



ENGENHARIA DE SOFTWARE

INTRODUÇÃO A MODELAGEM

DE SOFTWARE

UML - UNIFIED MODELING LANGUAGE

Profº Ms Gustavo Molina
msc.gustavo.unip@gmail.com

Prof. Ms Gustavo Molina



- Graduado em Sistemas de Informação pelo MACKENZIE.
- Licenciado em Matemática pela UNIP.
- Pós – Graduado em Plataforma de Desenvolvimento Web pelo CLARETIANO.
- Pós – Graduado em IA pela faculdade Serra Geral
- Pós – Graduado em Gestão e Governança de Tecnologia da Informação pela UNIP
- Mestre em Engenharia Elétrica pela FEI
- Doutorando em Ciências da Educação pela Ivy Enber Christian University

Prof. Ms Gustavo Molina



<https://www.linkedin.com/in/gustavo-molina-a2798418/>



<http://lattes.cnpq.br/8512452850609937>



msc.gustavo.unip@gmail.com



<https://github.com/gustavomolina17>

UML - Linguagem de Modelagem Unificada

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

- 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, Microsoft, Unysis, etc)
- 2015 – Foi lançada a UML 2.5
- Especificação atual adotada pela OMG.

UML

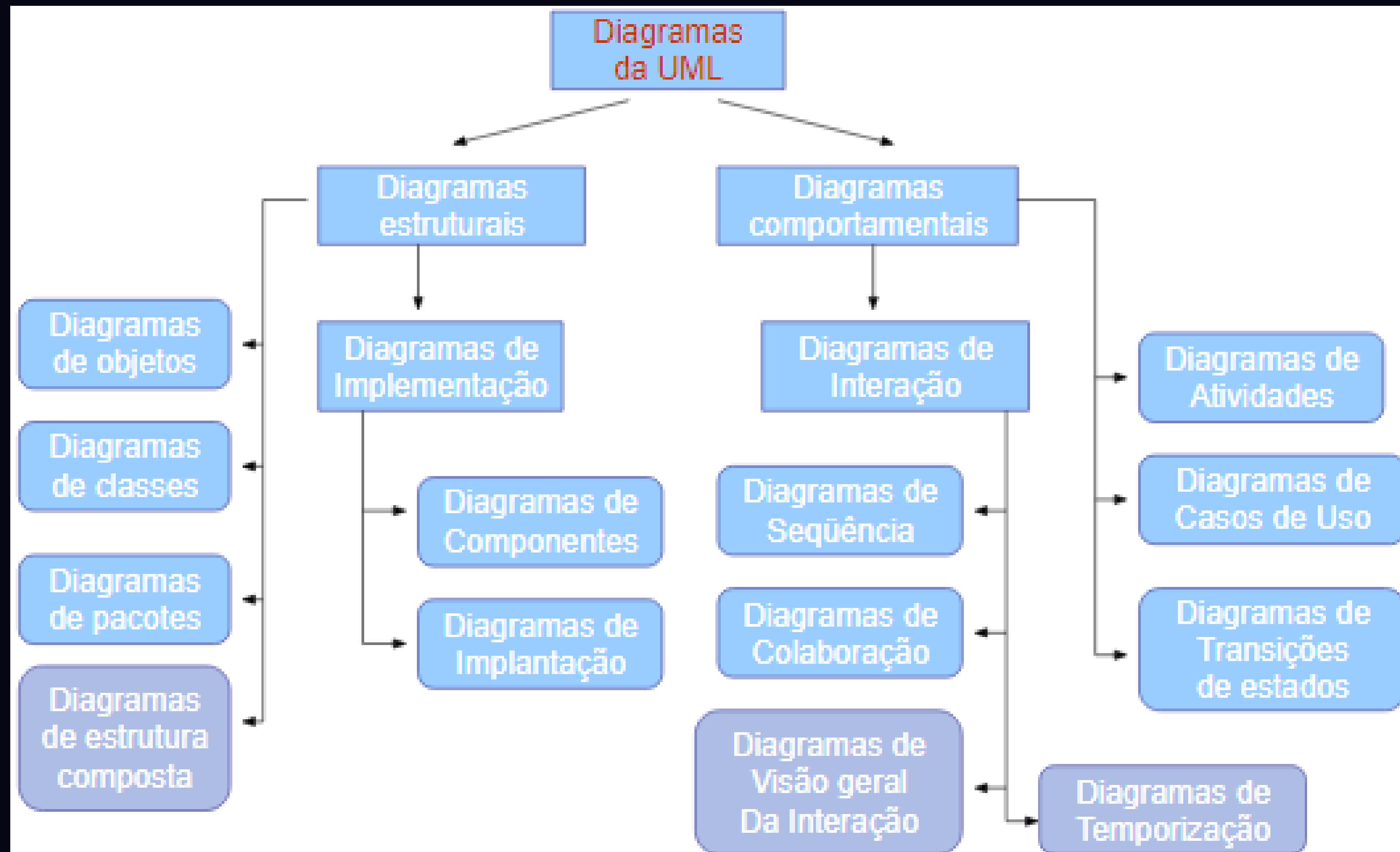
- é uma linguagem visual para modelar sistemas Orientados a Objetos.
- Define elementos gráficos que podem ser utilizados na modelagem de sistemas.
- Através dos elementos definidos na linguagem podem-se construir diagramas para representar diferentes perspectivas de um sistema.
- Cada elemento gráfico 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

- 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.

- 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: Características 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: Características de concorrência, sincronização e desempenho do sistema.

- Diagramas:
 - Os documentos gerados em um processo de desenvolvimento 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.



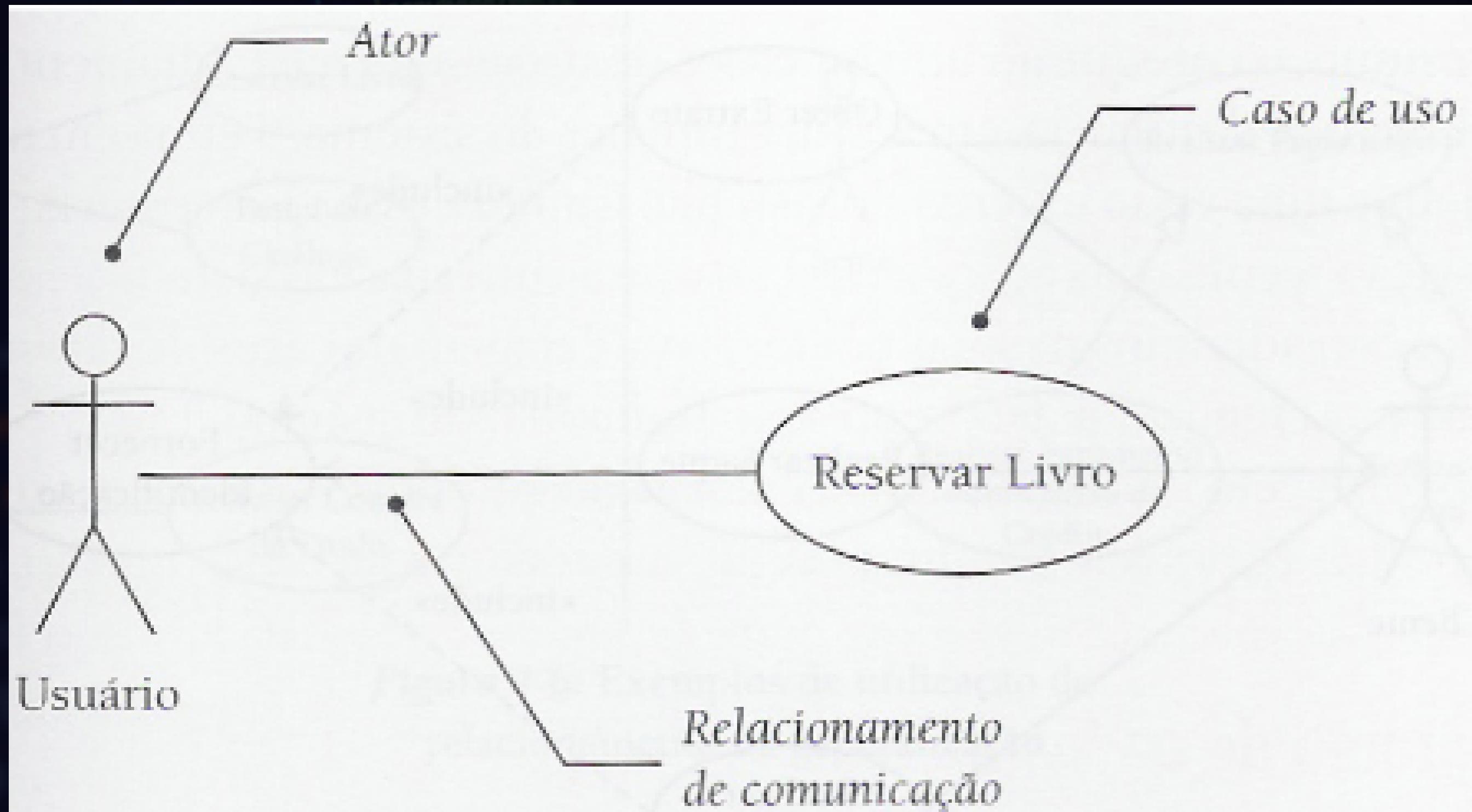
1º MODELO:

