AESO

Faculdades Integradas Barros Melo

Introdução a Modelagem de Software

UML – Unified Modeling Language

Prof. Thiago Affonso de M. N. Viana

Histórico

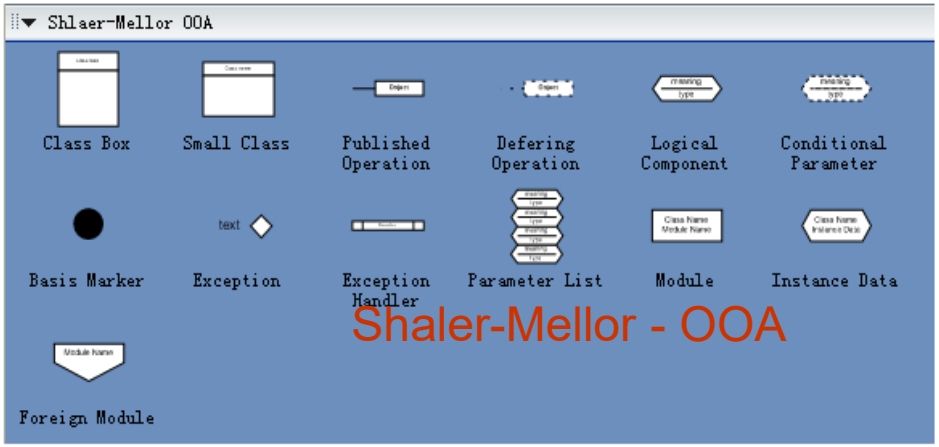
- 1950/60 – Sistemas ad hoc
  - Época dos fluxogrmas
- 1970 – Programação estruturada
  - Tom de Marco, Edward Yourdon
- 1980 – Análise estruturada moderna
  - Necessidade de interface mais sofisticada
  - DFD, DTE, DER
- 1990 – Análise Orientada a Objetos
  - Shaler, Mellor, Booch, Jacobson, Rumbaugh

AESO

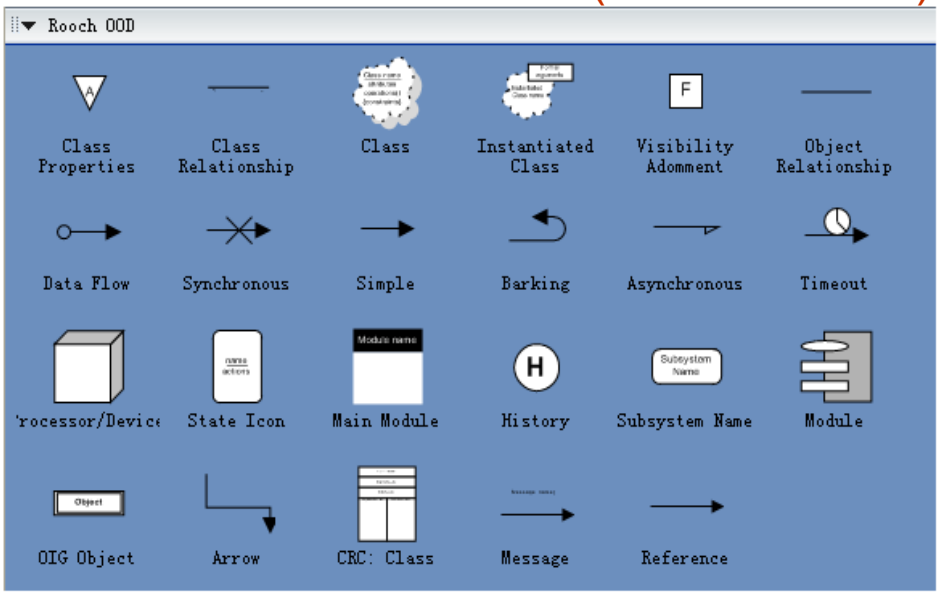
Faculdades Integradas Barros Melo

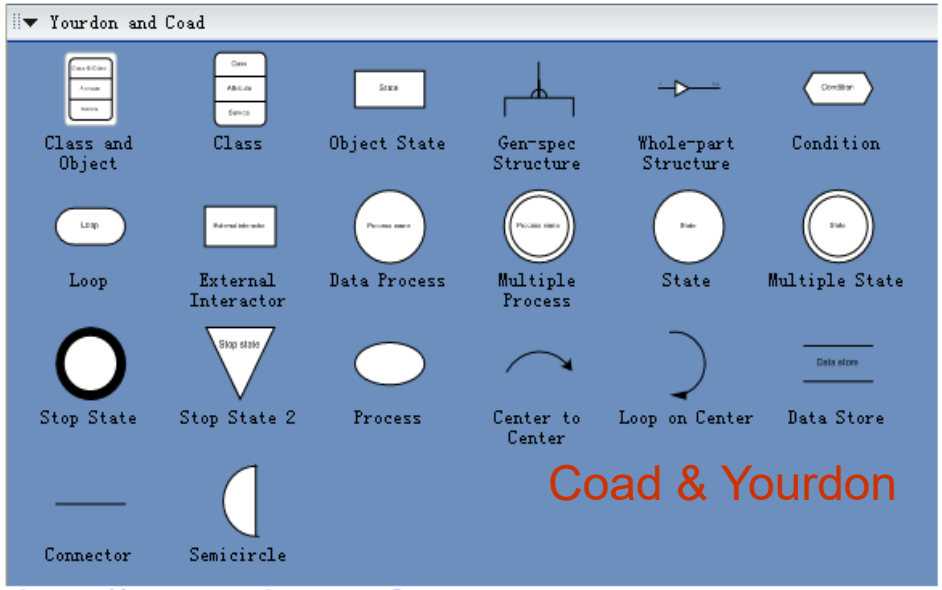
Técnicas para modelagem

Booth (Booch Method)

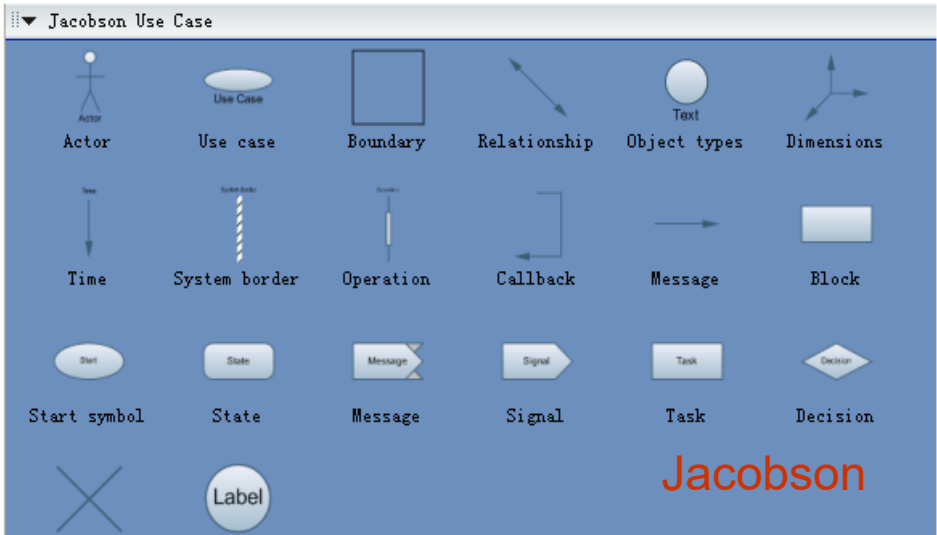


Shaler-Mellor - OOA





Coad & Yourdon



Jacobson

http://www.edrawsoft.com

Histórico – Técnicas de modelagem OO

Ano	Autor (Técnica)
1990	Shaler & Mellor
1991	Coad & Yourdon (OOAD – Object-oriented Analysis and Design)
1993	Grady Booch (Booch Method)
1993	Ivar Jacobson (OOSE – Object Oriented Software Engineering)
1995	James Rumbaugh et al (OMT – Object Modeling Technique)
1996	Wirfs-Brock (Responsability Driven Design)
1996	Surge a UML como a melhor candidata de notações, diagramas e formas de representação

UML – Linguagem de Modelagem Unificada

- Principais autores do processo: Grady Booch, James Rumbaugh, Ivar Jacobson
- Chamados os 3 amigos
- Aproveitar o melhor das caracterísiticas das notações preexistentes
- *Notação da UML é uma união das diversas notações prexistentes com alguns elementos removidos e outros adicionados com o objetivo de torna-la mais expressiva.*

UML – Linguagem de Modelagem Unificada

- 1997 – A UML foi aprovada pela OMG (Object Management Group)
- A definição passa por constantes melhorias e conta com diversos colaboradores comerciais(Digital, HP, IBM, Oracle, Microsof, Unysis, etc)
- 2003 – Foi lançada a UML 2.0
  - Especificação atual adotada pela OMG

UML – Linguagem de Modelagem Unificada

- UML
  - é uma linguagem visual para modelar sistemas Orientados a Objetos
  - Define elementos gráficos que podem ser utilizados na modelagem de sistemas
  - Através dos elementos definidos na linguagem podem-se construir diagramas para representar diferentes perspectivas de um sistema
  - Cada elemento gráfico possui uma
    - Sintaxe: forma predeterminada de desenhar o elemento
    - Semântica: O que significa o elemento e com que objetivo deve ser usado
  - A sintaxe e a semântica são extensíveis

UML – Linguagem de Modelagem Unificada

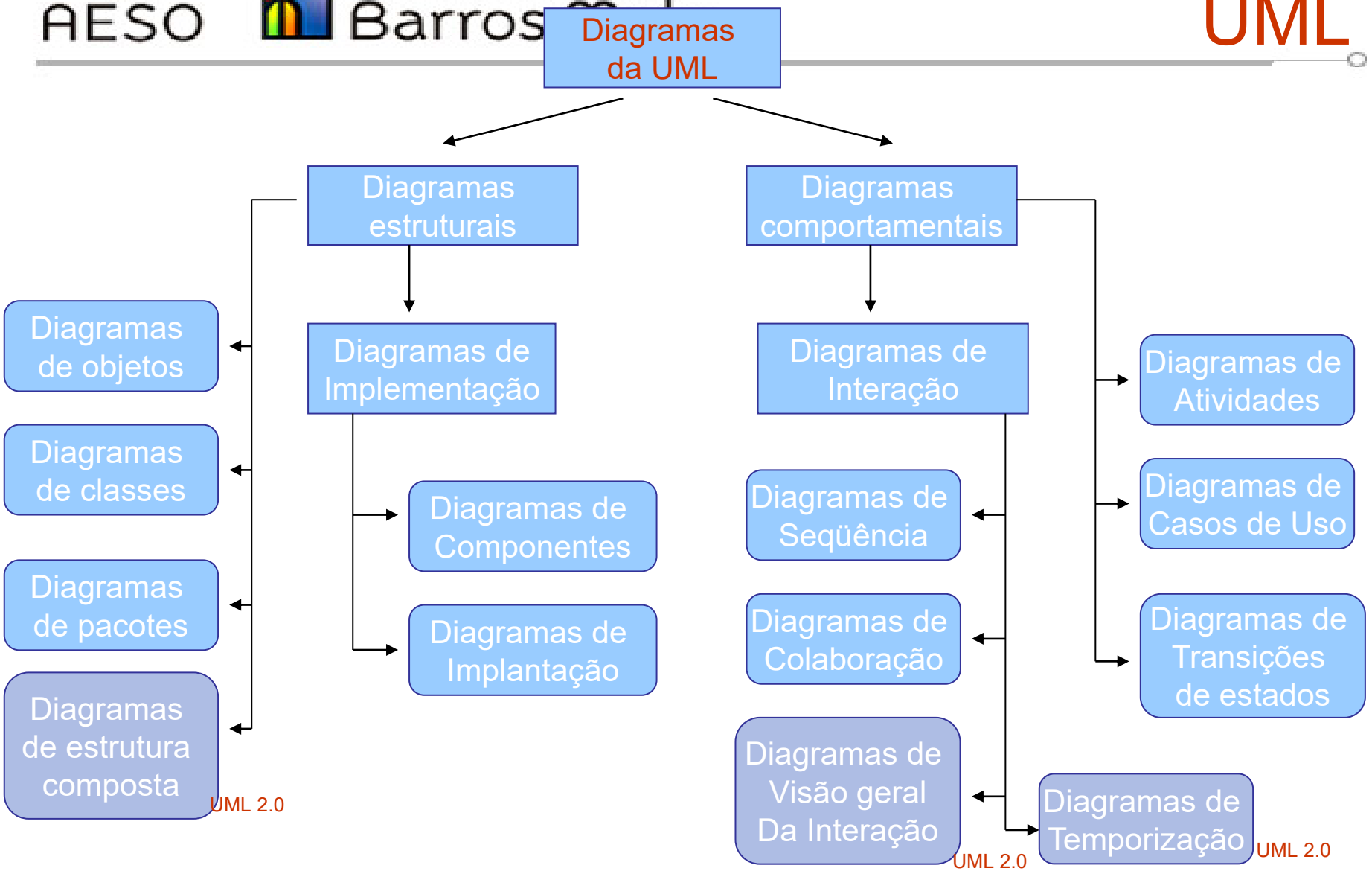
- UML
  - É independente de linguagens de programação e de processo de desenvolvimento
  - Definição completa:
    - www.uml.org
    - Especificação de leitura complexa voltada a pesquisadores ou desenvolvedores de ferramentas de suporte

UML – Linguagem de Modelagem Unificada

- Visões de um sistema
  - Um sistema complexo pode ser examinado a partir de diversas perspectivas.
  - Autores da UML definem 5 visões:
    - Visão de Casos de uso: Visão externa do sistema que define a interação entre o sistema e agentes externos.
    - Visão de Projeto: Caracterísiticas estruturais e comportamentais do sistema.
    - Visão de Implementação: gerenciamento de versões construídas pelo agrupamento de módulos e subsistemas.
    - Visão de Implantação: Distribuição física do sistema.
    - Visão de Processo: Caracterísiticas de concorrência, sincronização e desempenho do sistema.

UML – Linguagem de Modelagem Unificada

- Diagramas:
  - Os documentos gerados em um processo de desenvoleimento são chamados de artefatos na UML
  - Os artefatos compõe as visões do sistema
  - A UML define 13 diagramas
  - Esta quantidade de diagramas é justificada pela necessidade de analisar o sistema por meio de diferentes perspectivas
  - Cada diagrama fornece uma perspectiva parcial do sistema.


Diagramas da UML
Diagramas estruturais
Diagramas comportamentais
Diagramas de objetos
Diagramas de classes
Diagramas de pacotes
Diagramas de estrutura composta
UML 2.0
Diagramas de Implementação
Diagramas de Componentes
Diagramas de Implantação
Diagramas de Interação
Diagramas de Seqüência
Diagramas de Colaboração
Diagramas de Visão geral Da Interação
UML 2.0
Diagramas de Atividades
Diagramas de Casos de Uso
Diagramas de Transições de estados
Diagramas de Temporização
UML 2.0

UML – Mecanismos gerais

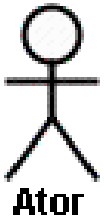
- Componentes da UML
  - Blocos de construção básicos
  - Regras que restringem como os blocos de construção podem ser associados
  - Mecanismos de uso geral
    - Estereótipos, Notas explicativas, Etiquetas valoradas, Restrições, Pacotes, OCL

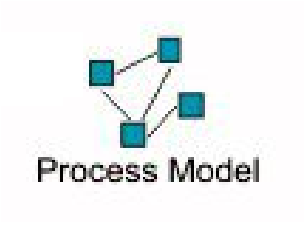
UML – Mecanismos gerais

- Estereótipos
  - Estende o significado de determinado elemento em um diagrama
    - Existem estereótipos predefinidos
    - O usuário pode definir um estereótipo
  - Um estereótipo deve ser documentado para evitar ambigüidades
  - Estereótipos gráficos: Ícones gráficos
  - Estereótipos textuais: Rótulo junto ao símbolo que representa.

UML – Mecanismos gerais

- Estereótipos Gráficos





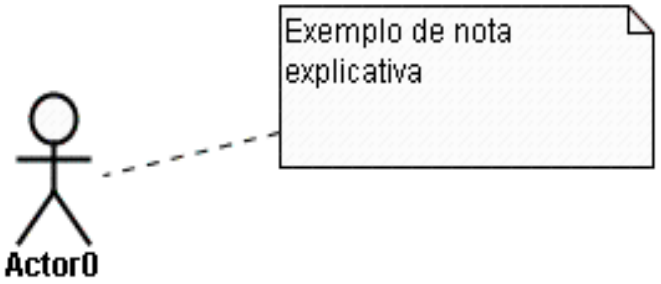
- Estereótipos Textuais

<<document>>
<<interface>>
<<entity>>

<<satisfaz>>
<<realiza>>

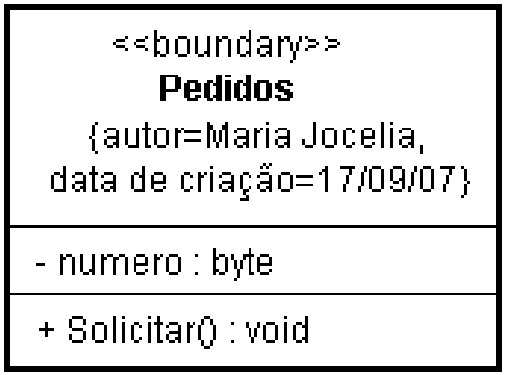
UML – Mecanismos gerais

- Notas explicativas
  - Comenta ou esclarece alguma parte do diagrama
    - Textuais
    - Linguagem de restrição de objetos (OCL)
  - Não modificam nem estendem o significado do elemento
  - Não deve ser usado em excesso



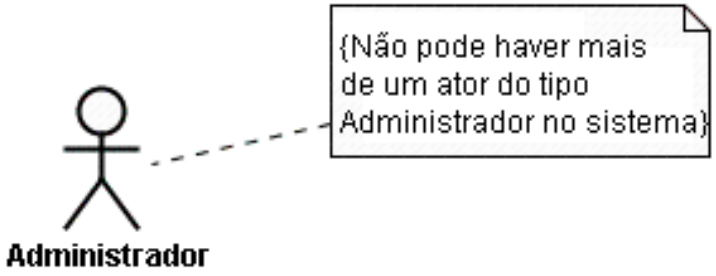
UML – Mecanismos gerais

- Etiquetas valoradas (tagged value)
  - Os elementos da UML tem 3 propriedades predefinidas: nome, lista de atributos e lista de operações
  - Etiquetas valoradas são usadas para definição de outras propriedades além das 3 predefinidas
  - Na UML 2.0 somente pode-se usar uma etiqueta valorada como um atributo usado sobre
    estereótipo
  - Notação
    - `{tag=valor}`



UML – Mecanismos gerais

- Restrições
  - Podem estender ou alterar a semântica natural de um elemento gráfico
  - Podem ser especificadas formalmente (OCL) ou informalmente (texto livre)
  - Restrições devem aparecer dentro de notas explicativas

