Universidade Federal do Espírito Santo

Centro Tecnológico

Departamento de Informática

Disciplina: ***INF15978 – Engenharia de Software I***

Prof.: ***Monalessa Perini Barcellos***

(monalessa@inf.ufes.br)

1

Análise de Requisitos

Conceitos da Orientação a Objetos

a) Objetos: entidades que interagem entre si, onde cada uma delas desempenha um papel específico.

O carro do João

João

b) Classes: descrevem um conjunto de objetos com as mesmas propriedades (atributos e associações) e o mesmo comportamento (operações).
Objetos são instâncias das classes

Carro A

Carro

Carro B

Carro C

João

Pessoa

Maria

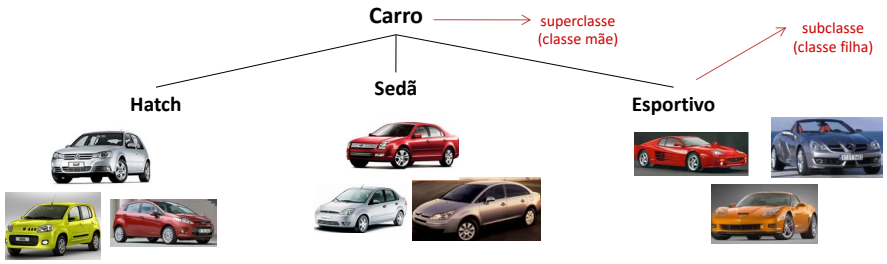
Cecy

2

Análise de Requisitos

c) Ligações e Associações: relacionamentos entre objetos e classes (respectivamente).

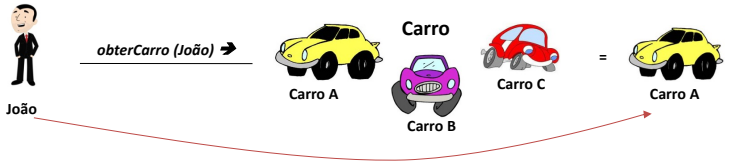
d) Herança: descrevem um conjunto de objetos com as mesmas propriedades (atributos e associações) e o mesmo comportamento (operações).



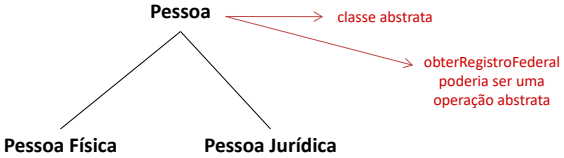
3

Análise de Requisitos

e) Mensagens e Métodos: forma de comunicação entre os objetos.



d) Classes e Operações Abstratas: classes que não possuem instância e operações que não são implementadas nas classes (são apenas assinatura).



4

Análise de Requisitos

Modelagem Conceitual Estrutural

Diagrama de Classes

Tem por objetivo descrever as informações que o sistema deve representar e gerenciar.

Devem ser concebidos com foco no domínio do problema e não no domínio da solução.

Para elaborar um diagrama de classes, é preciso:

a) **Identificar classes**

b) **Identificar atributos e associações**

c) **Especificar Hierarquias de Generalização/Especialização**

Análise de Requisitos

a) Identificação de Classes

Objetos

O carro do João

João

Classes descrevem um conjunto de objetos com as mesmas propriedades (atributos e associações) e o mesmo comportamento (operações).

Objetos são instâncias das classes

Classes

João

Pessoa

Maria

Cecy

Carro A

Carro

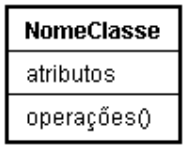
Carro B

Carro C

Análise de Requisitos

a) Identificação de Classes

Notação:



Como encontrar as classes?

Procurar no documento de requisitos e modelos de casos de uso:

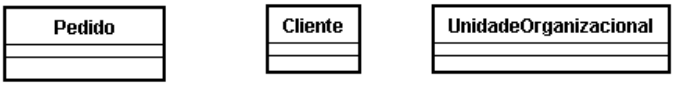
-Agentes (ex.: pessoa e organização)

- Papéis (ex.: autor, contratante e analista)

-Objetos (ex.: livro, contrato e curso)

- Eventos (ex.: locação, matrícula e reserva).

Exemplos:



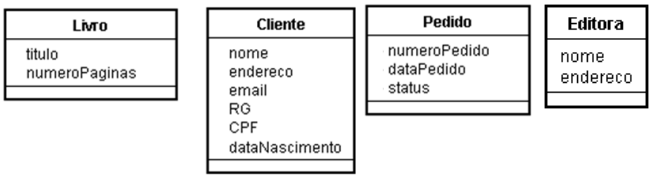
7

Análise de Requisitos

b) Identificação de Atributos e Associações

Atributos descrevem propriedades dos objetos de uma classe.