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

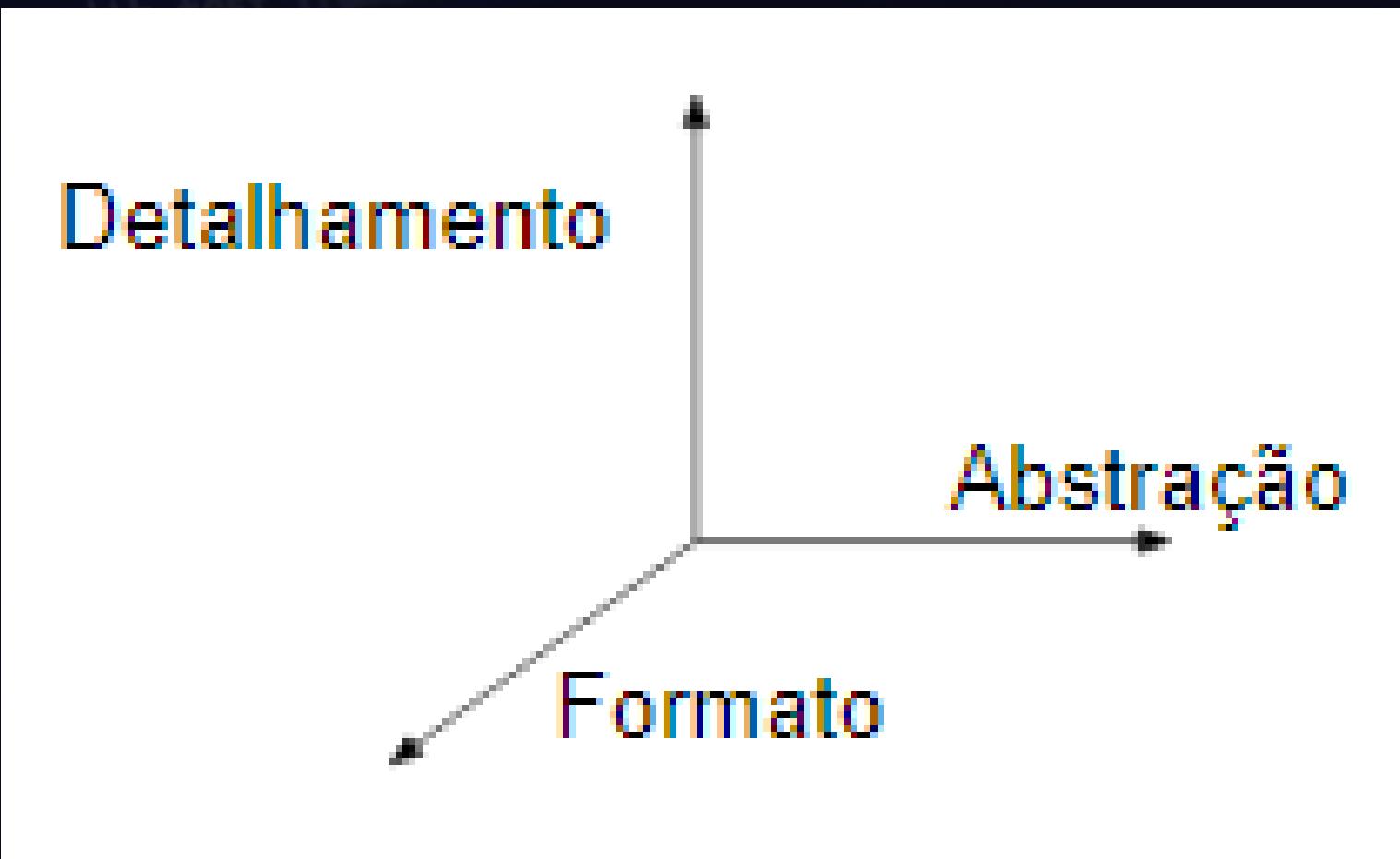


Casos de Uso

- 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 uso 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) O sistema valida a senha e exibe as opções de operações possíveis.
- 5) O Cliente opta por realizar um saque.
- 6) O 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 uso 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 a realização de um saque.	Requisita o total a ser sacado.
Fornecer 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: Descreva as interações em detalhes.
-
- Grau de abstração
- Existência ou não de menção a aspectos relativos à 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

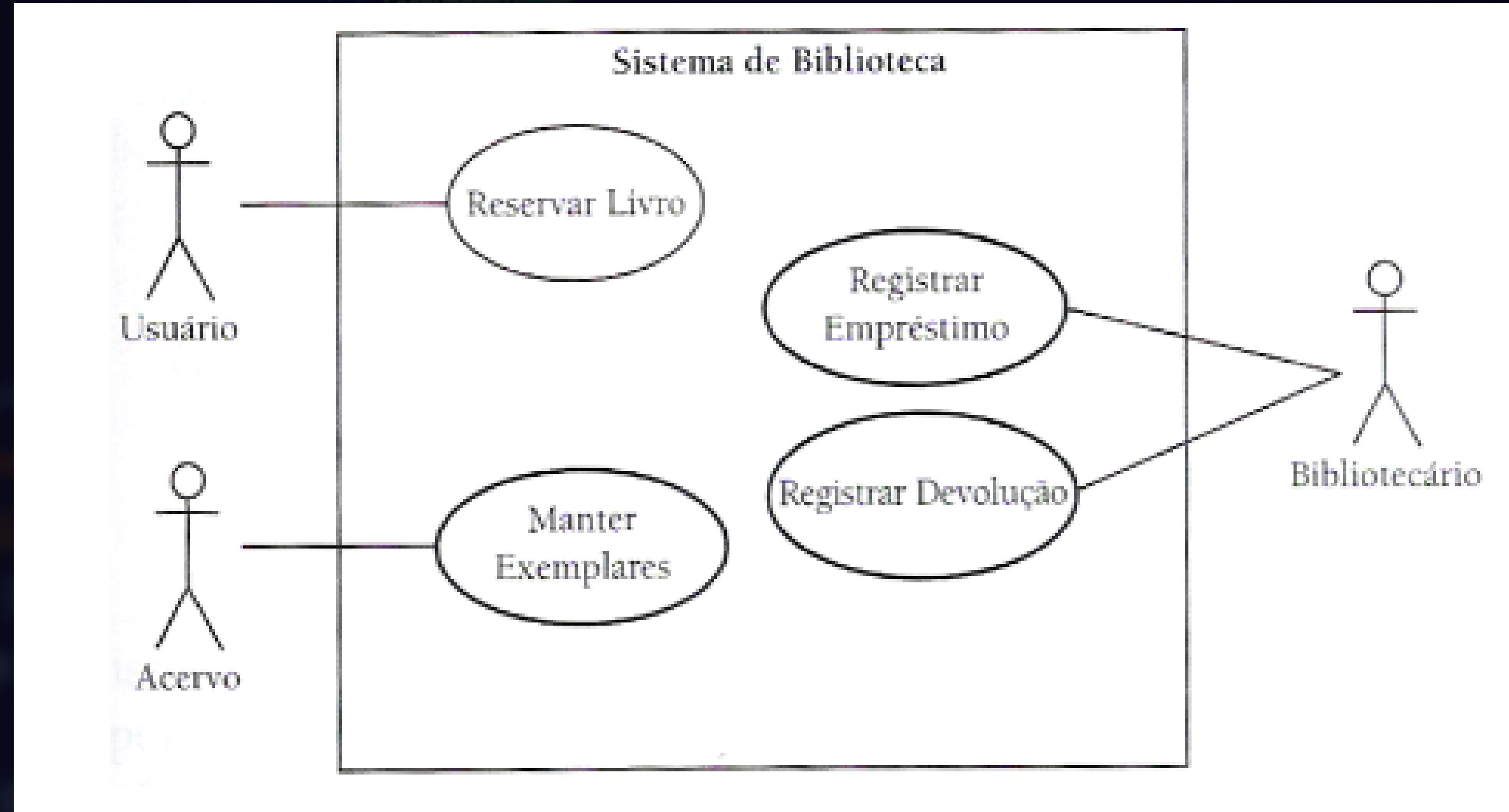
- Cenários
- É a descrição de uma das maneiras pelas quais o caso de uso pode ser utilizado.
- É um episódio de utilização de uma funcionalidade.
- Pode ser utilizada posteriormente na fase de testes.
- Pode ser vista como uma instância de um caso de uso.

- **Cliente insere seu cartão no caixa eletrônico.**
- **O sistema apresenta a tela de requisição de senha do Cliente.**
- **Cliente digita sua senha**
- **O 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

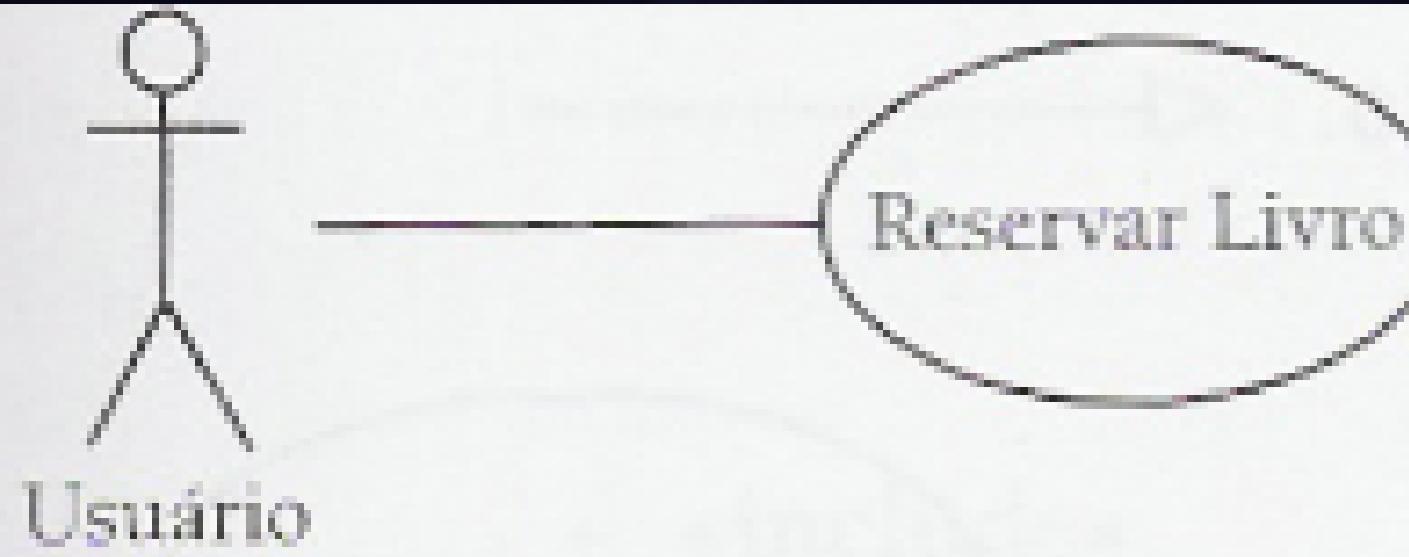


Relacionamentos

- 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

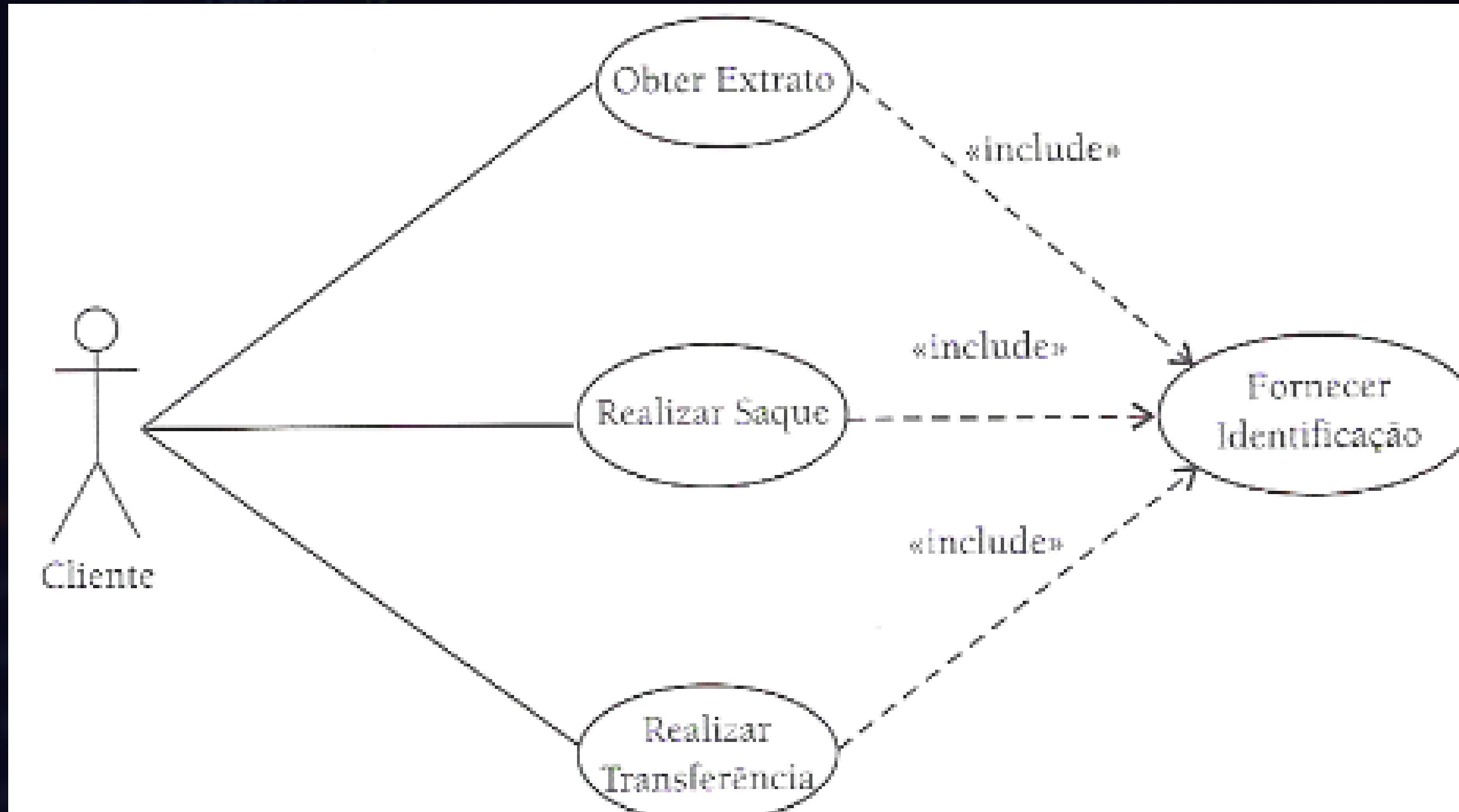
- Comunicação:
- Informa a que caso de uso o ator está associado.
- Representa as trocas de informação entre os atores e casos de uso.
- É o mais utilizado nos MCU.
- Um ator pode estar associado a vários casos de uso.
- Um caso de uso pode estar associado a vários fatores.