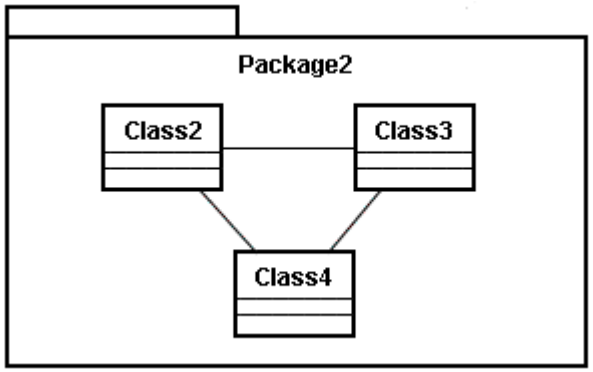


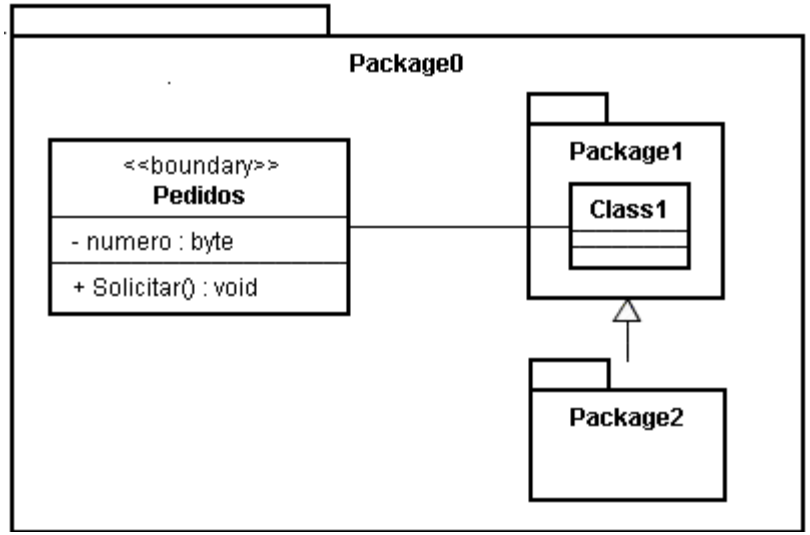
UML – Mecanismos gerais

- Pacotes
  - Agrupa elementos semanticamente relacionados
  - Um pacote se liga a outro através de uma relacionamento de dependência
  - A dependência pode ser especificada através de um estereótipo
  - Pode agrupar outros pacotes

UML – Mecanismos gerais

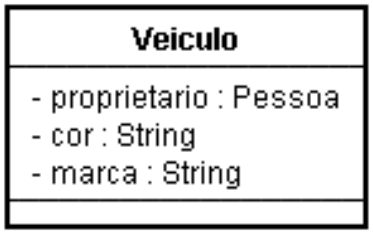
- Pacotes





UML – Mecanismos gerais

- OCL (Linguagem de restrição de objetos)
  - Linguagem formal para especificar restrições sobre diversos elementos em um modelo
  - Consiste de:
    - Contexto: Domínio no qual a declaração em OCL se aplica
    - Propriedade: um componente do contexto
    - Operação: O que deve ser aplicado sobre a propriedade
  - Exemplo:



Context Veículo

inv: self.proprietário.idade >= 18

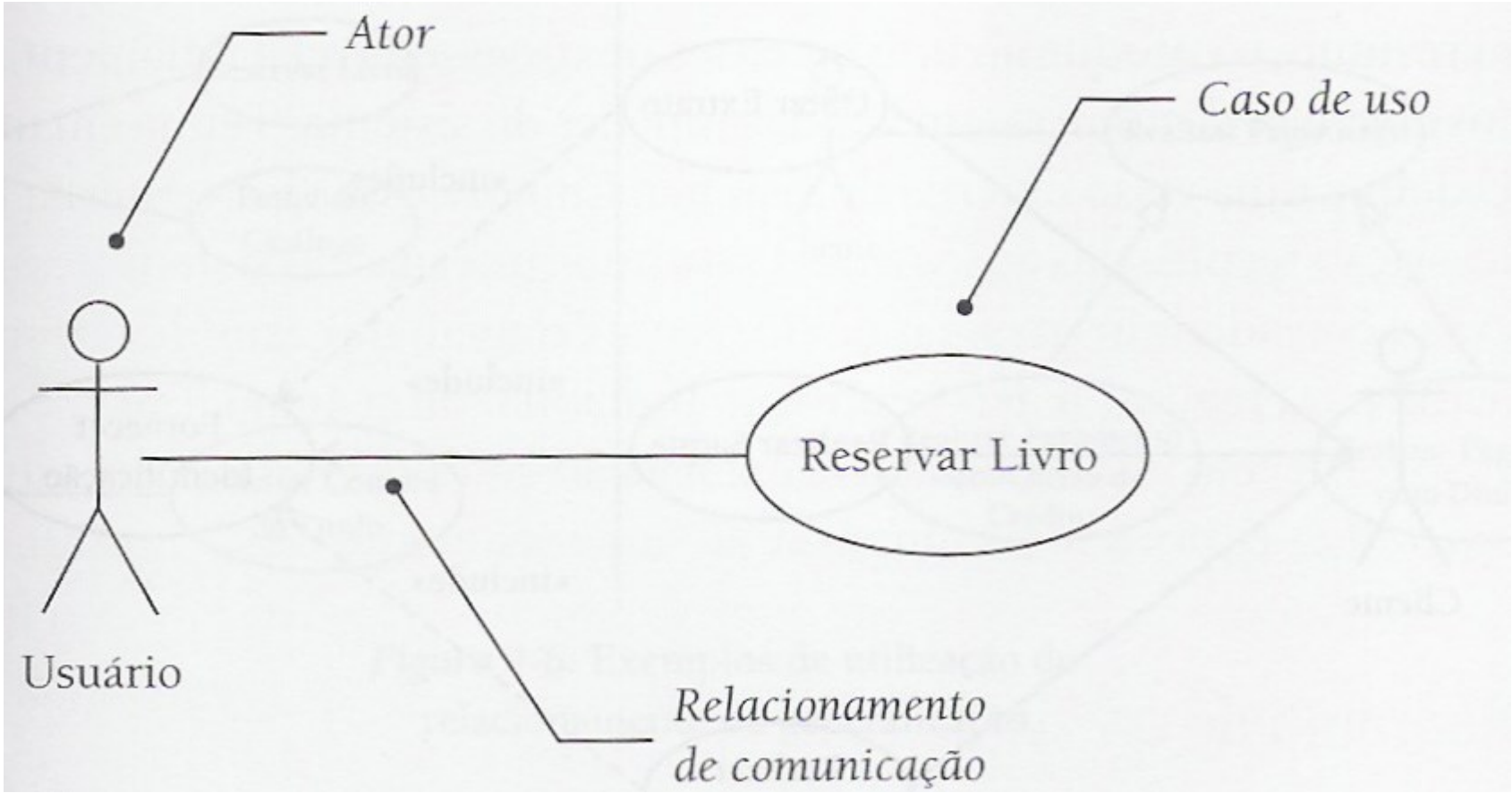
1º Modelo:

Diagrama de Casos de Uso

Introdução

- Modelos de Casos de Uso
  - É uma representação das funcionalidades eternamente observáveis do sistema e dos elementos externos ao sistema que interagem com eles
  - É um modelo de análise que representa um refinamento dos requisitos funcionais.
  - Idealizado por Ivar Jacobson em 1970 e inserida na UML na década de 90.
  - É o modelo mais popular para a documentação de requisitos funcionais
  - O MCU representa os possíveis usos de um sistema.
  - Componentes: *Casos de Usos, Atores e Relacionamentos*

Notação da UML

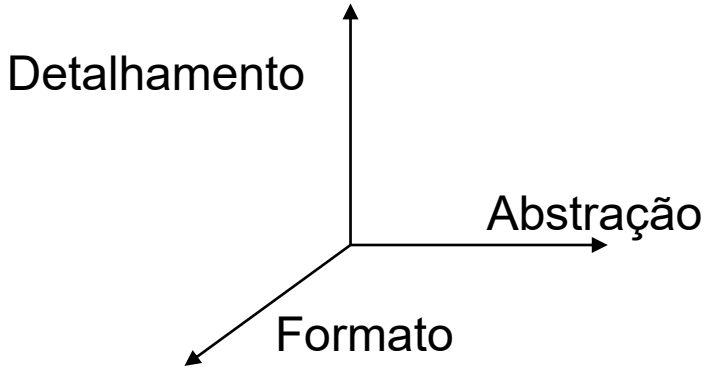


Casos de Uso

- É a especificação de uma seqüência completa de interações entre um sistema e um ou mais agentes externos a este sistema.
  - Representa uma determinada funcionalidade de um sistema conforme percebida externamente.
  - Representa também os agentes externos que interagem com o sistema
  - Não revela a estrutura e o comportamento interno do sistema.
- Completa representa um relato fim a fim de um dos usos do sistema para alcançar um objetivo util.
  - *Ex: Entrar no sistema **não** é um caso de uso*

Casos de Uso

- Um MCU pode conter vários casos de uso
- Cada caso de uso se define pela descrição narrativa das interações entre o agente externo e o sistema.
- Há 3 dimensões para variações das descrições dos casos de uso



Casos de Uso

- Formato: estrutura utilizada para organizar a sua narrativa textual:
  - Contínuo, numerado, tabular

- Formato contínuo

> Este caso de uso inicia quando o Cliente chega ao caixa eletrônico e insere seu cartão. O sistema requisita a senha do Cliente. Após o Cliente fornecer sua senha e esta ser validada, o Sistema exibe as opções de operações possíveis. O Cliente opta por realizar um saque. Então o Sistema requisita o total a ser sacado. O Cliente fornece o valor da quantidade que deseja sacar. O sistema fornece a quantia desejada e imprime o recibo para o Cliente. O Cliente retira a quantia e o recibo , e o caso de usa termina.

Casos de Uso

- Formato numerado

1) Cliente insere seu cartão no caixa eletrônico.
2) O sistema requisita a senha do Cliente.
3) Cliente fornecer sua senha
4) Sistema valida a senha e exibe as opções de operações possíveis.
5) O Cliente opta por realizar um saque.
6) Sistema requisita o total a ser sacado.
7) O Cliente fornece o valor da quantidade que deseja sacar.
8) O sistema fornece a quantia desejada e imprime o recibo para o Cliente.
9) O Cliente retira a quantia e o recibo , e o caso de usa termina.

Casos de Uso

- Formato Tabular

Cliente	Sistema
Insere seu cartão no caixa eletrônico.	Requisita a senha do Cliente.
Fornecer sua senha	Valida a senha e exibe as opções de operações possíveis.
Solicita realização de um saque.	Requisita o total a ser sacado.
Fornece o valor da quantidade que deseja sacar.	Fornece a quantia desejada e imprime o recibo para o Cliente.
Retira a quantia e o recibo	

Tenta prover alguma estrutura à descrição

Casos de Uso

- Grau de detalhamento
  - Sucinto: Não detalha as interações
  - Expandido: Descreve as interações em detalhes

- Grau de abstração
  - Existência ou não de menção a aspectos relativos a tecnologia durante a descrição de um caso de uso
    - Real: Se compromete com a solução do projeto
      - Ex: O usuário insere o seu cartão magnético
    - Essencial: É abstrato no sentido de não mencionar aspectos relativo ao uso de tecnologias
      - Ex: O usuário fornece sua identificação

Casos de Uso

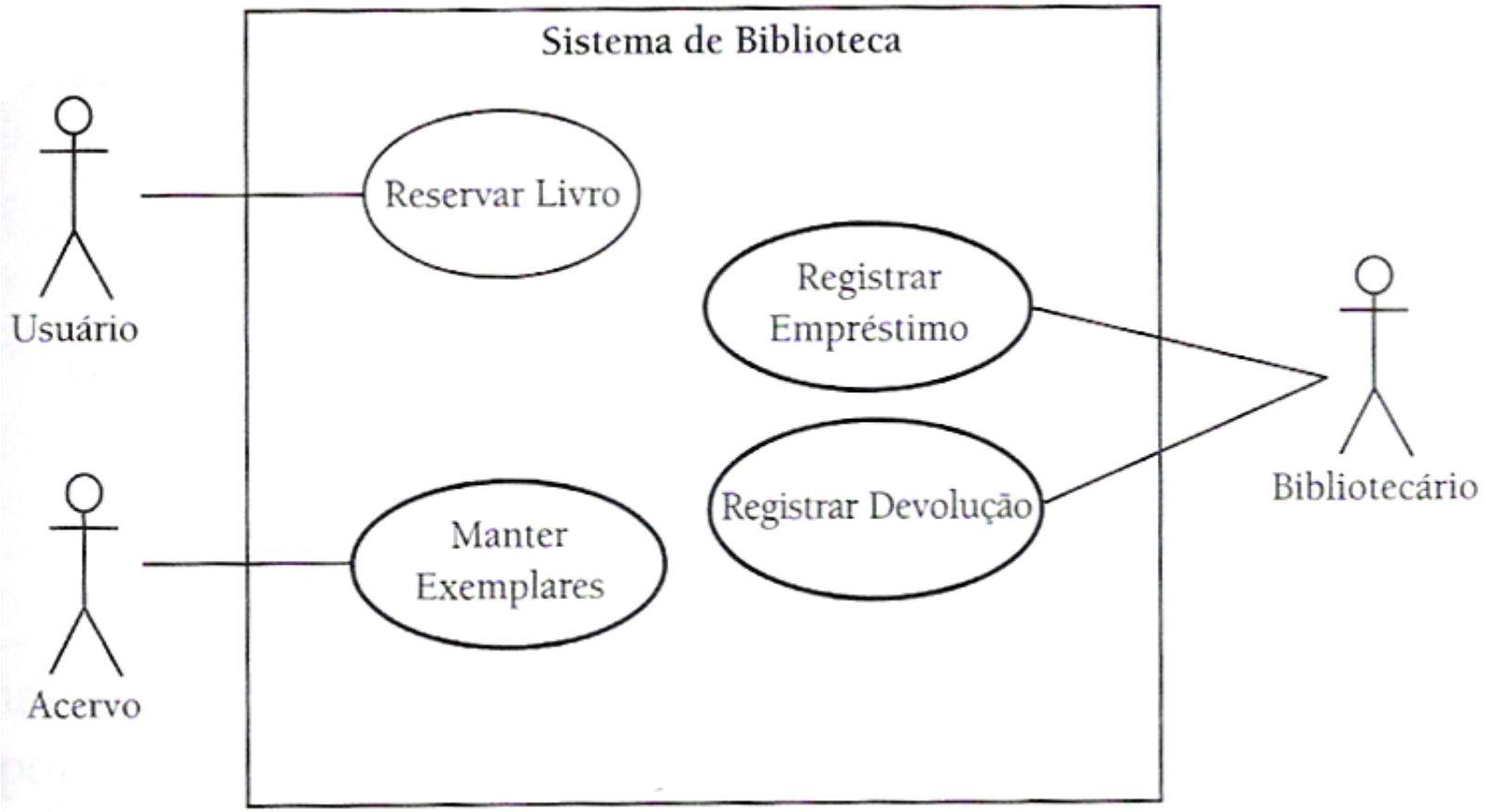
- Cenários
  - É a descrição de uma das maneiras pelas quais o caso de uso pode ser utilizada.
  - É um episódio de utilização de uma funcionalidade.
  - Pode ser utilizada posteriormente na fase de testes
  - Pode ser vista como uma instância de um caso de uso.

- Cliente insere seu cartão no caixa eletrônico.
- O sistema apresenta a tela de requisição de senha do Cliente.
- Cliente digita sua senha
- Sistema valida a senha e exibe o menu com as opções de saque, pagamento ou transferência.
- O Cliente seleciona a opção saque.
- Sistema apresenta tela com a requisição do valor a ser sacado.
- O Cliente digita o valor da quantidade que deseja sacar.
- ...

Atores

- É qualquer elemento externo ao sistema que interage com o mesmo
  - Atores não fazem parte do sistema
  - Atores trocam informações com o sistema
- Um ator representa um papel representado em relação ao sistema
- Categorias
  - Cargos
  - Organizações ou divisões de uma organização
  - Outros sistemas de software
  - Equipamentos que o sistema se comunica
- Atores podem ser Primários ou Secundários

Diagrama de Casos de Uso

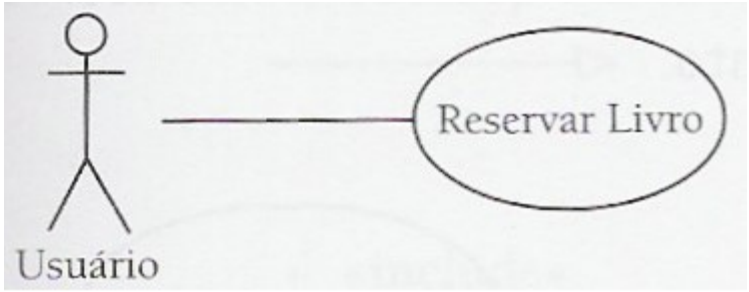


Relacionamentos

- Componente responsável por representar a interação entre os atores e casos de usos. **(Ator <--> Caso de Uso)**
- Também representa ligações entre casos de uso ou entre atores. **(Ator <--> Ator; Caso de Uso <-->Caso de Uso)**
- Tipos de Relacionamentos no MCU:
  - Comunicação
  - Inclusão
  - Extensão
  - Generalização

Relacionamentos

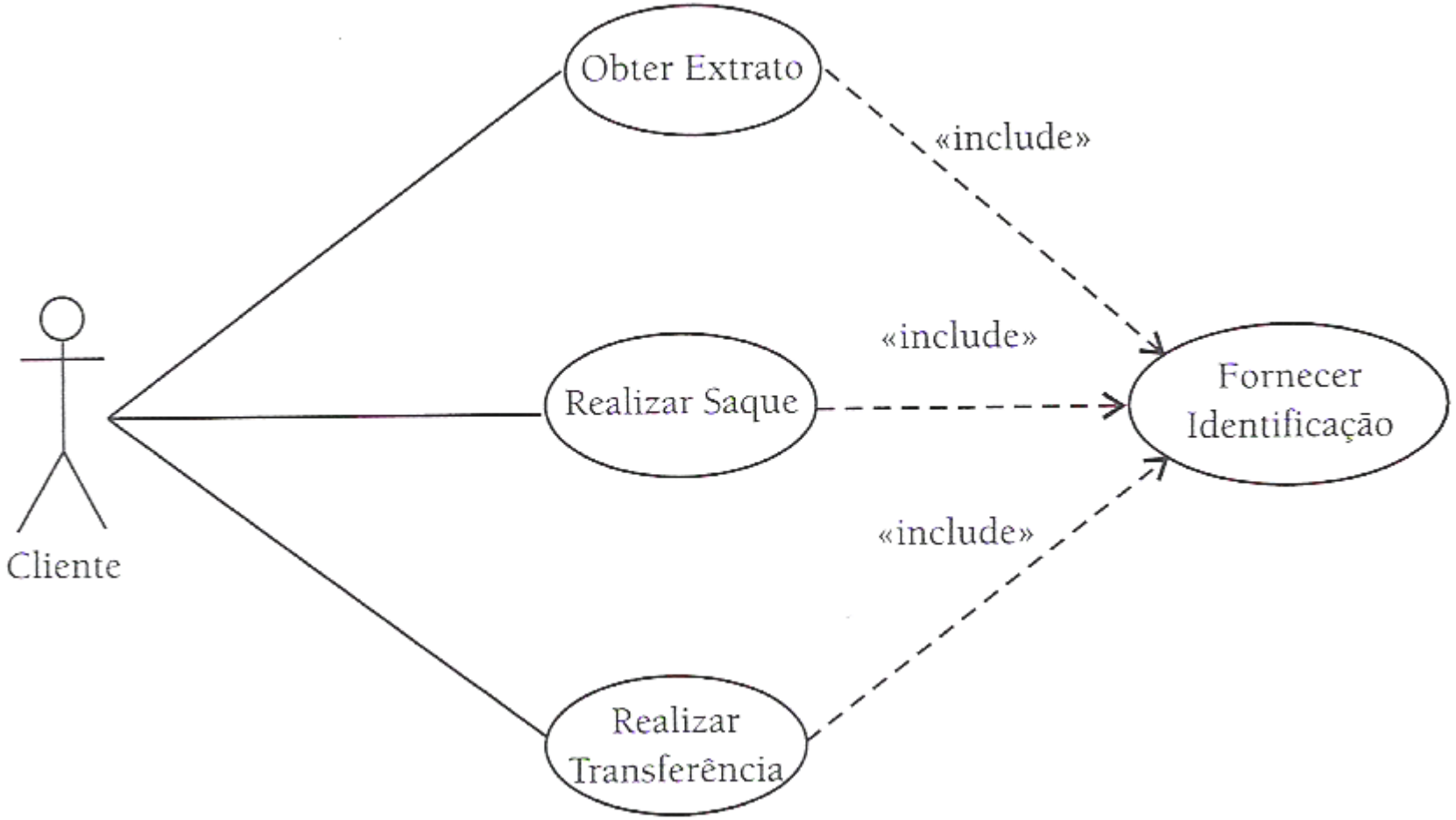
- Comunicação:
  - Informa a que caso de uso o ator está associado
  - Representa as trocas de informação entre os atores e casos de uso
  - É o mais utilizado nos MCU
  - Um ator pode estar associado a vários casos de uso
  - Um caso de uso pode estar associado a vários atores



