

# UML

Unified Modeling Language  
Linguagem de Modelagem Unificada

**Prof. Me André Santana**

andre.santana@ifrj.edu.br

# Planejamento

- Carga Horária: 20 horas
- Quantidade de Aulas: 5
- Aula 1: Introdução a linguagem UML e Diagrama de Caso de Uso
- Aula 2: Diagrama de Classes/Objetos
- Aula 3: Diagrama de Seqüência

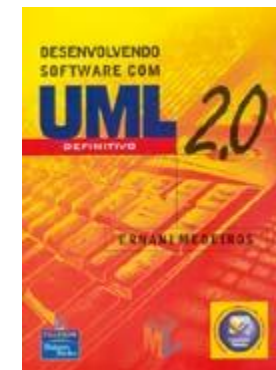
# Planejamento

- Aula 4: Diagrama de Colaboração / Estados / Atividades
- Aula 5: Desenvolvimento de atividade avaliativa

# Metodologia

- Aulas expositivas, com slides
- Abordagem da linguagem UML de forma prática
  - Um exemplo será utilizado (caso de estudo)
- Atividade final de modelagem

# Livros





# A linguagem UML

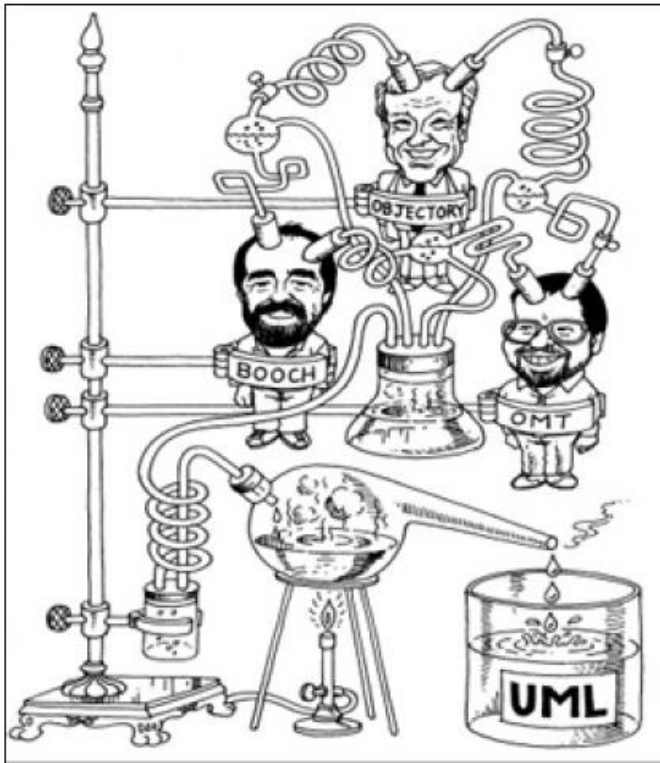
- UML (Unified Modeling Language) – Linguagem de Modelagem Unificada
- É uma linguagem de modelagem (visual), não uma linguagem de programação
- É uma linguagem de modelagem não proprietária
- Permite a utilização de diagramas padronizados para especificação e visualização de um sistema

# De onde surgiu?

- Da união de três metodologias de modelagem:
  - Método de Booch, de Grady Booch;
  - Método OMT (Object Modeling Technique) de Ivar Jacobson;
  - Método OOSE (Object Oriented Software Engineering) de James Rumbaugh.
- Os “três amigos”.



# UML



OOSE

Rumbaugh

“Fundadores” da UML

# De onde surgiu?

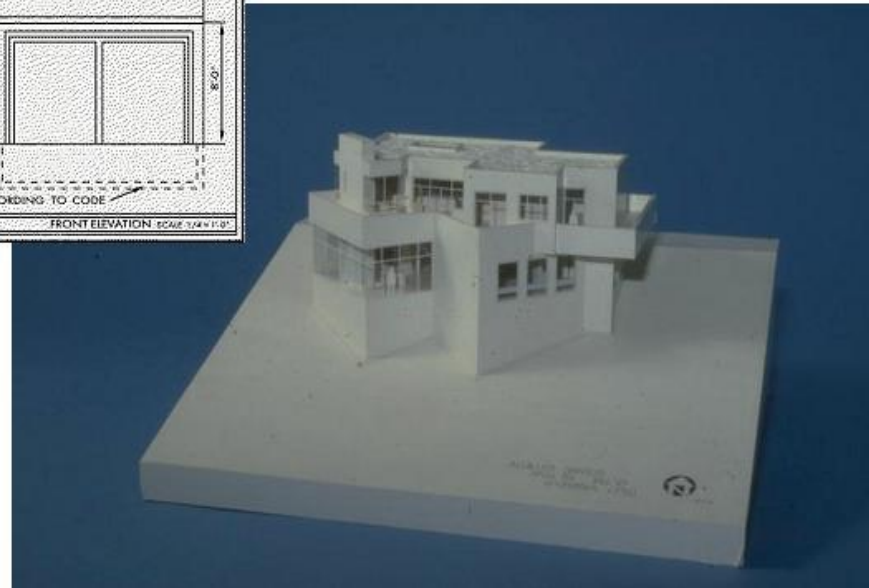
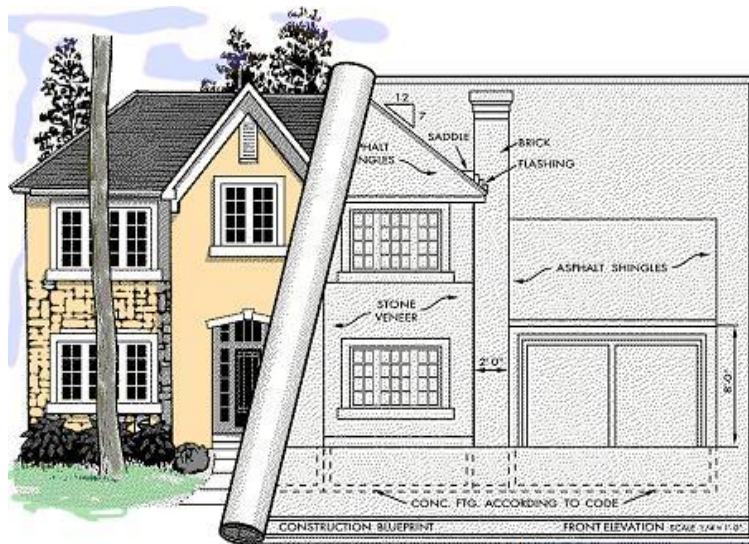
- A primeira versão foi lançada em 1996
- Em 1997 a UML foi adotada pela a OMG (Object Management Group – Grupo de gerenciamento de Objetos) como linguagem padrão de modelagem.

# O que é modelagem?

- Atividade de construir modelos que expliquem as características ou comportamentos de um sistema.
- A UML pode ser usada com todos os processos durante o ciclo de desenvolvimento do projeto
  - Análise de requisitos;
  - Análise de sistema;
  - Design;
  - Programação e
  - Testes.

# Por que usar UML?

- Desenvolver o modelo de uma aplicação antes de construí-la, é tão essencial quanto ter uma planta para a construção de uma casa.
  - Analisar o projeto sobre vários aspectos;
  - Diminui a possibilidade de erros.



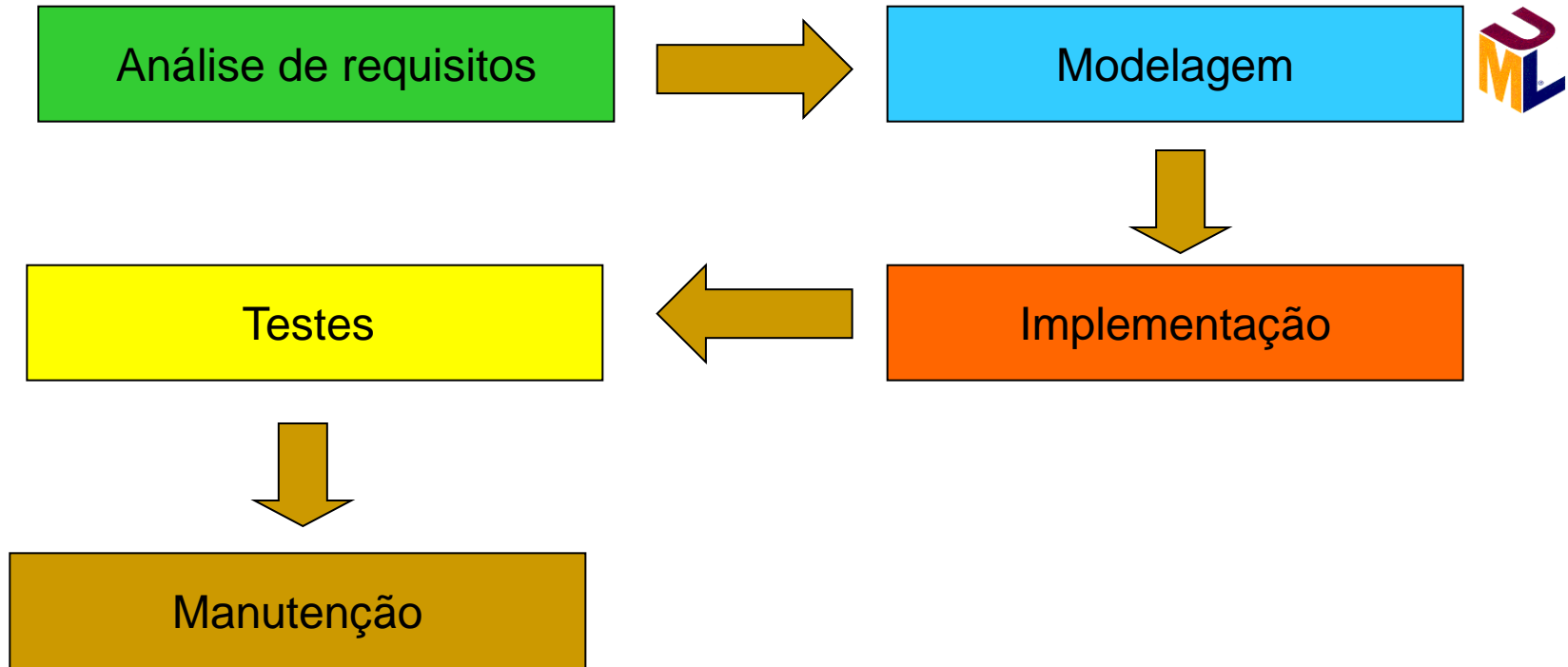
# Por que usar UML?

- Bons modelos são essenciais para a comunicação entre os times de projetos e para assegurar a beleza arquitetural.
  - Facilita a programação;
  - Todo o time entende a modelagem, facilitando assim a manutenção.

# Por que usar UML?

- Ter um rigoroso padrão de linguagem de modelagem é um fator essencial para o sucesso de um projeto.
  - Sistemas são dinâmicos;

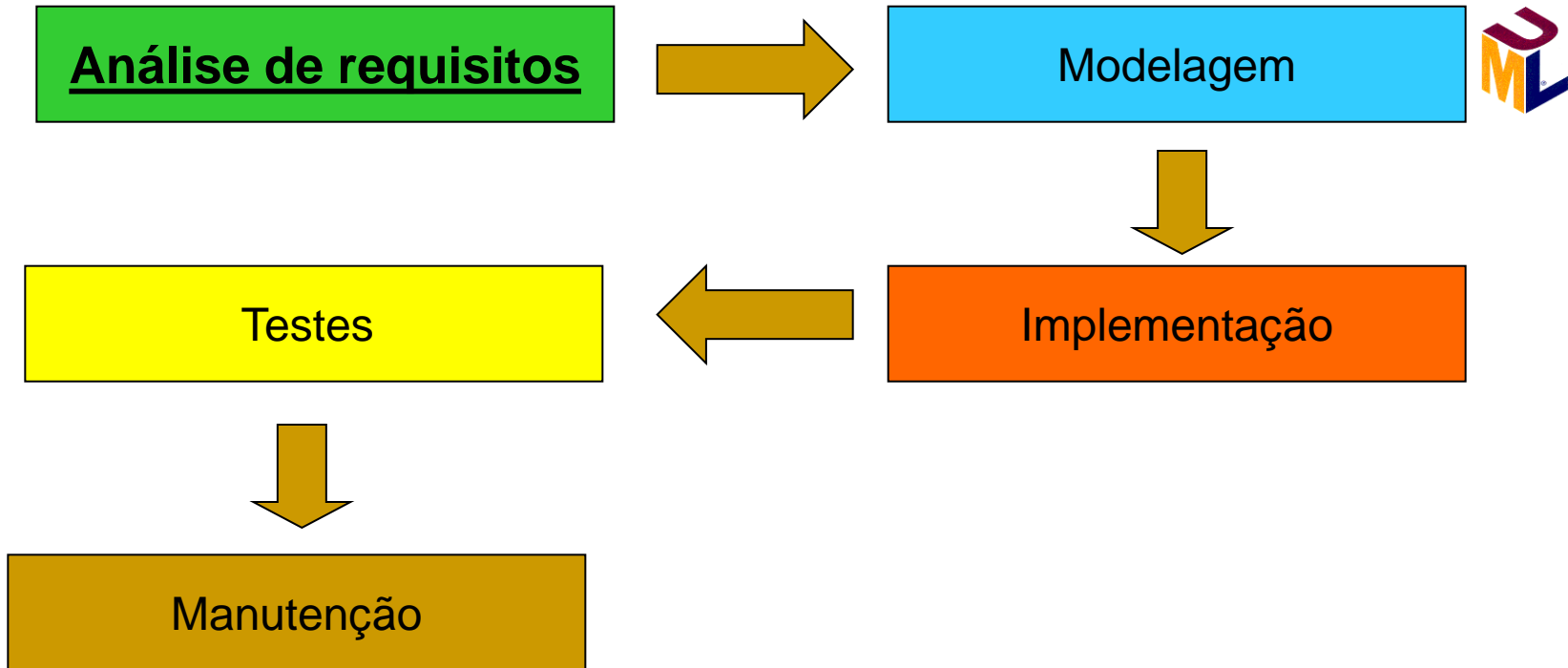
# E onde fica a modelagem?



