




Laboratório de Engenharia de Software

# Aula 14

## Modelagem Física – Parte II

Alessandro Garcia  
LES/DI/PUC-Rio  
Abril 2011

### Última aula...



Laboratório de Engenharia de Software

- Para que serve modelagem?
- O que é um modelo?
- O que é modelagem conceitual?
- O que é o modelo da arquitetura? Para que serve?
- O que é um modelo físico?

Set 2009LES/DI/PUC-Rio2 / 36

## UML (Unified Modelling Language)



- UML: foi desenvolvida para...
  - modelagem orientada à objetos: classes, objetos, e diferentes relacionamentos
  - modelagem **conceitual**: domínio da aplicação ("entidades do mundo real")
- **Modelagem física** com UML requer ligeiras adaptações
  - utiliza-se **notação semelhante**
- Um **modelo físico** de uma estrutura de dados descreve todas as possíveis instâncias desse conceito em memória física
- Um **exemplo físico** ilustra exatamente **uma** dessas possíveis instâncias

Set 2009

© LES/DI/PUC-Rio

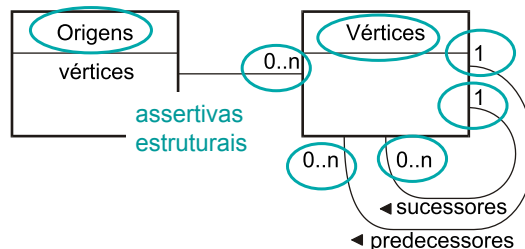
3 / 28

## Do modelo conceitual para...



- ... modelo físico:
  - É o resultado de uma **transformação do modelo conceitual**, tornando-o realizável no meio físico alvo
  - Estabelecem-se **interdependências** a serem realizadas no **meio físico escolhido**, tais como:
    - memória principal, determinada linguagem de programação, ou determinado sistema de gerência de banco de dados
  - Transformações devem **preservar** as características do modelo conceitual
  - Exemplo –

**Grafo direcionado**



Set 2009

© LES/DI/PUC-Rio

4 / 28

Laboratório de Engenharia de Software

## Linguagem de representação

- Estruturas de dados físicas podem ser modeladas usando uma representação **semelhante** a dos **diagramas de classes** da UML2
  - dois tipos básicos de relacionamentos

Referência

```

struct {
    tpPessoa Pessoa;
    tpCargo Cargo;
    tpSalario Salario;
} tpEmpregado
          
```

nome da referência usado no Tipo 1  
(outro exemplo: ponteiros)

- Tipo 1 e Tipo 2** são agregados de 1 ou mais elementos de dados: ou seja, são estruturas (struct)

Set 2009
© LES/DI/PUC-Rio
5 / 28

Laboratório de Engenharia de Software

## Linguagem de representação física

- Estruturas de dados físicas podem ser modeladas usando uma representação **semelhante** a dos **diagramas de classes** da UML2
  - dois tipos básicos de relacionamentos

Referência

Agregação

possibilidades:  
 definido, exemplo: 20  
 restrito, exemplo: 0..20  
 indefinido, exemplos: \*  
 + (1 ou mais)  
 3..\*

nome da referência usado no Tipo 1

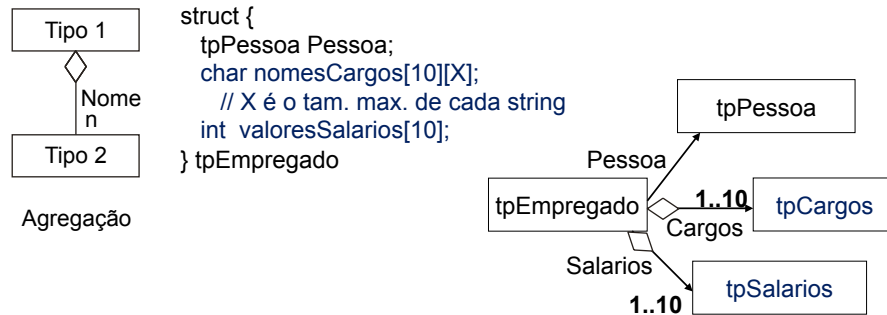
- Tipo 1 e Tipo 2** são agregados de 1 ou mais elementos de dados: ou seja, são estruturas (struct)