Relacionamentos

- Inclusão:
  - Somente entre Casos de Usos
  - Quando dois ou mais casos de usos incluem uma sequencia comum de interações, esta sequencia pode ser descrita em outro caso de uso
  - Vários caos de uso podem incluir o comportamento deste caso de uso comum.
  - Ex: *Obter Extrato*, *Realizar Saque* e *Transferência* incluem *Validar Senha*

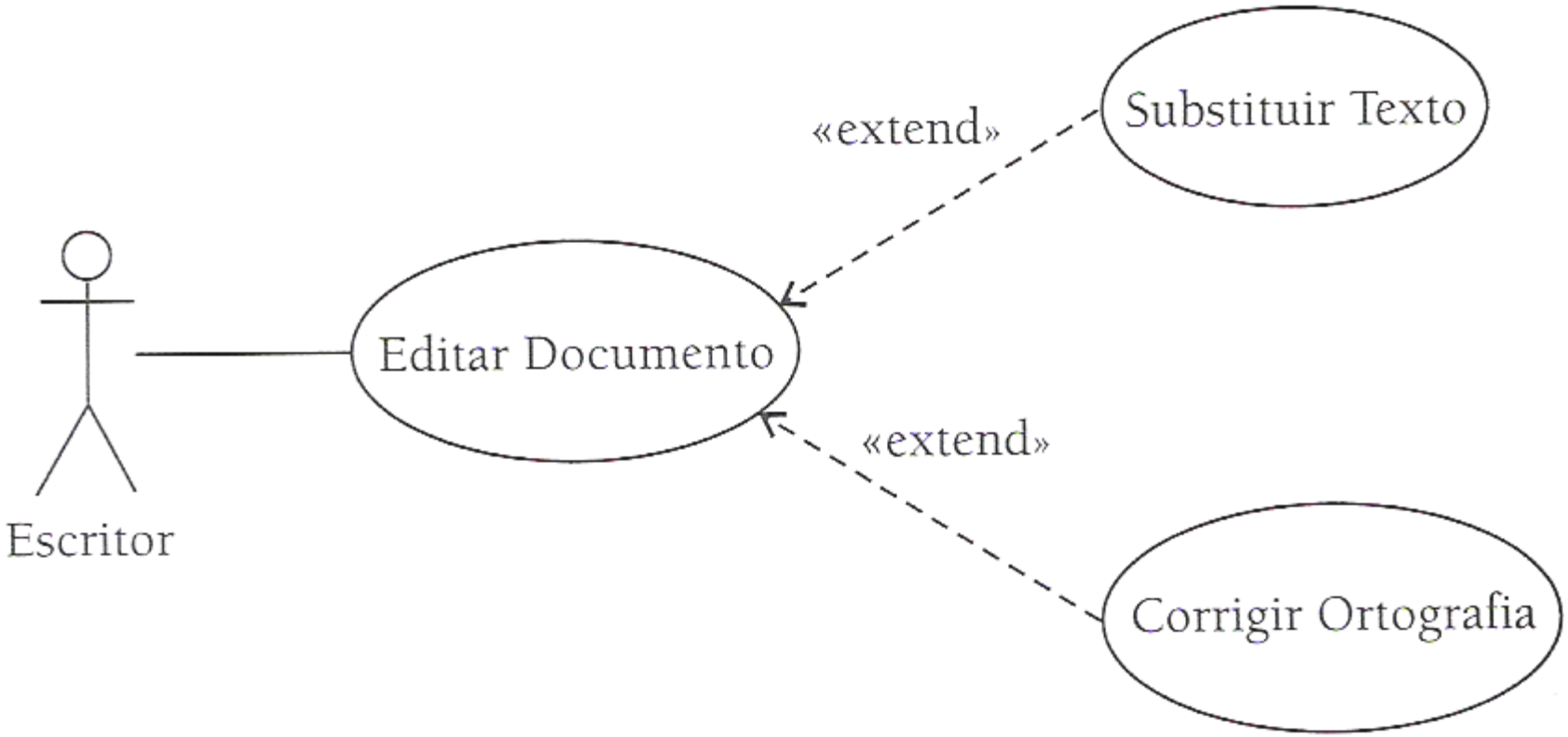
Diagrama de caso de Uso - Inclusão



Relacionamentos

- Extensão:
  - Somente entre Casos de Usos
  - Modelar situações em que diferentes seqüências de interações podem ser inseridas em um mesmo caso de uso. Estas seqüências representam um comportamento eventual.
  - A existência de um caso de uso estendido deve ser independente da existência de casos de uso que estendam o primeiro
  - Exemplo: *Realizar Saque* e *Transferência* podem ser estendidos por *Consultar Extrato*

Diagramas de Caso de Uso - Extensão



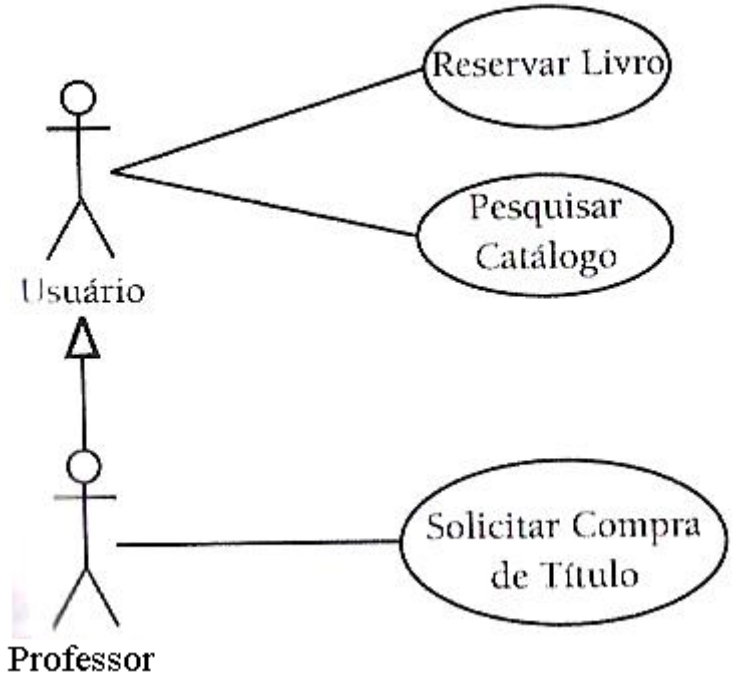
Relacionamentos

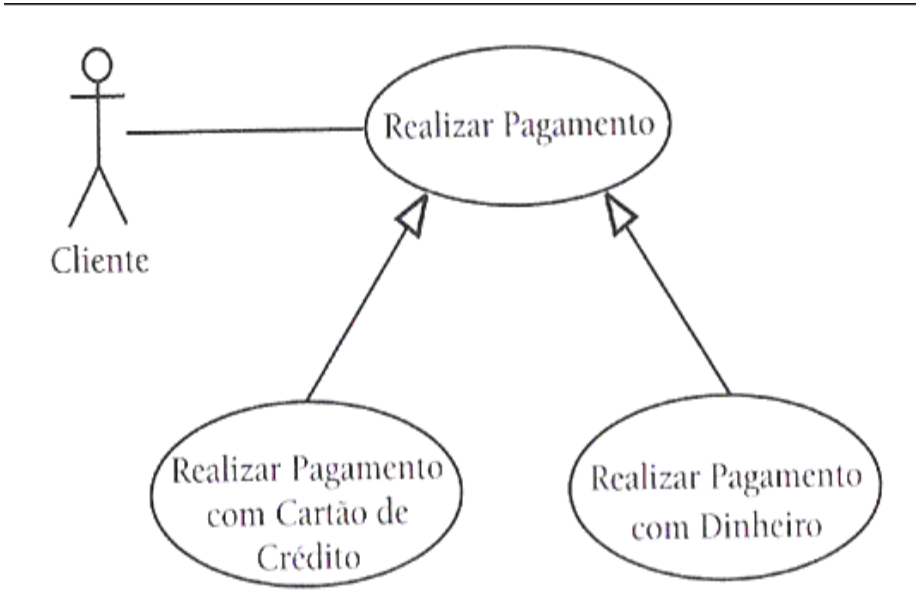
- Generalização:
  - Pode existir entre dois casos de Uso ou entre dois atores
  - Permite que um caso de uso (ou um ator) herde comportamento de outro caso de uso (ou ator)
  - O caso de uso mais genérico pode ser Abstrato ou concreto.
  - É recomendado que o caso de uso pai sempre seja abstrato para evitar problemas na especificação
  - O caso de uso pai é utilizado apenas para representar a natureza dos casos de uso filho.
  - Não há especificação de comportamento para o caso de uso abstrato.

Relacionamentos

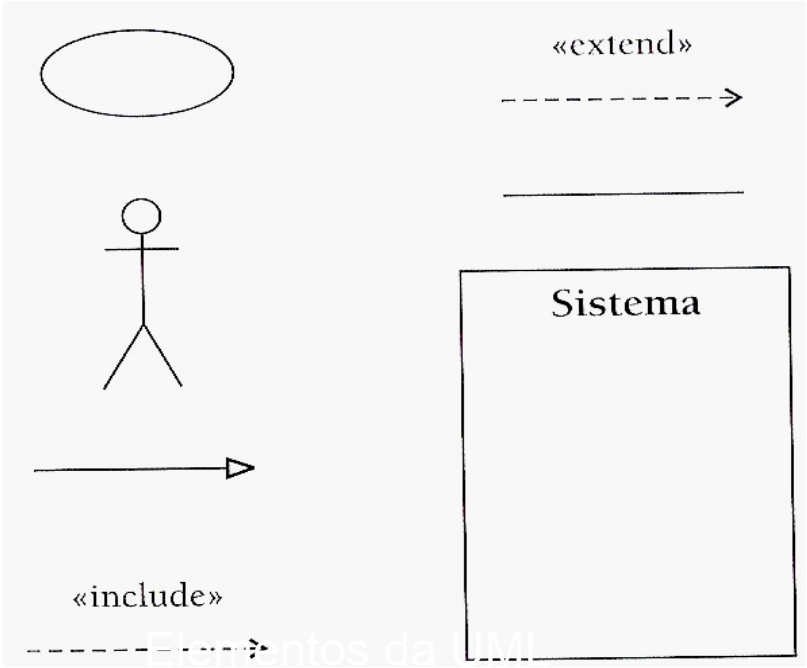
- Generalização:
  - Exemplos:
    - *Requisitar Produtos* é generalização de *Requisitar produtos na loja*, *Requisitar produtos pela internet*
    - *Usuário* é generalização de *Professor* e *Aluno*

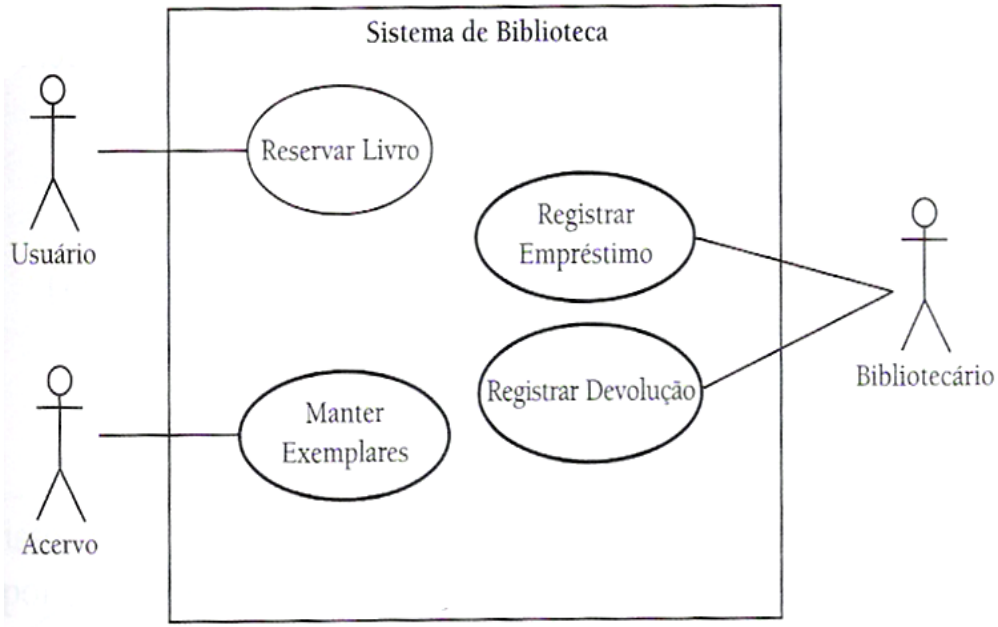
Diagramas de Caso de Uso - Generalização





Diagramas de Casos de Uso





Quando usar relacionamentos

- Inclusão
  - Quando o mesmo comportamento se repete em mai de um caso de uso
  - O Comportamento comum necessariamente contido em todos os cenários do caso de uso *inclusor*
  - O caso de uso *inclusor* não está completo sem o caso de uso *incluso*
- Extensão
  - Quando um comportamento eventual do caso de uso tiver que ser descrito
  - Para estender o comportamento do caso de uso sem modificar o original

Quando usar relacionamentos

- Generalização de caso de uso
  - Identifica-se vários casos de uso com o mesmo comportamento
  - Se o comportamento do pai difere em alguma coisa do do filho, não use generalização, mas extensão.
- Generalização de Ator
  - Precisa definir um ator que desempenha papel já desempenhado por outro ator em relação ao sistema, mas que também possui comportamento particular
- A legibilidade tem preferência sobre a formalização
  - Nunca use muitos relacionamentos de extensão, inclusão e generalização

Quando usar relacionamentos

- DFD X MCU
  - DFD:
    - Modelo funcional representa uma visão do comportamento interno do sistema, mesmo que em alto nível
    - Processos se comunicam. Trocam informações.
    - Identifica as funções do sistema
  - MCU:
    - Representa uma visão externa
    - Não existe troca de informações entre casos de uso
    - Identifica os objetivos do usuário

Identificação dos elementos do MCU

- Atores
  - Identificar quais as fontes de informação a ser processadas
  - Identificar os destinos das informações geradas
  - Se o sistema for uma empresa, identificar as áreas que serão afetadas
  - Perguntas a ser respondidas para identificação:
    - Quais órgãos, departamentos ou pessoas usarão o sistema?
    - Que equipamentos se comunicarão com o sistema?
    - Quem vai ser informado sobre os resultados do sistema?
    - Quem tem interesse em um determinado requisito?

Identificação dos elementos do MCU

- Casos de Usos Primários
  - Representam os objetivos dos atores
  - Perguntas a ser respondidas para a identificação:
    - Quais as necessidades e objetivos de cada ator em relação ao sistema?
    - Que informações o sistema deve produzir?
    - O sistema deve realizar alguma ação que ocorre regularmente no tempo?
    - Para cada requisito funcional, existe um (ou mais) caso(s) de uso para atendê-lo?

Identificação dos elementos do MCU

- Casos de Usos Primários que podem surgir:
  - Casos de uso opostos: desfazem o resultado.
    - Ex: Efetuar Pedido X Cancelar Pedido
  - Casos de uso que precedem outro caso de uso: pré-requisitos pra realização de um caso de uso
    - Ex: Realizar um pedido → Cadastro realizado
  - Casos de uso que sucedem outro caso de uso:
    - Ex: Realizar compra por internet -> Agendar entrega
  - Casos de uso temporais: Tarefas automáticas
    - Ex: Gerar folha de pagamento mensal automaticamente
  - Casos de uso relacionado a uma condição interna
    - Ex: Notificar o usuário que chagaram novos e-mails

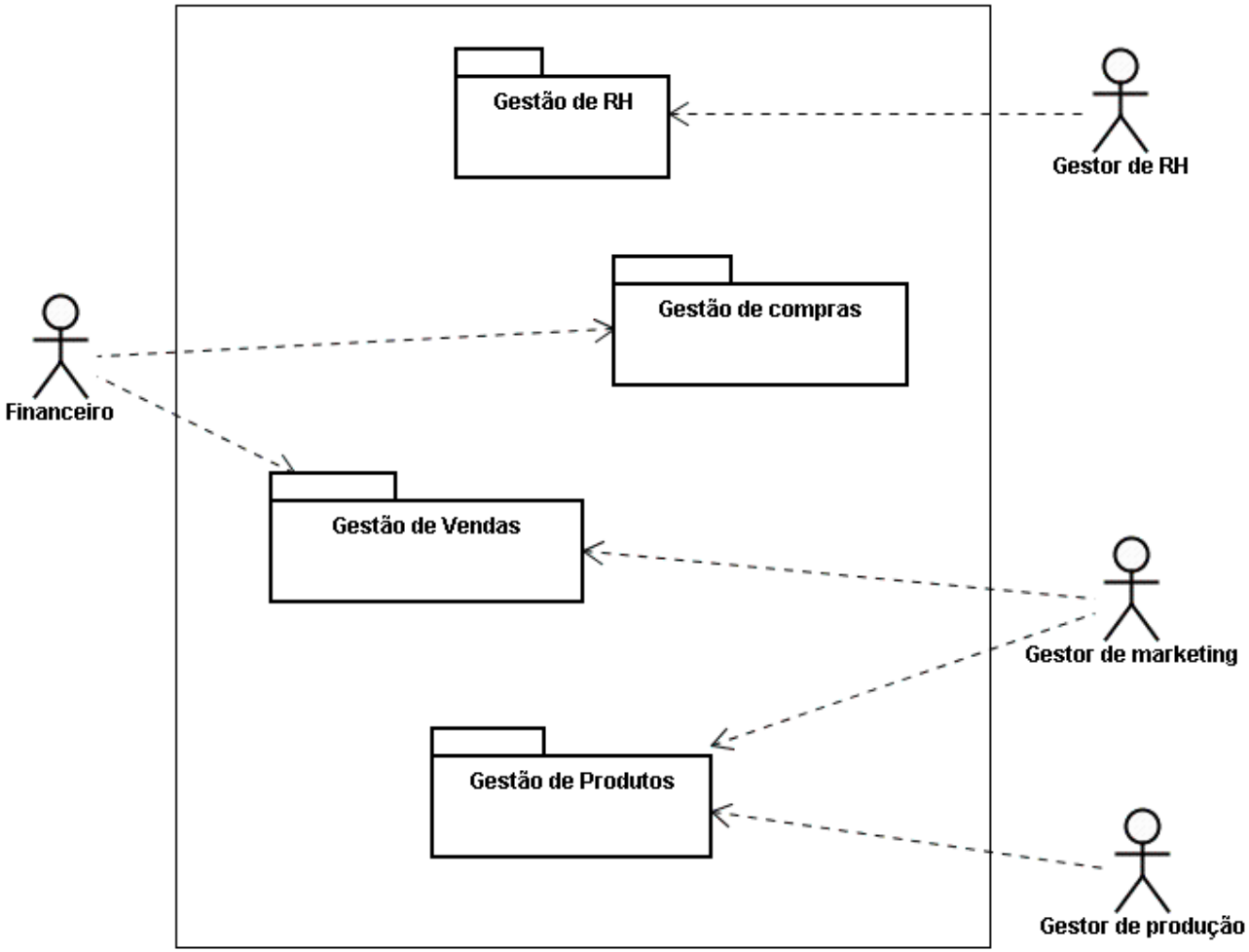
Identificação dos elementos do MCU

- Casos de Usos Secundários
  - Não traz benefícios diretos para os atores
  - Necessário para que o sistema funcione adequadamente
  - Deve ser explicitamente definido para evitar ambigüidades
  - Categorias:
    - Manutenção de cadastros
    - Manutenção de usuários e seus perfis
    - Manutenção de informações provenientes de outros sistemas
- Iniciar pelos MCU Primários - Objetivos

Construção do MCU

- Critérios para divisão de diagramas
  - Diagrama que exibe um caso de uso e seus relacionamentos
  - Diagrama que exibe todos os casos de uso para um ator
  - Diagrama que exibe todos os casos de uso que serão implementados em uma iteração
  - Diagrama que exibe todos os casos de uso de uma divisão da organização

Construção do MCU



Construção do MCU

- Documentação de Atores
  - Nome: Papel desempenhado pelo ator
  - Breve descrição: uma ou duas frases

Construção do MCU

- Documentação de Casos de Uso
  - Usar os itens de descrição que realmente sejam úteis e inteligíveis para o usuário.
  - Sugestão:
    - Nome: Mesmo nome do DCU; Cada caso de uso deve ter um nome único.
    - Identificador: Identifica os casos de uso em atividades que exijam referência cruzada ou rastreamento.
      - Ex: CSU01, CSU02
    - Importância: Categorias de importância (Riscos X Prioridades).
    - Sumário: Breve declaração do objetivo do ator ao usar o caso de uso.