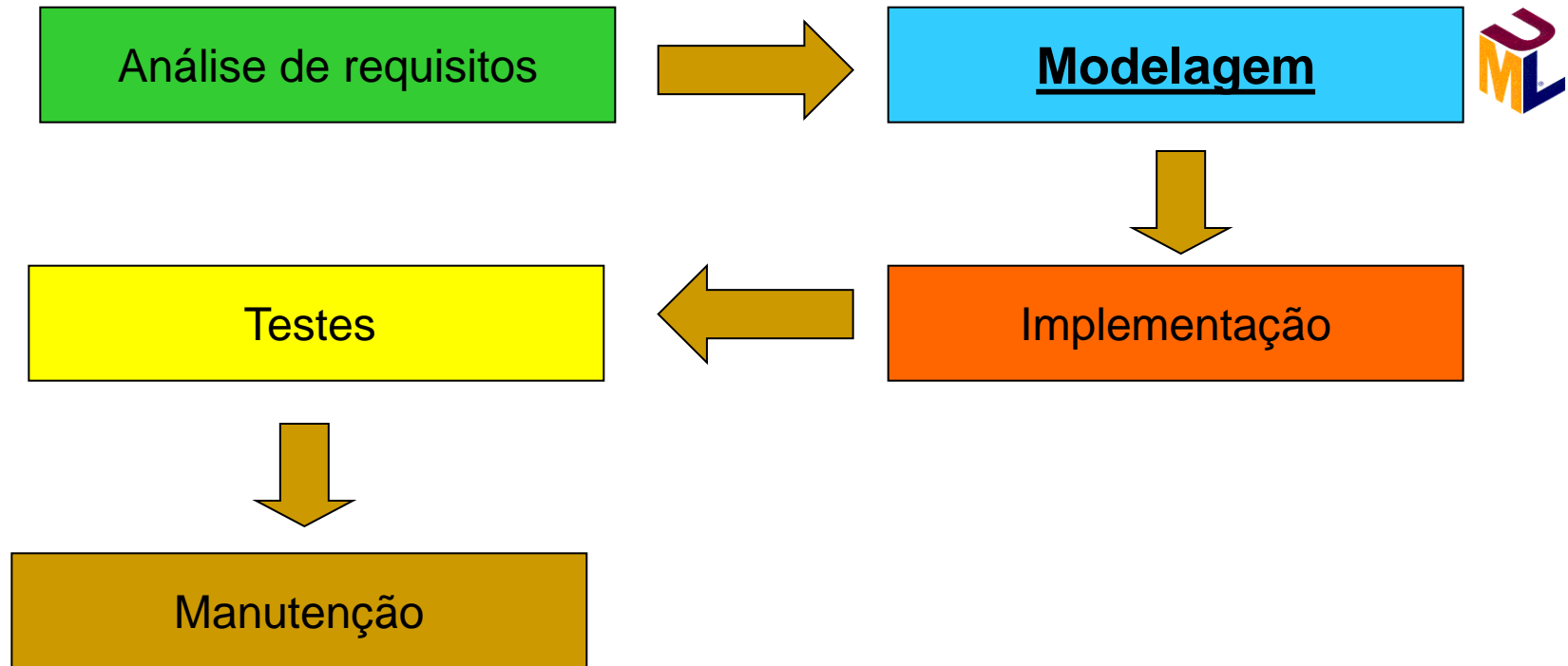
**Modelo de desenvolvimento mais comum.  
Todos os modelos são derivados dessa idéia**