Exemplos:



Nota: **Não se representam os atributos que indicam relacionamentos** (ex.: atributo editora na classe Livro)

8

Análise de Requisitos

Especificando Atributos

Notação: **visibilidade nome: tipo [multiplicidade] = valorInicial {propriedades}**

Visibilidade

\+ público : o atributo pode ser acessado por qualquer classe;

\# protegido: o atributo só é passível de acesso pela própria classe ou por uma de suas especializações;

\- privado: o atributo só pode ser acessado pela própria classe;

~ pacote: o atributo só pode ser acessado por classes declaradas dentro do mesmo pacote da classe a que pertence o atributo.

A visibilidade só é determinada na fase de Projeto.

Análise de Requisitos

Especificando Atributos

Notação: **visibilidade nome: tipo [multiplicidade] = valorInicial {propriedades}**

Tipo

Atributos possuem *tipos*, que podem ser primitivos ou específicos de domínio.

Exemplos de *tipos primitivos*: String, Boolean, Integer, Float, Currency, Date, Time, DateTime, Year, YearMonth.

Exemplos de *tipos específicos de domínio*: CPF, ISBN, CNPJ, atributos enumerados (sexo, dias da semana, status etc)

Análise de Requisitos

Especificando Atributos

Notação: **visibilidade nome: tipo [multiplicidade] = valorInicial {propriedades}**

Multiplicidade

Informada quando um atributo for opcional ou quando puder ter mais do que uma ocorrência.

Deve ser informada, indicando o valor mínimo e o valor máximo.

Exemplos:

nome: String → instâncias da classe têm obrigatoriamente um e somente um nome.

carteira: String [0..1] → instâncias da classe têm uma ou nenhuma carteira.

telefones: Telefone [0..*] → instâncias da classe têm um ou vários telefones.

pessoasContato: String [2] → instâncias da classe têm exatamente duas pessoas de contato.

Análise de Requisitos

Especificando Atributos

Notação: **visibilidade nome: tipo [multiplicidade] = valorInicial {propriedades}**

Valor Inicial

Valor que, quando não informado outro valor, será atribuído ao atributo.

Exemplo: **origem: Ponto = (0,0)** → a origem, quando não informado outro valor, será (0,0).

Propriedades

Propriedades dos atributos que vale a pena serem destacadas.

Exemplo: **numSocio: int {readonly}**

Análise de Requisitos

Especificando Atributos

Na fase Análise e Especificação de Requisitos, geralmente são especificados apenas:

nome: tipo [multiplicidade]

Tipo e *multiplicidade* podem ser tratados apenas na fase Projeto de Sistema, mas pode-se optar por representá-los na fase de Análise para tornar o modelo mais expressivo. Nesse caso, considera-se alguns tipos primitivos, os quais devem ser ajustados na fase Projeto de Sistema de acordo com as tecnologias adotadas.

Valor Inicial e *Propriedades* costumam ser tratados na fase Projeto de Sistema.

Visibilidade sempre é tratada na fase Projeto de Sistema.

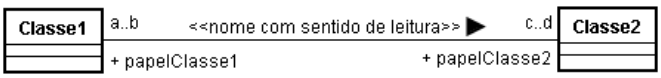
13

Análise de Requisitos

Associações são relacionamentos que ocorrem entre instâncias de duas ou mais classes.

Exemplo: Cliente efetua Pedido

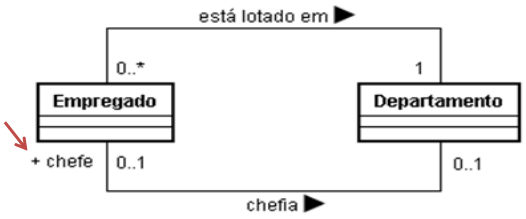
Notação:



14

Análise de Requisitos

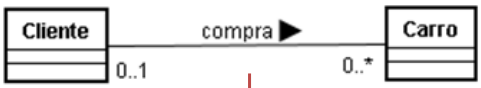
Papéis



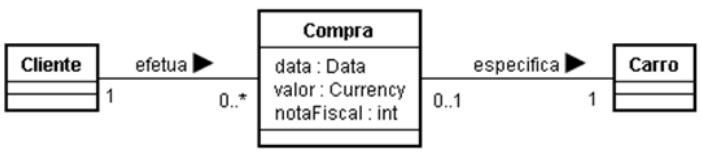
15

Análise de Requisitos

Evento Lembrado



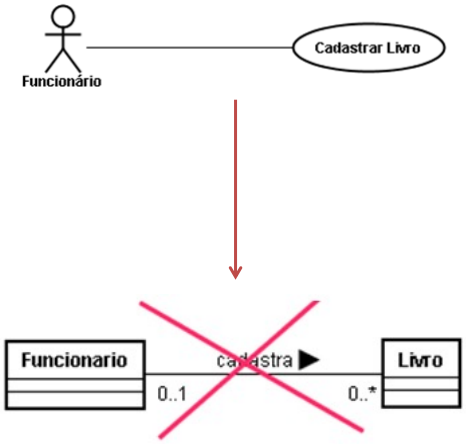
Para uma concessionária, a compra é um evento importante para o negócio e precisa ser registrado. Então, nesse caso, ela é tratada como uma classe e não como uma associação.