Construção do MCU

- Documentação de Casos de Uso
  - Sugestão (cont.)
    - Ator Primário: Nome do ator – Um único ator
    - Atores secundários: Nome dos atores – Vários atores
    - Precondições: Hipóteses assumidas como verdadeiras para que o caso de uso inicie
    - Fluxo principal: Descrição de seqüência de passos que normalmente acontece. (Não usar jargões computacionais).
    - Fluxos alternativos: Descreve situações quando o ator resolve usar o caso de uso de forma diferente ou descrever escolhas exclusivas ente si. (não é obrigatório)
    - Fluxos de exceção: Descreve o que acontece quando algo inesperado ocorre (erro, uso inválido, cancelamento)

Construção do MCU

- Documentação de Casos de Uso
  - Sugestão (cont.)
    - Pós-condição: Estado que o sistema alcança após um caso de uso ter sido executado. Escrito no pretérito.
    - Regras de negócios: Políticas, condições ou restrições do domínio da organização.
    - Histórico: Autor, data, modificações no conteúdo do caso de uso.
    - Notas de implementação: Capturar idéias de implementação do caso de uso. Não é usada na validação.

Documentação suplementar ao MCU

- Modelo de casos de uso capturam apenas os requisitos funcionais do sistema
- Requisitos Não Funcionais, Regras de Negócios e Requisitos de interface são capturados nas especificações suplementares
  - Utiliza-se texto informal ou descrição estruturada
  - Utilizar um identificador. Ex:
    - RN01 para Regras de Negócios
    - RNF01 para Requisitos Não Funcionais
    - RI01 para Requisitos de Interface
  - Pode-se utilizar tabelas para a documentação.

MCU no processo Iterativo

- Divida os casos de uso em grupos
- Cada grupo é uma iteração
- A cada iteração um grupo de casos de uso é detalhado e desenvolvido
- Ordem de desenvolvimento:
  - Risco alto e Prioridade alta
  - Risco alto e Prioridade baixa
  - Risco baixo e Prioridade alta
  - Risco baixo e Prioridade baixa

MCU no processo Iterativo

- Procedimento utilizado no processo iterativo:
  - Concepção: Identifique atores e casos de uso.
  - Elaboração:
    - Desenhe os diagramas de casos de uso
    - Escreva os casos de uso em formato de alto nível e essencial
    - Ordene a lista de casos de uso de acordo com prioridades e riscos
    - Planejamento (Elaboração para construção)
      - Associe cada grupo de casos de uso a uma iteração da fase de construção
  - Construção (n-esima iteração)
    - Detalhe os casos de uso
    - Implemente estes casos de uso

EXEMPLO DE CASO DE USO

Descrição da situação

- Uma faculdade precisa de uma aplicação para controlar alguns processos acadêmicos, como inscrições em disciplinas, lançamento de notas, alocação de recursos a turmas, etc.
- Após a elicitação inicial dos requisitos, os analistas chegam a seguinte lista de requistos não funcionais:

RFs

- RF1. O sistema deve permitir que os alunos visualizem as notas obtida por semestre letivo
- RF2. O sistema deve permitir o lançamento das notas das disciplinas lecionadas em um período letivo e controlar os prazos e atrasos nestes lançamentos
- RF3. O sistema deve manter informações cadastrais sobre disciplinas no currículo escolar.
- RF4. O sistema deve permitir a abertura de turmas para uma disciplina assim como a definição de salas e laboratórios e horários e dias da semana em que haverá aulas.
- RF5. O sistema deve permitir que o aluno realize inscrição nas disciplinas do semestre.
- RF6. O sistema deve permitir o controle do andamento das inscrições dos alunos.
- RF7. O sistema deve permitir comunicação com o sistema de RH para coletar dados dos professores.
- RF8. O sistema deve se comunicar com o sistema de faturamento para informar inscrições do alunos.
- RF9. O sistema deve manter informações cadastrais sobre o alunos e o seu histórico escolar.

RFNs

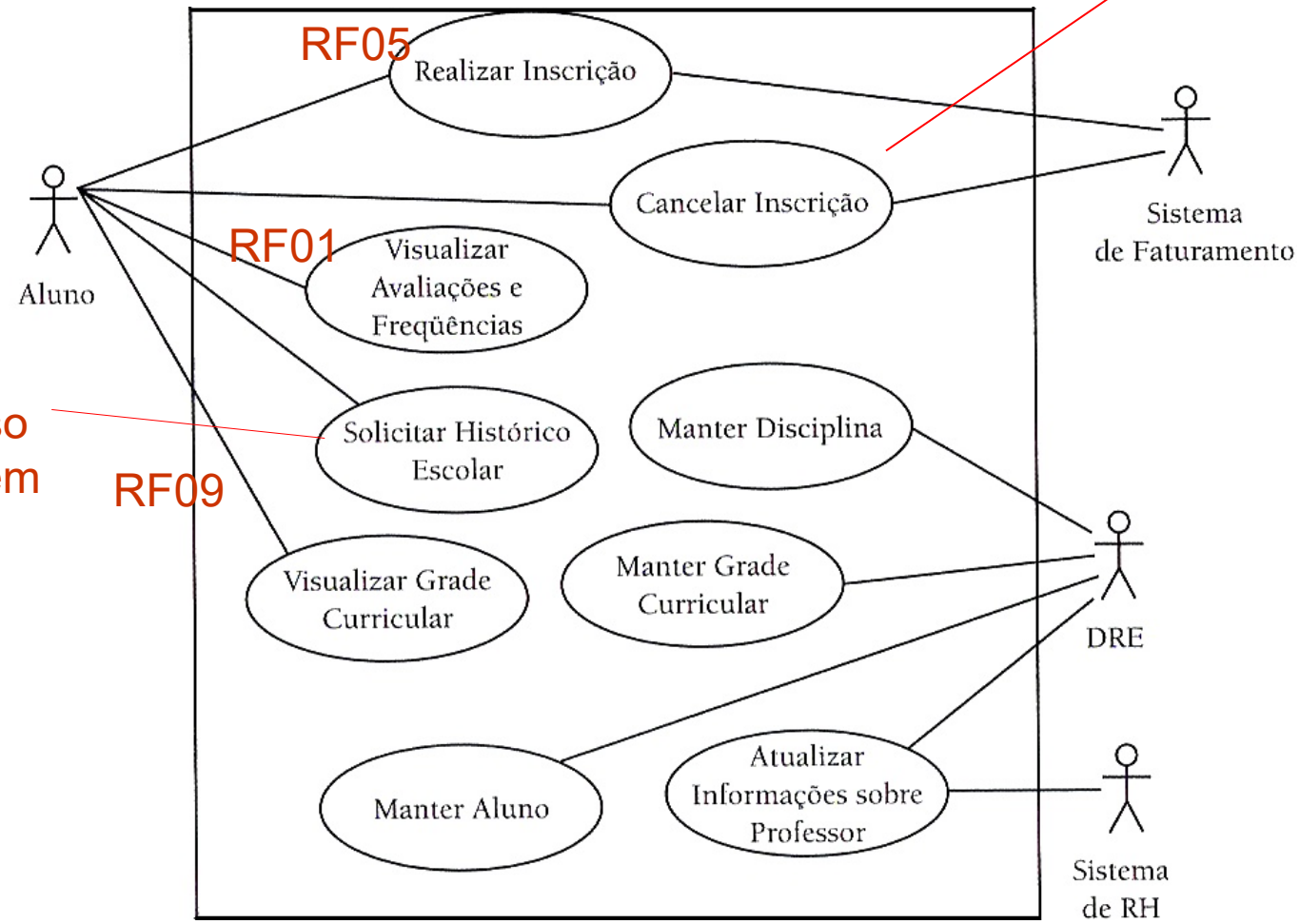
- RFN1. Quantidade máxima de inscrições em um período letivo
  - O aluno só pode se inscrever em 20 créditos por semestre
- RFN2. Quantidade de alunos por disciplinas
  - Em uma disciplina só podem ser matriculados 40 alunos no máximo
- RFN3. Habilitação pra lecionar disciplina
  - Um professor só pode lecionar uma disciplina para o qual esteja habilitado
- ...
- RFN6. Política de avaliação de alunos
  - A nota de um aluno em uma disciplina é obtida pela média aritmética de duas notas de avaliações no semestre e pela freqüência de aulas:
    - Se o aluno obtiver nota >= 7.0 será aprovado
    - Se o aluno obtiver nota >= 5.0 nota <= 7.0 deverá fazer avaliação final
    - Se o aluno obtiver nota < 5.0 será reprovado por média
    - Se um aluno tiver frequencia < 75% será reprovado por faltas

Documentação do MCU

- Atores
  - Aluno: Indivíduo que está matriculado da faculdade, que tem interesse em se inscrever em disciplinas do curso
  - Professor: .....*aqui a definição de professor....*
  - Coordenador: ....*aqui a definição de coordenador....*
  - Departamento de Registro Escolar: Departamento da faculdade interessado em manter informações sobre os alunos matriculados e sobre seu histórico.
  - Sistemas de RH: Sistema legado responsável por manter informações sobre os recursos humanos da escola, como os professores.
  - Sistema de faturamento: ...*aqui a definição de sistema de faturamento...*

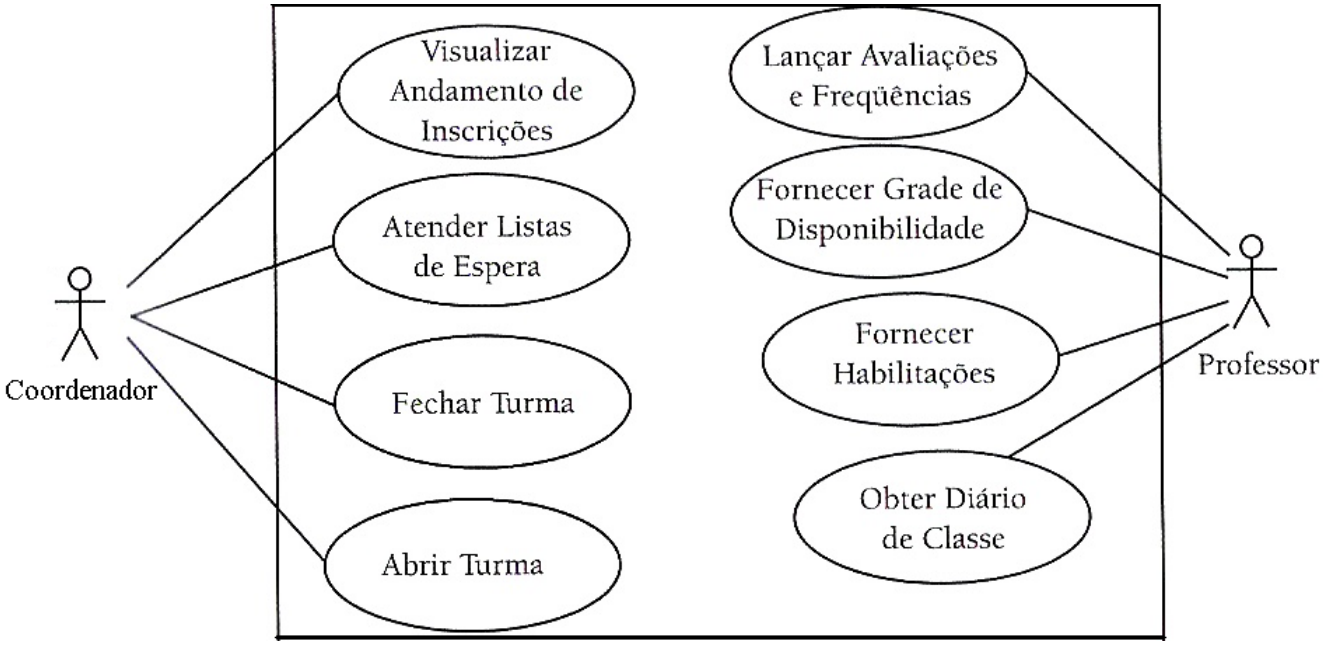
Diagrama de caso de uso

Casos de uso opostos



Casos de uso que precedem ou sucedem outro

Diagrama de caso de uso



Descrição do Caso de Uso no formato Essencial e Expandido

Realizar Inscrição (CSU01)

Sumário: Aluno usa o sistema para realizar inscrição em disciplinas

Ator Primário: Aluno

Ator Secundário: Sistema de faturamento

Precondições: O aluno está identificado pelo sistema

Fluxo Principal:

1. O aluno solicita a realização da inscrição
2. O sistema apresenta as disciplinas para as quais o aluno tem pré-requisitos (conforme a RN03), excetuando-se as que este já tenha cursado
3. O aluno define a lista de disciplinas que deseja cursar no próximo semestre letivo e as relaciona para inscrição
4. Para cada disciplina selecionada, o sistema designa o aluno para uma turma que apresente uma oferta para tal disciplina.
5. O sistema informa as turmas para as quais o aluno foi designado. Para cada turma o sistema informa o professor, horário, local da aula.
6. O aluno confere as informações fornecidas. Aqui é possível que o caso de uso retorne ao passo 3, conforme o aluno queira atualizar a lista de disciplinas
7. O sistema registra a inscrição do aluno e envia os dados para o sistema de faturamento e o caso de uso termina

Fluxo alternativo (4): Inclusão em lista de espera

a. Se não há oferta disponível para alguma disciplina selecionada pelo aluno (conforme RN02), o sistema reporta o fato ao aluno e fornece a possibilidade de inserir em uma lista de espera.

b. Se o aluno aceitar, o sistema o insere na lista de espera e apresenta a posição em que o aluno foi inserido na lista.
O caso de uso retorna ao passo 4

c. Se o aluno não aceitar, o caso de uso prossegue a partir do passo 4.

Fluxo de Exceção (4): Violação de RN01

a. Se o aluno atingiu a quantidade máxima de inscrições possíveis em um semestre letivo(conforme RN01), o sistema informa ao aluno a quantidade de disciplinas que ele pode selecionar e o caso de uso retorna ao passo 2.

Pós-condições: O aluno foi inscrito em uma das turmas de cada uma das disciplinas desejadas, ou foi adicionado a uma ou mais listas de espera.

Regra de negócios: RN01, RN02, RN03

Diagrama de Classes
e
Diagrama de Objetos

Introdução

- Externamente ao sistema, os usuários visualizam resultados de cálculos, relatórios produzidos, confirmações de requisições, etc.
- Internamento, o sistema orientado a objetos é composto por um conjunto de objetos que cooperam entre si.
- Cooperação:
  - Aspecto estrutural : Apresenta como o sistema está internamente estruturado.
  - Aspecto dinâmico: Apresenta as interações entre os objetos.
- O aspecto estrutural de um sistema é representado pelo *diagrama de classes*.

Introdução

- Desenvolvimento inclui:
  - Requisitos –> Análise –> Projeto -> Implementação
- O Modelo de classes (MC) evolui durante o desenvolvimento iterativo
- Estágios de abstração
  - Análise
    - Atenção sobre o que o sistema deve fazer
  - Especificação
  - Implementação

Introdução

- Classes de Análise (MCA)
  - Atenção sobre o que o sistema deve fazer
  - Não leva em consideração restrições associadas a tecnologia a ser utilizada na resolução de um problema
  - O MCU e o MCA são os dois principais modelos da fase de análise
- Classes de especificação(MCE)
  - É um detalhamento do modelo de classes de análise
  - É também conhecido como Modelo de classes de projeto
  - A atenção é sobre como o sistema deve funcionar
  - Novas classes começam a aparecer
  - Ex: Analogia com uma casa: Classes de análise são salas, quartos, banheiro, porta. Classes de projeto são encanamento, parte elétrica, encaixe das portas.
  - Parte visível X parte menos evidente do modelo

Introdução

- Classes de implementação (MCI)
  - É a implementação das classes em uma linguagem de programação (C, Java, C#, etc.)
  - Construído na implementação.
  - É o próprio código fonte como um modelo.

