC07".

## Casos de Uso Identificados



## Especificação dos casos de uso: Solicitar reserva de leito

Nome	UC001: Solicitar reserva de leito
Atores	Autor Principal: Médico Assistente. Autor de Suporte: Médico Intensivista.
Descrição	Caso de uso executado quando o médico assistente necessita realizar uma solicitação de reservas de leito.
Pré-condições	- Paciente deve estar cadastrado. - Médico Assistente tem que estar autorizado.
Pós-condições	Reserva de leito solicitada.

## Especificação dos casos de uso: Solicitar reserva de leito

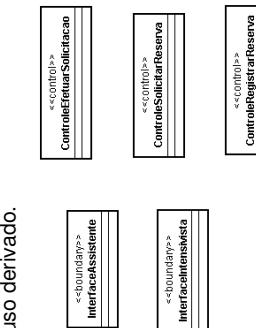
Ruivo Básico	Ações do sistema
Ações dos atores	
1 - Médico Assistente seleciona a opção de solicitação da reserva de leito.	
2 - Exibe o tela de solicitação da reserva de leito com as seguintes informações: nome do paciente, data de nascimento, sexo, diagnóstico, equipamentos necessários, urgência (só atributos).	
3 - O Médico Assistente preenche as informações.	
4 - O Médico Assistente confirma as informações.	
5 - Verifica as informações conforme as Regras de Negócio RN001 e RN002. Caso haja uma regra de negócio não atendida vai para 5.a.	
6 - Grava todas as informações.	
7 - Registra a data da solicitação.	
8 - Envia a solicitação ao Médico intensivista.	
<b>Regras de Negócio</b>	
[RN001] O campo "Nome do paciente" é obrigatório.	
[RN002] A data da nascimento" deve ter o formato dd/mm/aaaa.	

## Especificação dos casos de uso: Solicitar reserva de leito

Ruivo Alternativo 1	Ações do sistema
Ações dos atores	
1 - A qualquer momento o usuário seleciona cancelar.	
	2 - Desconsidera informações digitadas e encerra o caso de uso.
Ruivo Alternativo 2	Ações do sistema
Ações dos atores	
1 - A qualquer momento o usuário seleciona limpar.	
	2 - Limpa todas as informações digitadas.
Ruivo de Exceção	Ações do sistema
Ações dos atores	
5.a - Caso RN01 não seja atendida, apresenta a mensagem "O paciente já está em leito". Caso o RN02 não seja atendida, apresenta a mensagem "Data de nascimento deve ter o formato dd/mm/aaaa" e volta para o Fluxo Básico.	

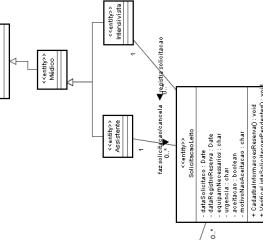
## Diagrama de Classes: Controle e Interface

- Definir uma classe de interface para cada ator.
- Definir uma classe para cada caso de uso e para cada caso de uso derivado.



## Diagrama de Classes: Entidades

- Definir quais dados participam dos casos de uso.
- Agrupar os dados que têm afinidade, identificar os possíveis relacionamentos.
- Identificar as possíveis generalizações e especializações.
- Identificar os métodos que podem atuar sobre os dados de cada classe, a partir das ações apresentadas nos casos de uso..



## Dicionário de Informações Controle de Leitos de UTI

Deve incluir a descrição dos seus atributos.

Pessoas: Identifica as características básicas de qualquer pessoa					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio
nome	Nome completo	60	Alfanumérico	(X)60	Continuo
sexo	Refer-se ao sexo	1	Alfabético	A	Discreto F - Feminino M - Masculino
dataNascimento	Refer-se a data de nascimento	8	Data	dd/mm/aaaa	Continuo d = dia, m = mês, a = ano.
diagnostico	Diagnóstico a ser utilizado pelo médico para a análise.				Continuo

Como é uma especialização, deve incluir a descrição dos atributos das classes acima na hierarquia (Pessoa).

## Dicionário de Informações Controle de Leitos de UTI

Como é uma especialização, deve incluir a descrição das atributos das classes acima na hierarquia (Pessoa, Profissional de Saúde e Médico).