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 casos de uso podem incluir o comportamento deste caso de uso comum.
- Ex: Obter Extrato, Realizar Saque e Transferência incluem Validar Senha

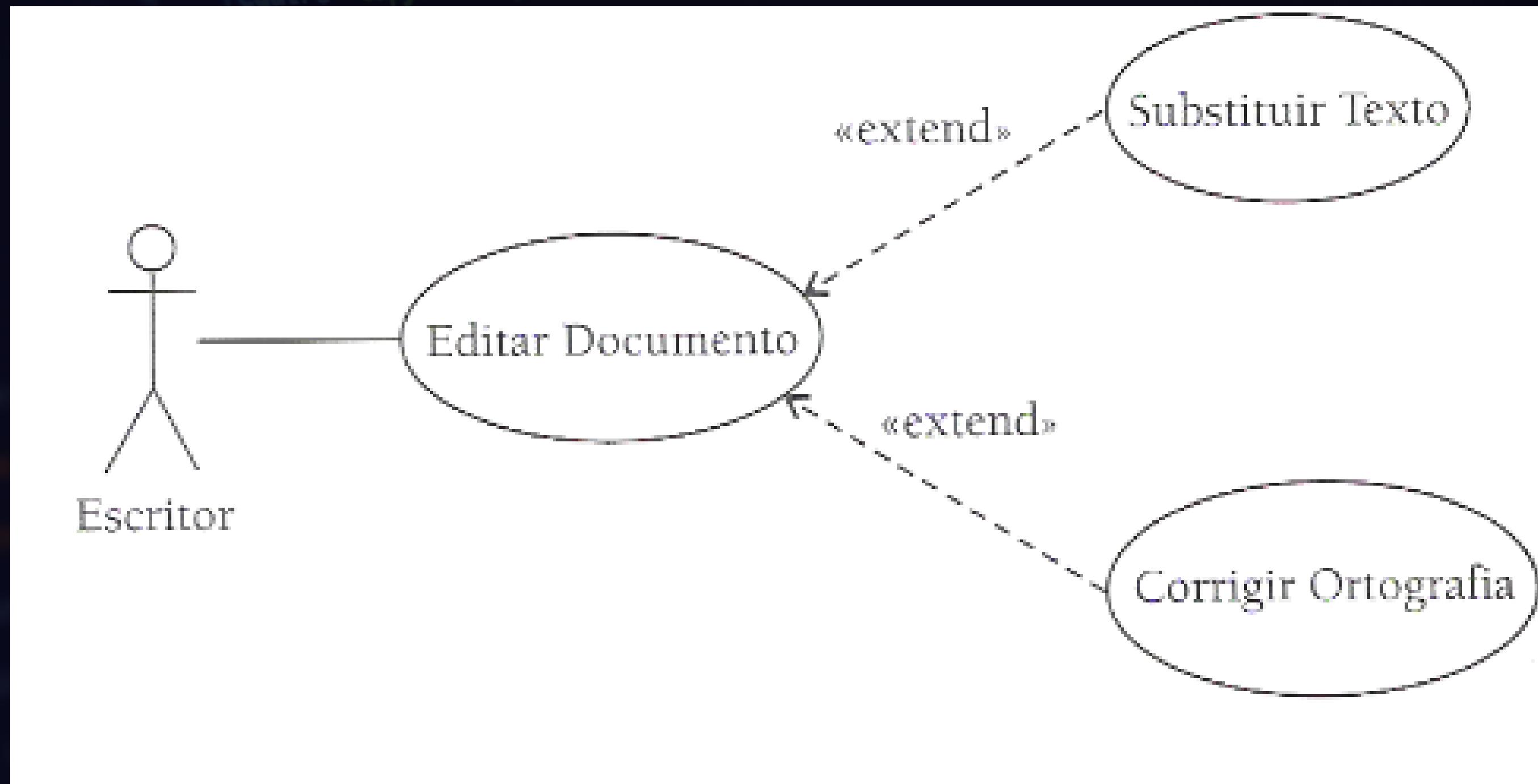
Diagrama de caso de Uso - Inclusão



Relacionamentos

- Extensão:
- Somente entre Casos de Usos.
- Modelar situações em que diferentes sequências de interações podem ser inseridas em um mesmo caso de uso. Estas sequências 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



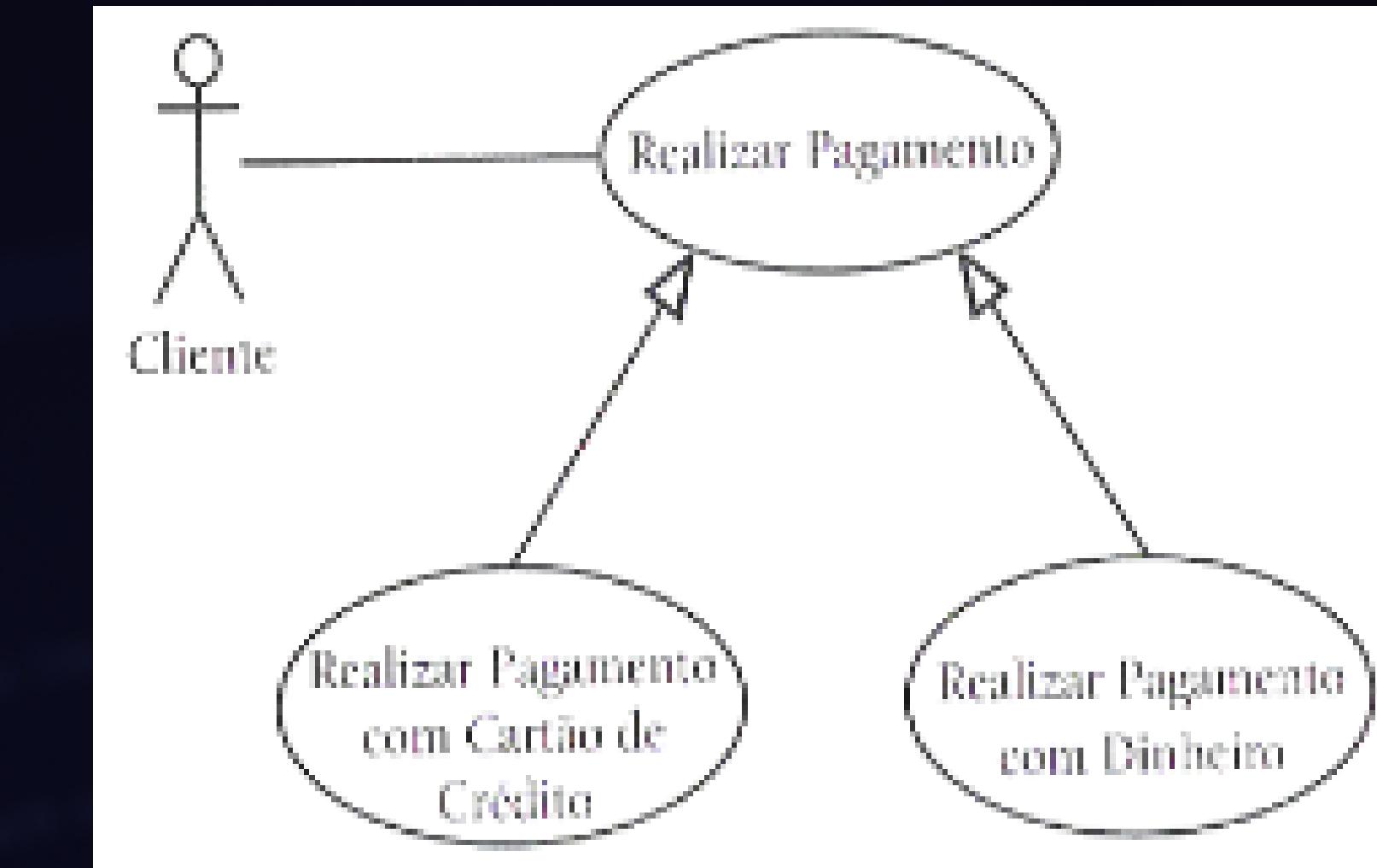
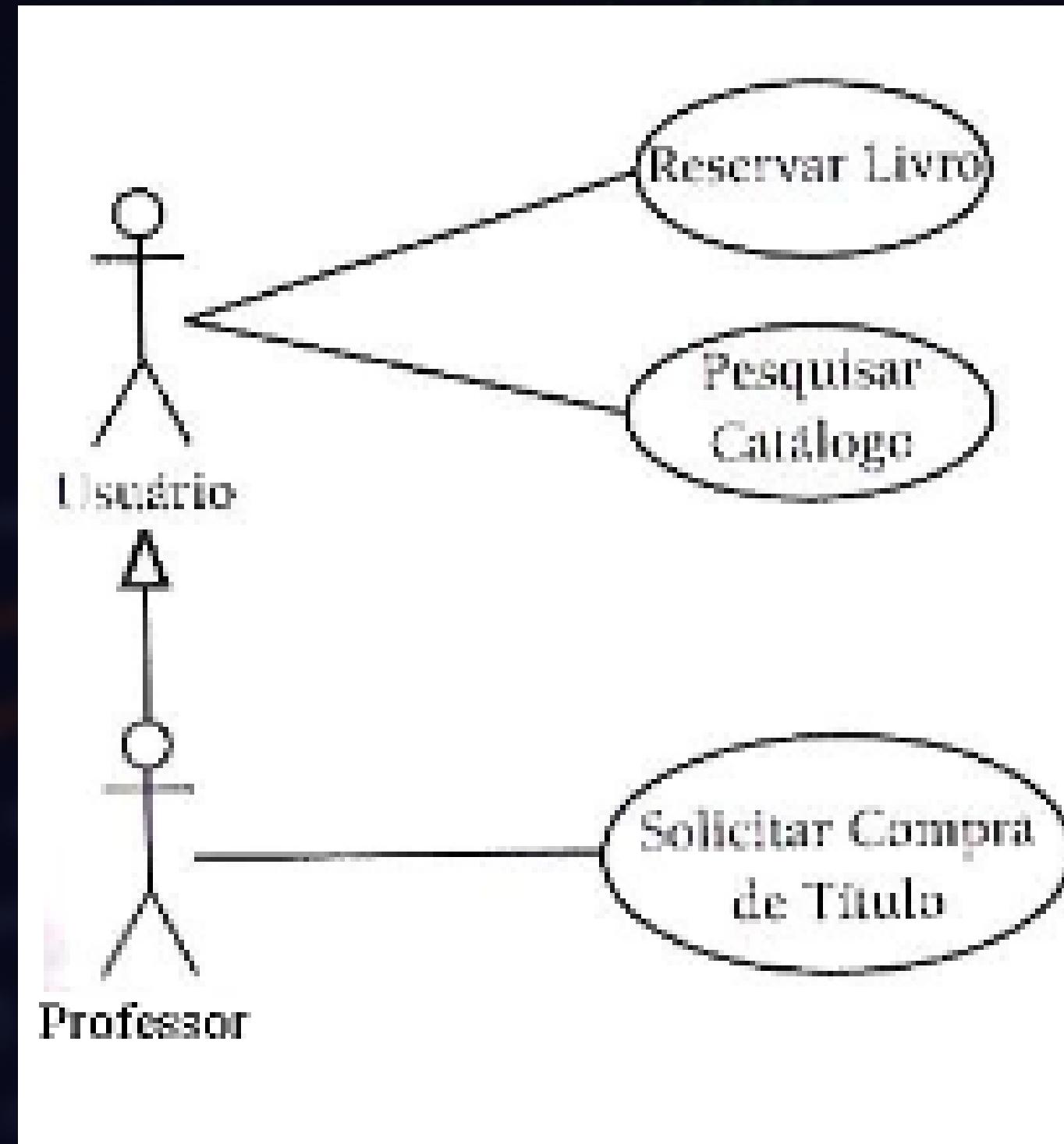
Relacionamentos

- 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 do 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 do filho.
- Não há especificação de comportamento para o caso de uso abstrato.