Set 2009
© LES/DI/PUC-Rio
6 / 28

## Linguagem de representação física



- Estruturas de dados físicas podem ser modeladas usando uma representação **semelhante** a dos **diagramas de classes** da UML2
  - dois tipos básicos de relacionamentos



- Tipo 1 e Tipo 2** são agregados de 1 ou mais elementos de dados: ou seja, são estruturas (struct) ou classes

## Outro Exemplo

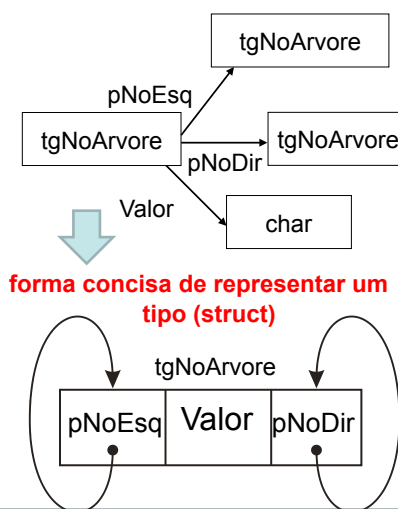


- Modelando fisicamente a estrutura de uma Arvore (ignorando a cabeça)

```
typedef struct tgNoArvore {
    struct tgNoArvore * pNoEsq ;
    /* Ponteiro para filho à esquerda


    struct tgNoArvore * pNoDir ;
    /* Ponteiro para filho à direita

    char Valor ;
    /* Valor do nó */
} tgNoArvore ;
```



Laboratório de Engenharia de Software

## Esta aula: especificação



- Aula Anterior: Modelo conceitual, exemplo conceitual e modelagem da arquitetura
- Objetivo dessa aula
  - Concluir apresentação da linguagem gráfica para a modelagem física de estruturas de dados
  - Motivar o uso de assertivas com modelos
- Referência básica:
  - Capítulo 9.4
- Referência complementar
  - Silva, R.P.; *UML2 em Modelagem Orientada a Objetos*; Florianópolis, SC: Visual Books; 2007


Set 2009

© LES/DI/PUC-Rio

9 / 28

Laboratório de Engenharia de Software

## Sumário



- Linguagem de representação gráfica para modelos físicos
- Modelo físico
- Exemplo físico
- Modelagem de cabeças de estrutura de dados
- Assertivas estruturais

Set 2009

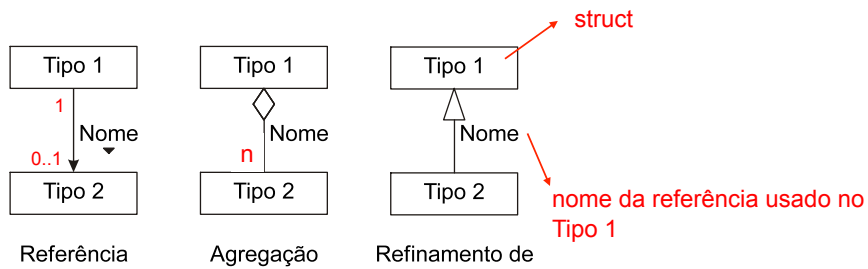
© LES/DI/PUC-Rio

10 / 28

## Linguagem de representação



- Estruturas de dados físicas podem ser modeladas usando uma representação **semelhante** a dos **diagramas de classes** da UML2
  - quatro tipos de relacionamentos



- Tipo 1 e Tipo 2** são agregados de 1 ou mais elementos de dados: ou seja, são estruturas (struct) ou classes