Intensivista: Médico responsável pela solicitação de reservas de vagas nos Leitos de UTI					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio
nome	Nome completo	60	Alfanumérico	(X)60	Continuo
sexo	Refer-se ao sexo	1	Alfabético	A	Discreto F - Feminino M - Masculino
dataNascimento	Refer-se a data de nascimento	8	Data	dd/mm/aaaa	Continuo d = dia, m = mês, a = ano.
acelatado	Indica se foi aceito ou não	3	Alfabético	AAA	Discreto [Sim   Não]
motivoNaoAcelatado	Motivos para a não aceitação	60	Alfanumérico	(X)60	Continuo

Como é uma especialização, deve incluir a descrição dos atributos das classes acima na hierarquia (Pessoa).

## Dicionário de Informações Controle de Leitos de UTI

Deve incluir a descrição dos seus atributos.

## Dicionário de Informações Controle de Leitos de UTI

Deve incluir a descrição dos seus atributos.

Solicitação de leito: Refer-se a solicitação de uma reserva de leito na UTI					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio
dataSolicitação	Data da solicitação da reserva de leito	8	Alfanumérico	dd/mm/aaaa	Continuo d = dia, m = mês, a = ano.
dataRegistroReserva	Data do registro da reserva de leito	8	Alfanumérico	dd/mm/aaaa	Continuo d = dia, m = mês, a = ano.
equipamentoNecessarios	Necessidades especiais do paciente				
urgencia	Grau da urgência da solicitação	5	Alfabético	AAAAA	Discreto Alto Médio Baixo
aceitacao	Indica se foi aceito ou não	3	Alfabético	AAA	Discreto [Sim   Não]
motivoNaoAcelatado	Motivos para a não aceitação	60	Alfanumérico	(X)60	Continuo

Como é uma especialização, deve incluir a descrição dos atributos das classes acima na hierarquia (Pessoa).

Obrigado.

## Análise e Projeto de Sistemas

### **Diagrama de Instâncias ou Diagrama de Objetos**

### **Diagrama de Instâncias ou Diagrama de Objetos**

### **Diagrama de Instâncias ou Diagrama de Objetos**

Os objetos não apresentam métodos, apenas atributos.

O nome do objeto pode estar em uma das três formas:

<<entity>>	: Paciente
	name = Ronaldo
	sexo = M
	dataNascimento = 11/04/1978
	diagnóstico = fratura
<<entity>>	Paciente 1
	name = Ronaldo
	sexo = M
	dataNascimento = 11/04/1978
	diagnóstico = fratura
<<entity>>	Paciente
	name = Ronaldo
	sexo = M
	dataNascimento = 11/04/1978
	diagnóstico = fratura

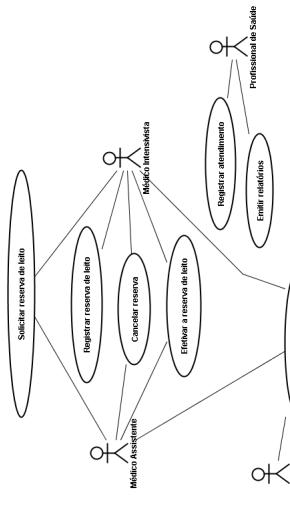
- Utilizado para melhor entendimento do modelo de classes, assim como para a sua validação.
- Não é um diagrama independente, sendo um complemento para o Diagrama de Classes.
- Fornecê uma visão dos objetos armazenados pelos objetos, em um determinado instante de tempo.
- Simula as possibilidades de preenchimento de um objeto, a partir das definições do Dicionário de Informações.

**Prof. Laudelino Cordeiro Bastos**

### **Diagrama de Instâncias ou Diagrama de Objetos**

- As ligações entre os objetos (vínculos - *links*) nada mais são do que instâncias das associações entre as classes representadas no Diagrama de Classes.
- Uma ligação entre objetos não possui cardinalidade (multiplicidade).

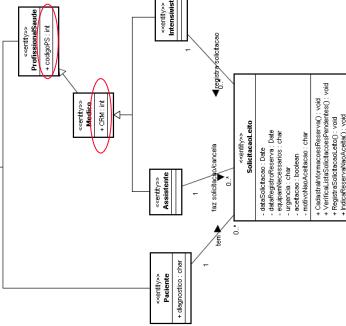
### **Diagrama de Casos de Uso Visão Geral - Controle Leitos de UTI**



## **Estudo de Caso: Controle de Leitos de UTI**

## Diagrama de Classes

Adicionado códigoPS  
e CRM.



## Dicionário de Informações Controle de Leitos de UTI

Deve incluir a descrição dos seus atributos.

Pessoa: identifica as características básicas de qualquer pessoa					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio
nome	Nome completo	60	Alfanumérico	(X)60	Continuo
sexo	Refere-se ao sexo	1	Alfabético	A	Discreto F - Feminino M - Masculino
dataNascimento	Refere-se a data de nascimento	8	Data	dd/mm/aaaa	Continuo d = dia, m = mês, a = ano.