- O nível de abstração diminui a cada estágio

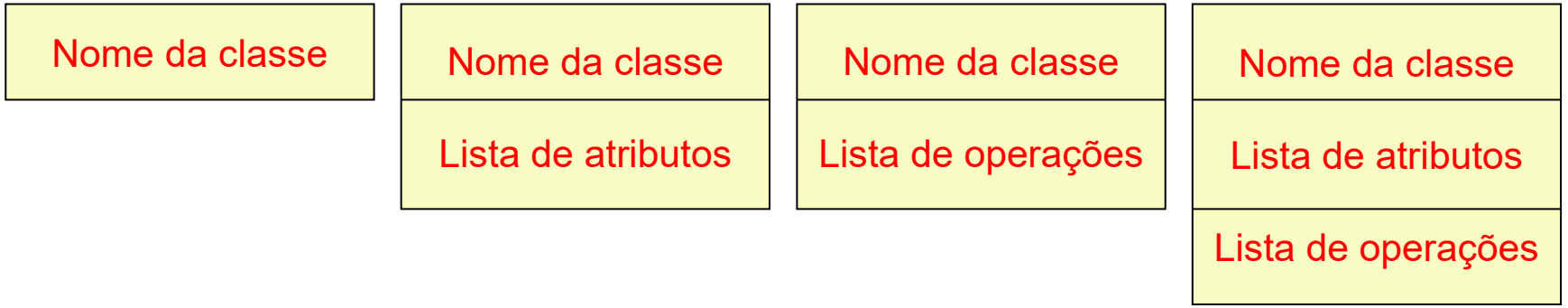
Análise Projeto Implementação

Introdução

- Notação para o MC (recomendações)
  - Identificadores: Espaços em branco e preposições são removidas
  - Nomes de classes e relacionamentos:
    - Palavras começando por letra maiúscula
    - Ex: Cliente, Pedido, ItemPedido
  - Nomes de atributos e operações
    - Palavras começando com letra minúscula
    - Duas (ou mais) palavras separadas por letra maiúscula
    - Siglas inalteradas
    - Ex: quantidade, precoUnitario, CPF

Diagrama de Classes

- Classes
  - Representada por uma caixa com 3 compartimentos no máximo:



  - O grau de abstração determina quando usar uma notação

Diagrama de Classes

- Classes
  - Exemplo:

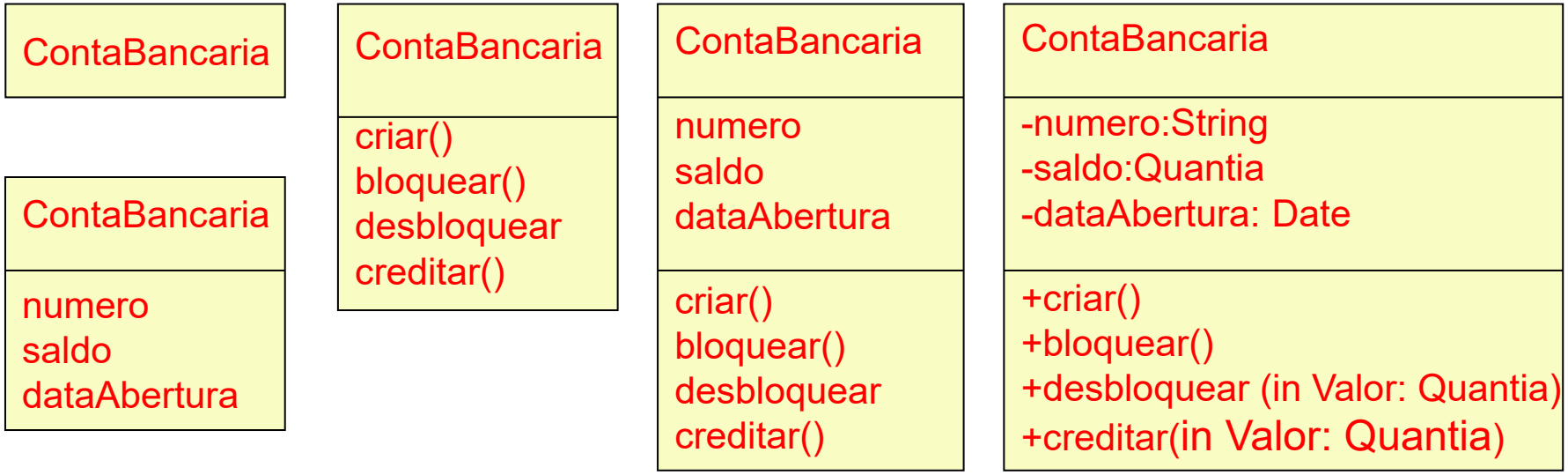


Diagrama de Classes

- Classes
  - Os atributos correspondem à descrição dos dados armazenados pelos objetos de uma classe.
    - Cada objeto tem os seus próprios valores
  - As operações correspondem a descrição das ações que os objetos de uma classe sabem realizar.
    - Objetos de uma classe compartilham as mesmas operações

Diagrama de Classes

- Associações
  - Objetos podem se relacionar com outros, possibilitando a troca de mensagens entre eles.
  - O relacionamento entre objetos são representados no diagrama de classes por uma *Associação*.
  - Uma Associação é representada por uma linha ligando as classes.
  - Ex: Um cliente compra produtos

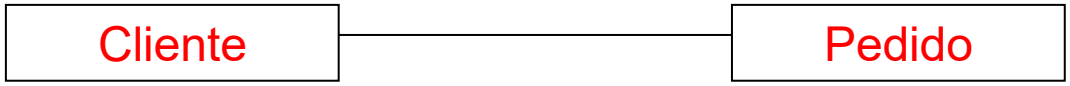


Diagrama de Classes

- Relacionamentos
  - Associação
  - Agregação e Composição
  - Generalização e Especialização

Diagrama de Classes

- Associações
  - Características das associações:
    - Multiplicidade
    - Nome
    - Direção de leitura
    - Papéis
    - Tipo de participação
    - Conectividade

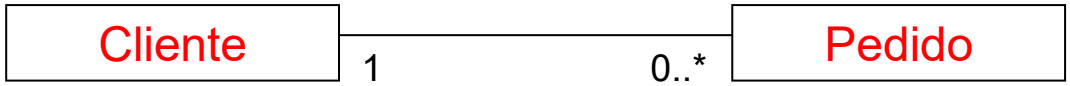
Diagrama de Classes

- Multiplicidade:
  - Representa as informações dos limites inferior e superior da quantidade de objetos aos quais outro objeto pode estar associado.

Nome	**Simbologia**
Apenas Um	**1** *(ou* **1..1***)*
Zero ou Muitos	**0..*** *(ou **)*
Um ou Muitos	**1..***
Zero ou Um	**0..1**
Intervalo específico	**1i..1s**

Diagrama de Classes

- Multiplicidade:

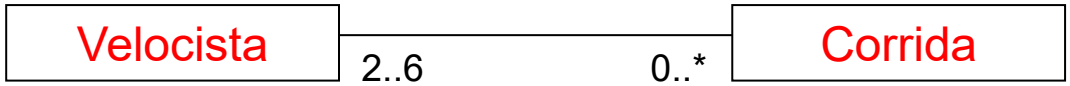


  - Pode haver algum objeto da classe Cliente que está associado a *vários objetos* da classe Pedido (representado por * do 0..*)
  - Pode haver algum objeto da classe Cliente que *NÃO está associado* a classe Pedido (representado por 0 do 0..*)
  - Objetos da classe pedido está associado a UM e somente um objeto da classe Cliente

Cliente José tem os pedidos 1, 2 e 3
Cliente Ana tem os pedidos 4 e 5
Cliente Maria não tem pedidos
O pedido 1 está associado somente a José

Diagrama de Classes

- Multiplicidade:



  - O velocista pode participar de várias corridas (*) ou não participar de nenhuma (0)
  - Em uma corrida deve haver no mínimo DOIS velocistas e no máximo SEIS velocistas

  - Uma lista de intervalos também pode ser especificada na multiplicidade de uma associação. Ex: [1,3,5..9,11]
  - Os valores especificados em uma multiplicidade devem sempre estar em ordem crescente.

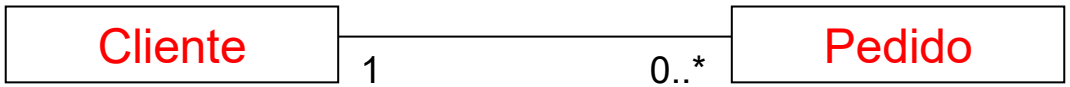
Diagrama de Classes

- Multiplicidade:
  - As associações podem ser agrupadas em 3 tipos. Estes tipos são denominados *Conectividade*:

Conectividade	**Multiplicidade de um extremo**	**Multiplicidade do outro extremo**
Um para Um	**0..1** ou **1**	**0..1** ou **1**
Um para Muitos	**0..1** ou **1**	* ou **1..*** ou **0..***
Muitos para Muitos	* ou **1..*** ou **0..***	* ou **1..*** ou **0..***

Diagrama de Classes

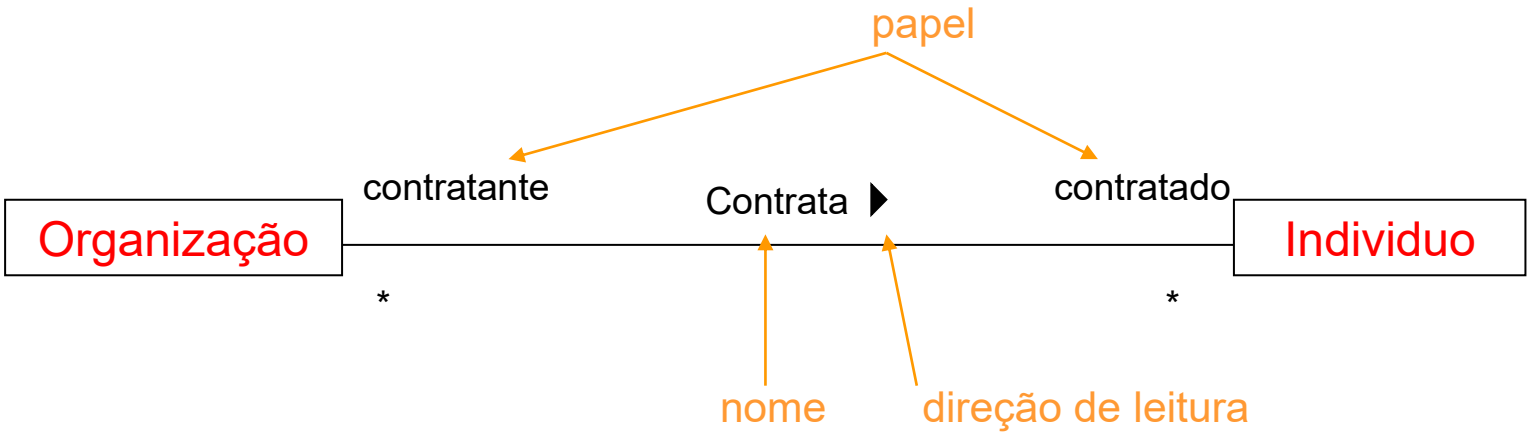
- Participações
  - Necessidade ou não da existência dessa associação entre objetos.
  - Obrigatória:
    - Se o valor mínimo da multiplicidade é igual a **Um**
  - Opcional
    - Se o valor mínimo puder ser **Zero**



Para objetos da classe pedido a participação é **obrigatória**: Um objeto da classe Pedido só existe se estiver associado a classe Cliente.

Diagrama de Classes

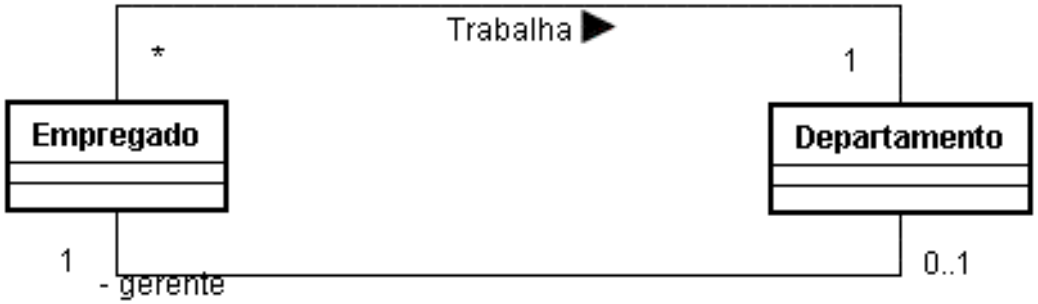
- Nome da associação, direção de leitura e papéis
  - Servem para esclarecer melhor o significado de uma associação
  - Só usar quando o significado de uma associação não for clara. Evitar usar em associações claras ou óbvias.



  - Uma organização (faz o papel de contratante) contrata indivíduos (faz o papel de contratado)

Diagrama de Classes

- Nome da associação, direção de leitura e papéis
  - Podemos representar mais de uma associação entre objetos



  - Uma organização precisa saber quem são os empregados e quem é o gerente

Diagrama de Classes

- Classes Associativas
  - Classes ligadas a associações em vez de estar ligada a outras classes.
  - Necessário quando se quer manter informações sobre a associação de duas ou mais classes.
  - Pode estar ligada associação de qualquer conectividade.
  - Pode ser substituída por uma classe com associação para as outras duas classes.

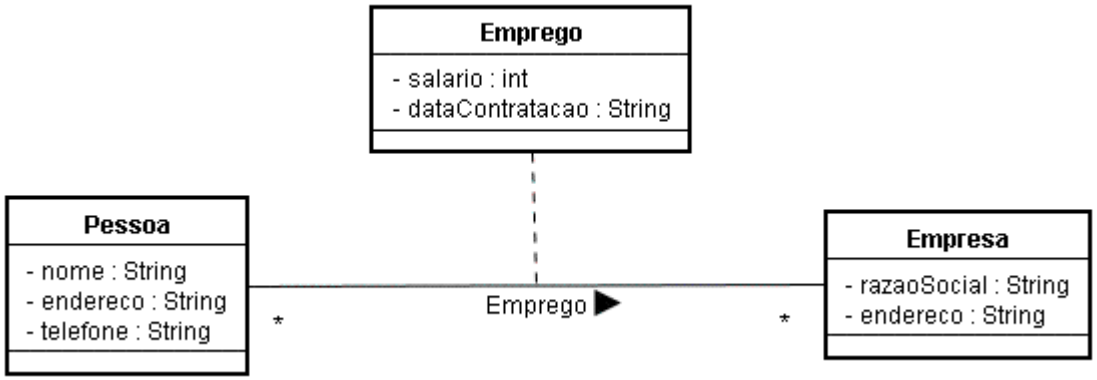


Diagrama de Classes

- Associações reflexivas (auto-associação)
  - Associa objetos da mesma classe
  - Cada objeto tem um papel distinto na associação
  - O uso de papeis é importante neste caso

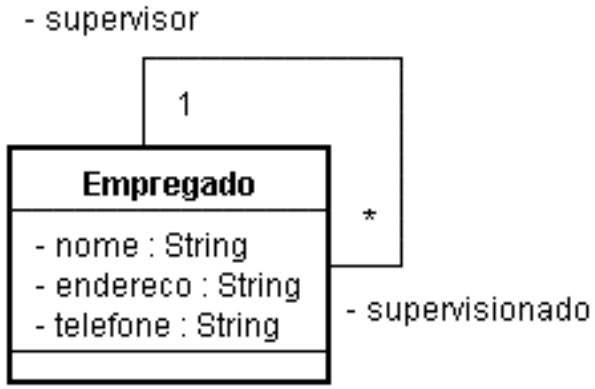
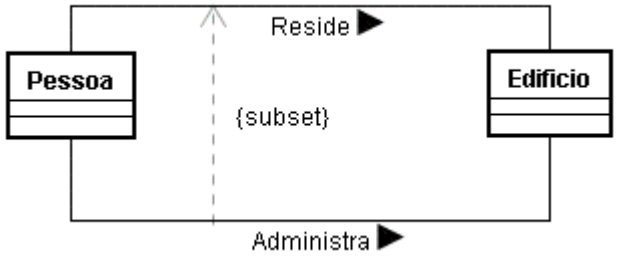


Diagrama de Classes

- Restrições sobre as associações



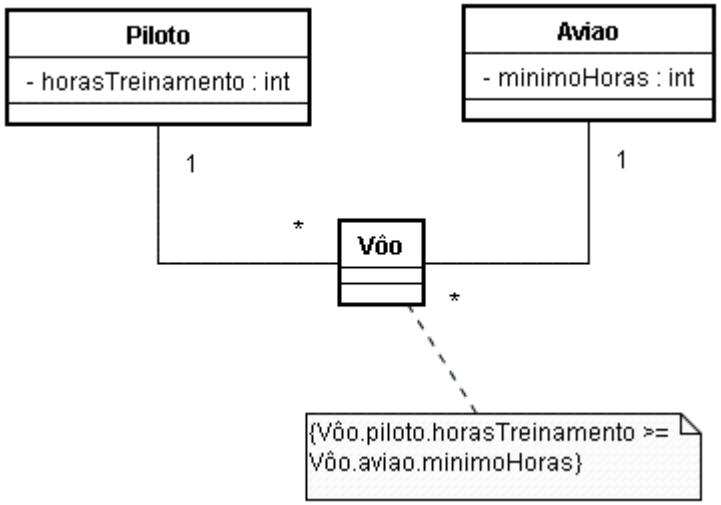
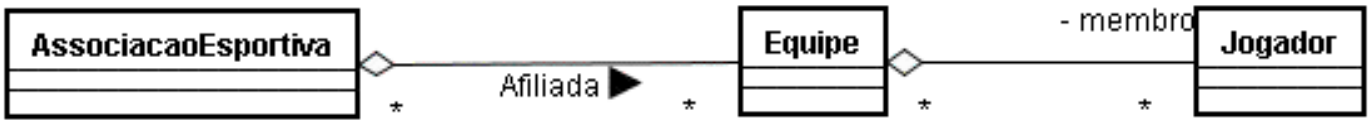


Diagrama de Classes

- Agregações e Composições
  - Representa uma relação *todo-parte*
  - Uma relação *todo-parte* significa que um objeto está contido em outro. Ou um objeto contém outro.
  - Características:
    - São assimétricas: Se A é parte de B, B não pode ser parte de A
    - Propagam comportamentos: O comportamento do todo se aplica as partes.
    - As partes são normalmente criadas e destruídas pelo todo. Isto é no Todo são definidas as operações de *Adicionar* e *Remover* as partes.
  - Tipos de relacionamentos todo-parte:
    - Agregação
    - Composição

Diagrama de Classes

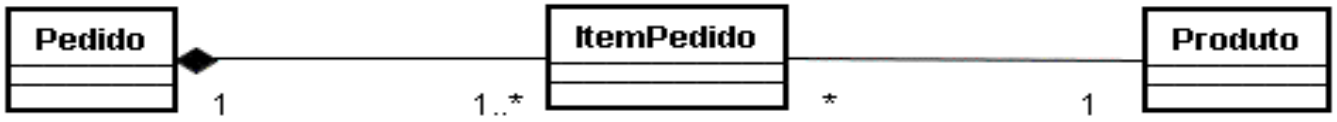
- Agregações
  - Notação:



  - Uma associação é formada por diversas equipes. Cada Equipe é formada por diversos Jogadores.

Diagrama de Classes

- Composições
  - Notação:



  - Um pedido inclui vários itens. Cada item diz respeito a um produto.