Set 2009

© LES/DI/PUC-Rio

11 / 28

## Como implementar?



- Herança em C:

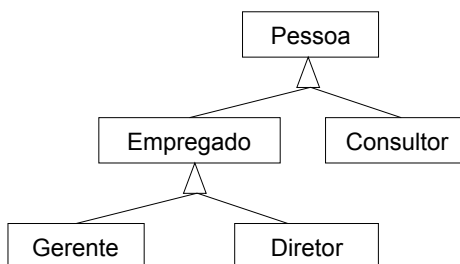
```
typedef struct {
    /* entram aqui os dados de pessoa */
} tpPessoa

struct {
    tpPessoa Pessoa ;
    /* entram aqui os dados de empregado */
} tpEmpregado

struct {
    tpEmpregado Empregado
    /* entram aqui os dados de gerente */
} tpGerente

struct {
    tpEmpregado Empregado
    /* entram aqui os dados de diretor */
} tpDiretor

...}
```




Set

DI/PUC-Rio

12 / 28

Laboratório de Engenharia de Software

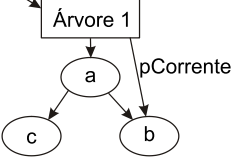


## Se fossem modelar por completo...

### ... a estrutura de dados **Árvore**:

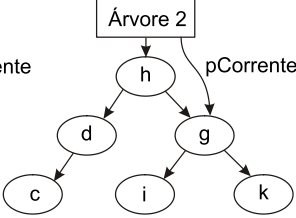
o que falta neste modelo?

Uso 1  
Uso 2  
Uso 3



Árvore 1

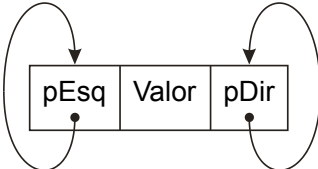
$a(c * b)$




Árvore 2

$h(d(c \text{ nil}) * g(i \ k))$

Modelo de árvore binária

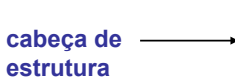


Laboratório de Engenharia de Software



## Como representar cabeças de estruturas?

cabeça de estrutura

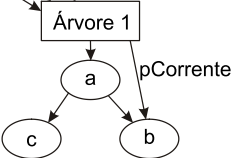


Árvore

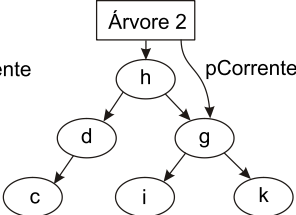
pRaiz pCorrente

Árvores

Uso 1  
Uso 2  
Uso 3



Árvore 1



Árvore 2

Set 2009

© LES/DI/PUC-Rio

14 / 28

## Cabeça de estrutura



- Cada **estrutura de dados** deve poder ser tratada como se fosse **uma unidade**
  - independente da sua complexidade e da diversidade de componentes
- Para tal pode-se utilizar uma **cabeça de estrutura**
  - todas as referências para a estrutura referenciam a cabeça
  - as referências internas à estrutura são desconhecidas ao cliente
- Vantagens
  - melhor encapsulamento da estrutura
    - a cabeça da estrutura passa a ser uma interface de acesso
    - reduz o risco de uso acidental ou deliberadamente incorreto
  - permite tratar estruturas vazias
  - permite mover as estruturas na memória
    - somente a cabeça precisa ficar fixa