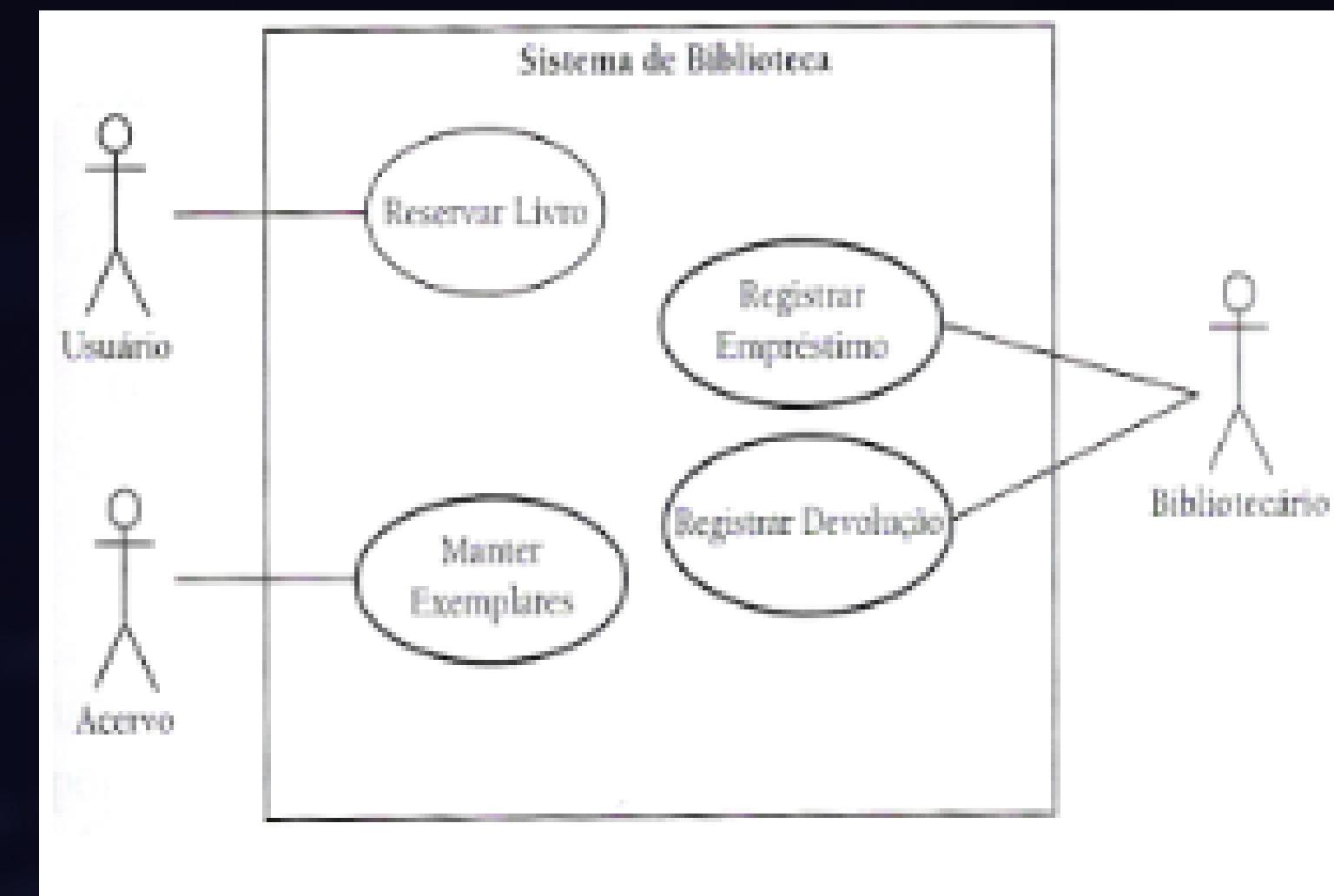
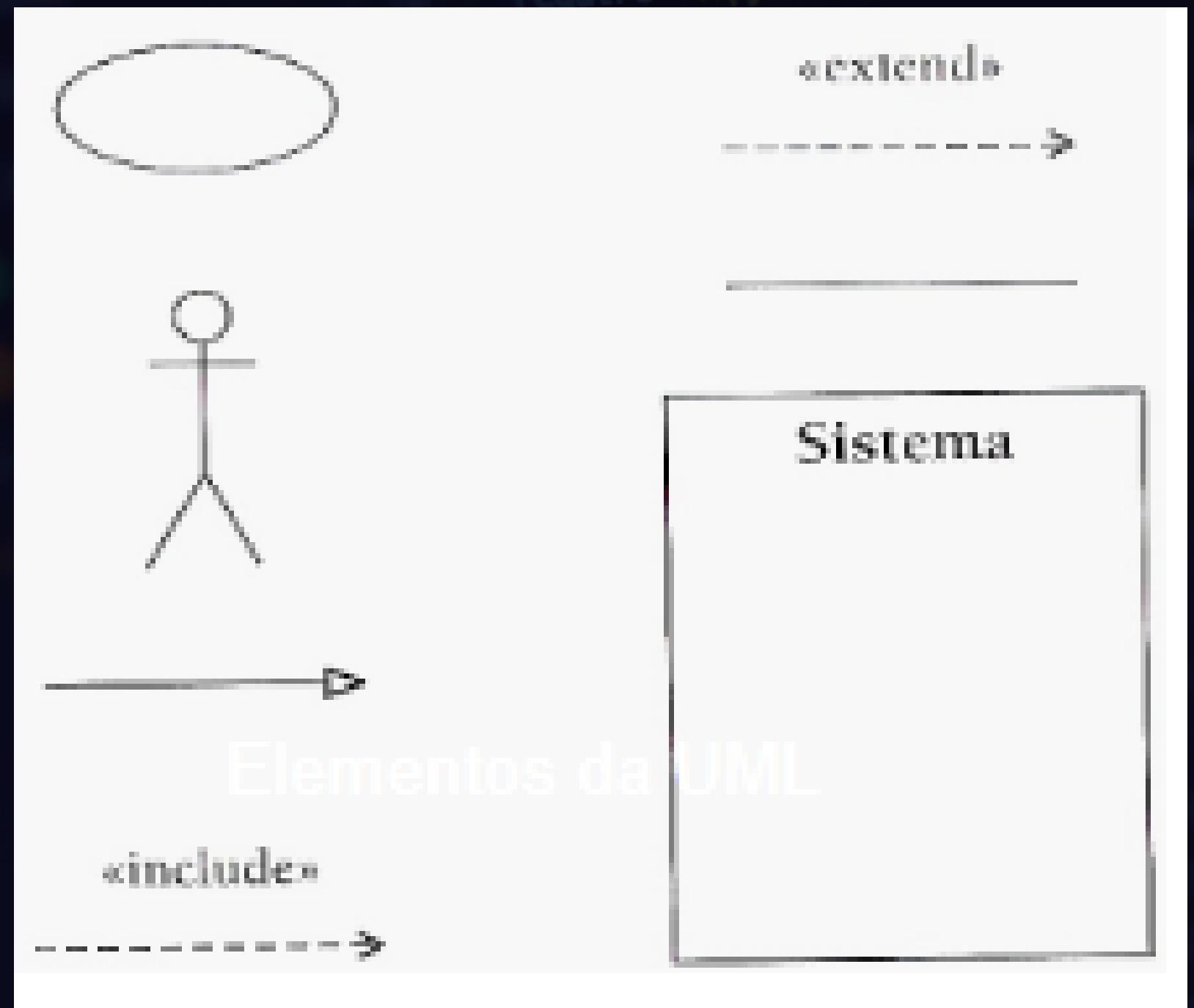
Relacionamentos

- Generalização:
- Exemplos:
- *Requisitar Produtos* é a 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 Caso de Uso



Quando usar relacionamentos

- Inclusão
 - Quando o mesmo comportamento se repete em mais de um caso de uso.
 - O Comportamento comum é necessariamente contido em todos os cenários do caso de uso inclusor.
 - O caso de uso inclusor não está completo sem o caso de uso incluído.
- 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

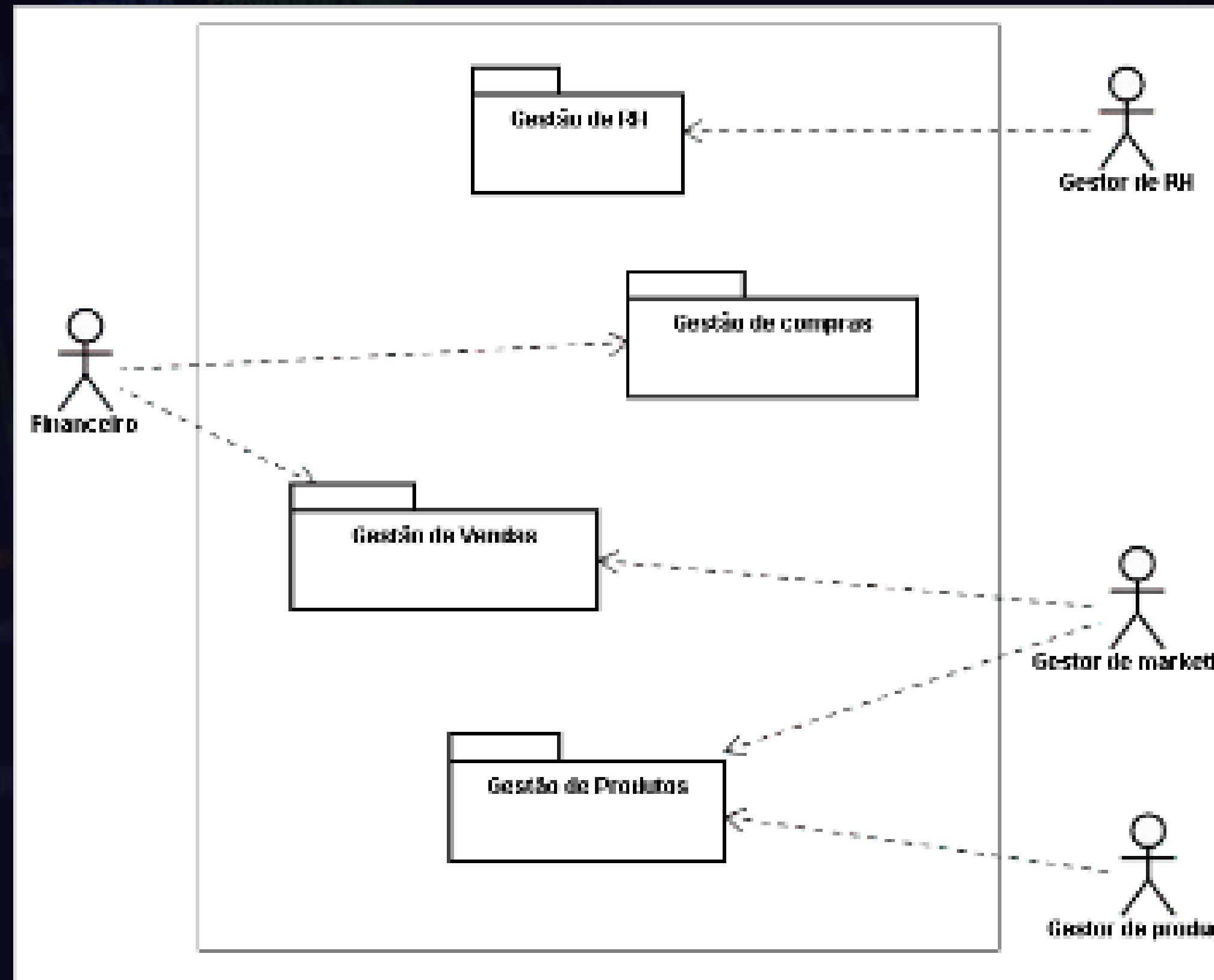
- Generalização de caso de uso
- Identifica-se vários casos de uso com o mesmo comportamento.
- Se o comportamento do pai difere em alguma coisa do do filho, não use generalização, mas extensão.
- Generalização de Ator
- Precisa definir um ator que desempenha papel já desempenhado por outro ator em relação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

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?