## Dicionário de Informações Controle de Leitos de UTI

Como é uma especialização, deve incluir a descrição dos atributos das classes acima na hierarquia (Pessoa, Profissional de Saúde e Médico).

Paciente: Pessoa que precisa de uma cama na UTI.

Profissional de Saúde: Médico responsável pela solicitação de reservas de vagas de Leitos de UTI

CRM: códigoPS e CRM.

## Dicionário de Informações Controle de Leitos de UTI

Como é uma especialização, deve incluir a descrição dos atributos das classes acima na hierarquia (Pessoa, Profissional de Saúde e Médico).

Paciente: Pessoa que precisa de uma cama na UTI.

CRM: códigoPS e CRM.

## Dicionário de Informações Controle de Leitos de UTI

Como é uma especialização, deve incluir a descrição dos atributos das classes acima na hierarquia (Pessoa, Profissional de Saúde e Médico).

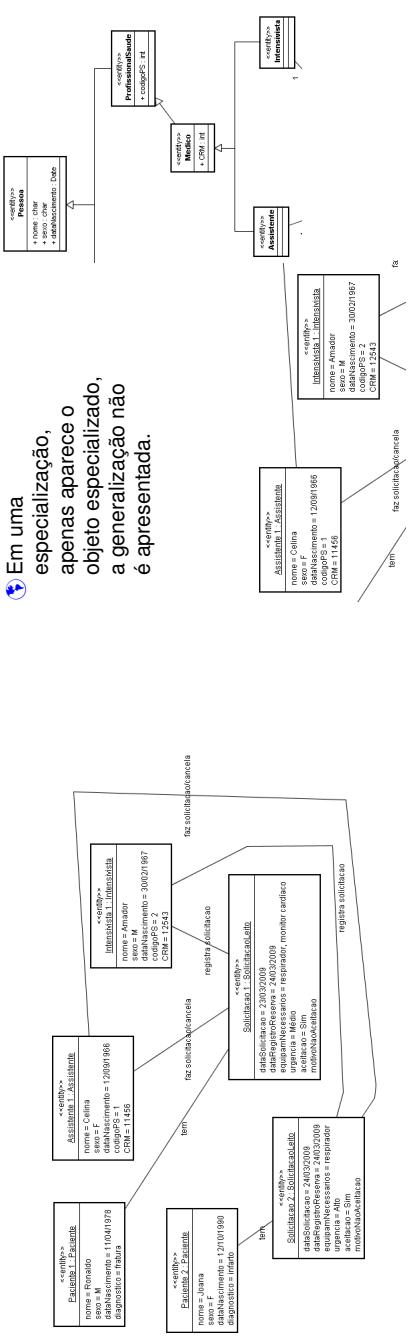
Intensivista: Médico responsável pela solicitação de reservas de vagas de Leitos de UTI

CRM: códigoPS e CRM.

SolicitaçãoLeito: Refere-se à solicitação de uma cama na UTI					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio
dataSolicitacao	Data da solicitação da reserva de leito	8	Alfanumérico	dd/mm/aaaa	Continuo d = dia, m = mês, a = ano.
dataRegistroReserva	Data do registo da reserva de leito	8	Alfanumérico	dd/mm/aaaa	Continuo d = dia, m = mês, a = ano.
equipamentoNecessario	Necessidades especiais do paciente	60	Alfanumérico	(X)60	Continuo
urgencia	Grau da urgência da solicitação	5	Alfabético	AAAAA	Discreto Alto Médio Baixo
aceitacao	Indica se foi aceito ou não	3	Alfabético	AAA	Discreto [Sim   Não]
motivoNaAcelacao	Motivos para a não aceitação	60	Alfanumérico	(X)60	Continuo

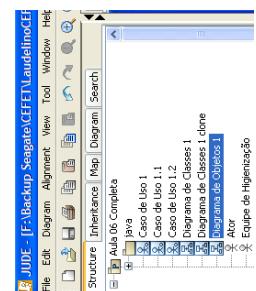
## Diagrama de Objetos (Instâncias)

### Diag. de Classes e Diag. de Objetos



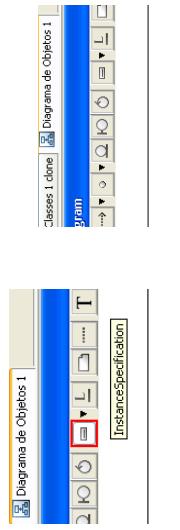
## Criando um objeto

Crie um novo *Class Diagram*, mas nomeie como Diagrama de Objetos.