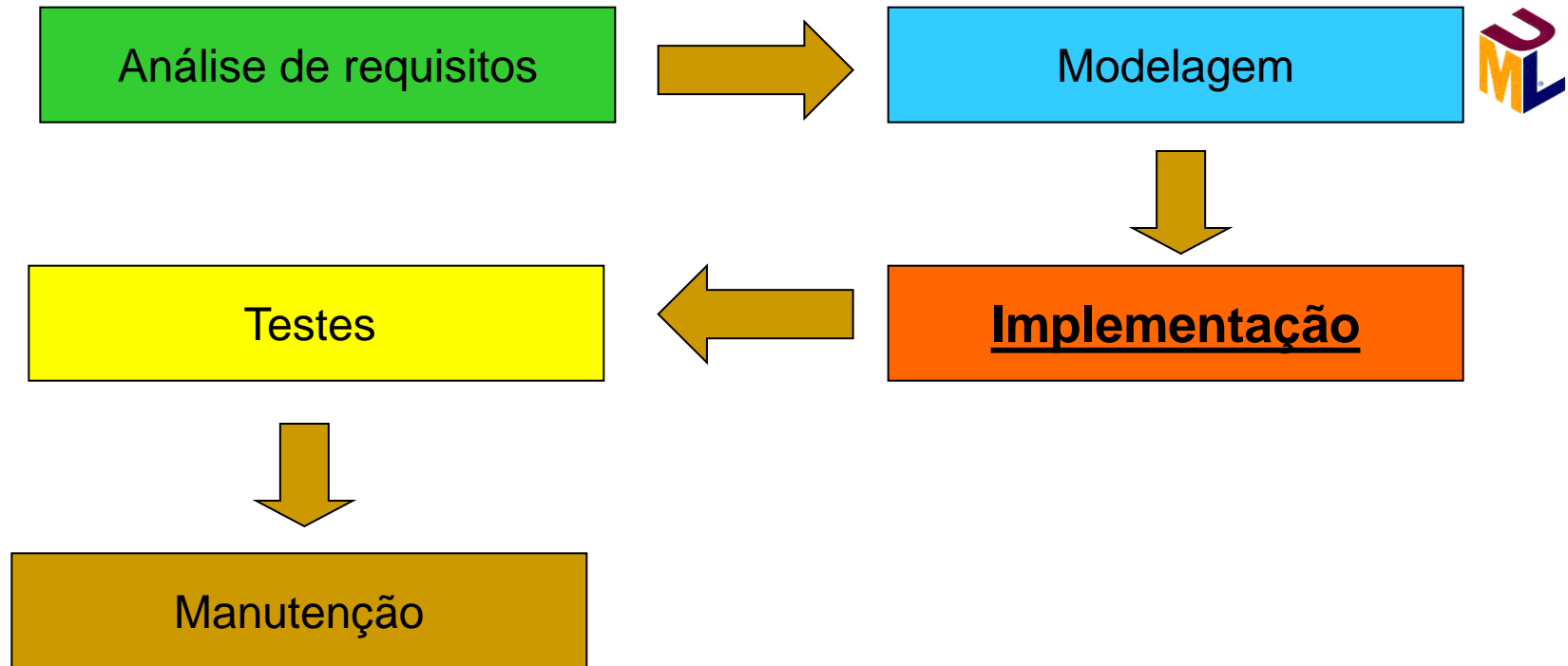
# Fases do modelo



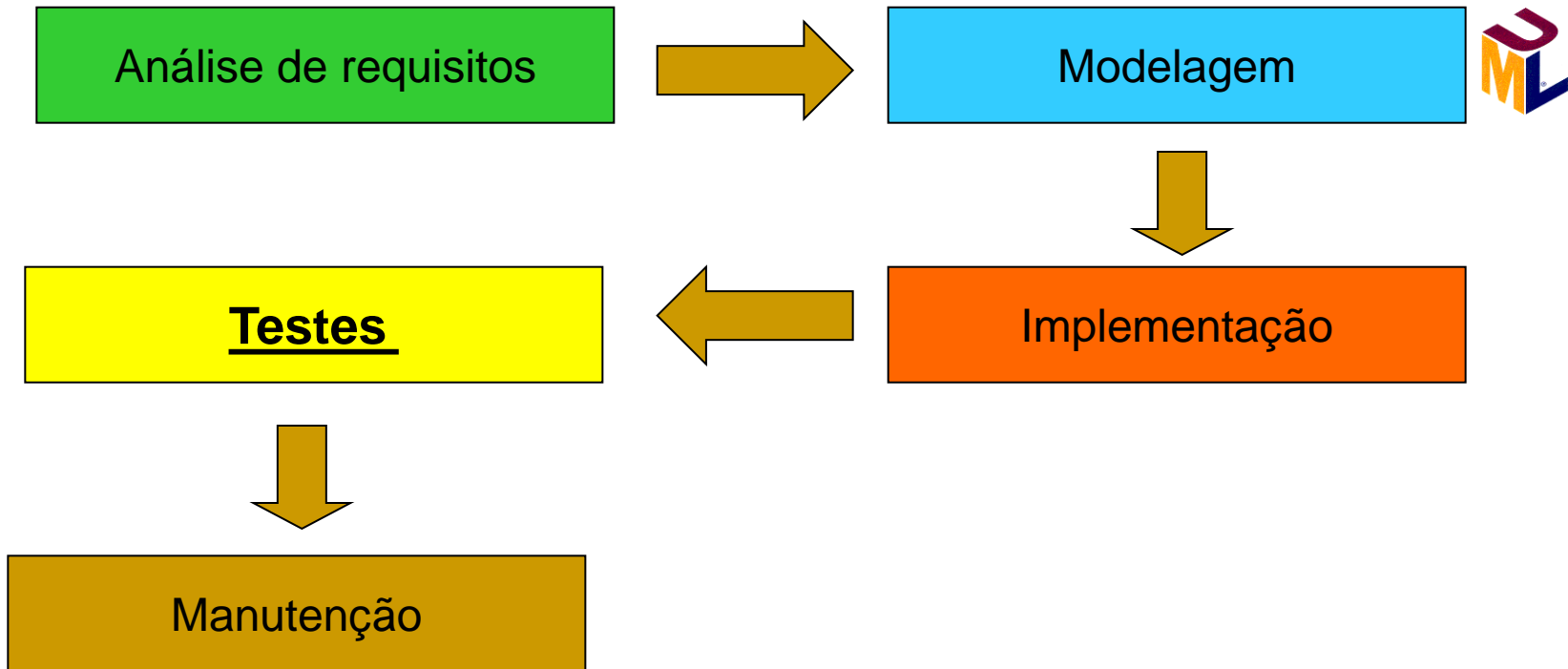
# Fases do modelo



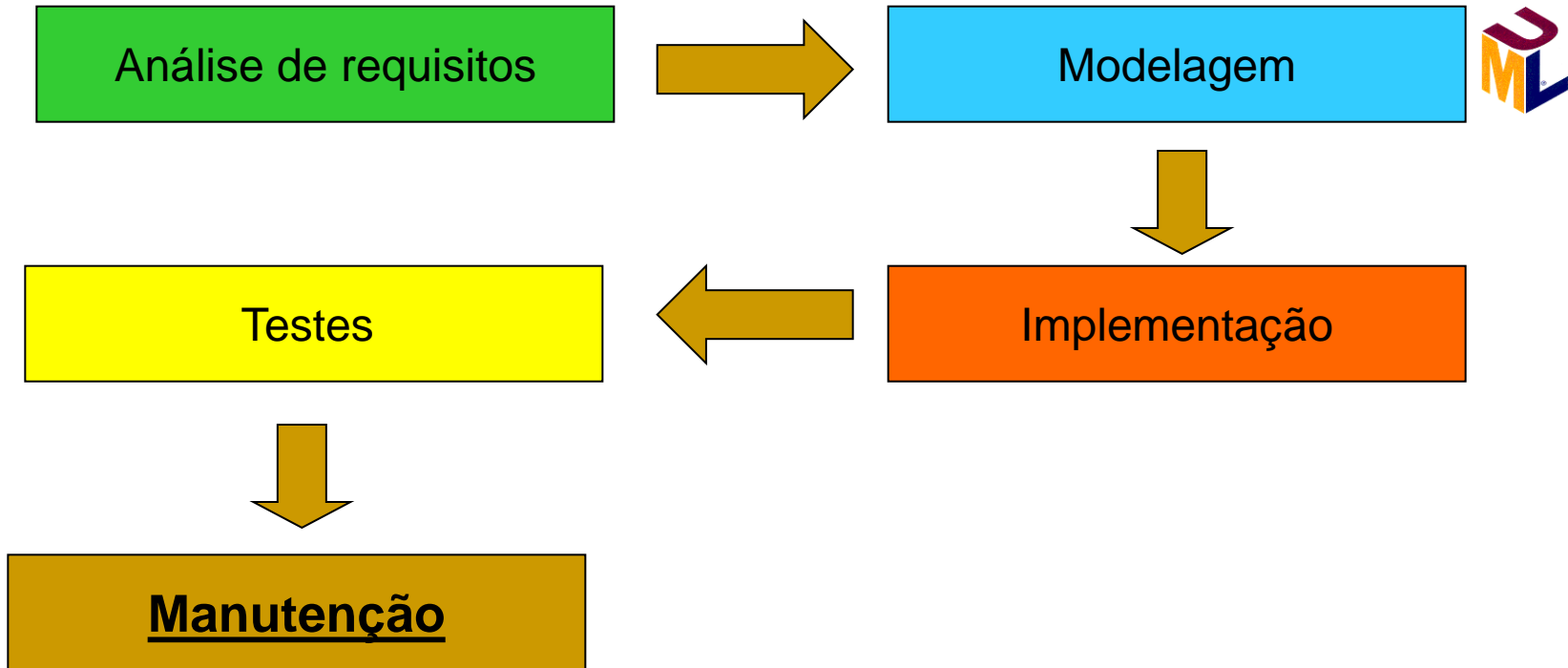
# Fases do modelo



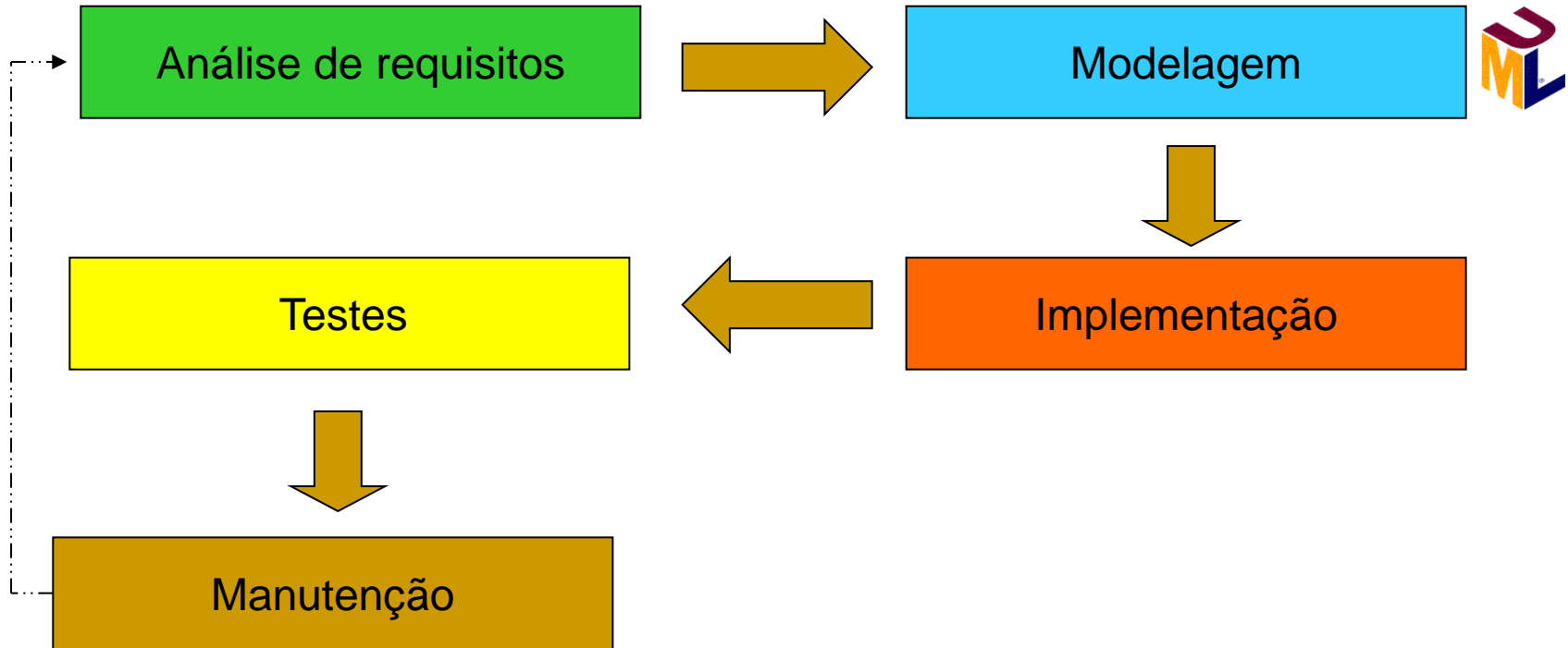
# Fases do modelo



# Fases do modelo



# Recomeçando o ciclo



# Modelos

- Tipos de Modelagens
  - Estrutural;
  - Comportamental.
- Modelos Proporcionam:
  - Visualização do sistema;
  - Especificação da estrutura ou comportamento do sistema;
  - Guia para a construção do sistema;
  - Documentação das decisões tomadas.

# Diagramas UML

## ■ Representação Gráfica de um Conjunto de Elementos.

### ■ Estrutural (Estática)

- Diagrama de Classes
- Diagramas de Objetos
- Diagrama de Caso de Uso
- Diagrama de Componentes

### ■ Dinâmica

- Diagrama de Estados
- Diagrama de Atividades
- Diagrama de Colaboração
- Diagrama de Seqüência



# Ferramentas CASE

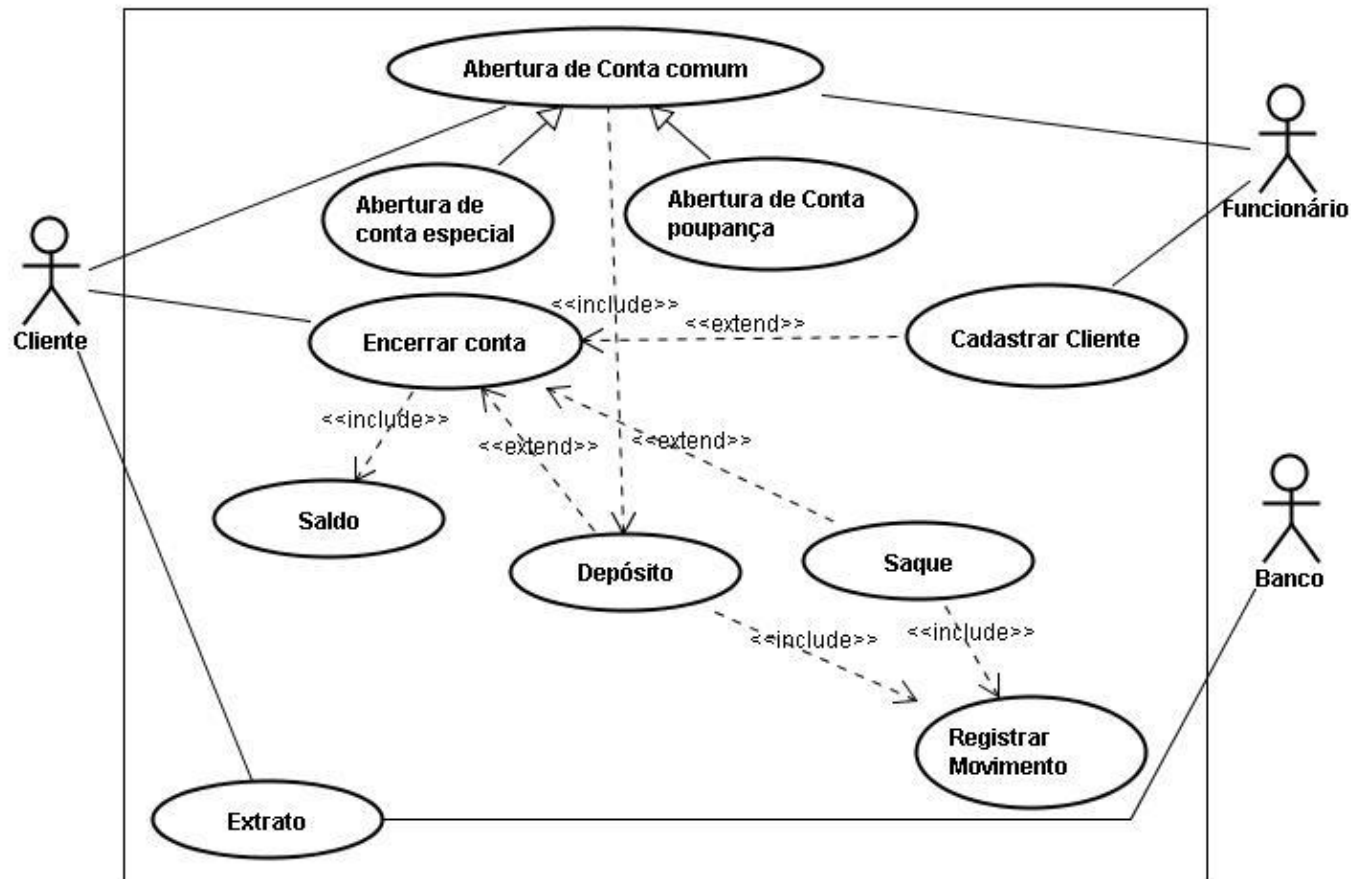
- Auxiliam na construção e gerenciamento de diagramas UML
  - ❑ Rational Rose
  - ❑ MS Visio
  - ❑ PowerDesign
  - ❑ ArgoUML
  - ❑ Jude
  - ❑ Poseidon

# Resumo dos diagramas

# Diagrama de Caso de Uso

- Diagrama mais geral da UML;
- Usado geralmente nas fases de Levantamento e Análise de Requisito do Sistema;
- Mostra como o sistema irá se comportar.

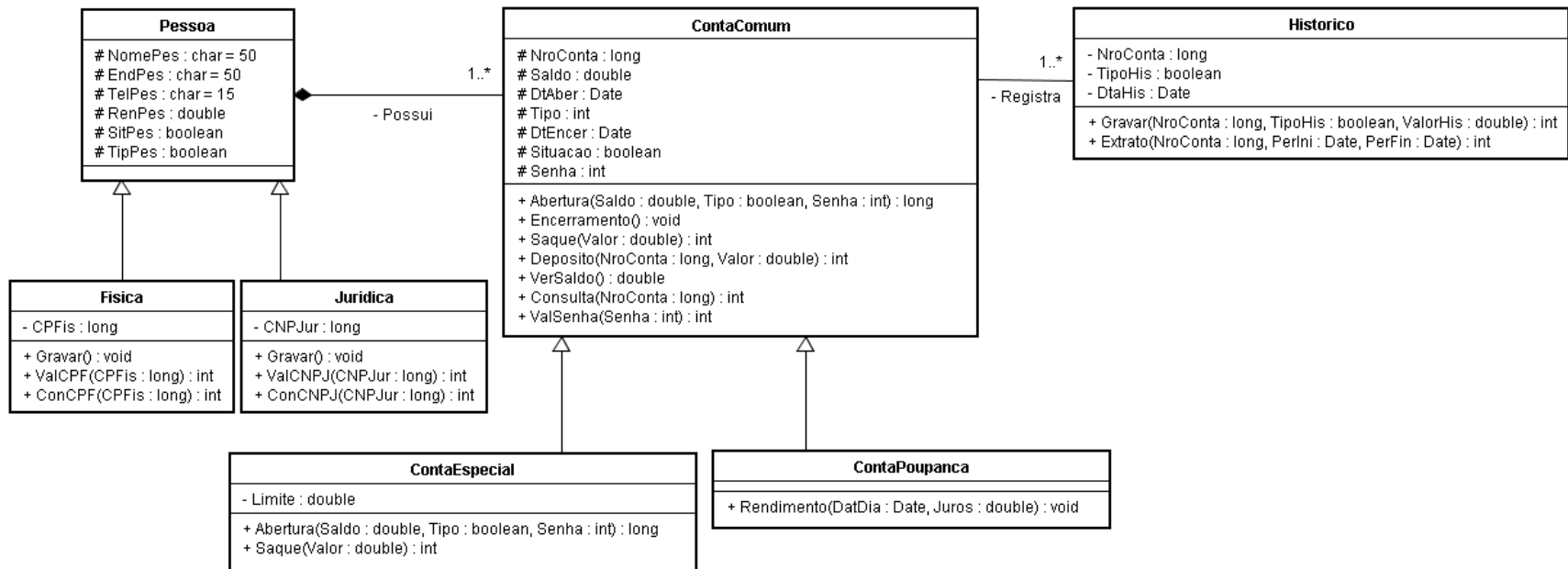
# Diagrama de Caso de Uso



# Diagrama de Classes

- Diagrama mais utilizado da UML;
- Serve de apoio para a maioria dos outros diagramas.
- Define a estrutura de classes do sistema;
- Estabelece como as classes se relacionam.