- 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 ocorra regularmente no tempo?
 - Para cada requisito funcional, existe um (ou mais) caso(s) de uso para atendê-lo?

Construção do MCU



Construção do MCU

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

- 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 sequência 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 entre si. (não é obrigatório).
- Fluxos de exceção: Descreve o que acontece quando algo inesperado ocorre (erro, uso inválido, cancelamento).

- 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.

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 requisitos 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.

RFs

- 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 as inscrições dos 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 para 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

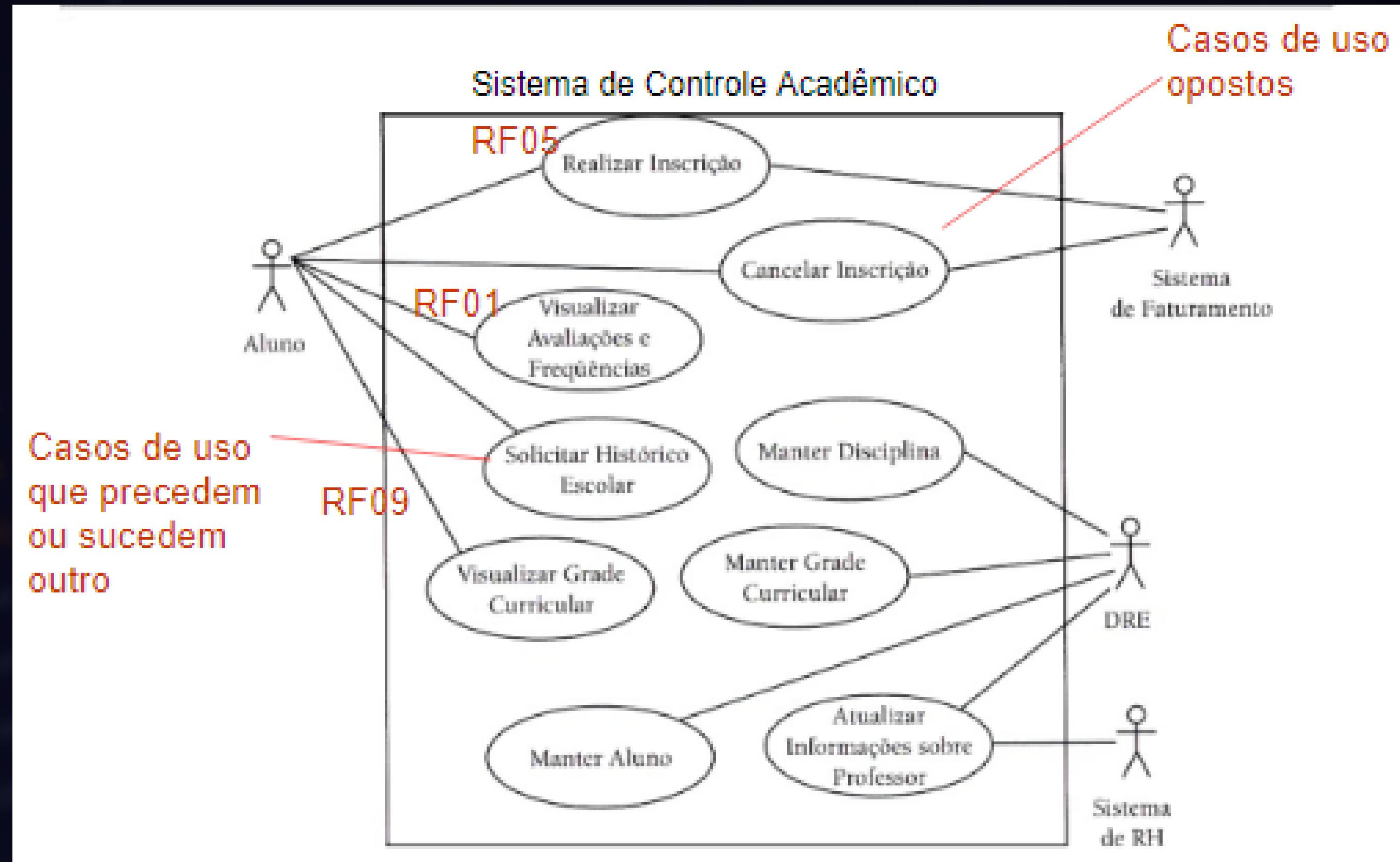


Diagrama de caso de uso



Descrição do Caso de Uso no formato Essencial e Expandido

Realizar Inscrição (CSU01)

Sumário: Aluno usa o sistema para realizar inscrição em disciplinas

Autor Primário: Aluno

Autor 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

1. 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.
2. 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
3. Se o aluno não aceitar, o caso de uso prossegue a partir do passo 4.

Fluxo de Exceção (4): Violação de RN01

1. Se o aluno atingir 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

```
9   require 'capybara/rspec'  
10  require 'capybara/rails'  
11  
12  Capybara.javascript_driver = :webkit  
13  Category.delete_all; Category.create(name: "Category 1")  
14  Shoulda::Matchers.configure do |config|  
15    config.integrate do |with|  
16      with.test_framework :rspec  
17      with.library :rails  
18    end  
19  end  
20  
21  # Add additional libraries to the gemspec here.  
22  #  
23  # Requires supporting files contained in 'support/' directory  
24  #  
25  # run as spec  
26  # in _spec.rb will boot the application  
27  # run twice. It is recommended to use  
28  # end with _spec.rb. You can configure  
29  # option on the command line.  
30  #  
31  # found for 'mongoid'
```

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.

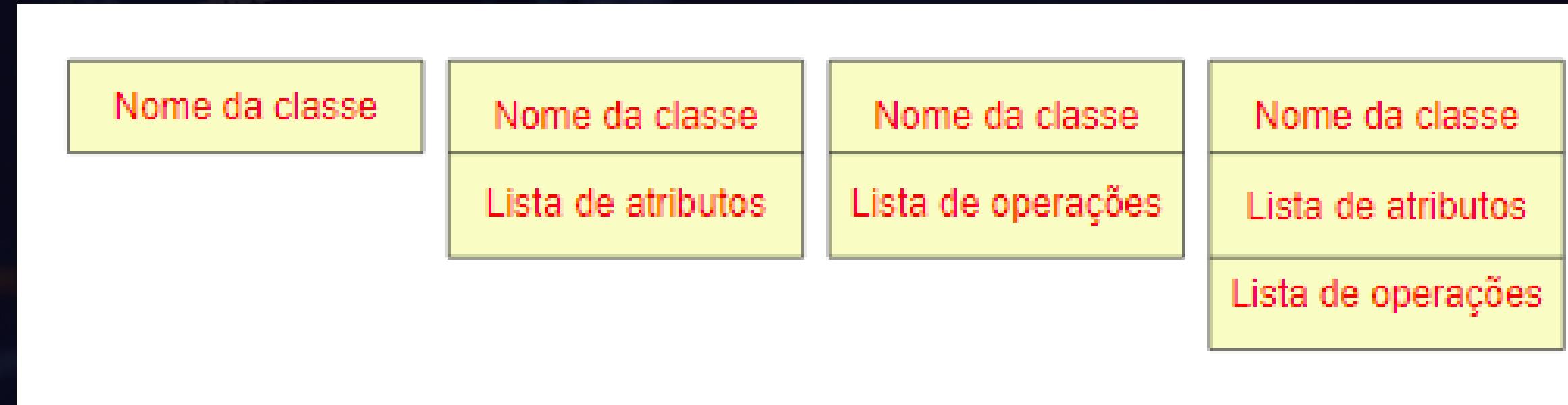
Introdução

- 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

Diagrama de Classes

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



- O grau de abstração determina quando usar uma notação.

- Classes
- Exemplo:

Diagrama de Classes

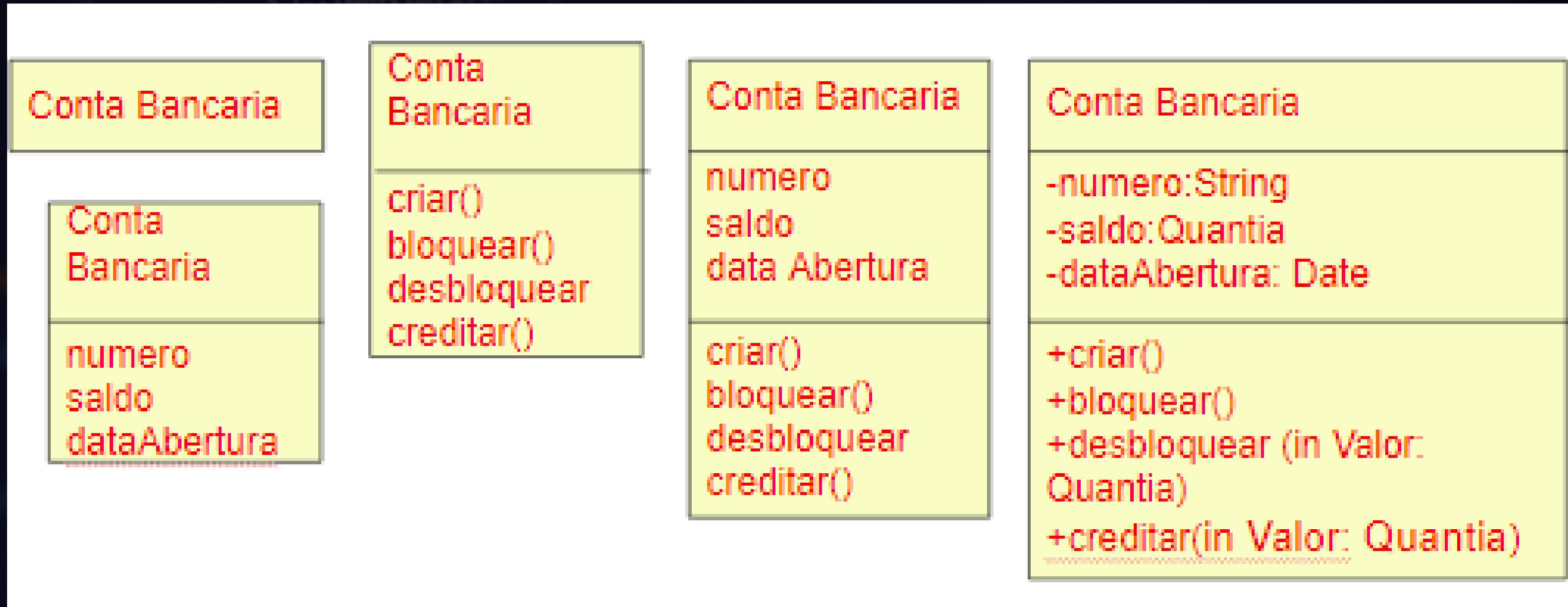


Diagrama de Classes

- Classes
- Os atributos correspondem à descrição dos dados armazenados pelos objetos de uma classe.
- Cada objeto tem os seus próprios valores.
- As operações correspondem à 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 é representado 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.
- Associações
 - Características das associações:
 - Multiplicidade
 - Nome
 - Direção de leitura
 - Papéis
 - Tipo de participação
 - Conectividade