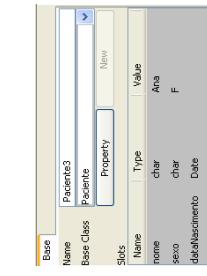
## Criando um objeto

Clique sobre o ícone *InstanceSpecification* e crie um objeto.



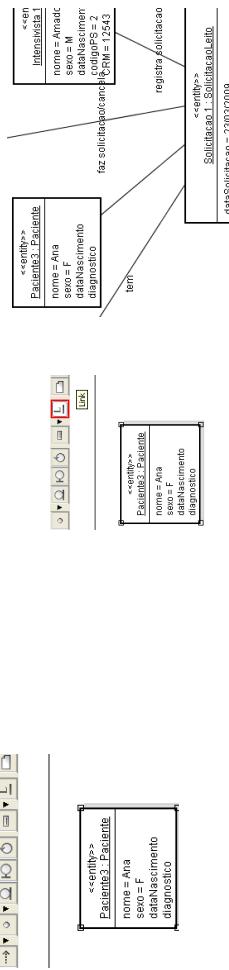
## Criando um objeto

Preencha a aba *Base Class*.

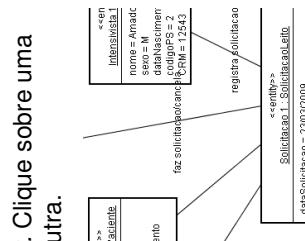


## Criando uma associação

Clique sobre o botão *Link*. Clique sobre uma classe e depois sobre a outra.



## Criando uma associação



Obrigado.



**MINISTÉRIO DA EDUCAÇÃO**  
**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**  
**CURSO DE ENGENHARIA DE COMPUTAÇÃO**  
**MATRIZ CURRICULAR**