Diagrama de Classes

- Agregações x Composições
  - As diferenças não são muito bem definidas
  - Diferenças mais marcante:
    - Na agregação, a destruição do objeto Todo não implica na destruição do objeto Parte. Na composição a destruição do Todo implica na destruição das partes.
      - *Ex: Se uma equipe deixar de existir o jogador ainda pode continuar a existir.*
    - Na composição, os objetos parte pertencem a um único todo. Por outro lado na agregação pode ser que um objeto parte participe como componente de vários outros objetos.
      - *Ex: Um item de produto só pode pertencer a um único pedido.*

Diagrama de Classes

- Generalizações e Especializações
  - Usa-se vários termos: SuperClasse e SubClasse, Supertipo e SubTipo, Classe Base e Classe Herdeira.
  - Representa o conceito de Herança.
  - Não somente atributos e operações são herdados, mas as associações também.
  - Notação:

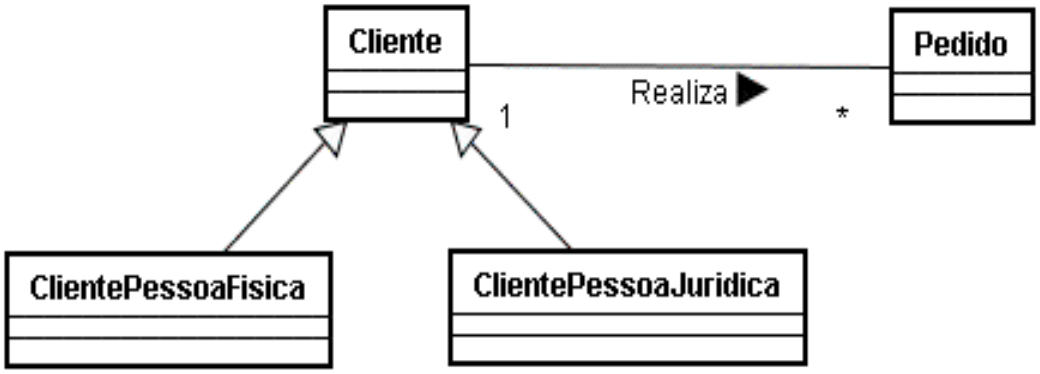


Diagrama de Classes

- Generalizações e Especializações
  - Classes Abstratas:
    - É usada para organizar a hierarquia de classes.
    - Não geram objetos diretamente
    - Muito utilizada nas Classes de Projetos
    - Notação: O nome é definido em *Itálico*

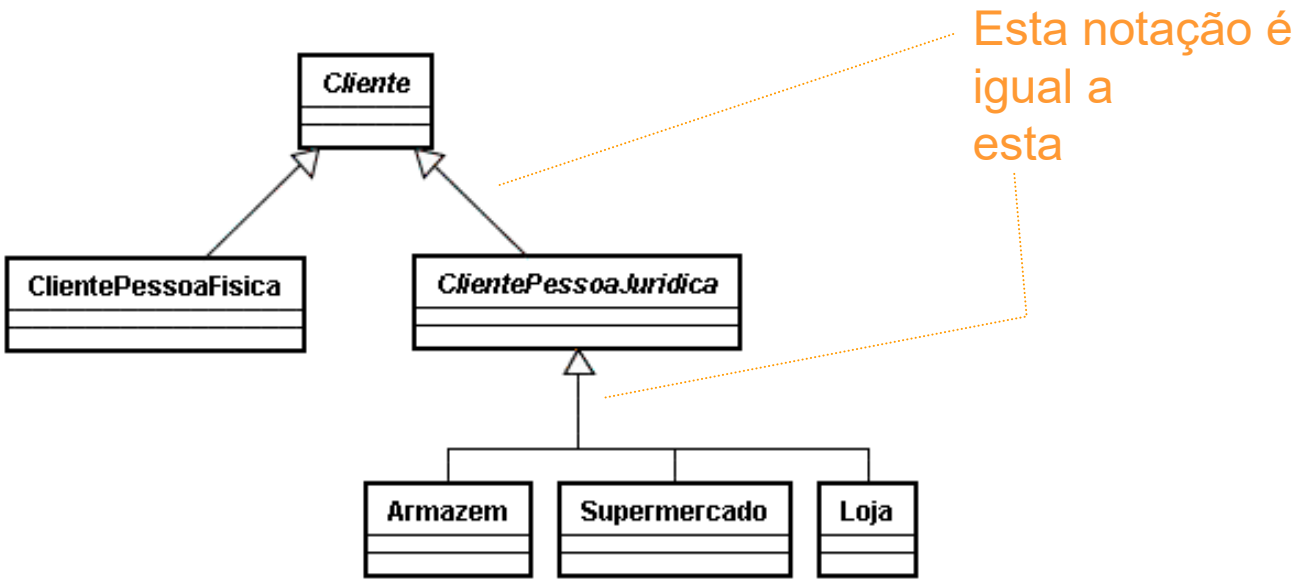


Diagrama de Classes

- Herança X Associação
  - O relacionamento de herança acontece entre classes
  - Os relacionamentos de Associação, Agregação / Composição e Associação ocorre entre as instâncias das classes (os objetos).
- Propriedades de relacionamentos de herança
  - Transitividade
    - Se A é uma generalização de B e B é uma generalização de C, então C herda características de B e A.
  - Assimetria
    - Se A é uma generalização de B, B não pode ser uma generalização de A
- Deve-se evitar hierarquias muito profundas, com mais de 3 níveis, pois dificulta a leitura.

Diagrama de Classes

- Restrições de Generalização e Especialização:
  - Sobreposta: Podem ser criadas subclasses que herdem de mais de uma subclasse
    - Ex: Atleta – (Nadador e Corredor)
  - Disjunta: As subclasses só podem herdar de uma subclasse
    - Ex: Figura geométrica – (Elipse, Quadrado, Circulo)
  - Completa: Todas as subclasses possíveis foram enumeradas.
    - Ex: Indivíduo – (Homem e Mulher)
  - Incompleta: Nem todas as subclasses foram enumeradas na hierarquia
    - Ex: Figura geométrica – (Elipse, Quadrado, Circulo)

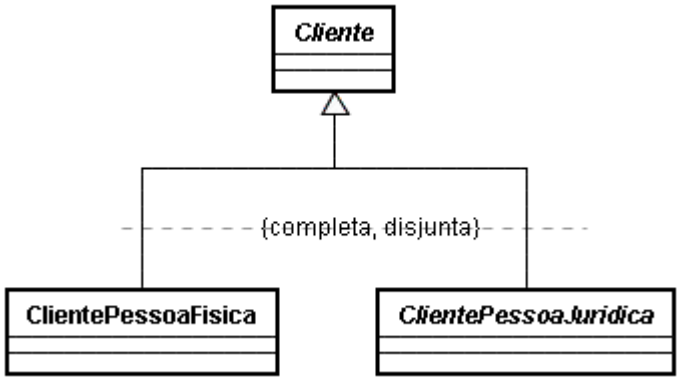


Diagrama de Objetos

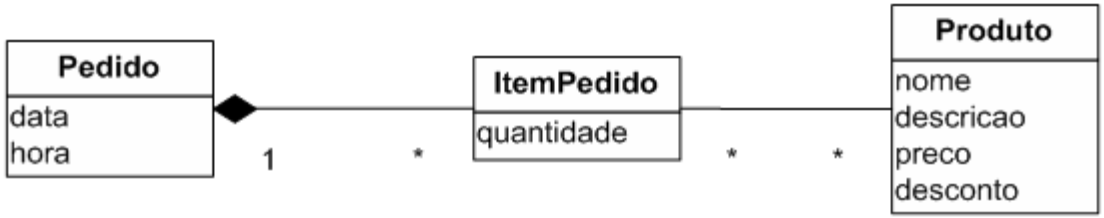
- São instâncias dos diagramas de classes, assim como os objetos são instâncias das classes.
- São estruturas estáticas
- Notação: Definido como “Nome do objeto” + “: (dois pontos)” + “Nome da classe”

O nome da objeto é opcional

Formato	Exemplo
:NomeClasse	:Pedido
nomeObjeto: NomeClasse	umPedido: Pedido

Diagrama de Objetos

- Exemplo: Diagrama de Classes



- Diagrama de Objeto

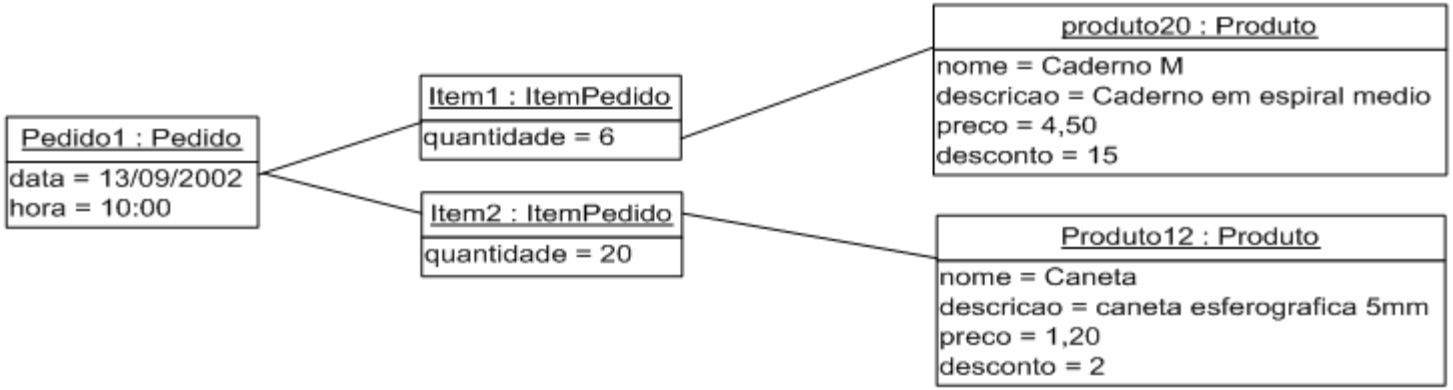
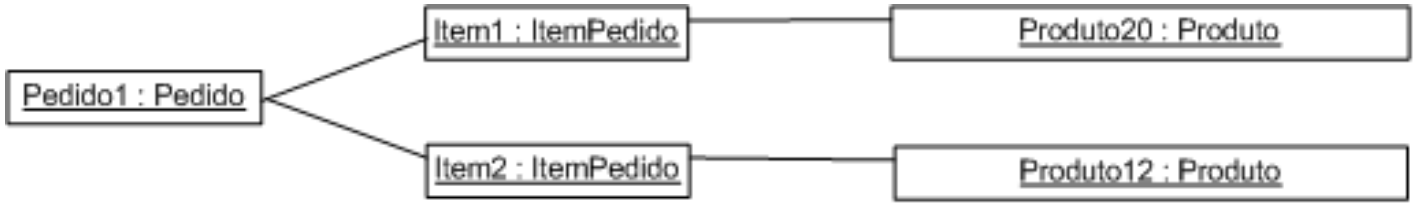


Diagrama de Objetos

- Exemplo 2: Diagrama de objetos



- A utilidade prática dos diagramas de objetos é ilustrar a formação de relacionamentos complexos

Técnicas para identificação de Classes

- Uma das tarefas mais difíceis é a identificação de classes necessárias e suficientes para compor um sistema.
- Identificar as classes significa “saber quais são os objetos que irão compor o sistema”
- Atividades da identificação:
  - Definir classes candidatas
  - Eliminar as classes desnecessárias

Técnicas para identificação de Classes

- Técnicas
  - Análise textual de Abbott
  - Análise dos Casos de Uso
  - Identificação dirigida a responsabilidades
  - Padrões de análise

Técnicas para identificação de Classes

- Análise textual de *Abbott*
  - Utiliza-se diversas fontes de informação: Documento de Requisitos, Modelo de Negócios, Glossários, etc.
  - Destacam-se os termos como sujeito, substantivos e verbo
  - Elimina-se os sinônimos e classifica os termos:

Parte do Texto	**Componente**	**Exemplo**
Nome próprio	Objeto	Maria
Nome simples	Classe	aluno
Verbos de Ação	Operação	registrar
Verbo Ser	Herança	é um
Verbo Ter	Todo-Parte	tem um

Técnicas para identificação de Classes

- Vantagem
  - Simplicidade
- Desvantagem
  - Resultado depende da completude do documento fonte
  - Pode-se gerar classes candidatas que nunca serão classes
  - A linguagem natural é imprecisa e classes importantes podem não ser identificadas.

Técnicas para identificação de Classes

- Análise dos Casos de Uso
  - O modelador identifica as classes necessárias para produzir o comportamento que está documentado na descrição do caso de uso
  - Justificativa: uma classe só deve existe se ela participar do comportamento externo visível do sistema.

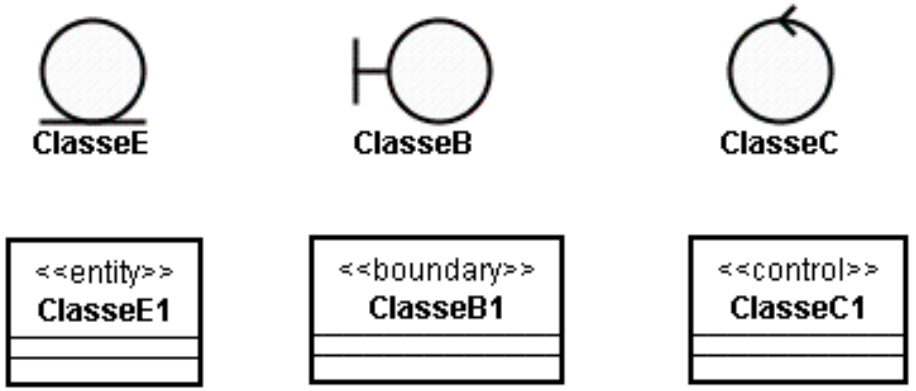
Técnicas para identificação de Classes

- Análise dos Casos de Uso
  - Passo a passo:
    - Suplemente as descrições dos casos de uso
    - Para cada caso de uso:
      - Identifique as classes a partir do comportamento(*)
      - Distribua o comportamento do caso de uso pelas classes identificadas
    - Para cada classe de análise resultante:
      - Descreva suas responsabilidades
      - Descreva atributos e associações
    - Unifique as classes de análise em um ou mais Diagrama de Classes

  - (*) As classes são identificadas pelo uso da categorização BCE

Técnicas para identificação de Classes

- Análise dos Casos de Uso
  - Categorização BCE: Os objetos são divididos em 3 categorias:
    - Objetos de Fronteira (Boundary)
    - Objetos de Controle (Control)
    - Objetos de Entidade (Entity)
  - Notação UML



Técnicas para identificação de Classes

- Análise dos Casos de Uso
  - Objetos de Fronteira

    - Permite ao sistema interagir com seu ambiente
    - Tipos principais: Interface com usuário, Interface com sistemas externos, Comunicação com dispositivos
    - Responsabilidades (comunicação):
      - Notificar aos demais objetos de eventos gerados pelo ambiente
      - Notificar os atores sobre o resultado de interações entre os objetos
    - Os nomes devem lembrar qual o canal de comunicação com o mundo externo
      - Ex: FormularioInscricao, LeitoraCartao, SistemaFaturamento
    - Durante a análise estas classes devem representar apenas os pontos de comunicação.

Técnicas para identificação de Classes

- Análise dos Casos de Uso
  - Objetos de Controle
    - É uma ponte de comunicação entre os objetos de fronteira e os objetos de entidade
    - Responsabilidades
      - Realizar monitorações para responder a eventos externos
      - Coordenar a realização de um caso de uso
      - Criar associações entre objetos Entidade
      - Manter valores acumulados ou derivados durante a realização de um caso de uso
      - Manter o estado da realização de um caso de uso
    - Os nomes devem lembrar o caso de uso que a classe é responsável por coordenar
      - Ex: GerenciadorContas, ControladorInscricao, ControladorReservas, MarcadorTempo, etc.

Técnicas para identificação de Classes

- Análise dos Casos de Uso
  - Objetos de Entidade

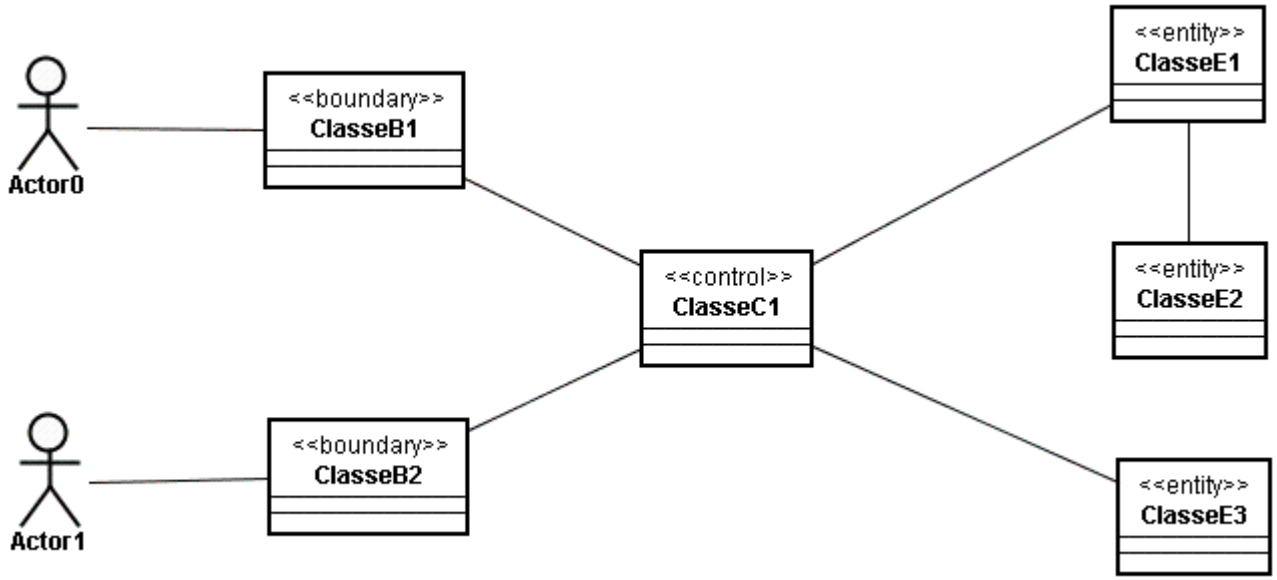
    - Servem como um repositório para as informações manipuladas pelo sistema
    - Dizem respeito a lógica do negócio
    - Os nomes lembram as informações do sistema:
      - Produto, Cliente, Pedido, ItemPedido, etc
    - Responsabilidades:
      - Informar valores de seus atributos aos requisitantes
      - Realizar cálculos ou impor restrições relativas as regras de negócios
      - Criar e destruir objetos partes

Técnicas para identificação de Classes

- Análise dos Casos de Uso
  - Para cada caso de uso pode-se utilizar as regras:
    - Adicionar um objeto de fronteira para cada ator do caso de uso: encapsula a comunicação nos objetos de fronteira
    - Adicionar um objeto de controle para cada caso de uso: Isto por que um caso de uso é responsável por um negócio específico.