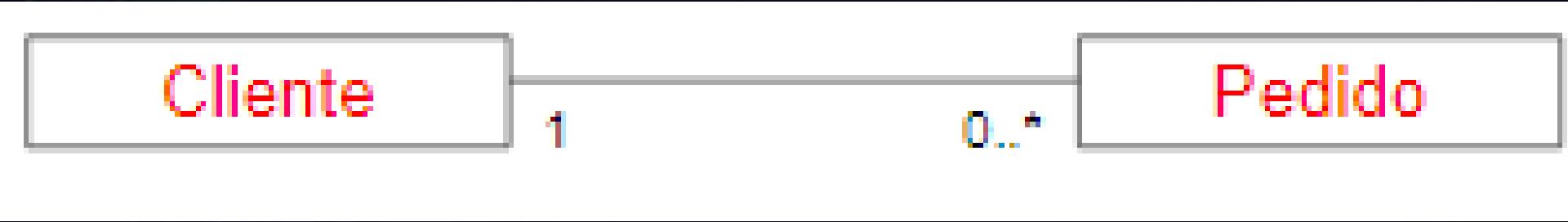
Diagrama de Classes

- 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:

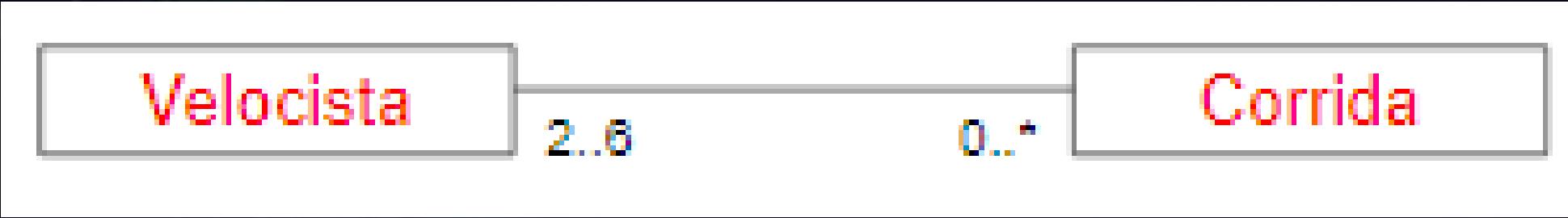


- Pode haver algum objeto da classe Cliente que está associado a vários objetos da classe Pedido (representado por * do 0..*)
- Pode haver algum objeto da classe Cliente que NÃO está associado a classe Pedido (representado por 0 do 0..*)
- Objetos da classe pedido está associado a UM e somente um objeto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

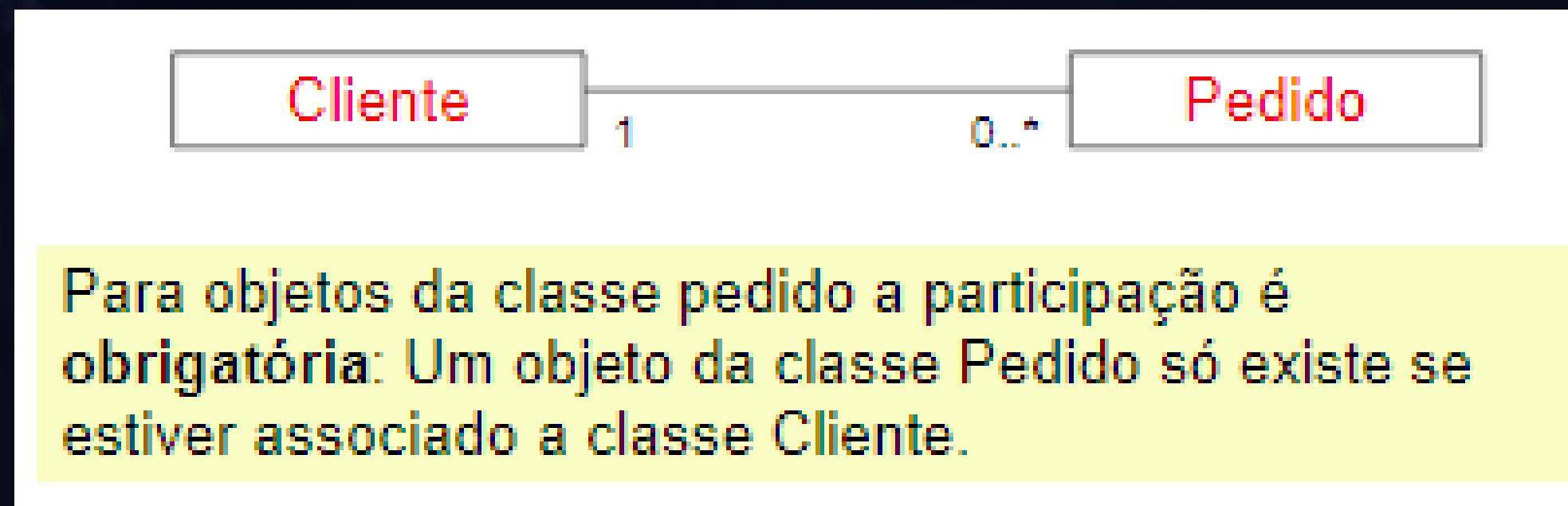


Diagrama de Classes

- 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).

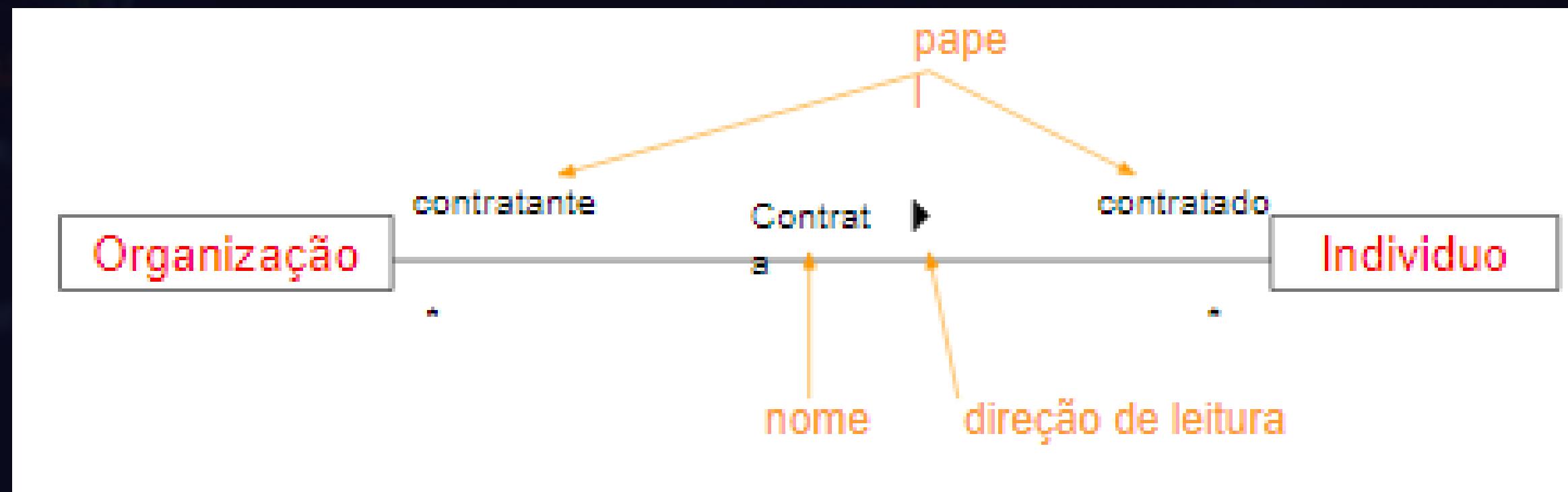


Diagrama de Classes

- 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.

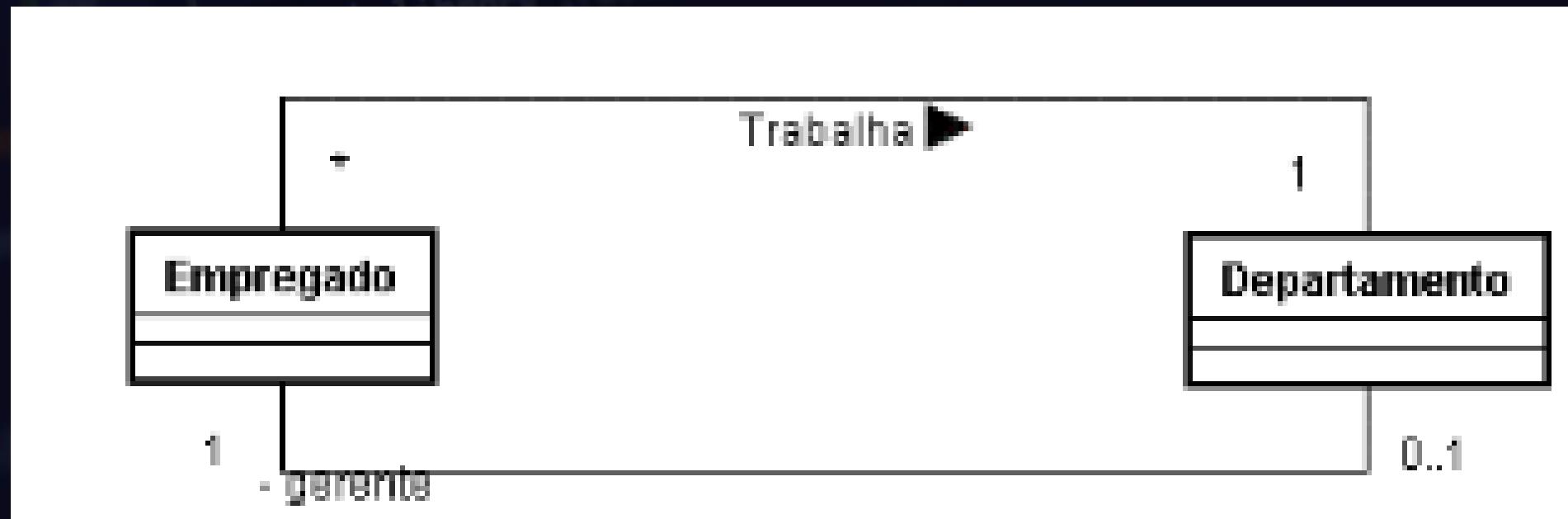


Diagrama de Classes

- 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 à associação de qualquer conectividade.
- Pode ser substituída por uma classe com associação para as outras duas 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

