



FUNDAÇÃO UNIVERSIDADE ESTADUAL DE MARINGÁ

Centro de Tecnologia - CTC

Departamento de Informática - DIN

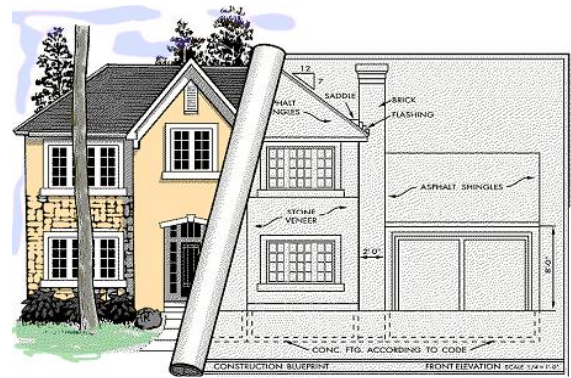
Programa de Pós-Graduação em Ciência da Computação – PCC

ESTÁGIO DE DOCÊNCIA II

Disciplina: Engenharia de Software I
 Professora: Itana Maria de Souza Gimenes
 Curso: Informática
 Turma: 2º Ano
 Mestranda: Edna Tomie Takano Yanaga
 Datas: 03/04/2006 e 05/04/2006

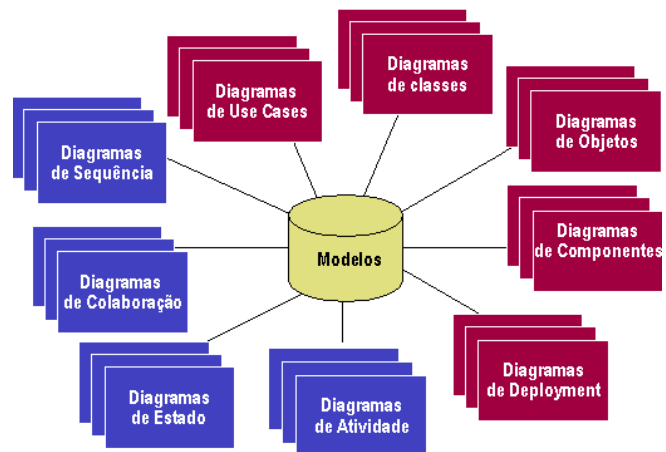
1. INTRODUÇÃO

- Impossível construir uma casa sem primeiramente definir sua planta.
- Impossível construir um *software* sem inicialmente definir sua arquitetura.
- Extremamente importante ter uma representação visual de seu sistema antes que ele entre na etapa de implementação.



2. UNIFIED MODELLING LANGUAGE (UML)

- ❖ Linguagem de modelagem visual utilizada para entender, projetar, navegar, configurar, manter e controlar informações sobre um sistema.
- ❖ Linguagem de modelagem visual utilizada para especificar, visualizar, construir e documentar artefatos de *software*.
- ❖ UML não é um processo de desenvolvimento. Porém, esta pode ser utilizada em qualquer processo.



2.1. ELEMENTOS DE MODELAGEM

Elementos	Descrição
Processo	Descreve <i>quem</i> está fazendo <i>o quê</i> , <i>como</i> e <i>quando</i> .
Worker (“ <i>quem</i> ”)	Descreve o comportamento do indivíduo no negócio e as responsabilidades.
Artefatos (“ <i>o quê</i> ”)	Informação que é produzido, modificado ou usado por um processo.

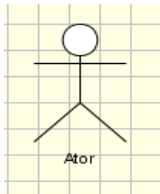
Atividades (“como”)	Unidade de trabalho que o indivíduo com um determinado papel deve executar e produzir um resultado no contexto do projeto.
Workflow (“quando”)	Seqüência de atividades que produzem um resultado de valor.