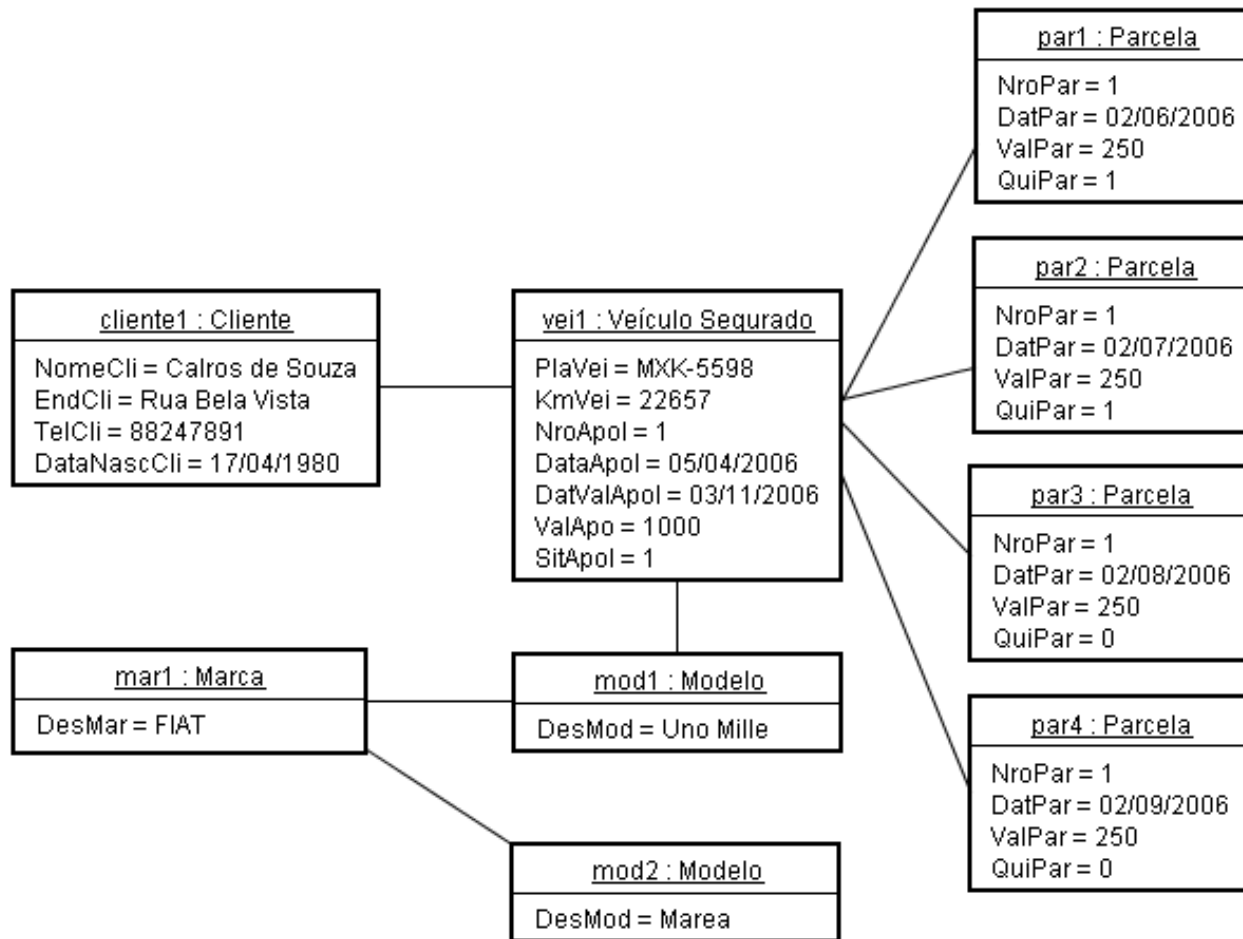
# Diagrama de Classes



# Diagrama de Objetos

- Complemento do Diagrama de Classes
- Exibe os valores armazenados pelos objetos de um Diagrama de Classes.

# Diagrama de Objetos

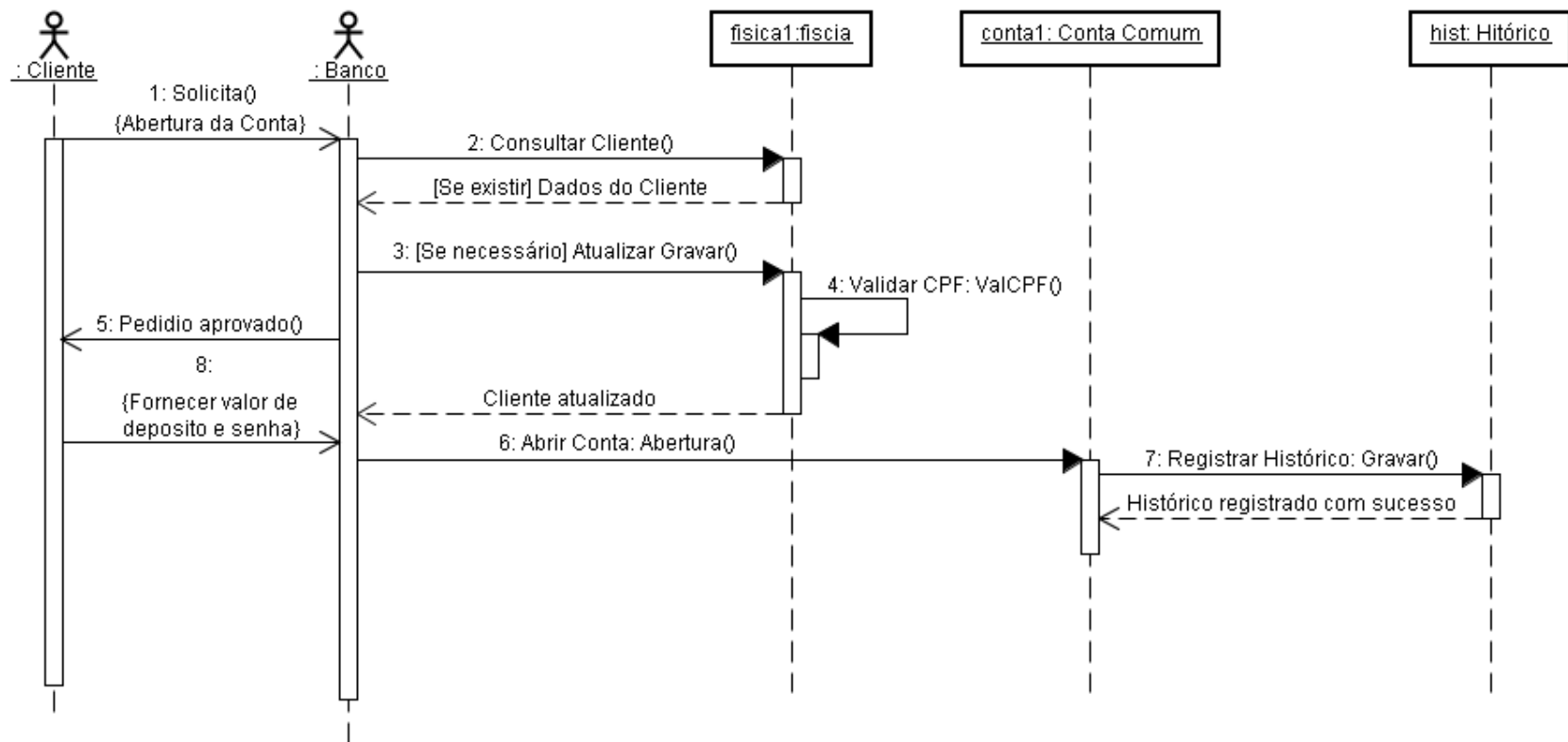




# Diagrama de Seqüência

- Preocupa-se com a ordem temporal em que as mensagens são trocadas
- Baseia-se em um Caso de Uso
- Costuma identificar o Evento gerador do processo modelado, bem como, o Ator responsável por este evento.

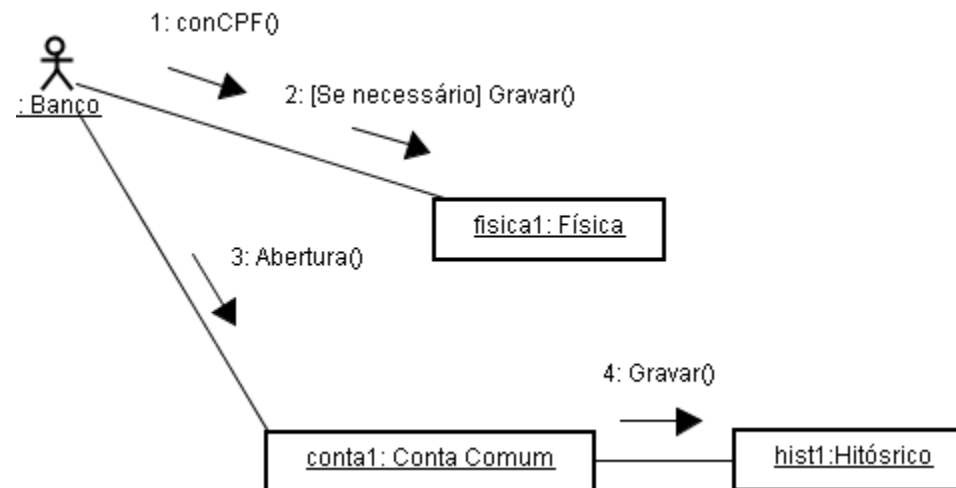
# Diagrama de Seqüência



# Diagrama de Colaboração

- Amplamente associado ao diagrama de seqüência, um complementa o outro.
- Não se preocupa com a temporalidade, mas sim, em como os objetos estão vinculados e quais mensagens trocam entre si.

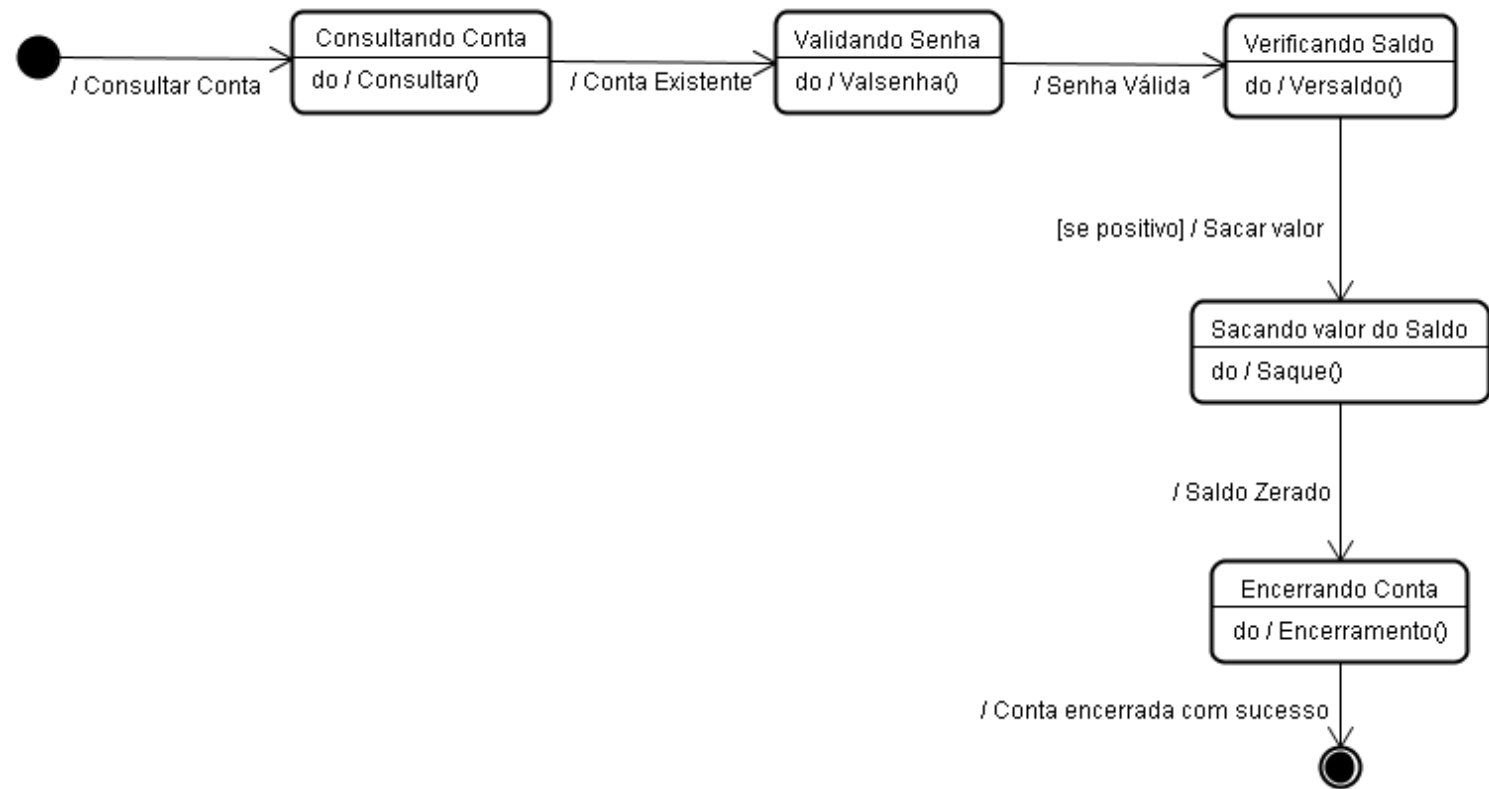
# Diagrama de Colaboração



# Diagrama de Estados

- Procura acompanhar as mudanças sofridas por um Objeto dentro de um determinado processo.
- O Diagrama de Estados é utilizado normalmente para acompanhar os estados por que passa uma instância de uma classe.