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

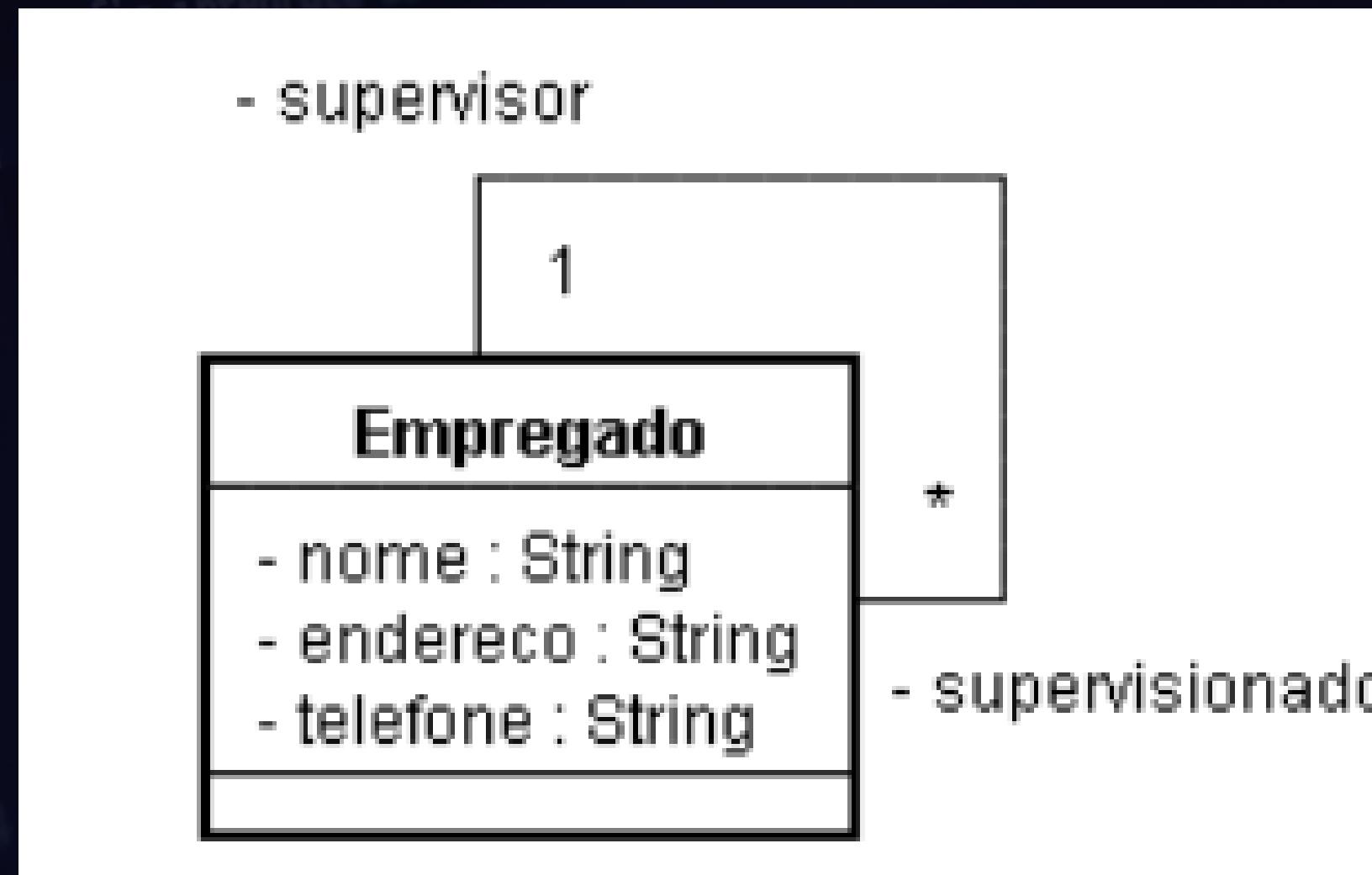


Diagrama de Classes

- Diagrama de Classes

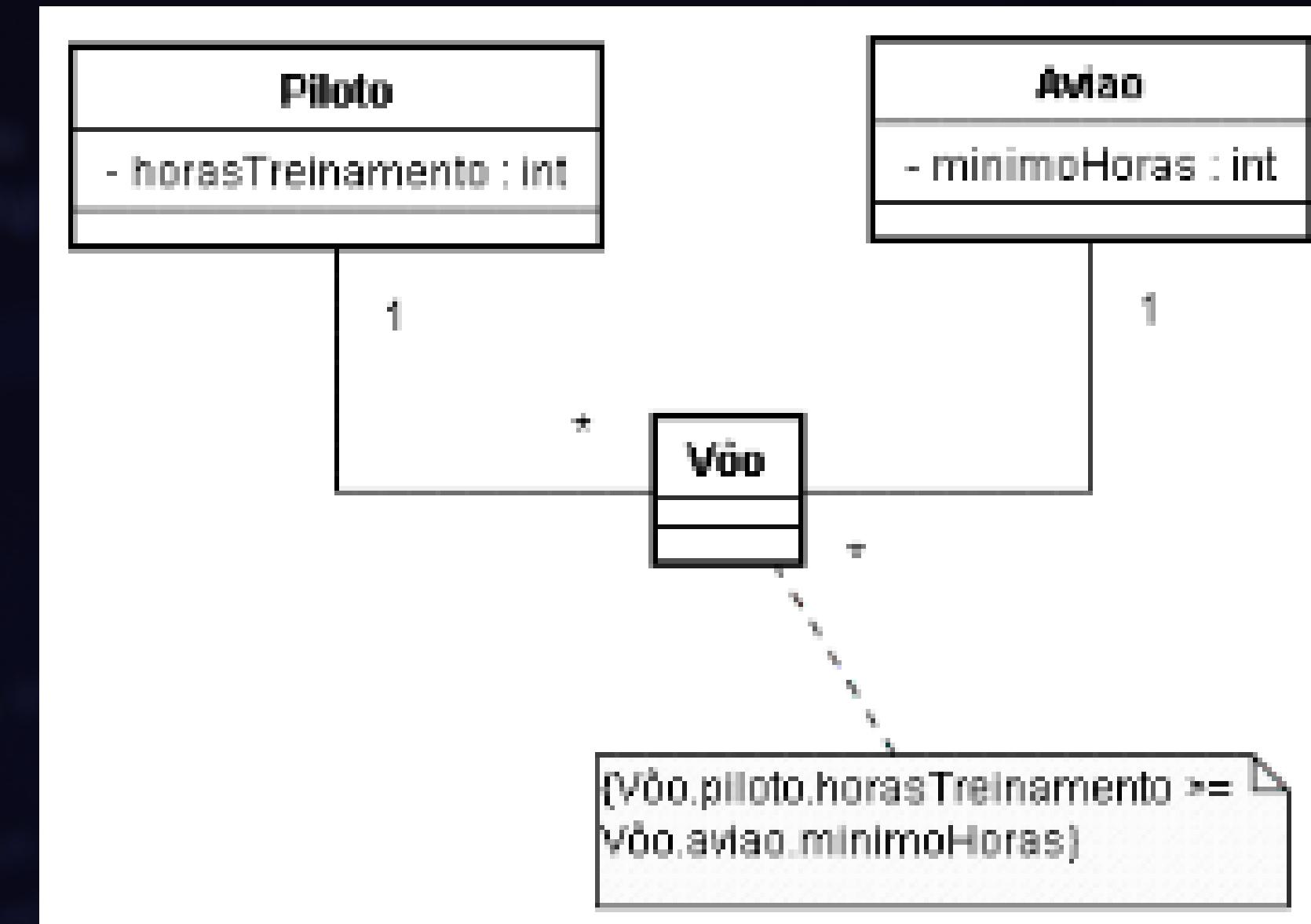
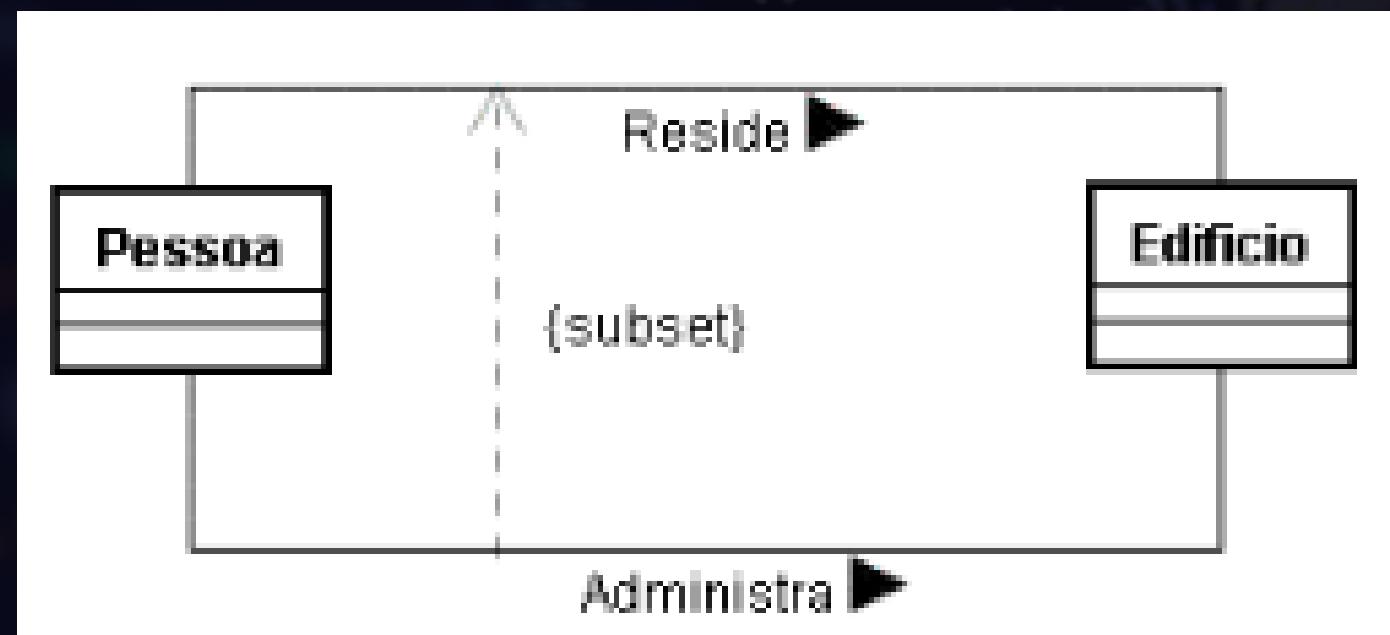


Diagrama de Classes

- 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 às 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.