2.2. DIAGRAMA DE USE CASE

- ❖ Técnica utilizada para descrever e definir os requisitos funcionais de um sistema.
- ❖ Auxiliar na comunicação entre os analistas e o cliente.
- ❖ Descrever um cenário que mostra as funcionalidades do sistema do ponto de vista do usuário.
- ❖ O Diagrama de *Use Case* é composto de:

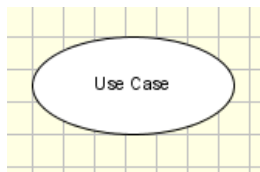


➤ Atores



- Um ator é um usuário do sistema, que pode ser um usuário humano ou um outro sistema computacional.
- Categorias de atores:
 - Pessoas (Empregado, Cliente, Gerente, Vendedor, etc).
 - Organizações (Empresa Fornecedora, Administradora de Cartões, etc).

➤ Use Cases

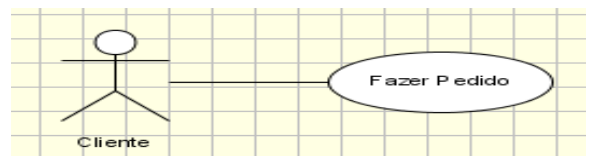


- Um *Use Case* é a especificação de uma seqüência de interações entre um sistema e os agentes externos que utilizam este sistema.
- Representa quem faz o que (interage) com o sistema, sem considerar o comportamento interno do sistema.
- Um *Use Case* é uma funcionalidade do sistema.

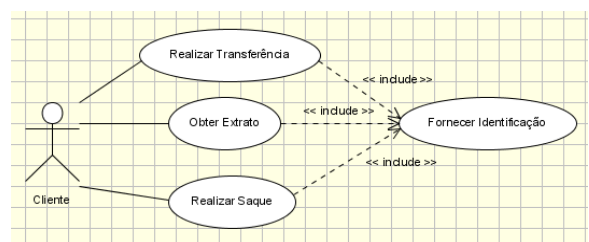
➤ Relacionamentos

- *Use Cases* e atores não existem sozinhos... Precisamos estabelecer seus relacionamentos!!
- Um ator deve estar relacionado a um ou mais *use cases* do sistema.
- Pode haver relacionamentos entre os *use cases* de um sistema.
- A UML define alguns tipos de relacionamentos no modelo de *use case*:

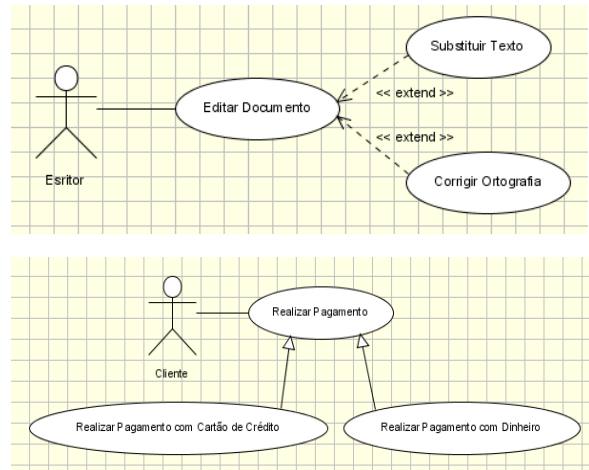
- ✓ Associação: Define uma funcionalidade do sistema do ponto de vista do usuário.



- ✓ Inclusão: Um relacionamento <<include>> de um use case A para um use case B indica que B é essencial para o comportamento de A.



- ✓ Estender: Um relacionamento <<extend>> de um use case A para um use case B indica que o use case A pode ser acrescentado para descrever o comportamento de B (não é essencial). A extensão é inserida no ponto de extensão do use case B.
- ✓ Generalização: Use case B *é um* use case A (A é uma generalização de B, ou B é uma especialização de A). Um relacionamento entre um use case genérico para um mais específico, que herda todas as características de seu pai.