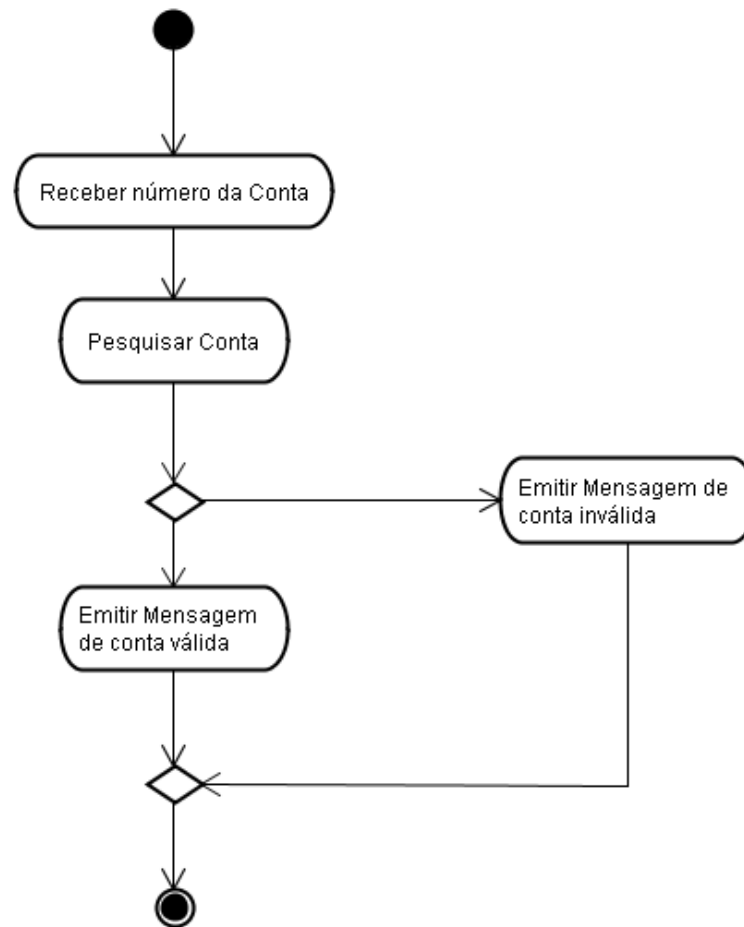
# Diagrama de Estados



# Diagrama de Atividades

- Preocupa-se em descrever os passos a serem percorridos para a conclusão de uma atividade específica.
- O Diagrama de Atividades concentra-se na representação do fluxo de controle de uma atividade

# Diagrama de Atividades

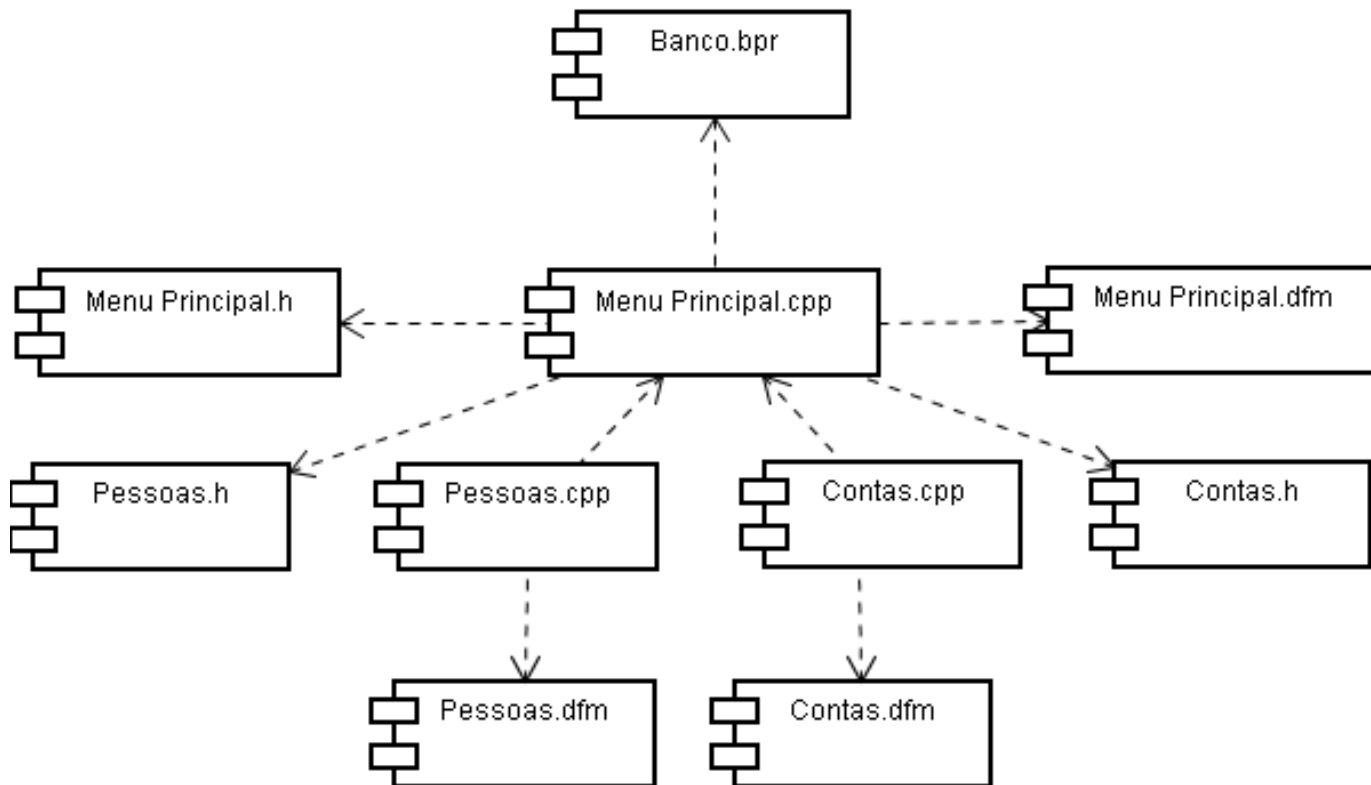




# Diagrama de Componentes

- Amplamente associado a linguagem de programação que será utilizada para desenvolver o sistema modelado.
- Este diagrama representa os componentes do sistema quando este for implementado em termos de módulos de código-fonte, bibliotecas, arquivos de ajuda, módulos executáveis, etc.

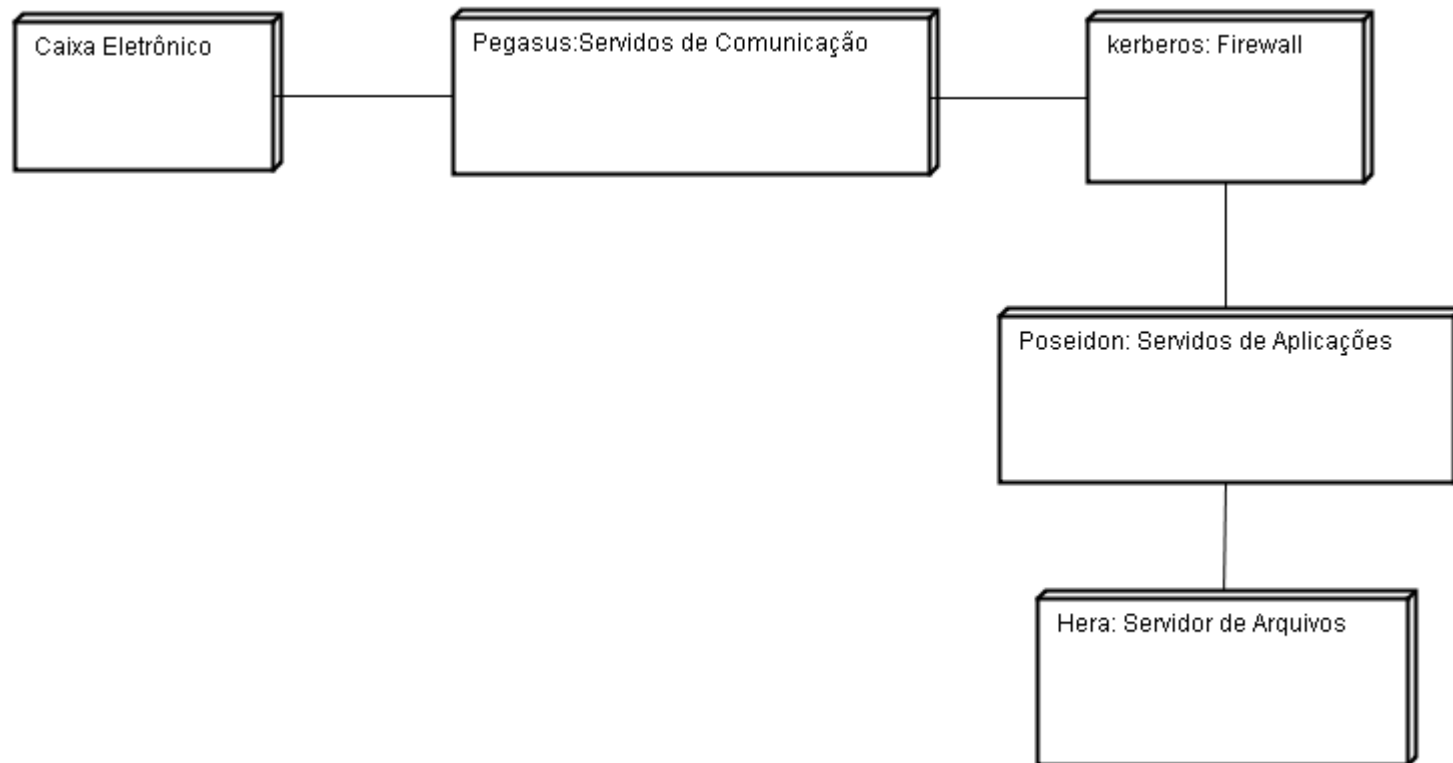
# Diagrama de Componentes



# Diagrama de Implantação

- Determina as necessidades de hardware do sistema, as características físicas como servidores, estações, topologias e protocolos de comunicação, ou seja, todo o aparato físico sobre o qual o sistema deverá ser executado.

# Diagrama de Implantação



# Outros diagramas

- Diagrama de Pacotes

- Tem por objetivo representar os sub-sistemas englobados por um sistema de forma a determinar as partes que o compõem.

- Diagrama de Interação Geral

- Fornece uma visão geral dentro de um sistema ou processo de negócios

- Diagrama de Tempo

- Descreve a mudança no estado ou na condição de uma instância de uma classe ou seu papel durante o tempo.

# Diagrama de Casos de Uso

Interagindo com o Usuário

# Diagrama de Casos de Uso

- Procura, por meio de uma linguagem simples, possibilitar a compreensão do comportamento externo do sistema por qualquer pessoa, tentando apresentar o sistema através de uma perspectiva do usuário.

# Diagrama de Casos de Uso

- Dentre todos os diagramas da UML, é o mais abstrato e, portanto o mais flexível e informal.
- Geralmente é modelado no início da modelagem do sistema, ainda nas etapas de levantamento e análise de requisitos.
  - O que é modelado primeiro.



# Diagrama de Casos de Uso

- Tem por objetivo apresentar uma visão externa geral das funções e serviços que o sistema deverá oferecer ao usuário.
  - Sem se preocupar como essas funções serão implementadas.
- Um caso de uso descreve, as operações que o sistema deve cumprir para cada usuário.
  - Irá existir um caso de uso para cada tarefa que o sistema deve executar.

# Diagrama de Casos de Uso

- No entanto, Um caso de uso não diz como o sistema FAZ determinada tarefa, apenas o que o sistema FAZ, deixando para outros diagramas essa tarefa.