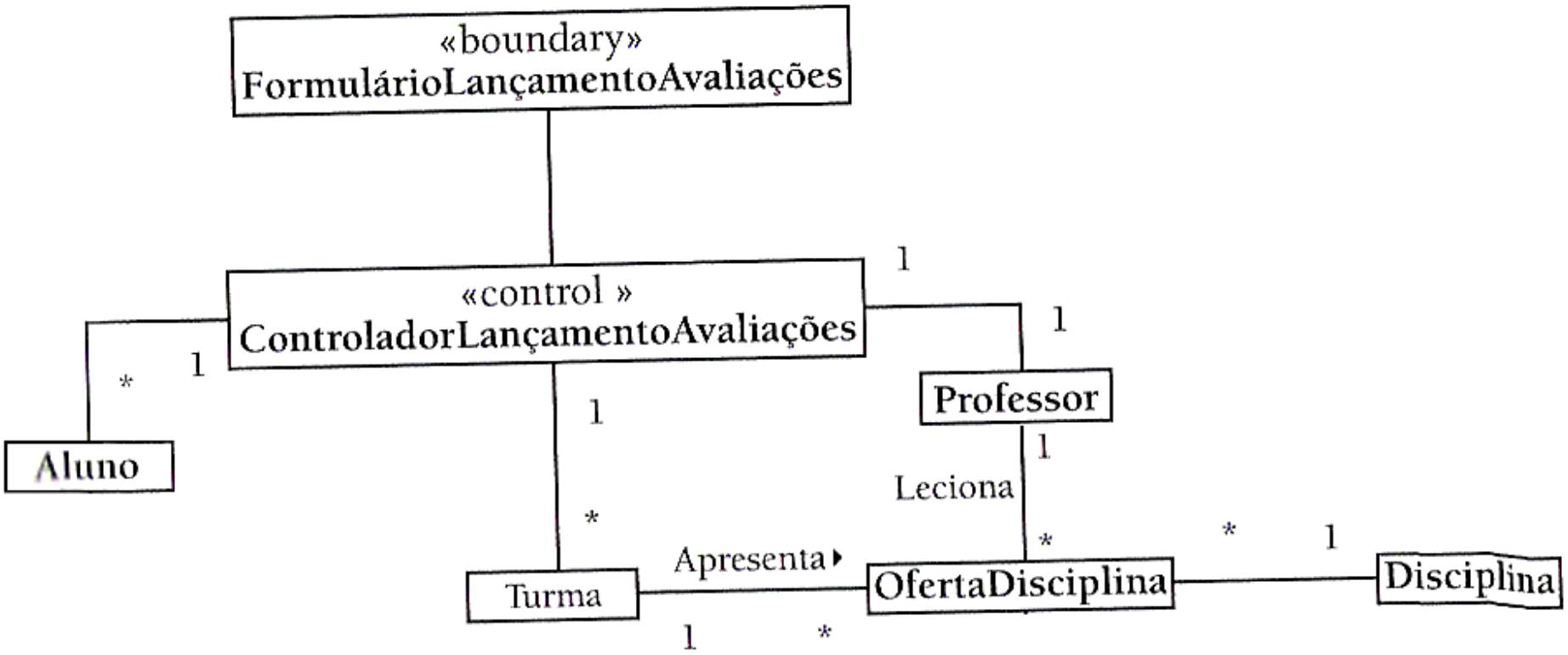
Técnicas para identificação de Classes

- Análise dos Casos de Uso



Técnicas para identificação de Classes

- Análise dos Casos de Uso
  - Ex. de um DCA para caso de uso LancarAvaliacao



Técnicas para identificação de Classes

- Técnica Identificação dirigida a responsabilidades
  - Enfatiza o encapsulamento da estrutura e do comportamento dos objetos
  - O comportamento do objeto é definido de tal forma que ele possa cumprir com suas responsabilidades.
  - Uma responsabilidade é uma obrigação que o objeto tem para com o sistema no qual está inserido.
  - Um objeto pode ter responsabilidades que não pode cumprir sozinho, então ele necessita da colaboração de outros objetos
  - Esta técnica é chamada Modelagem CRC (Classes, Responsabilidades e Colaboradores).

Técnicas para identificação de Classes

- Técnica Identificação dirigida a responsabilidades
  - Modelagem CRC
    - Analisa-se o cenário de um caso de uso
    - Já inicia com um conjunto de classes candidatas
    - Identifica-se as responsabilidades de cada classe
    - Identifica-se os colaboradores
    - O processo se dá em reuniões chamadas *sessões CRC*, cuja saída final é o preenchimento de cartões CRC.

Nome da Classe	
1ª responsabilidade	Colaborador
2ª Responsabilidade	Colaborador
...	

Técnicas para identificação de Classes

- Técnica Identificação dirigida a responsabilidades
  - Modelagem CRC
    - Ex:

ContaBancaria	
1 - Conhecer o seu cliente	Cliente
2 – Conhecer o seu número	Transação
3 - Conhecer o seu saldo	
4 - Manter um histórico de transações	
5 – Aceitar saques e depósitos	

DIAGRAMAS DE INTERAÇÃO

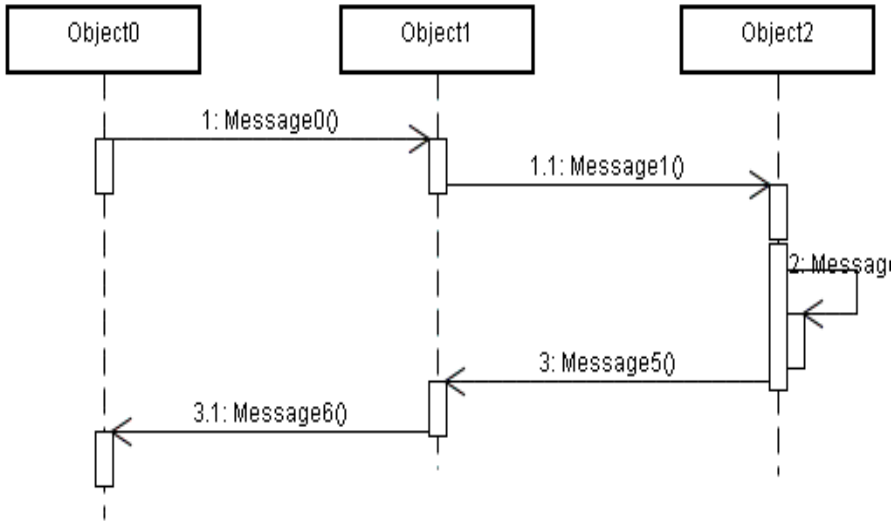
Introdução

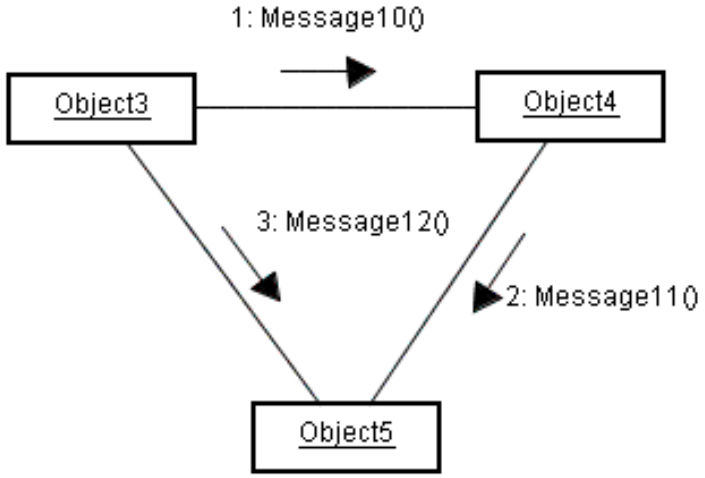
- Portanto: Modelos de casos de uso e classes são representações incompletas do sistema
- O modelo de interação permite a representação do comportamento dinâmico de um SSOO.
- Com o modelo de interação:
  - as classes, responsabilidades e colaboradores da técnica CRC podem ser validadas.
  - O modelo de classe pode ser refinado pois definimos as operações de cada classe.

Elementos da modelagem de interação

- A interação entre objetos para dar suporte a funcionalidade de um caso de uso denomina-se realização de casos de uso.
- Diagramas de interação:
  - Diagrama de Seqüência:
    - Ênfase na troca de mensagens entre objetos na ordem temporal
  - Diagrama de Comunicação (ou diagrama de colaboração)
    - Ênfase nos relacionamentos existentes entre os objetos
  - Diagrama de visão geral da Interação
    - Apresenta uma visão geral de diversas interações entre objetos
- Estes dois diagramas são equivalentes entre si, mas o Diagrama de seqüência é mais popular.
- O conjunto de todos os diagramas de interação contituem o modelo de interações.

Elementos da modelagem de interação





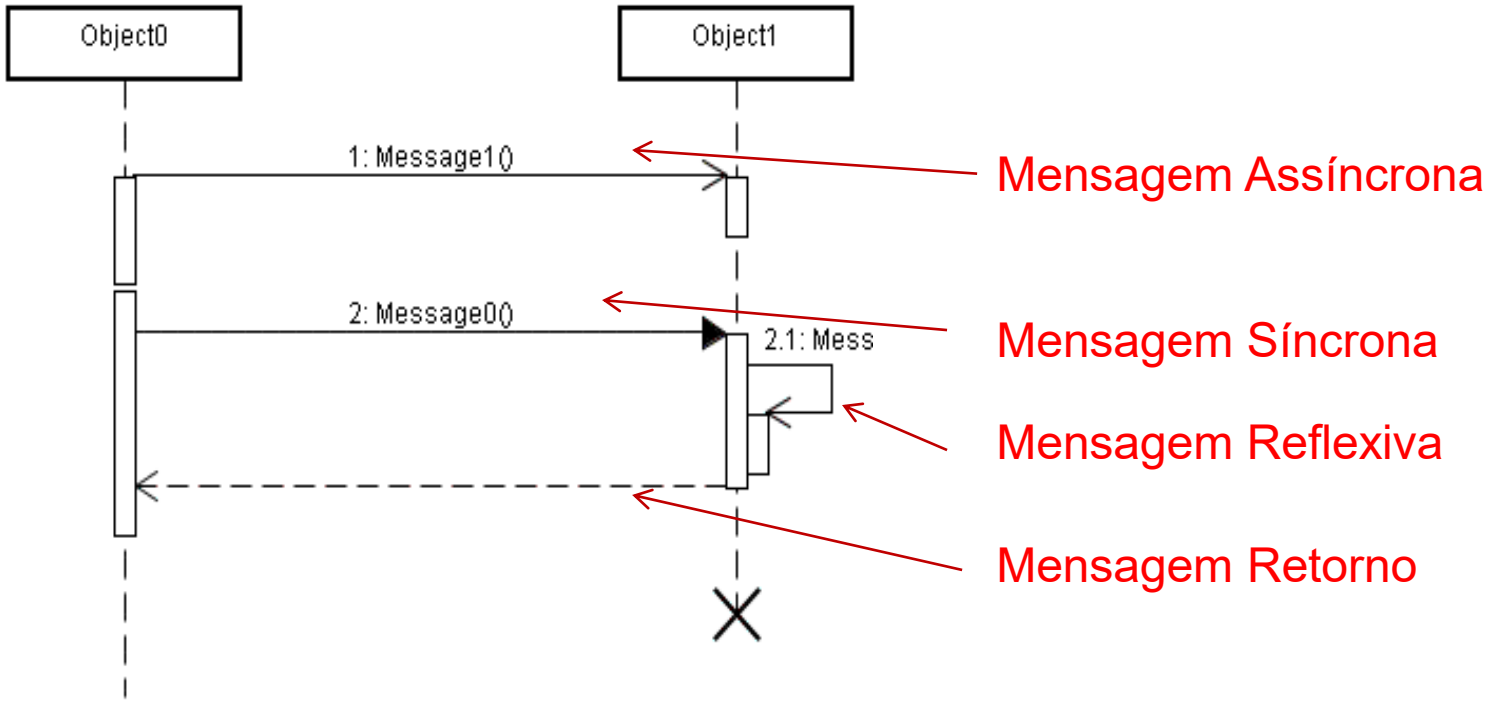
Elementos da modelagem de interação

- Mensagens
  - É o elemento mais importante das interações
  - Uma mensagem é uma solicitação de execução de uma operação em outros objeto
  - A mensagem deve ter informação suficiente para que o receptor execute a operação requisitada
  - Na UML, a sintaxe para representar mensagens é igual a todos os diagramas de interação

Elementos da modelagem de interação

- Mensagens
  - Mensagem Síncrona:
    - O objeto remetente fica aguardando que o receptor processe a mensagem antes de continuar o seu processamento.
  - Mensagem Assíncrona
    - O objeto remetente não espera a resposta para prosseguir com seu processamento
  - Mensagem de Sinal
    - É usada apenas para enviar um sinal
  - Mensagem de retorno
    - É usada para especificar o retorno de uma mensagem enviada anteriormente
- Mensagens reflexivas
  - Quando o objeto envia uma mensagem para si proprio

- Mensagens – Sintaxe UML



- Mensagens – Sintaxe UML
  - Nos diagramas de interação da fase de *análise* -> Usa apenas o nome da mensagem

  - Nos diagramas de interação da fase de *projeto* usa a assinatura completa da mensagem

- Mensagens – Sintaxe UML
  - expressao-sequencia: Indica a ordem das mensagens no diagrama:
    - 1,2,3 / 1.1, 1.2 / 1.1a, 1.1b
  - Controle: Indica se a mensagem tem alguma *condição* para ser enviada ou a *quantidade de vezes* que a mensagem deve ser enviada
    - Clausula condição (ou guarda)
      - [Senha é válida]: abrirJanelaPricipal()
      - [a>b]: trocar (a,b)
    - Clausula interação
      - *[Para cada f em F]: desenhar()
      - *[i:= 1..10]: mensagem()

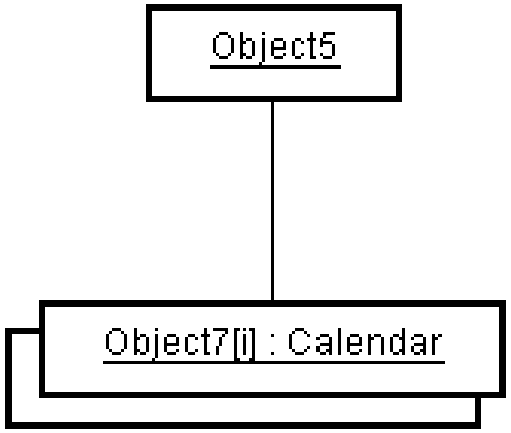
- Mensagens – Sintaxe UML
  - **v:=** : Usado para armazenar um valor de retorno que será usado em uma mensagem posterior.
    - X:= selecionar (e)
  - **nome** : é o nome da operação na classe receptora
  - **Argumentos:** Argumentos da operação na classe receptora

- Exemplos
  - 1: adicionarItem(item)
  - 3 [a>b]: trocar(a,b)
  - 2*: desenhar()
  - 1.2.1: x := selecionar(e)

- Atores
  - Atores podem participar do diagrama de interação
  - Mesma notação dos casos de uso
- Objetos
  - Mesma representação que no diagrama de objetos
- Classes
  - Na maioria das interações somente objetos sã representados nos DI
  - Pode-se usar classes para representar mensagens que disparam uma operação estática
  - A representação da UML é a mesma de classe de análise

Elementos da modelagem de interação

- Coleções ou multiobjetos: Representar coleções de objetos, tais como ItemPedido

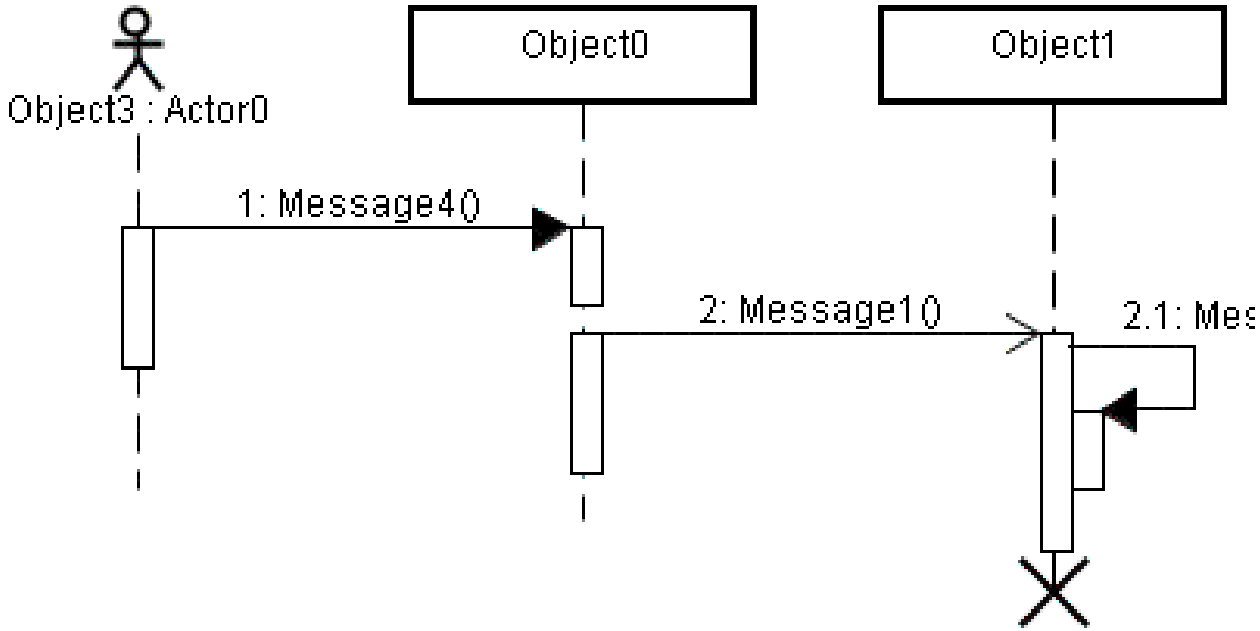


Diagramas de Seqüências

- Apresenta as interações em uma ordem temporal
- Linha de vida: Notação gráfica para representar a disposição dos objetos e suas interações
- Contém:
  - Cabeça: Ator, Classe, Objeto
  - Cauda: linha tracejada
- Ordem de disposição dos elementos:
  - Ator, Objeto de fronteira, Objeto de controle, Objeto de entidade
- Mensagens:
  - Assincronas, Sincronas, Retorno, Criação e Destruição, reflexivas
- A passagem no tempo ‘’e verificada observando-se a direção vertical no sentido de cima para baixo

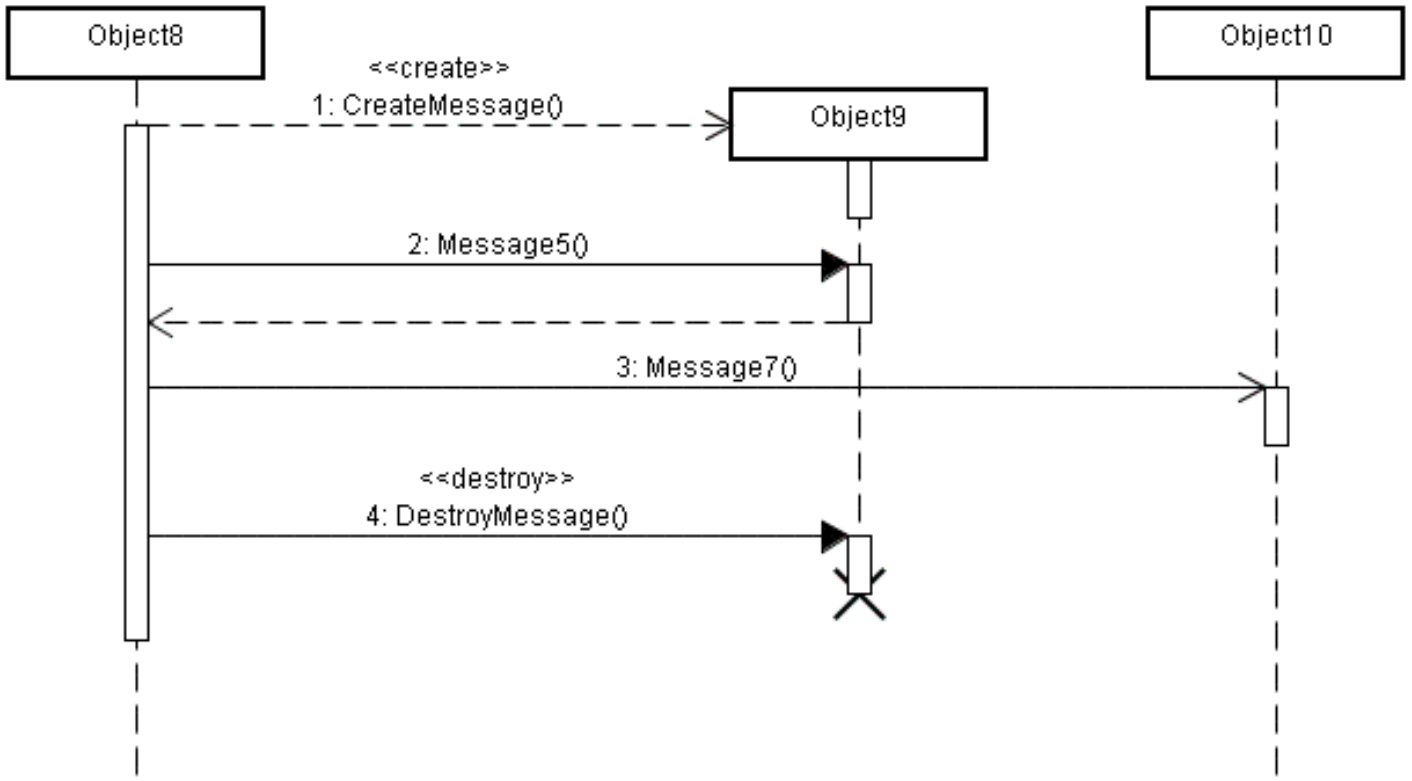
Diagramas de Seqüências

- Ocorrências de execução: corresponde ao tempo em que o objeto está ativo
  - O uso de ocorrência torna opcional o uso de mensagens de retorno