2.2.1. DIRETRIZES PARA ELABORAR CASOS DE USO

❖ Identificar os atores:

- Porque o sistema está sendo desenvolvido?
- Quem são os usuários do sistema?
- Quais sistemas externos vão interagir com o novo sistema?

❖ Identificar os *use cases* sob o ponto de vista dos atores:

- Principais tarefas de cada ator?
- Ator precisa ler/escrever/modificar alguma informação no sistema?
- Ator precisa ser informado em alguma situação?
- Quais informações cada ator precisa informar ao sistema?

2.2.2. EXERCÍCIO

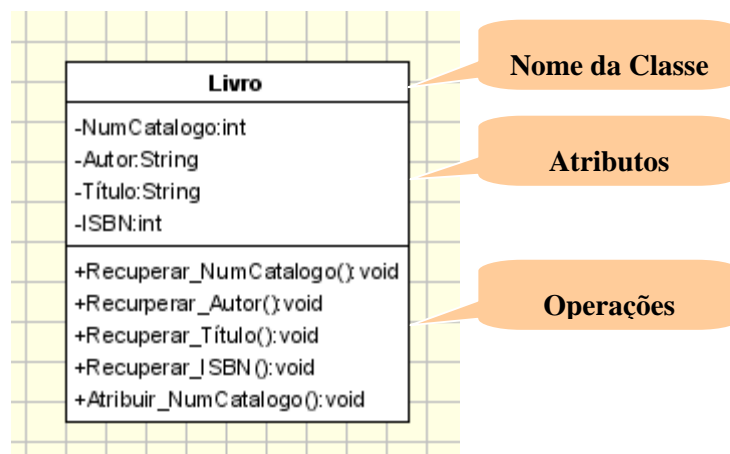
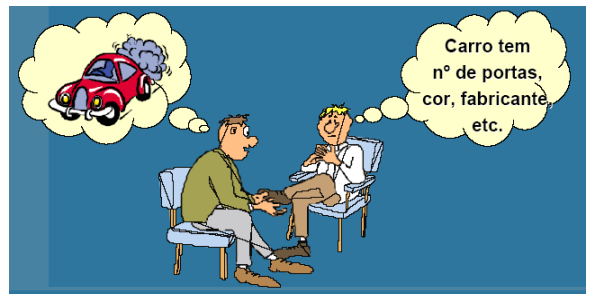
Da entrevista com o responsável da biblioteca de uma universidade resultou a seguinte descrição para um novo sistema:

“A atividade da biblioteca centra-se principalmente no empréstimo de publicações pelos alunos da universidade. O empréstimo é registrado pelos funcionários da biblioteca, que também consultam diariamente os empréstimos cujos prazos foram ultrapassados. Todo este processo é efetuado manualmente, sendo muito ineficiente. Espera-se que o novo sistema resolva esta situação. Os alunos necessitam de pesquisar os livros existentes na biblioteca. Caso um livro esteja requisitado é mostrada a data esperada de entrega”.

- Efetue o levantamento de requisitos e desenhe o respectivo diagrama de *use cases*.
- Sugestão:
 - ✓ Primeiro identifique os atores, em seguida identifique os respectivos *use cases* e, por fim, o desenho o diagrama.

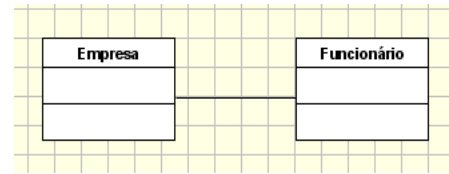
2.3. DIAGRAMA DE CLASSE

- ❖ Técnica utilizada para modelagem de classes e seus relacionamentos.
- ❖ Analisando os *use cases* podemos identificar classes e atributos.
- ❖ O que é um objeto?
 - É qualquer elemento do mundo real (concreto ou abstrato), constituído de atributos e operações.
- ❖ O que é uma classe?
 - Classe é a representação de um conjunto de objetos que compartilham a mesma estrutura e comportamento. Ela define:
 - ✓ Estrutura: De que o objeto é composto (atributos)
 - ✓ Comportamento: Como objetos reagem a eventos externos (operações).
- ❖ Uma classe é composta de:

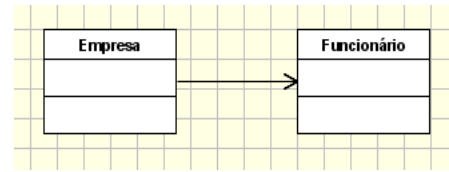


- ❖ Relacionamentos
 - As classes não vivem sozinhas... Precisamos estabelecer seus relacionamentos!!
 - É preciso definir a cardinalidade dos relacionamentos!!
 - ✓ Para expressar a multiplicidade entre os relacionamentos, um intervalo indica quantos objetos estão relacionados no *link*. O intervalo pode ser de zero para um (0..1), zero para vários (0..* ou apenas *), um para vários (1..*), dois (2), cinco para 11 (5..11) e assim por diante. É também possível expressar uma série de números como (1, 4, 6..12). Se não for descrito nenhuma multiplicidade, então é considerado o padrão de um para um (1..1 ou apenas 1).
 - A UML define alguns tipos de relacionamentos no modelo de classes. Os relacionamentos podem ser dos seguintes tipos:
 - ✓ Associação: É uma conexão entre classes, e também significa que é uma conexão entre objetos daquelas classes. Uma associação representa que duas classes possuem uma ligação (*link*) entre elas, significando, por exemplo, que elas "conhecem uma a outra", "estão conectadas com", "para cada X existe um Y" e assim por diante.

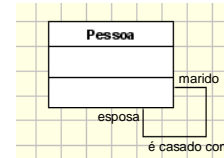
- Associação Normal: O tipo mais comum de associação é apenas uma conexão entre classes. É representada por uma linha sólida entre duas classes, podendo ser ligado de duas formas, ou seja, em suas duas direções.



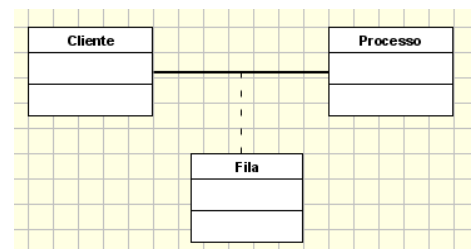
- Associação Direta: É uma associação normal com uma seta no final da associação indicando que esta só pode ser usada para o lado onde a seta aponta, ou seja, teremos apenas uma direção. Exemplo: Uma empresa possui um trabalhador.



- Associação Recursiva: É possível conectar uma classe a ela mesma através de uma associação e que ainda representa semanticamente a conexão entre dois objetos, mas os objetos conectados são da mesma classe. Uma associação deste tipo é chamada de associação recursiva.

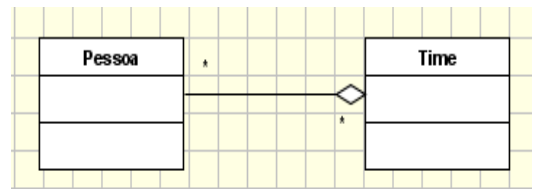


- Associação de Classe: Uma classe pode ser associada a uma outra associação. Este tipo de associação não é conectada a nenhuma das extremidades da associação já existente, mas na própria linha da associação. Esta associação serve para se adicionar informações extra a associação já existente.