# Componentes do Diagrama de Casos de Uso

- O Diagrama de Casos de Uso concentra-se em dois itens principais:
  - Atores
  - Casos de Uso

# Atores

- Casos de Uso descrevem interações entre o sistema e os **atores**.
- Os atores representam os papéis desempenhados pelos diversos usuários que poderão de alguma forma interagir com o sistema.
  - Pode ser também um hardware especial ou mesmo outro sistema que interaja com o software.

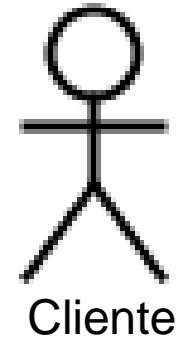
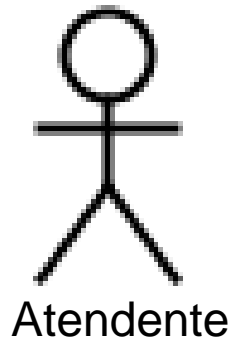
# Atores

- Dessa forma, o Ator é algo (usuário, outros sistema ou hardware), que não faz parte do sistema mas que interage em algum momento com ele.

# Atores

- Atores são representados por símbolos de “bonecos magros”, contendo uma breve descrição logo abaixo do seu símbolo que identifica qual o papel que o ator em questão assume dentro do diagrama.

# Exemplos de Atores



# Casos de Uso

- Os Casos de Uso referem-se aos serviços, tarefas ou funções que podem ser utilizadas de alguma maneira pelos usuários do sistema. Por exemplo:
  - ❑ Cadastrar uma venda;
  - ❑ Solicitar um saque de uma conta bancária;
  - ❑ Consultar um filme em uma locadora;
  - ❑ Etc.



# Representação dos Casos de Uso

- Os casos de uso são representados por elipses contendo dentro de si um texto descrevendo a que serviço o Caso de Uso se refere.
  - Não existe limites para descrever um Caso de uso;
  - Mas geralmente essa descrição dentro da elipse costuma ser sucinta.

# Exemplos de Casos de Uso



**Locação de Filmes**

**Consultar Gêneros**

**Cadastro de Clientes**

# Documentação de Casos de Uso

- Costuma descrever por meio de uma linguagem bastante simples, a função em linhas gerais do Caso de Uso.
  - Quais atores interagem com o mesmo;
  - Quais etapas devem ser executadas pelo Ator e pelo sistema para que o Caso de Uso execute sua função;
  - Quais parâmetros devem ser fornecidos e quais restrições e validações o Caso de Uso deve possuir.

# Documentação de Casos de Uso

- Não existe um formato específico.
  - ❑ Descrição passo a passo;
  - ❑ Através de tabelas;
  - ❑ Pseudo-código;
  - ❑ Até mesmo através de uma linguagem de programação, mesmo que fuja bastante do objetivo principal do Diagrama de Casos de Uso.

Nome do Caso de Uso	Abertura de Conta
Ator Principal	Cliente
Atores Secundários	Funcionário
Resumo	Este caso de Uso, descreve as etapas percorridas por um cliente para abrir uma conta corrente.
Pré-Condições	O pedido do cliente precisa ser aprovado
Pré-Condições	É necessário um depósito inicial
Ações do Ator	Ações do Sistema
1. Solicitar a abertura da conta	1. Consultar o saldo da conta
	2. Consultar o saldo da conta
	3. Se for necessário Gravar ou atualizar o cadastro do Cliente
	4. Avaliar o pedido
	5. Aprovar ou Reprovar o pedido
6. Escolher uma Senha para a conta	
	7. Abrir a conta
8. Informar o valor do depósito	
	9.Registrar o depósito
	10. Solicitar o cartão da compra

# Retirar dinheiro no Caixa Eletrônico

- O Cliente introduz o cartão no caixa eletrônico;
- O Sistema disponibiliza várias opções;
- O Cliente aperta o botão saque;
- O Cliente escolhe o tipo de conta:
  - Poupança;
  - Conta Corrente.
- O Cliente entra com o valor do saque;
- Em seguida o cliente informa a senha;
- O sistema verifica a senha e saldo em seu Banco de dados;
- O Caixa eletrônico libera o dinheiro para o usuário.

# Associações

- As associações representam as interações ou relacionamentos entre:
  - Os Atores que fazem parte do Diagrama;
  - Os Atores e os Casos de Uso e
  - Os Casos de Uso com outros Casos de Uso.
- Os relacionamentos entre os Casos de Uso, recebem um nome especial.
  - Inclusão;
  - Extensão e
  - Generalização.

# Associações

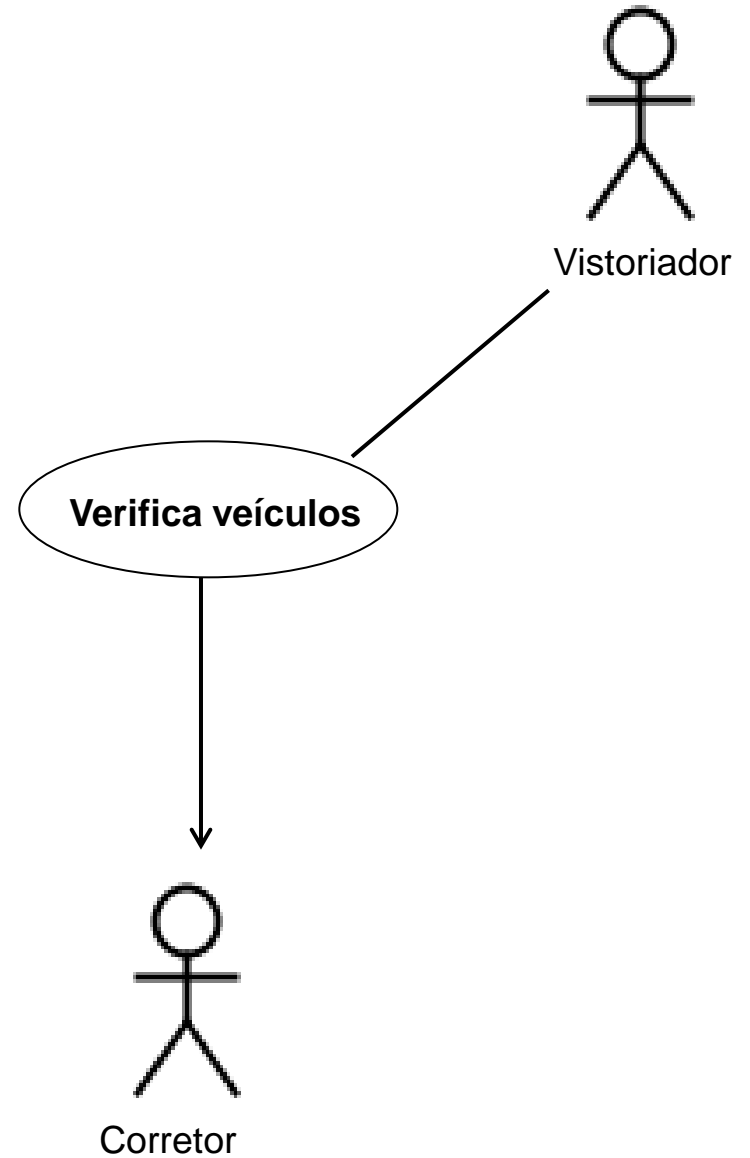
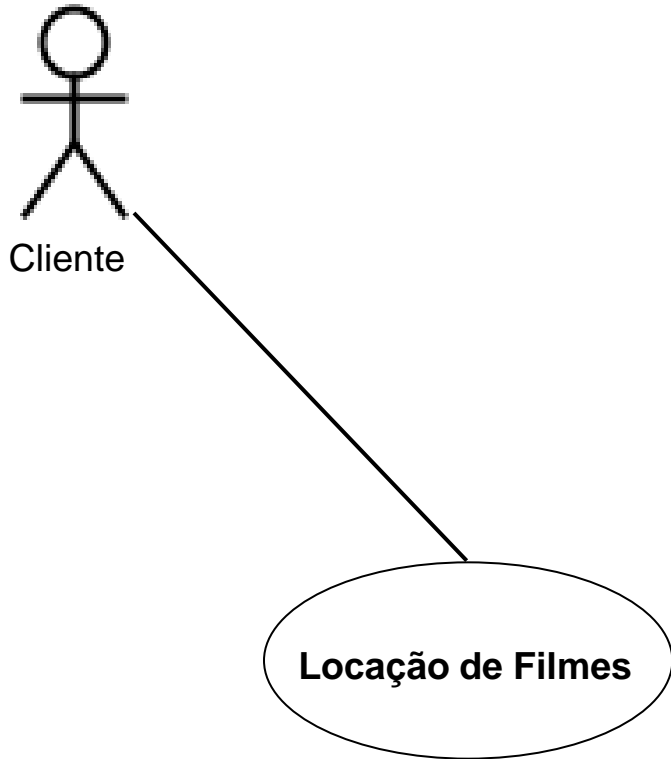
- Uma associação entre um Caso de Uso e um Ator demonstra que o Ator utiliza-se de alguma maneira, da função do sistema representada pelo Caso de Uso,
  - Seja requisitando a execução daquela função;
  - Seja recebendo o resultado produzido por ela a pedido de outro Ator.



# Associações

- A Associação entre um Ator e um Caso de Uso é representada por uma reta ligando o Ator ao Caso de Uso, podendo ocorrer nas que as extremidades da reta contenha **setas**, indicando a navegabilidade da Associação, demonstrando assim o sentido em que as informações trafegam.
- Quando a informação é transmitida nos dois sentidos, a reta passa a não possuir setas.

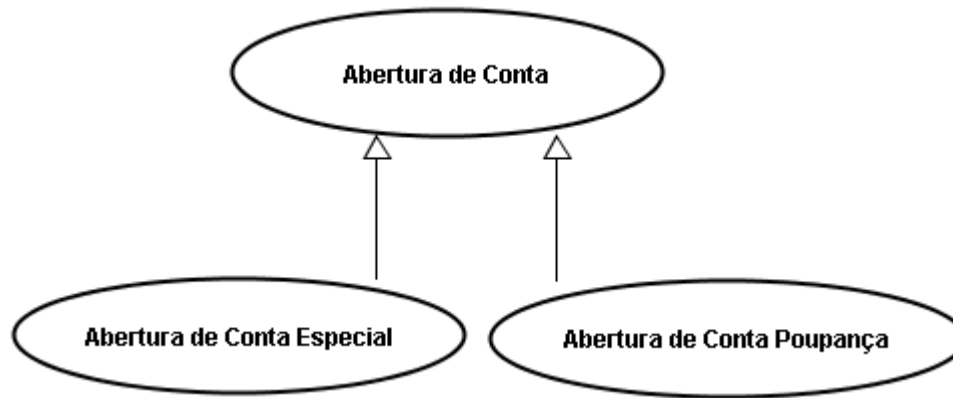
# Associações



# Especialização / Generalização

- Acontece quando dois ou mais Casos de uso possui características semelhantes, apresentando pequenas diferenças entre si.
- Dessa forma é importante definir um Caso de Uso Geral que descreve as características compartilhadas por todos os Casos de Uso em questão e então relacioná-los.

# Exemplos de Especialização / Generalização



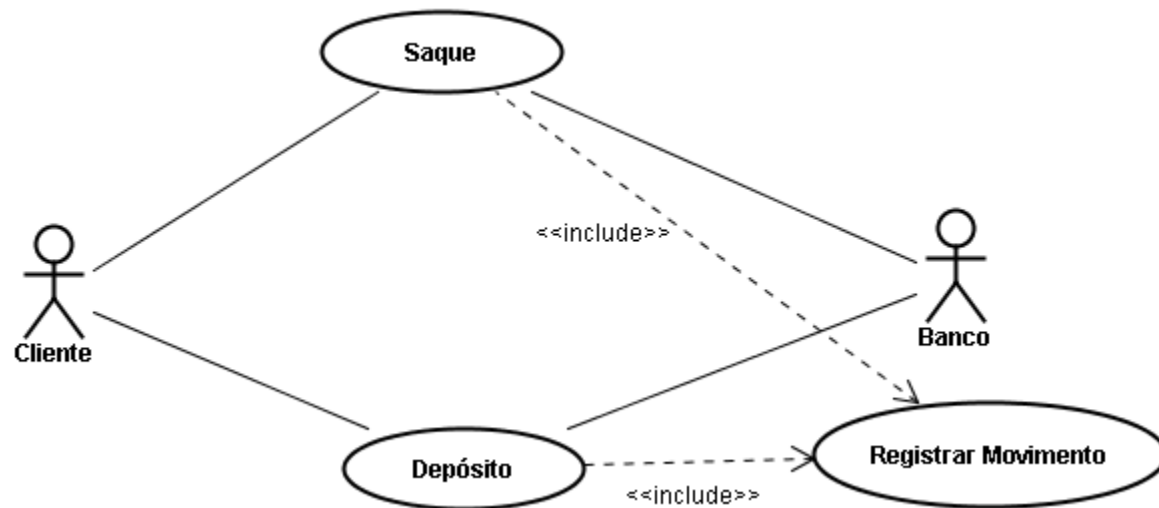
# Inclusão

- Costuma ser utilizada quando existe um serviço, situação ou rotina comum a mais de um Caso de Uso.
- Os relacionamentos de Inclusão indicam uma obrigatoriedade, ou seja, quando um determinado Caso de Uso possui um relacionamento de Inclusão com outro, a execução do primeiro obriga também a execução do segundo.