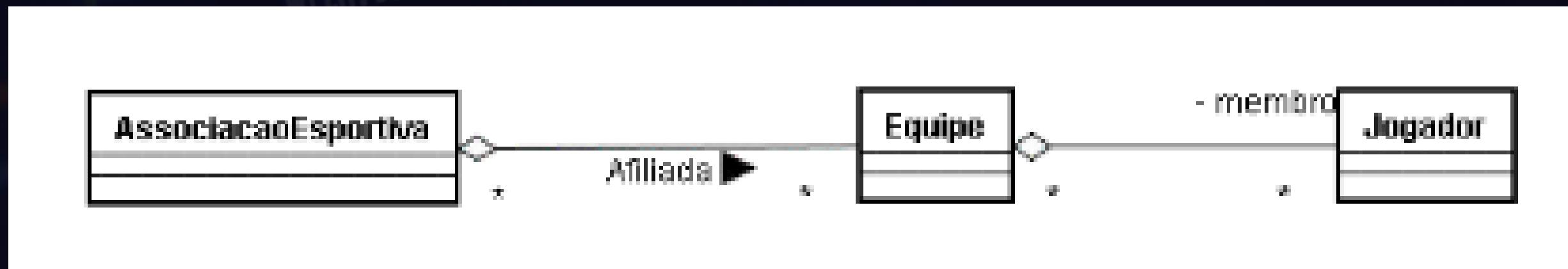


Diagrama de Classes

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

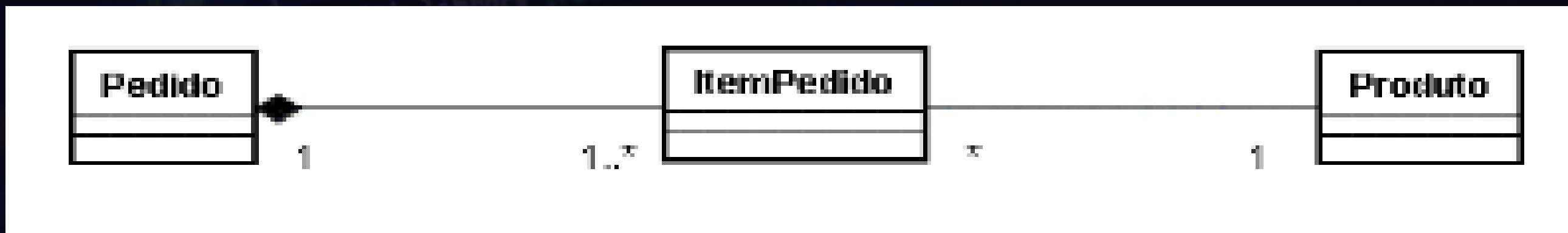


Diagrama de Classes

- 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 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:

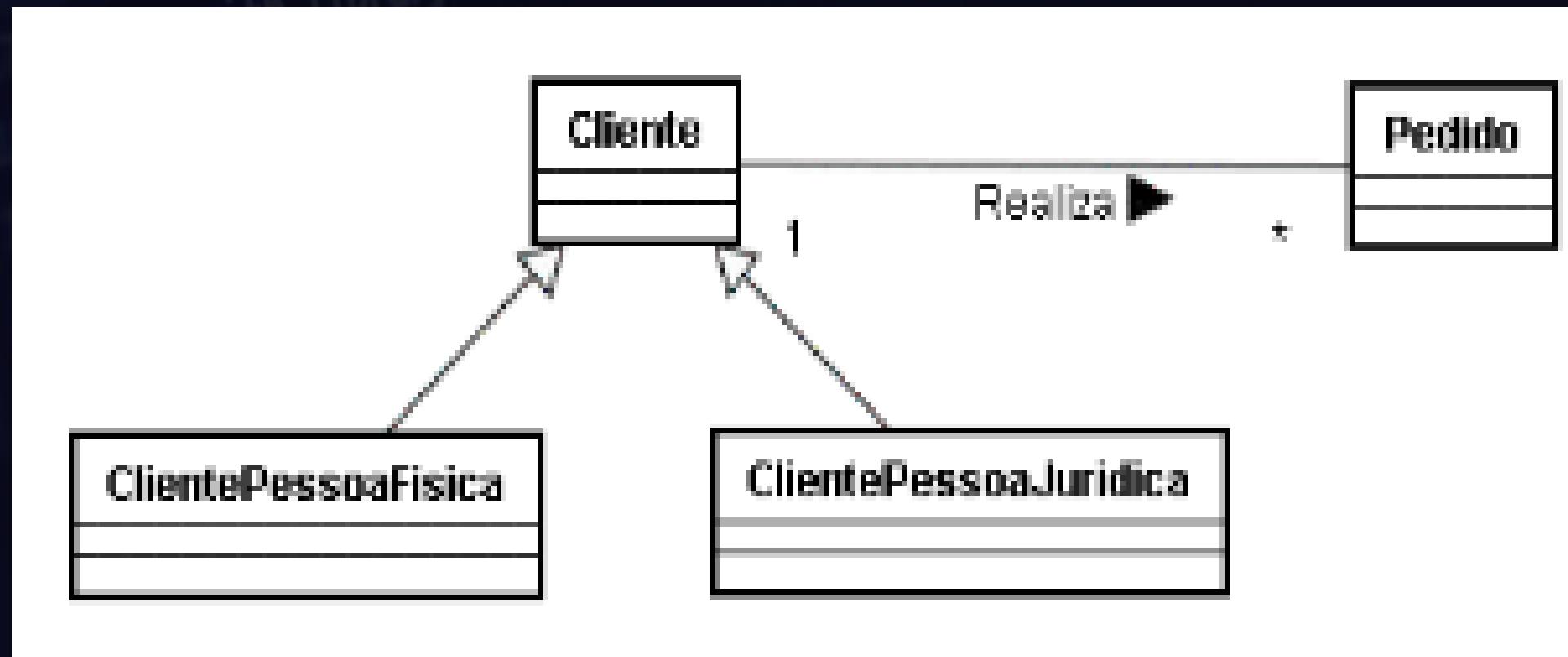


Diagrama de Classes

- 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

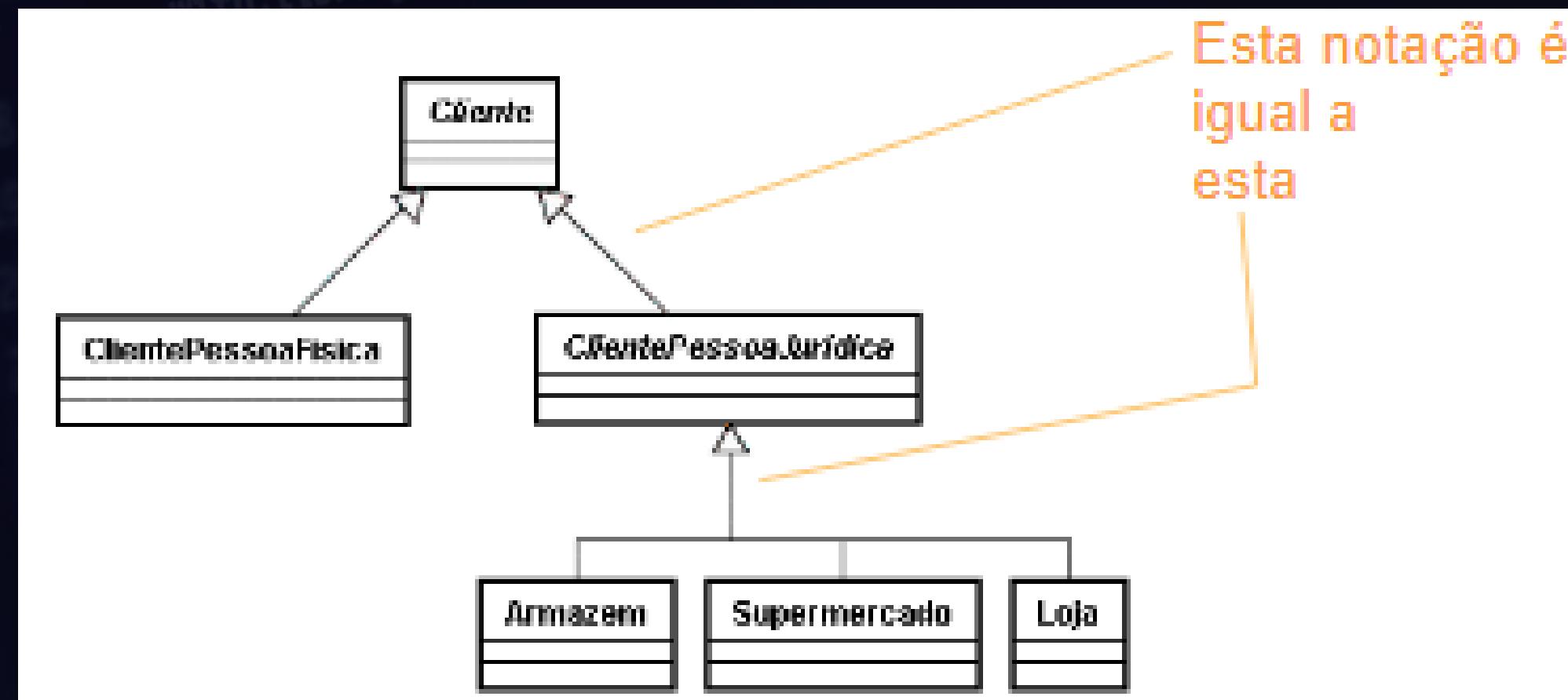
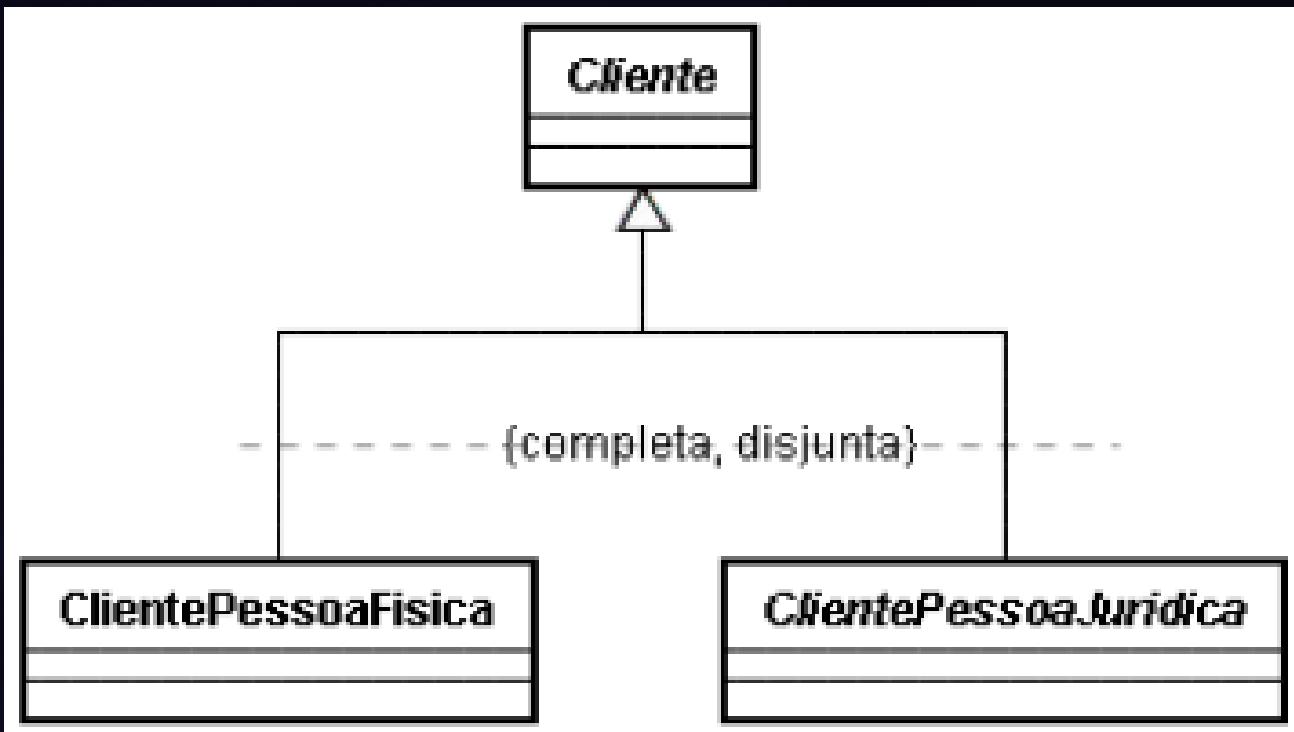


Diagrama de Classes

- 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 classe
- Ex: Atleta – (Nadador e Corredor)
- Disjunta: As subclasses só podem herdar de uma classe
- Ex: Figura geométrica – (Elipse, Quadrado, Círculo)
- 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, Círculo)



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.

Bibliografia

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