16

Análise de Requisitos

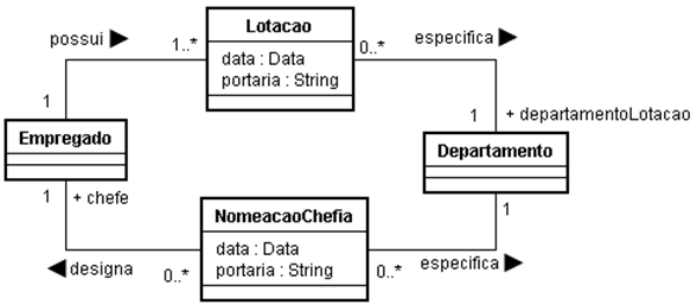
Mas…



17

Análise de Requisitos

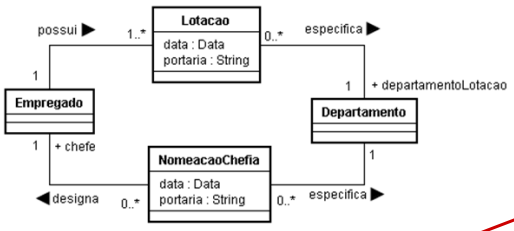
Registro de Históricos



18

Análise de Requisitos

Restrições para Complementar o Modelo



Um empregado pode ter mais de uma lotação vigente?

Um empregado pode ser chefe de mais de um departamento ao mesmo tempo?

Um departamento pode ter mais do que um chefe nomeado ao mesmo tempo?

Infelizmente, o modelo é incapaz de responder a essas perguntas.

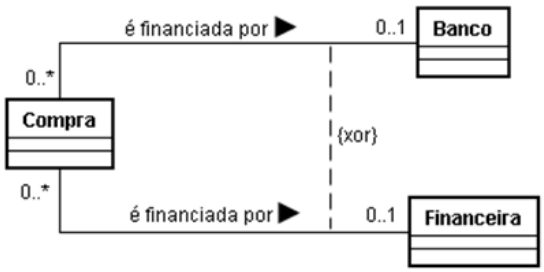
então

Restrições de Integridade:

- Um empregado só pode estar lotado em um único departamento em um dado momento.
- Um empregado só pode estar designado como chefe de um único departamento em um dado momento.
- Um departamento só pode ter um empregado designado como chefe em um dado momento.

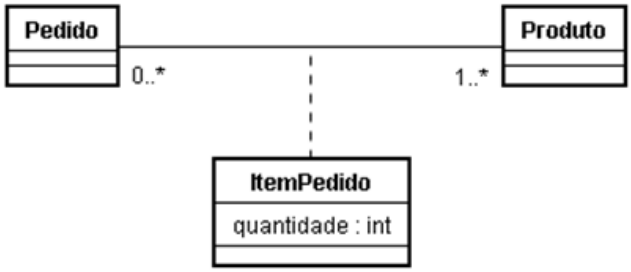
Análise de Requisitos

Exclusividade (XOR)



Análise de Requisitos

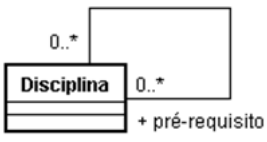
Classes Associativas



Análise de Requisitos

Mas…

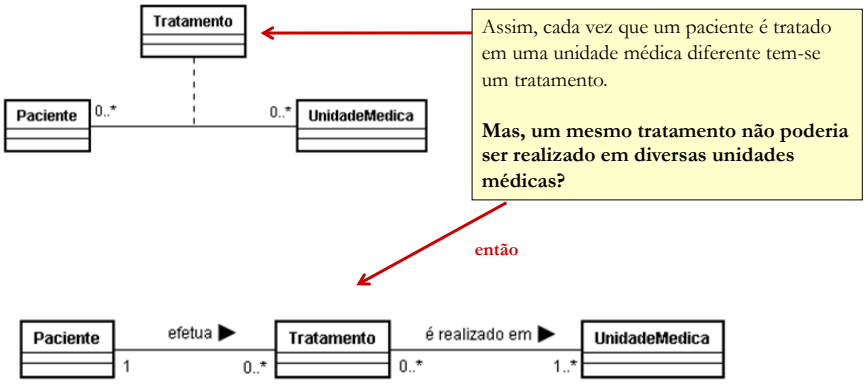
Não precisa de classe associativa.