1º Período		2º Período		3º Período		4º Período		5º Período		6º Período		7º Período		8º Período		9º Período		10º Período				
Tecnologia e Sociedade	1.1 2 2/0			Filosofia da Ciência e da Tecnologia	3.1 2 2/0	História da Técnica e da Tecnologia	4.1 2 2/0	Sociedade e Política no Brasil	5.1 2 2/0	Economia	6.1 2 2/0	Gestão de Pessoas	7.1 2 2/0	Gestão Financeira	8.1 2 2/0	Ciências Ambientais	9.1 2 2/0	Ética, Profissão e Cidadania	10.1 2 2/0			
ES61A	2			ES60A	2	ES60F	2	ES60G	2	GE60D	2	GE60B	2	GE60C	2	QB65A	2	IF60A	2			
	B 36			1.1 B 36		3.1 B 36	4.1 B 36				B 36		B 36		B 36		B 36		B 36			
Lógica para Computação	1.2 4 2/2	Química	2.1 6 4/2	Probabilidade e Estatística	3.2 4 4/0	Mecânica	4.2 4 4/0	Fenômenos de Transporte 1	5.2 2 1/1	Redes de Computadores 1	6.2 4 2/2	Redes de Computadores 2	7.2 4 2/2					Optativas				
IF61B	4	QB62A	6	MA65A	4	FI64C	4	FI66A	2	IF66B	4	IF67B	4			IF68A						
	P 72		B 108	2.3 B 72	1.5 1.6 B 72	2.5 B 36				4.3 PE 72		6.2 PE 72				Cursar 300h de disciplinas optativas disponíveis	IPE	360				
Fundamentos de Programação 1	1.3 6 3/3	Fundamentos de Programação 2	2.2 4 2/2	Estrutura de Dados 1	3.3 3 1/2	Estrutura de Dados 2	4.3 3 2/1	Teoria da Computação	5.3 3 2/1	Engenharia de Software	6.3 4 2/2	Sistemas Distribuídos	7.3 4 2/2									
IF61C	6	IF62C	4	IF63C	3	IF64C	3	IF65C	3	IF66C	4	IF67C	4									
	B 108	1.3 P 54		2.2 P 54	3.3 P 54	3.4 P 54	3.5 P 54	3.6 P 54	3.7 P 54	5.4 PE 72		6.2 6.4 PE 72										
Cálculo Diferencial e Integral 1	1.4 6 6/0	Cálculo Diferencial e Integral 2	2.3 4 4/0	Cálculo Diferencial e Integral 3	3.4 4 4/0	Análise de Sistemas Lineares	4.4 4 2/2	Análise e Projeto de Sistemas	5.4 3 3/0	Sistemas Operacionais	6.4 4 2/2	Sistemas Inteligentes 1	7.4 4 2/2	Sistemas Inteligentes 2	8.4 4 2/2	Processamento Digital de Imagens	9.4 4 2/2					
MA61A	6	MA62A	4	MA63A	4	EL65G	4	IF65D	3	IF66D	4	IF67D	4	IF68D	4	IF69D	4					
	B 108	1.4 B 72		2.3 B 72	3.4 3.8 PE 72	3.3 P 54	3.8 P 108	3.9 P 72	3.10 P 72	5.7 PE 72		3.3 PE 72		7.4 PE 72		2.2 P 6	IPE	72				
Matemática 1	1.5 6 6/0	Comunicação Gráfica	2.4 3 2/1	Matemática 2	3.5 4 4/0	Circuitos Digitais	4.5 6 4/2	Banco de Dados	5.5 4 2/2	Fundamentos de Controle	6.5 5 2/3	Processamento Digital de Sinais	7.5 4 2/2	EL66K	5	EL66D	4	IF68E	4			
MA61B	6	EL62O	3	MA63B	4	EL65A	6	IF65E	4	4.4 5.6 P 90	4.4 5.6 P 90	4.4 5.6 P 72	6.2 6.4 P 72	6.2 6.4 P 72								
	B 108			1.5 2.3 B 72		3.8 P 90		3.3 P 72														
Física 1	1.6 5 3/2	Física 2	2.5 5 3/2	Matemática Discreta	3.6 4 4/0	Eletônica Geral 1	4.6 5 2/3	Eletônica Geral 2	5.6 6 4/2	Sistemas Microcontrolados	6.6 4 2/2	Lógica Reconfigurável	7.6 4 2/2	EL66A	4	EL68A	4	Sistemas Embarcados	8.7 4 2/2			
FI61A	5	FI62A	5	IF63E	4	EL64O	5	EL65H	6	EL66H	4	EL68F	4									
	B 90	1.4 1.6 B 90			3.8 P 72		3.4 4.6 P 108		3.4 4.6 P 72	4.5 5.6 PE 72	4.6 5.6 PE 72	6.5 PE 72										
Física 3	2.6 5 3/2			Design de Intereração	3.7 3 2/1			Arquitetura e Organização de Computadores	5.7 4 2/2	Comunicação de Dados	6.7 2 2/0			Trabalho de Conclusão de Curso 1	7.7 4 1/3	Programação Matemática	8.6 4 2/2	Trabalho de Conclusão de Curso 2	10.3 4 1/3			
FI63A	5			IF63O	3			EL66C	4	EL66L	2	EL66L	2	IF67H	4	EL68G	4	IF60B	4			
	1.4 1.6 B 90							4.5 P 72		4.4 P 36		6.8 SIC 72		6.8 SIC 72		3.5 PE 72		7.7 SIC 72				
Oficina de Integração 1	2.7 3 1/2			Fundamentos de Análise de Circuitos Elétricos	3.8 5 3/2			Oficina de Integração 2	4.7 3 1/2	Oficina de Integração 3	6.8 3 1/2											
IF62J	3			EL63B	5			IF64J	3	IF66J	3	P6	SIC 54	P7								
P2				P4				P4		P6												
Atividades Complementares																						
IF60K																						
SIC 180h																						
Estágio Supervisionado																						
IF60J																						
SIC 360h																						

## LEGENDA

Ref - REFERÊNCIA NA MATRIZ  
 APS - ATIVIDADE PRÁTICA SUPERVISIONADA  
 T/P - AULAS TEÓRICAS/PRÁTICAS SEMANALIS  
 TT - TOTAL DE AULAS SEMANALIS  
 TA - TOTAL DE AULAS NO SEMESTRE  
 PR - PRÉ-REQUISITO

TC - TIPO DE CONTEÚDO  
 B - CONTEÚDOS BÁSICOS  
 P - CONTEÚDOS PROFISSIONALIZANTES  
 PE - CONTEÚDOS PROFISSIONALIZANTES ESPECÍFICOS  
 SIC - ATIVIDADE DE SÍNTSEDE INTEGRAÇÃO DE CONHECIMENTO

ATIVIDADES PRESENCIAIS 3705 hs  
 ATIVIDADES NÃO PRESENCIAIS 180 hs  
 ESTÁGIO 360 hs  
 CARGA HORÁRIA TOTAL 4245 hs

FRENTE/VERSO  
 CURSO 212 - GRADE 721  
 Atualização: maio/2011