# Inclusão

- Uma Associação de Inclusão é representada por uma reta tracejada com uma seta em uma das extremidades que aponta para o Caso de Uso incluído.
  - Possuir a expressão “include”, entre dois sinais de menor (<) e dois sinais de maior (>).

# Inclusão



# Extensão

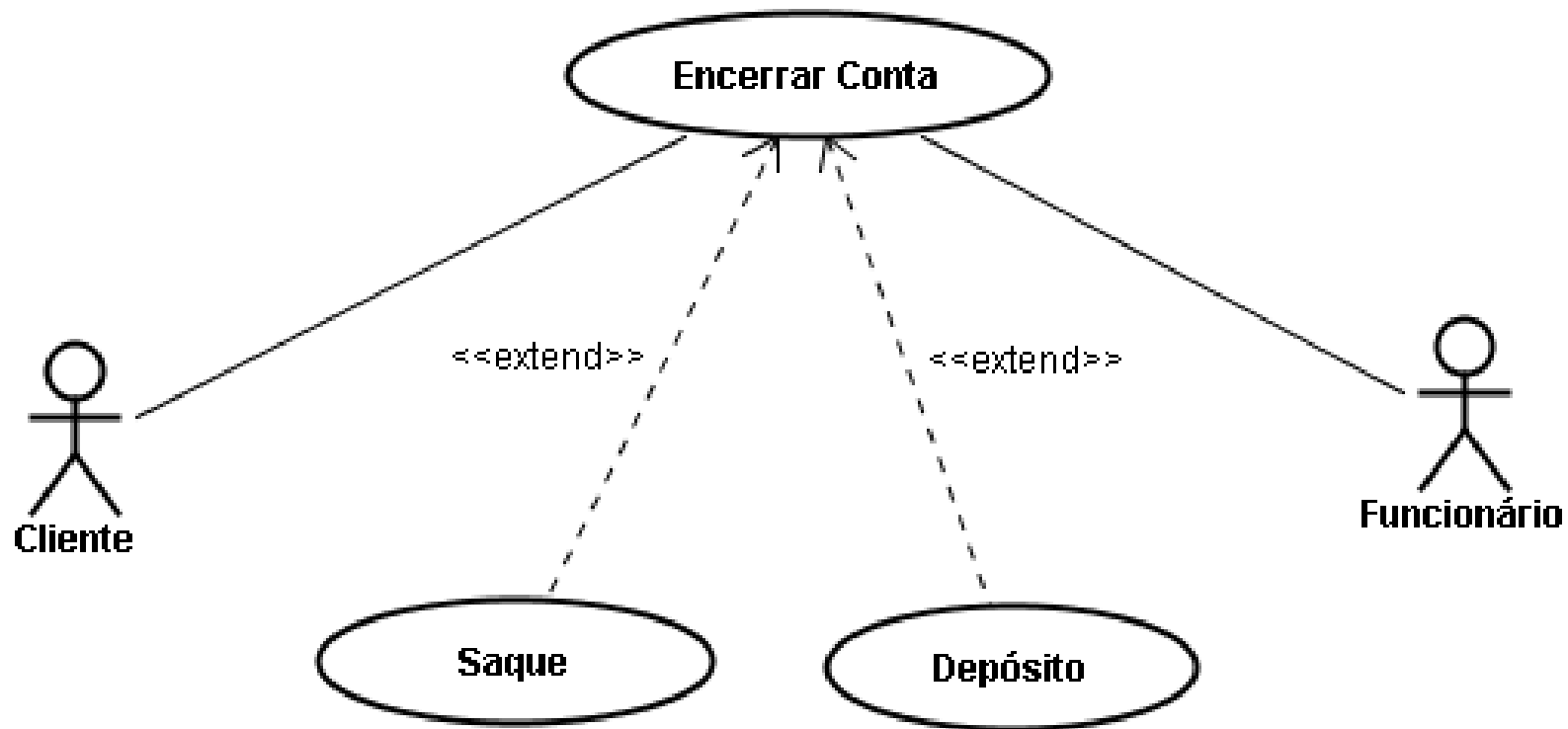
- Descreve cenários opcionais de um Caso de Uso.
  - Os Casos de uso estendidos descrevem cenários que somente acontecerão em uma situação específica, se uma determinada situação for satisfeita.
  - Dessa forma as Associações de Extensão necessita de um teste determinar se o Caso de Uso estendido será executado ou não.



# Extensão

- Em sua representação gráfica, é muito semelhante às associações de Inclusão.
  - Possuir a expressão “extend”, entre dois sinais de menor (<) e dois sinais de maior (>).

# Extensão



# Exercício 1

Desenvolva um Diagrama de Casos de Uso para um sistema de Vídeo Locadora equivalente ao módulo de locação de DVD's, de acordo com as afirmações abaixo:

- Ao realizar uma locação, o Cliente deve primeiro informar seu código para que o Atendente verifique se o mesmo já está cadastrado, se o Cliente não estiver cadastrado, então a locação deverá ser recusada e o Cliente deverá ser informado como proceder para se cadastrar.

# Exercício 1

- Caso o Cliente já esteja cadastrado, o Atendente deverá verificar se o Cliente já entregou todos os DVD's alugados anteriormente, se não a Locação será recusada.
- É de responsabilidade do Atendente o cadastro dos Filmes, registrando novos filmes adquiridos pela locadora.

# Exercício 2

Desenvolva o Diagrama de Caso de uso para um sistema de cursos de informática equivalente ao módulo de matrícula de acordo com os seguintes fatos:

- O Aluno primeiramente solicita informações ao Atendente sobre quais cursos a empresa oferece. Se o Aluno se interessar por algum curso, pedirá informações a respeito de quais turmas do curso em questão s encontram em aberto, qual o horário em que as aulas serão ministradas, data de início, número mínimos de alunos para que uma turma inicie.

## Exercício 2

- Caso o horário de alguma turma seja compatível com os horários do Aluno, este realizará a Matrícula em uma turma relativa ao curso em que se interessou. Caso o aluno nunca tenha feito nenhum curso na empresa e portanto não seja cadastrado, o Aluno deverá ser cadastrado antes de realizar a matrícula.

# Exercício 3

- Desenvolva o Diagrama de Caso de uso para um sistema de controle de apólice de seguros de acordo com os seguintes fatos:
  - Irá existir um cadastro de clientes e um cadastro de veículo, onde o cliente fornece as informações necessárias para que o corretor possa inserir no sistema.
  - Com relação ao veículo, um vistoriador analisa o veículo e informa ao corretor a situação do mesmo.
  - Em seguida o corretor consulta a Matriz, para saber valores e condições do seguro.
  - Logo que receber os valores da apólice, o corretor os repassa para o cliente, para que este decida, a quantidade de parcelas que deseja pagar a apólice.

# Exercício 3

- ❑ Assim que a apólice for gerada, será inserida automaticamente as parcelas a receber.
- ❑ Existirá também um controle de Sinistros, onde o Ator fornece as informações iniciais sobre o sinistro a secretária, que por sua vez insere os dados informador no sistema.
- ❑ Então o Vistoriador irá analisar a situação do veículo, que poderá acrescentar e/ou modificar as informações do sinistro.
  - Obs.: O mesmo ator poderá aparecer duas ou mais vezes no diagrama, para que o diagrama não fique poluído demais com os cruzamentos.



# Diagrama de Classes

A estrutura do projeto

# Diagrama de Classes

- É com certeza o mais importante e o mais utilizado diagrama da UML.
- Permite a visualização das classes que comporão o sistema com seus respectivos atributos e métodos, bem como os relacionamento entre as classes.

# Diagrama de Classes

- Apresenta uma visão estática de como as Classes estão organizadas;
- Preocupação apenas com a estrutura lógica.
- Serve como base para outros diagramas da UML.

# Persistência

- Em muitos casos é necessário preservar de forma permanente os objetos de uma Classe.
  - A Classe precisa ser Persistente.
- Uma Classe Persistente apresenta muitas semelhanças com uma entidade como as definidas no MER.
  - Modelo utilizado para definir as tabelas em banco de dados Relacional.

# Persistência

- Deve ficar claro que nem toda Classe é persistente, não sendo muitas vezes necessário preservar (armazenar) suas informações.

# Classes, Atributos e Métodos

- Classes costumam possuir atributos e atributos armazenam os dados dos Objetos da Classe.
- Métodos que são as funções que uma instância da Classe pode executar.

# Atributos

- Os valores dos Atributos podem variar de instância para instância.
  - É exatamente essa característica, que permite a identificação de cada Objeto.

# Atributos

- Cada atributo deverá conter um tipo de dados, ou seja a forma como a informação deverá ser armazenada.
  - Byte:
    - Tamanho em bits: 8
    - Faixa de valores: -128 a 127
  - Boolean:
    - Tamanho em bits: 8
    - Faixa de valores: true ou false
  - Int:
    - Tamanho em bits: 32
    - Faixa de valores: -2.147.482.648 a 2.147.843.467
  - Long:
    - Tamanho em bits: 64
    - Faixa: -9.223.372.036.854.775.802 a +9.223.372.036.854.775.802
  - Double:
    - Tamanho em bits: 64
    - Faixa: -1.79769313486231570E+308 a +1.79769313486231570E+308



# Atributos

- ❑ Char:
  - Texto.
- ❑ Date:
  - Data.

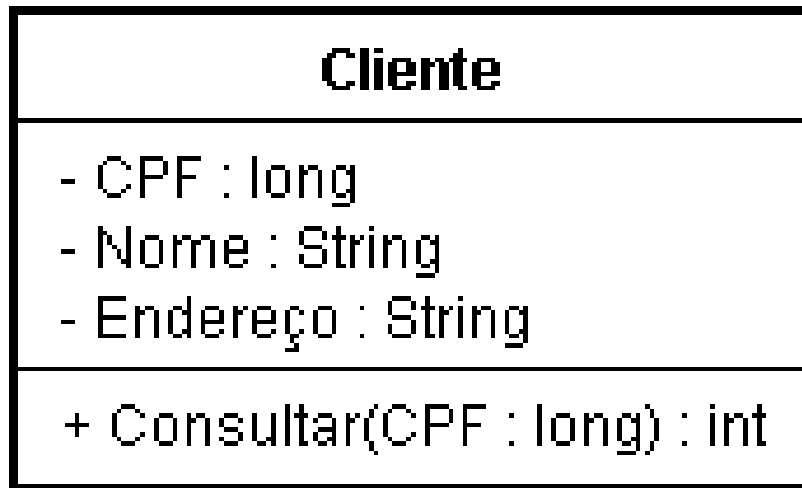
# Métodos

- Embora os Métodos sejam declarados no Diagrama de Classes, não é uma preocupação desse Diagrama, definir as etapas que estes métodos deverão percorrer quando forem chamados.
- Essa função é atribuída a outros Diagramas como:
  - Diagrama de Seqüência e
  - Diagrama de Atividades

# Representação de uma Classe

- Como já mostrado, anteriormente, uma Classe é representado por um retângulo com três divisões:
  - Na primeira parte → Nome da Classe;
  - Na segunda parte → Os Atributos da Classe;
  - Na terceira parte → Os Métodos da Classe.

# Representação de uma Classe



← Classe

← Atributos

← Métodos

# Tipos de visibilidade

## ■ Visibilidade Pública

- O atributo ou método que possuir essa visibilidade pode ser utilizado por qualquer Classe.
  - Símbolo (+), sinal de mais.

## ■ Visibilidade Protegida

- O atributo ou método que possuir essa visibilidade somente a classe possuidora ou as sub-classes terão acesso.
  - Símbolo (#), sustenido.

## ■ Visibilidade Privada

- Somente a Classe possuidora desse atributo ou método poderá utilizá-lo.
  - Símbolo (-), sinal de menos.

# Relacionamento

- As Classes costumam possuir relacionamento entre si, com o intuito de compartilhar informações e colaborarem umas com as outras para permitir a execução dos diversos processos executados pelo sistema.

# Associações

- Descreve um vínculo que ocorre normalmente entre duas Classes, chamado neste caso de **Associação Binária**.
- Em uma Associação determina-se que as instâncias de uma Classe estão de alguma forma ligadas às instâncias das outras Classes.