Ago 2008

Arndt von Staa © LES/DI/PUC-Rio

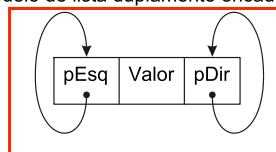
15 / 28

## Cuidado com Figuras

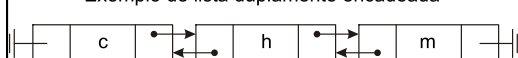
## Modelos demasiadamente abstratos



Modelo de lista duplamente encadeada



Exemplo de lista duplamente encadeada



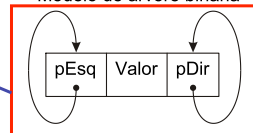
**Lista Duplamente Encadeada**

exemplo válido

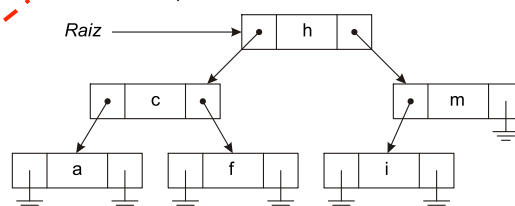
**Árvore Binária**

Quais problemas você identifica neste modelo?

Modelo de árvore binária



Exemplo de árvore binária



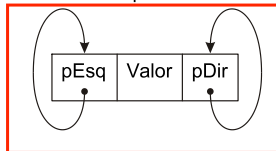


## Cuidado com Figuras

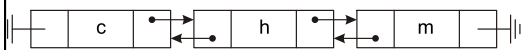
Use descrição de assertivas complementares



Modelo de lista duplamente encadeada



Exemplo de lista duplamente encadeada



**Lista Duplamente Encadeada**

1 pEsq == NULL

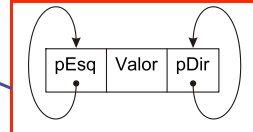
1 pDir == NULL

exemplo válido

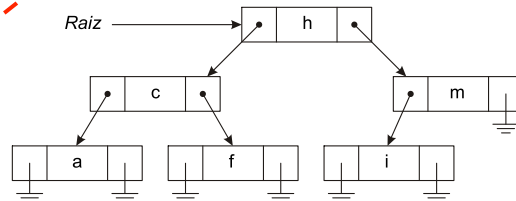
**Árvore Binária**

Quais problemas você identifica neste modelo?

Modelo de árvore binária



Exemplo de árvore binária



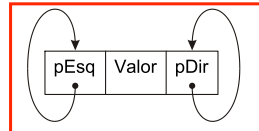
## Figuras não são suficientes



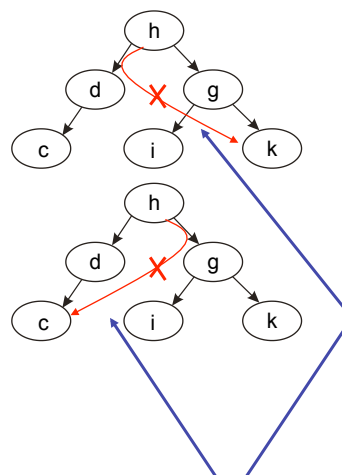
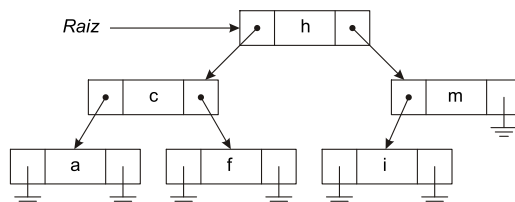
**Árvore Binária**

Quais problemas você identifica neste modelo?

Modelo de árvore binária



Exemplo de árvore binária




falta de detalhes importantes

Laboratório de Engenharia de Software

## Modelos devem ter assertivas adicionais

- Para resolver o problema da insuficiência de detalhes em figuras utilizam-se **assertivas estruturais**.
- Exemplo: lista duplamente encadeada**
  - Para cada nó N da lista
    - se  $N \rightarrow pEsq \neq \text{NULL}$  então  $N \rightarrow pEsq \rightarrow pDir == N$
    - se  $N \rightarrow pDir \neq \text{NULL}$  então  $N \rightarrow pDir \rightarrow pEsq == N$
- Exemplo: árvore**
  - para cada nó N da árvore
    - a referência para filho à esquerda de um nó tem como destino a raiz da sub-árvore à esquerda
    - a referência para filho à direita de um nó tem como destino a raiz da sub-árvore à direita
    - o conjunto de nós alcançáveis a partir da raiz da sub-árvore à esquerda é disjunto do conjunto de nós alcançáveis a partir da raiz da sub-árvore à direita