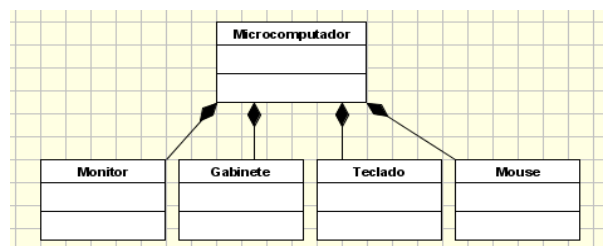


- ✓ Agregação: A agregação é um caso particular da associação. A agregação indica que uma das classes do relacionamento é uma parte, ou está contida em outra classe. As palavras chaves usadas para identificar uma agregação são: "consiste em", "contém", "é parte de". Existem tipos especiais de agregação que são:

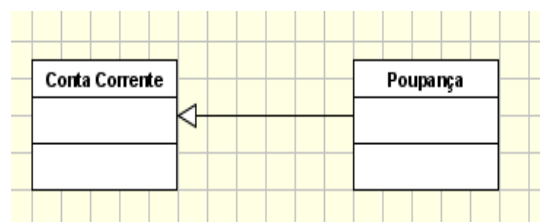
- Agregação Compartilhada: É dita compartilhada quando uma das classes é uma parte, ou está contida na outra, mas esta parte pode estar contida nas outras várias vezes em um mesmo momento.



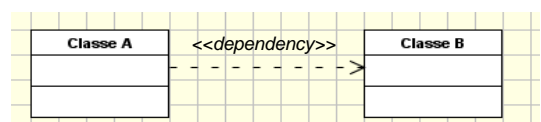
- Agregação de Composição: É uma agregação onde uma classe que está contida na outra "vive" e constitui a outra. Se o objeto da classe que contém for destruído, as classes da agregação de composição serão destruídas juntamente que as mesmas fazem parte da outra.



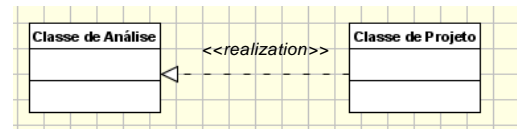
- ✓ Generalização: É um relacionamento de um elemento mais geral e outro mais específico. O elemento mais específico pode conter apenas informações adicionais. Uma instância (um objeto) do elemento mais específico pode ser usada onde o elemento mais geral seja permitido.



- ✓ Dependência e Refinamentos (Realização): Dependência é um relacionamento entre elementos, um independente e outro dependente. Uma modificação em um elemento independente



afetará diretamente elementos dependentes do anterior. Refinamento é um relacionamento entre duas descrições de uma mesma entidade, mas em níveis diferentes de abstração.



2.3.1. EXERCÍCIO

Considere a seguinte informação adicional à descrição apresentada no exercício de *use cases*. Esta informação consiste numa entrevista efetuada pelo consultor Paulo Bastos ao responsável da biblioteca João Almeida.

Paulo Bastos: Como é que funciona o processo de empréstimo de publicações?

João Almeida: Bom, neste momento as publicações disponíveis aos alunos são os livros e as revistas que assinamos. Um aluno dirige-se com as publicações ao balcão de atendimento para preencher uma ficha de empréstimo. Tem que efetuar uma ficha para cada publicação, preenchendo a cota e o título. Caso seja um livro, terá que escrever o(s) respectivo(s) autor(es).

Paulo Bastos: Existe alguma limitação no número de empréstimos?

João Almeida: Sim, no máximo um aluno pode efetuar 3 empréstimos.

Paulo Bastos: Qual é o procedimento quando chega uma nova publicação?

João Almeida: Bem... quando chega uma nova publicação esta é encaminhada para a responsável de catalogação onde será analisada e definida a sua área de conhecimento. Existem várias áreas pré-definidas como, por exemplo, Sociologia, Psicologia, Informática, etc. Novas áreas de conhecimento podem ser definidas.

Paulo Bastos: Existe alguma informação específica sobre cada uma das publicações?

João Almeida: Para os livros temos que registrar o seu número de identificação internacional, ISBN, e para as revistas registramos a sua periodicidade.

- Identifique classes e desenhe o respectivo diagrama.
 - ✓ Sugestão: Primeiro identifique os vários objetos, em seguida agrupe-os em possíveis classes e por fim desenhe o diagrama.

3. POSEIDON FOR UML COMMUNITY EDITION 4.1