Diagramas de Seqüências

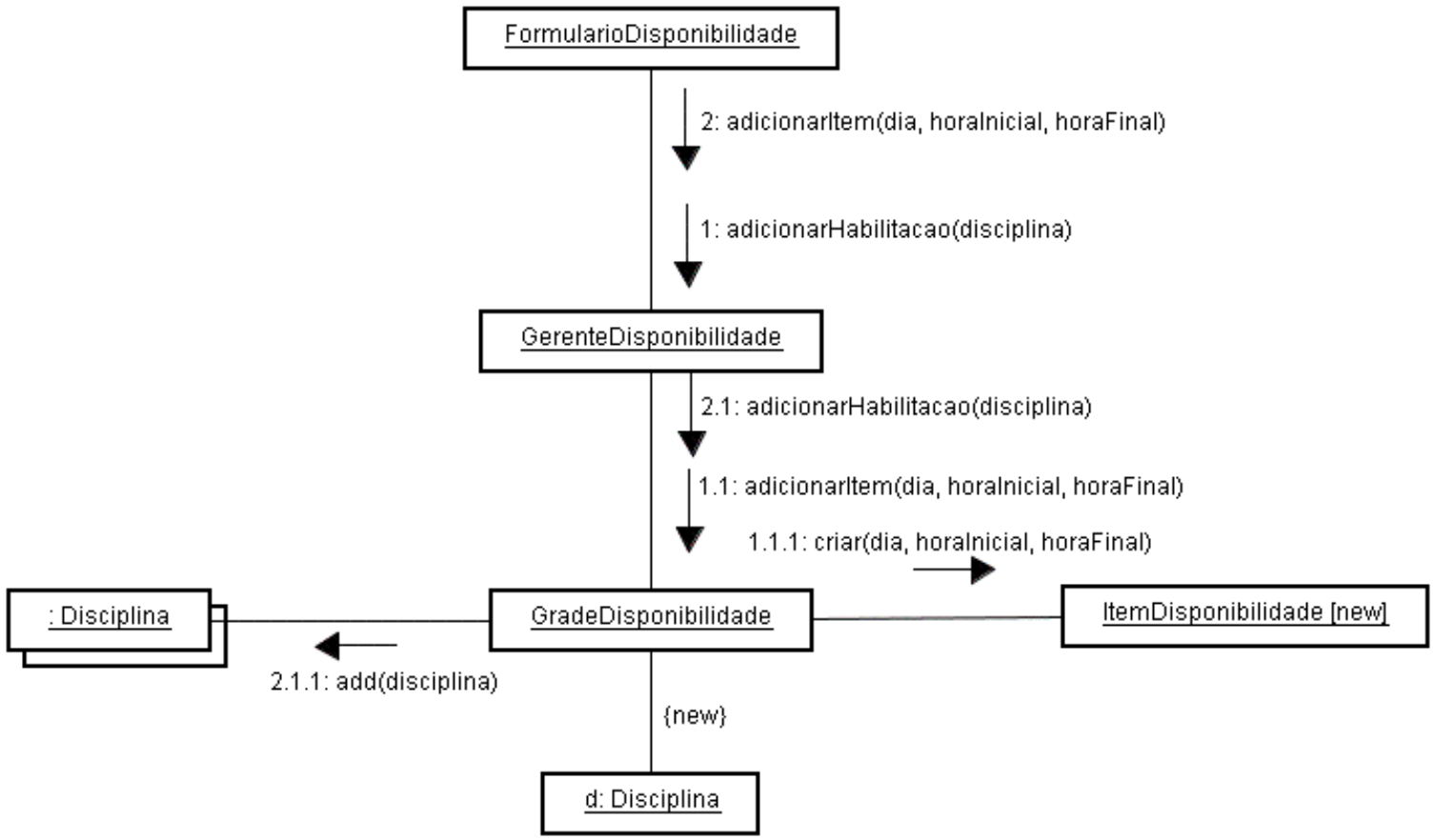
- Mensagens de Criação e Destruição



Diagramas de Comunicação

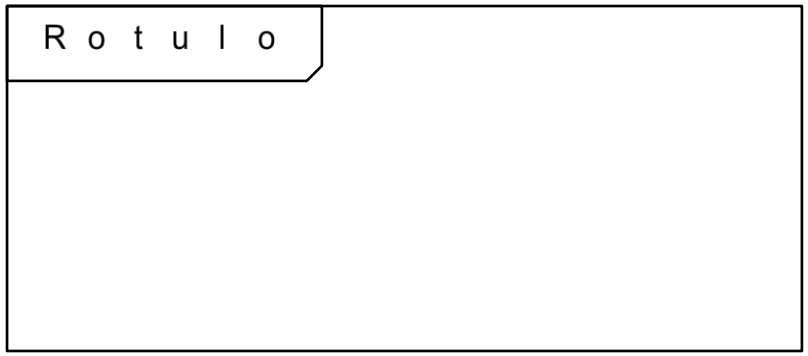
- Mostra os objetos relevantes para o caso de uso assim como as ligações entre os mesmos
- Estruturalmente é muito semelhante a um diagrama de objetos
- A diferença está nas ligações e mensagens trocadas entre os objetos.
- Diferente do diagrama de seqüência, o diagrama de comunicação não permite identificar a ordem de execução das mensagens.
- Todas as mensagens neste diagrama deve conter *obrigatoriamente* as expressões de seqüência.

Diagramas de Comunicação



Modularização da interação

- Inserido na UML 2.0
- Incluiu diversos elementos gráficos para construção modular de diagramas de interação
  - Quadros de interação, fragmentos combinados, referências, operadores, etc.
- Quadro de interação:
  - Serve para encapsular um diagrama de interação



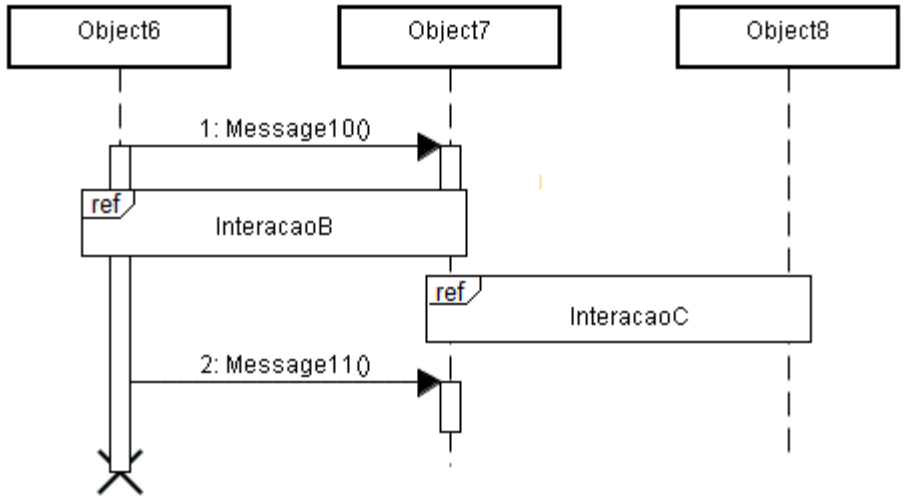
Um diagrama é posicionado no meio do quadro

ou

Uma referência a outro diagrama

Modularização da interação

- O rótulo pode ser o tipo e nome do diagrama:
  - Sd (diagrama de seqüência)
  - Comm (diagrama de comunicação)
  - Activity (diagrama de atividade)
- Ou uma referência a um diagrama separado: ref



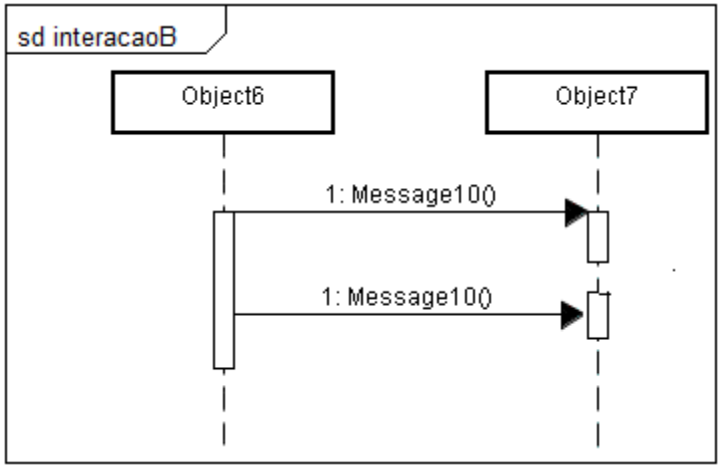


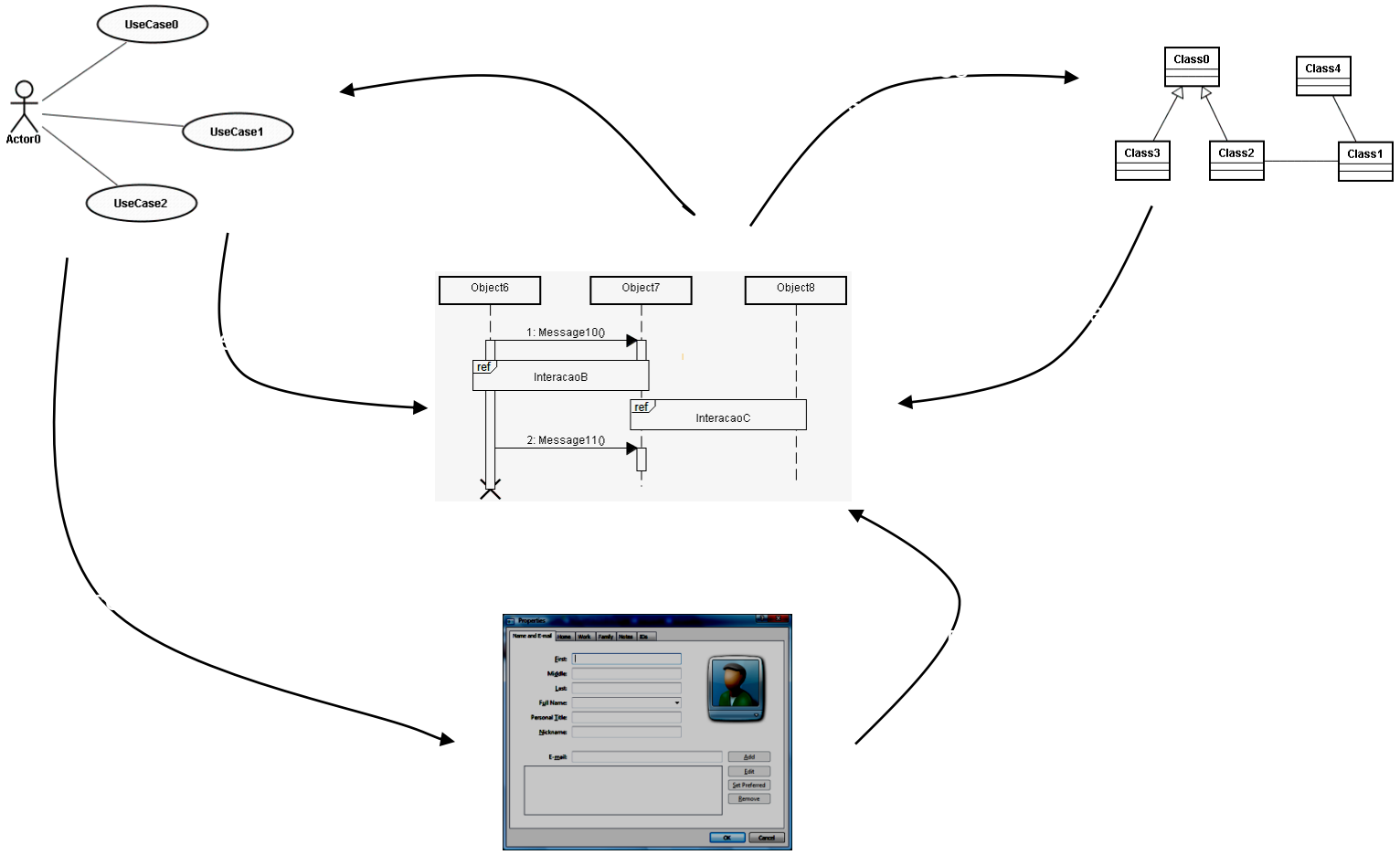
Diagrama de Visão da Interação

- Representado como um diagrama de atividades.
- Veremos posteriormente, ao estudar Diagrama de atividades

Procedimento para criação de um diagrama de interação

1. Para cada caso de uso, selecione um conjunto de cenários relevantes (fluxo principal, fluxo alternativo, fluxo de exceção)
2. Para cada cenário identifique os eventos do sistema
    1. Posicione os atores, objetos de fronteira, objetos de controle no diagrama
    2. Para cada passo do cenário, defina as mensagem enviadas de um objeto para o outro
    3. Defina as clausulas de condição e interação, caso existam
    4. Adicione multiobjetos e objetos de entidade, à medida que sua participação for necessária no cenário selecionado

Dependência dos artefatos produzidos



Estudo de caso - SCA

1. Para cada caso de uso, selecione um conjunto de cenários relevantes:
   a) Caso de Uso: Realizar Inscrição
   b) Cenário:

      1....

      2. O sistema apresenta as disciplinas para as quais o aluno tem pre-requisito (conforme RN03), excetuando-se as que já tenha cursado.

      3....

Estudo de caso - SCA

2. Posicione os atores, objetos de fronteira, objetos de controle no diagrama:
   - Identificamos os atores no MCU
   - Identificamos os obj no MCA

   Ator:Aluno
   Obj Fronteira:FormularioInscricao
   Obj Controle: ControleInscricao
   Obj Entidade: Aluno, disciplina.

Estudo de caso - SCA

3. Para cada passo do cenário, defina as mensagem enviadas de um objeto para o outro

– Usar um Diagrama de seqüência

Exercício

- Exercício: Dado as descrições de caso de uso, modelo de classes e protótipo, faça um ou mais diagrama de sequencia e um ou mais diagramas de comunicação para o caso de uso Solcitar Pedido

Descrição de casos de uso

Fluxo Principal

1. Cliente acessa o sistema
2. Sistema apresenta formulário de solicitação e solicita identificação do cliente
3. Cliente se identifica
4. Sistema acessa sistema SERASA para consultar situação do cliente
5. Sistema abre pedido e disponibiliza produtos para o cliente
6. Cliente seleciona o produto e insere na lista de itens pedidos.
7. O sistema retira os materiais selecionados da lista de materiais disponiveis
8. O Cliente confirma o pedido
9. O caso de uso se encerra

Descrição de casos de uso

Fluxo alternativo (6)

1. O cliente seleciona um ou mais itens e remove lista de itens pedidos
2. Os materiais selecionados retornam a lista de materiais disponiveis

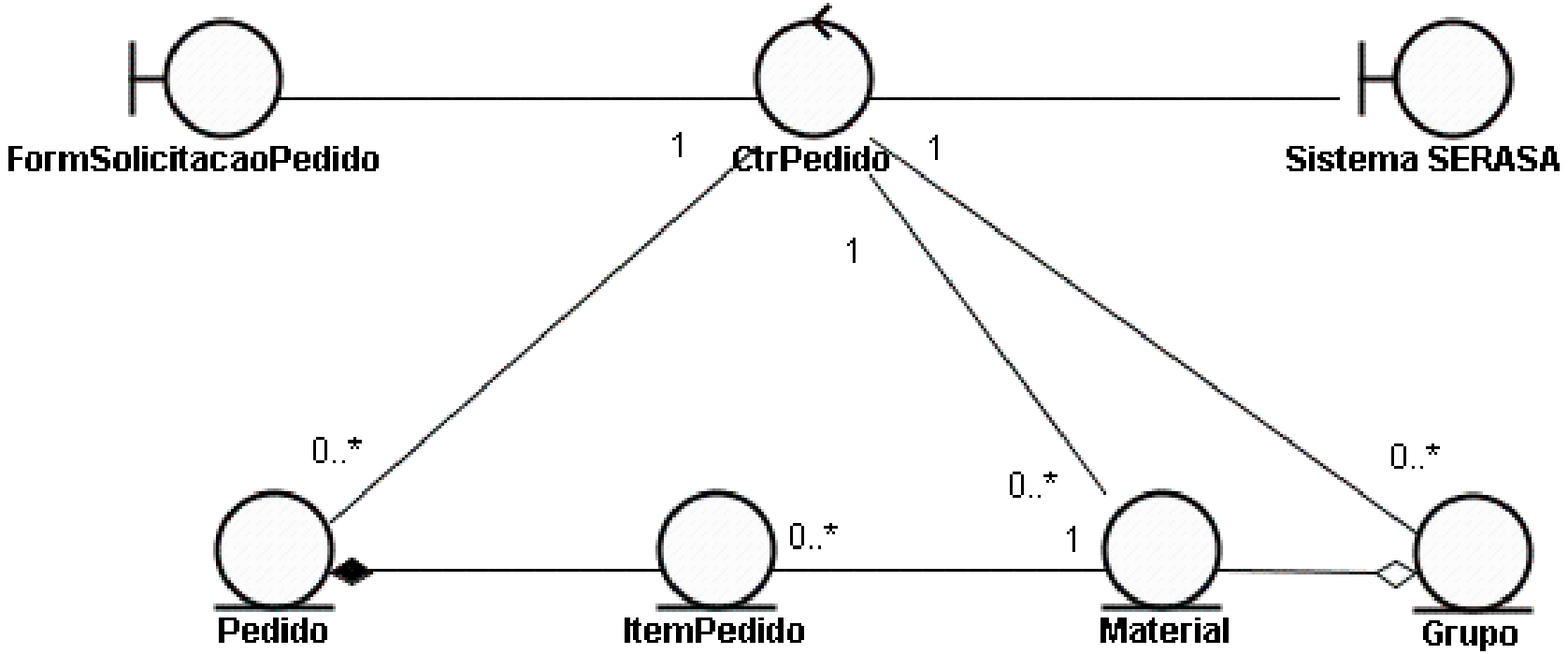
Fluxo alternativo (6)

1. O cliente remove todos os itens pedidos
2. Os materiais retornam a lista de materiais disponiveis

Fluxo alternativo (8)

1. Se o cliente não confirmar o pedido, o sistema deve cancelar o pedido automaticamente.

Diagrama de classes de análise



AESO

Faculdades Integradas Barros Melo

Nome

CPF

Identificação

Seja bem vindo. O seu pedido é o 89877-99 em 12/11/2007

Buscar

no grupo

P

Grupo Material	Material	Qtd
• Papelaria	• Lápis	• 100
• Papelaria	• Papel Oficio	• 12 resmas
• Livros	• Matemática	• 20
• Livros	• Português	• 45
• Informática	• CD	• 234
• Informática	• Mouse	• 23

Material	Qtd
• Lápis	• 2
• Português	• 1
• CD	• 5

Confirmar

Sair

Bibliografia

- BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivan. **UML**: **guia do usuário**. Rio de janeiro: Campus, 2000. 472p.