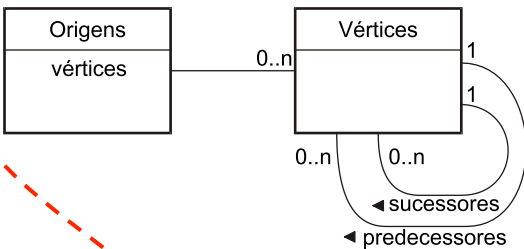


Abril 2009
© LES/DI/PUC-Rio
19 / 30

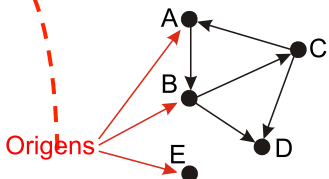
Laboratório de Engenharia de Software


## Outro exemplo: Grafo Dirigido

**Modelo Físico**  
(um possível)



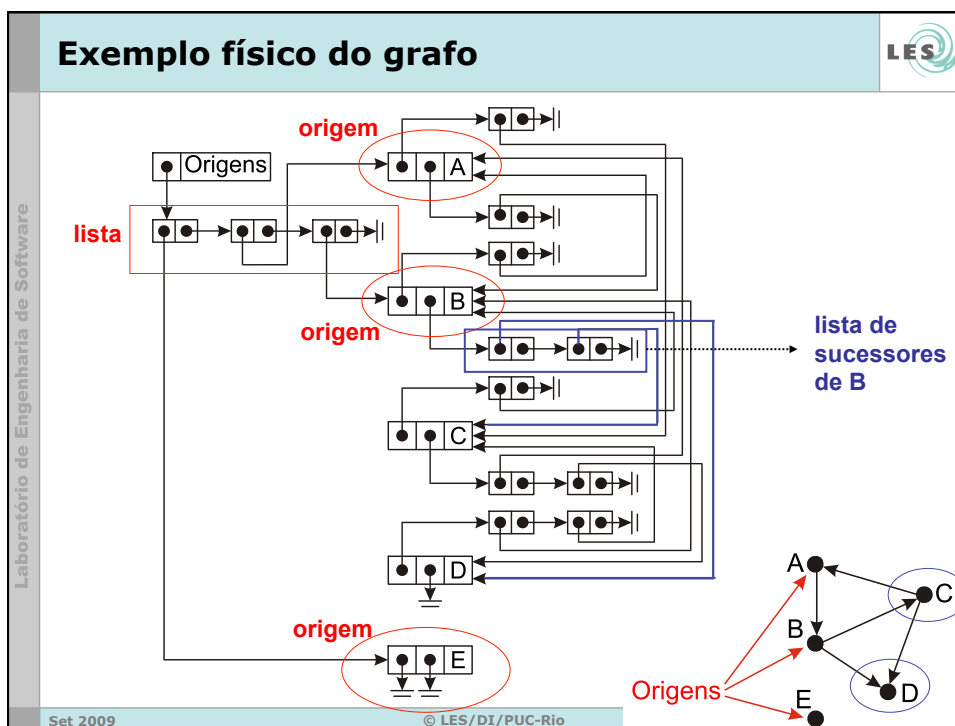
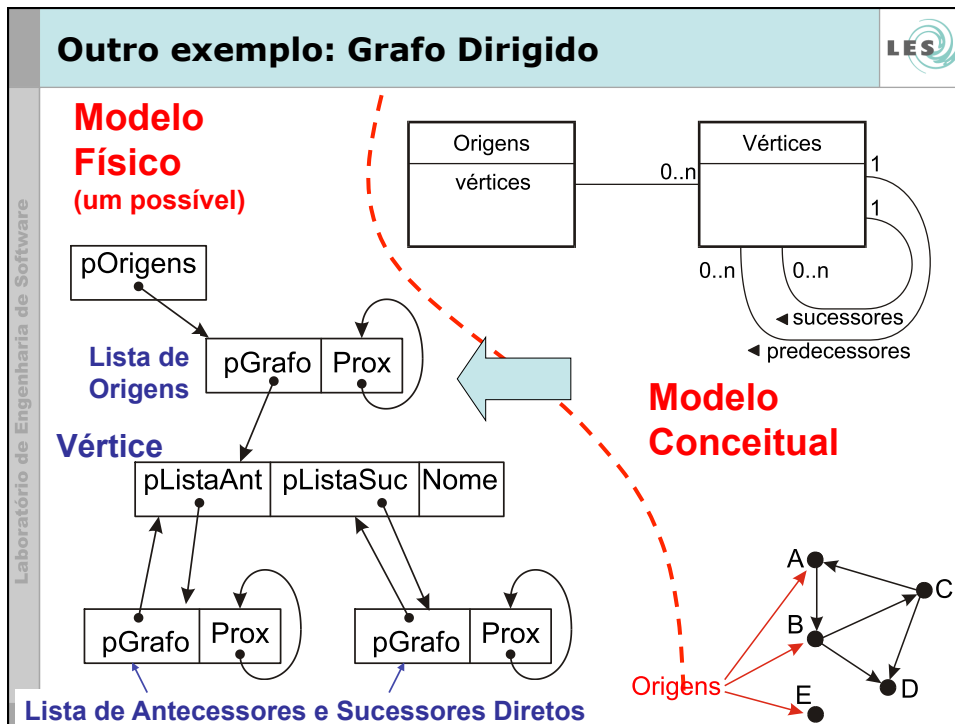
**Modelo Conceitual**