Análise de Requisitos

Classes Associativas x Eventos Lembrados

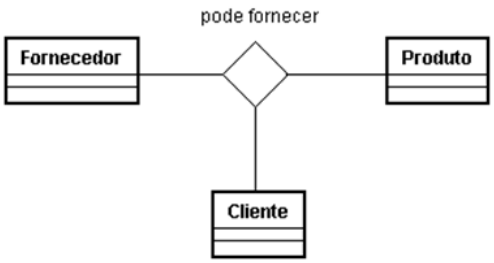
Situação: em um hospital, pacientes são tratados em unidades médicas. Um paciente pode ser tratado em diversas unidades médicas diferentes, as quais podem abrigar diversos pacientes sendo tratados.



23

Análise de Requisitos

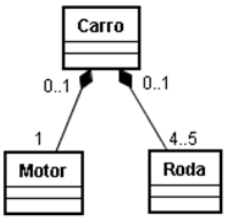
Associação Ternária



24

Análise de Requisitos

Agregação e Composição

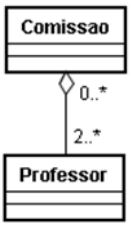


Composição

Um objeto-parte só pode ser parte de um único todo.

Agregação

Um objeto-parte pode ser parte de mais de um todo.

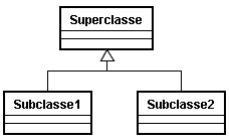


25

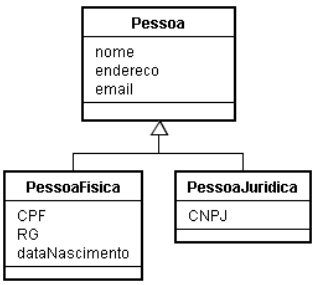
Análise de Requisitos

c) Identificação de Hierarquias de Generalização/Especialização

Notação:



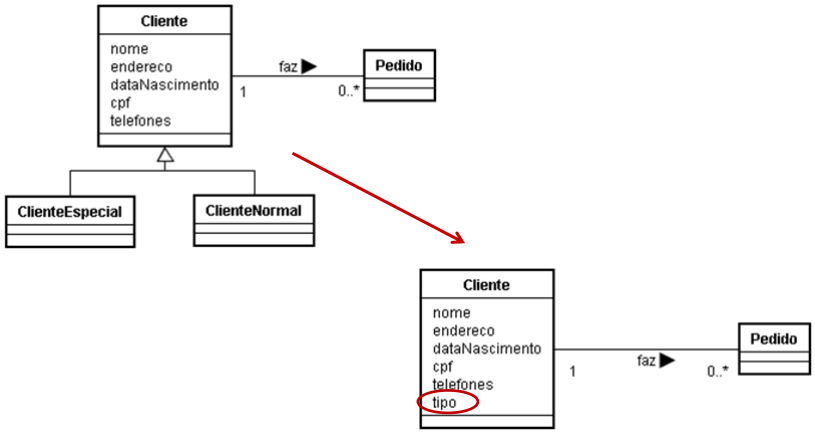
Exemplo:



26

Análise de Requisitos

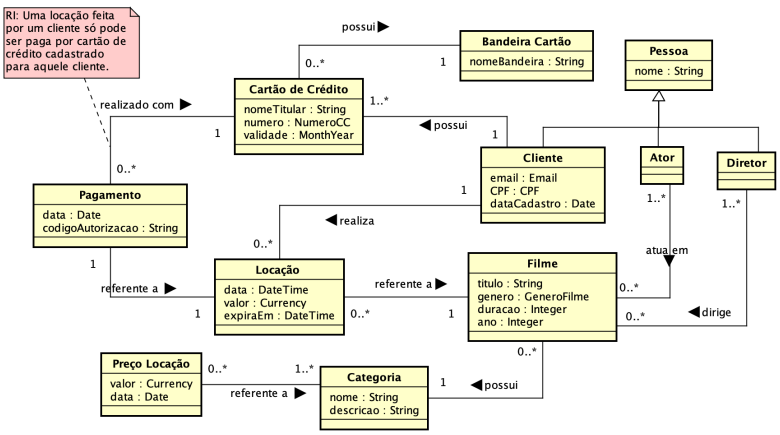
Uso ou Não da Herança



27

Análise de Requisitos

Exemplo



Exemplo para fins didáticos, considerando parte do escopo do projeto hipotético para a empresa Passatempo que foi usado em slides anteriores.

28

Universidade Federal do Espírito Santo

Centro Tecnológico

Departamento de Informática

Disciplina: ***INF15978 – Engenharia de Software I***

Prof.: ***Monalessa Perini Barcellos***

(monalessa@inf.ufes.br)

29