Set 2009
© LES/DI/PUC-Rio

10



Laboratório de Engenharia de Software

## Lembretes e Avisos...



- **Trabalho** - não esqueça: assume-se que vocês já estejam trabalhando...
  - Especificação e implementação dos módulos do jogo de damas
  - Raciocinar sobre regras do jogo de damas (requisitos) e quais funcionalidades para a realização de uma partida
    - não esqueça dos requisitos não funcionais
  - Planeje os principais módulos (arquitetura)
  - Agora:
    - Especificação da arquitetura
    - Melhoria da descrição dos requisitos para cada módulo
  - Linguagem para Scripts de Teste estará disponível esta semana
- **Envie para mim seus modelos da arquitetura**
  - **Próxima aula: esclarecimento de dúvidas do trabalho**


Set 2009

LES/DI/PUC-Rio

23 / 36

Laboratório de Engenharia de Software

## Processo de Desenvolvimento Modular



**No Trabalho 2, vocês devem seguir este processo! Relatar os passos a cada dia no [arquivo de relatório...](#)**

1. Descrição de requisitos do sistema
  1. descoberta dos requisitos funcionais e não-funcionais, além do enunciado do problema
  2. especificação dos requisitos (p.e. linguagem natural)
2. Definição da arquitetura do sistema
  1. especificação do modelo lógico
    - identificação dos módulos
    - identificação das interfaces e relacionamentos entre os módulos
3. Refinamento da arquitetura
  1. especificação do modelo físico (e.g. modelo da estrutura de dados)
4. Desenvolvimento dos módulos
  1. especificação das interfaces (\*.h) e requisitos associados
  2. especificação dos casos de teste
  3. especificação das funções (assertivas/contratos)
  4. implementação dos módulos e definição de novos requisitos emergentes

Set

5



# **Aula 14**

## **Modelagem Física – Parte II**

Alessandro Garcia  
LES/DI/PUC-Rio  
Abril 2011