# Multiplicidade

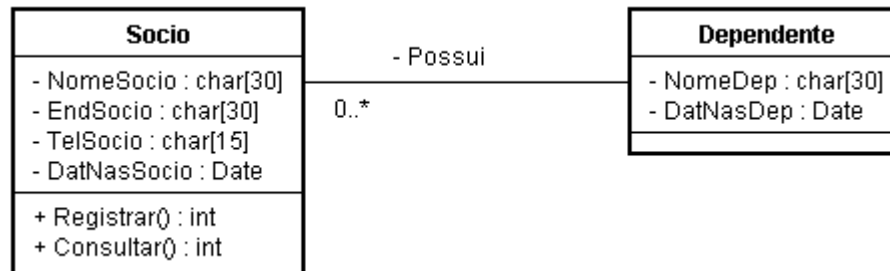
0..1	No mínimo zero (nenhum) e no máximo um. Indica que os Objetos da classe associada não precisam obrigatoriamente estar relacionados.
1..1	Um e somente um. Indica que apenas um objeto da classe se relaciona com os objetos da outra classe.
0..*	No mínimo nenhum e no máximo muitos. Indica que pode não haver não instâncias da classe participando do relacionamento.
*	Muitos. Indica que muitos objetos da Classe estão envolvidos no Relacionamento.
1..*	No mínimo um e no máximo muitos. Indica que há pelo menos um objeto envolvido no relacionamento, podendo haver muitos.
3..5	No mínimo 3 e no máximo 5. Indica que há pelo menos 3 instâncias envolvidas no relacionamento e que pode ser 4 ou 5 as instâncias envolvidas, mas não mais do que isso.



# Associação Binária

- Ocorre quando são identificados relacionamentos entre duas classes.
- Este tipo de Associação constitui-se na mais comum encontrada nos Diagramas de Classe.

# Representação da Associação Binária



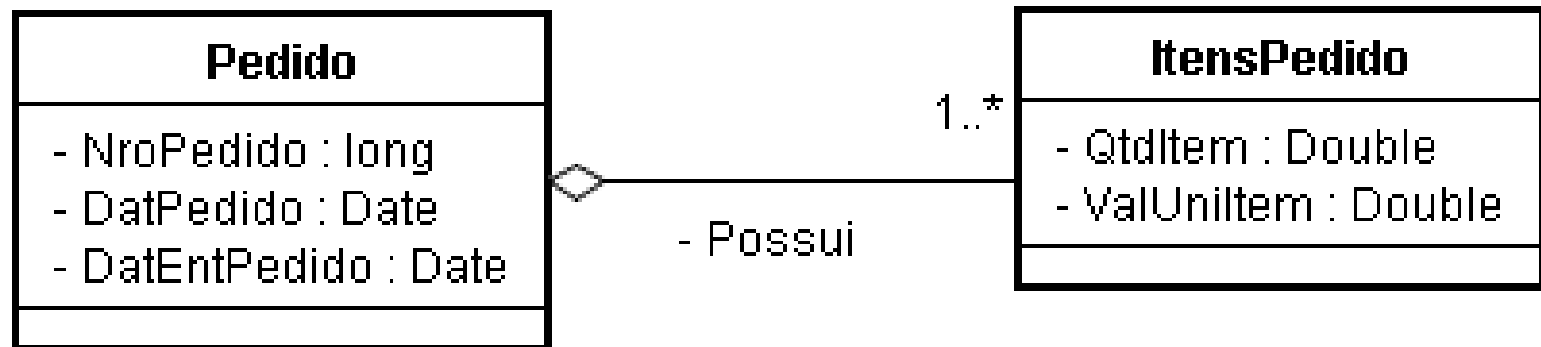
# Agregação

- É um tipo especial de associação onde tenta-se demonstrar que as informações de um objeto (chamado objeto-todo) precisam ser complementadas pelas informações contidas em um objeto de outra classe (chamado objeto-parte).

# Representação de Agregação

- O símbolo de agregação difere do de associação por conter um losango na extremidade da classe que contém os *objetos-todo*.

# Representação de Agregação



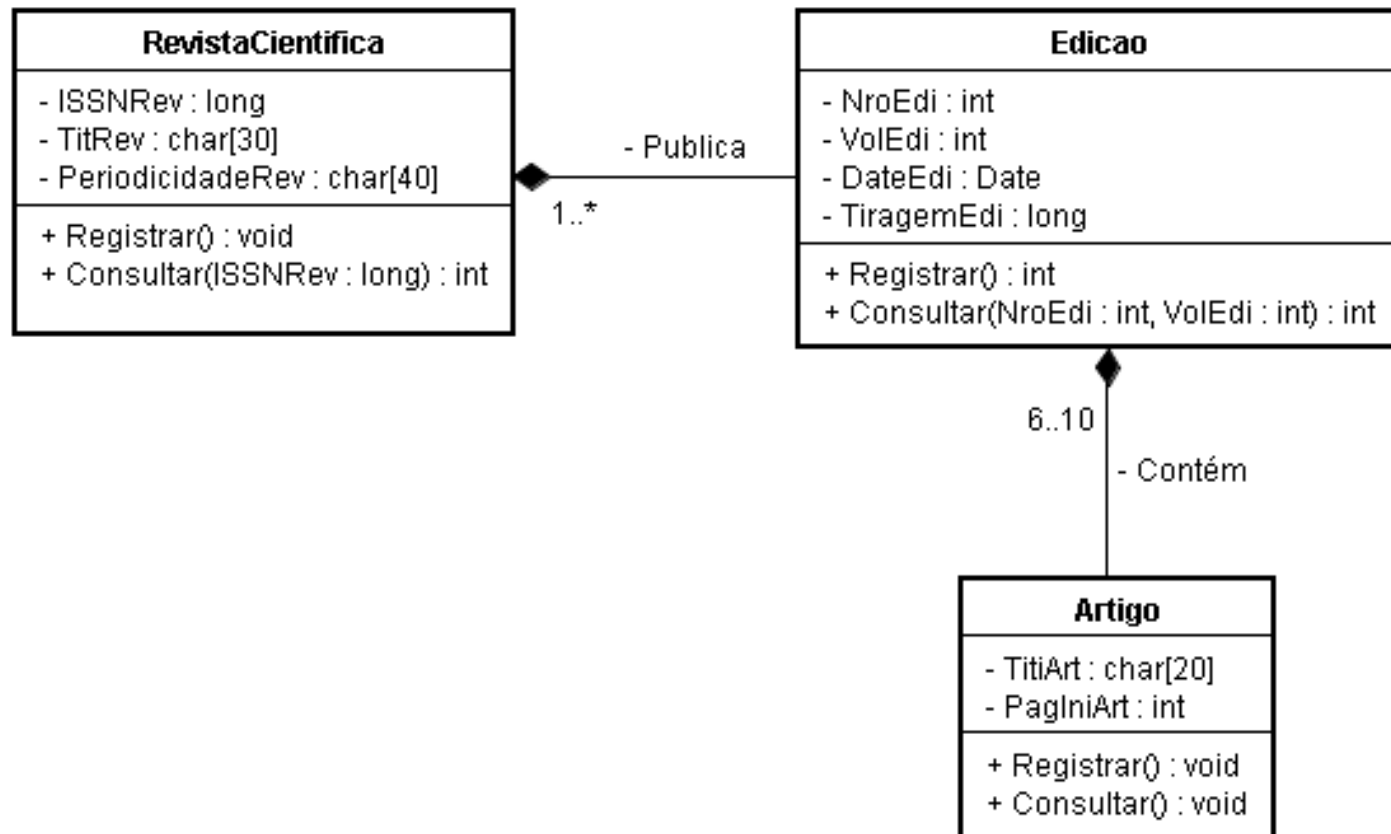
# Composição

- Constitui-se em uma variação do tipo agregação. Uma associação do tipo Composição tenta representar um vínculo mais forte entre os objetos-todo e objetos-parte.
- Tenta mostrar que os objetos-parte têm que pertencer exclusivamente a um único objeto-todo.

# Representação da Composição

- O símbolo usado para a associação de Composição é um losango preenchido, e da mesma forma que na Agregação, deve ficar ao lado do objeto-todo.

# Representação da Composição

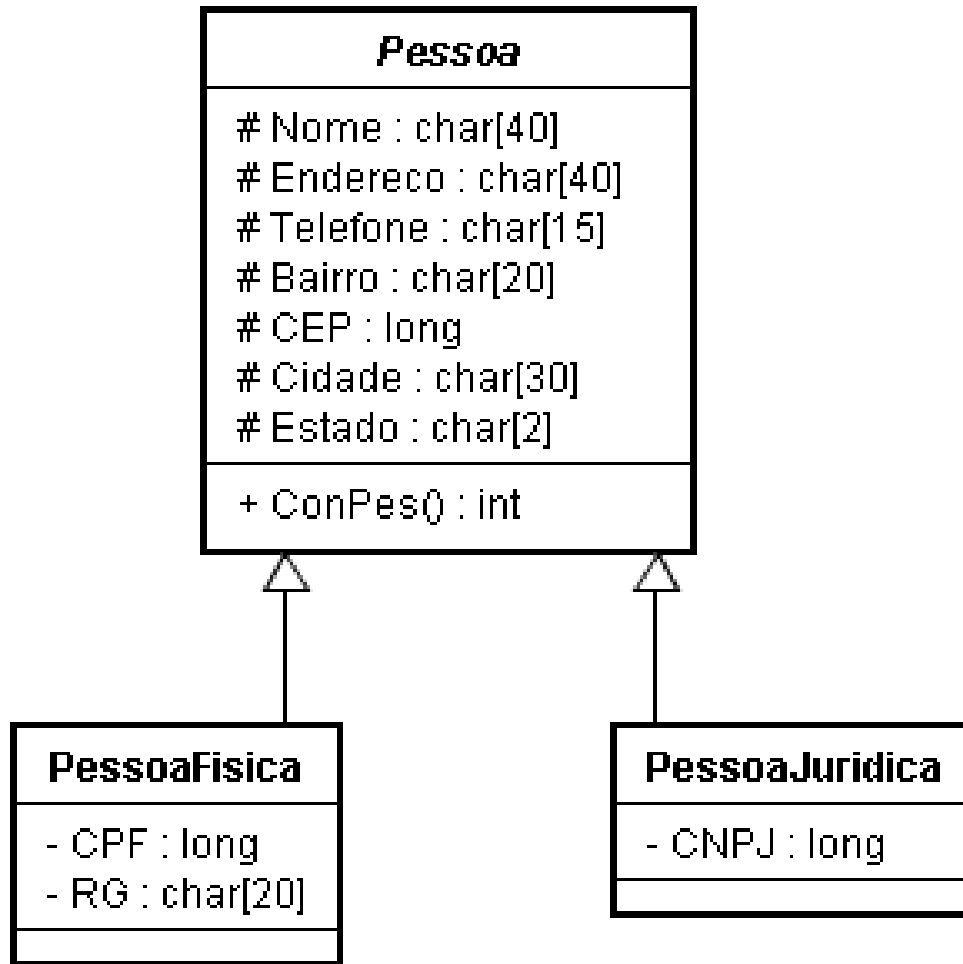




# Especialização / Generalização

- Similar à associação de mesmo nome utilizado no Diagrama de Casos de Uso. Seu objetivo é identificar classes-mãe (gerais) e classes filhas (especializadas).
- Permite também demonstrar a ocorrência de métodos polimórficos nas classes especializadas.

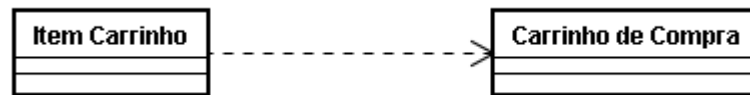
# Especialização / Generalização



# Dependência

- Não é um tipo comum de relacionamento, como o próprio nome diz, identifica um certo grau de dependência de uma classe em relação a outra.
- Representado por uma reta tracejada entre duas classes, contendo uma seta na extremidade do relacionamento que é dependente de alguma forma.

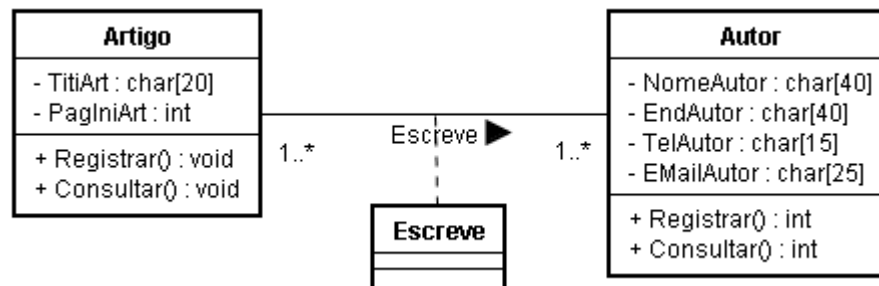
# Dependência



# Classe Associativa

- São classes produzidas quando da ocorrência de associações que possuem multiplicidade muitos (\*) em todas as suas extremidades.
- As Classes Associativas são necessárias nesses casos porque não existe um repositório que possa armazenar as informações produzidas pelas as associações.

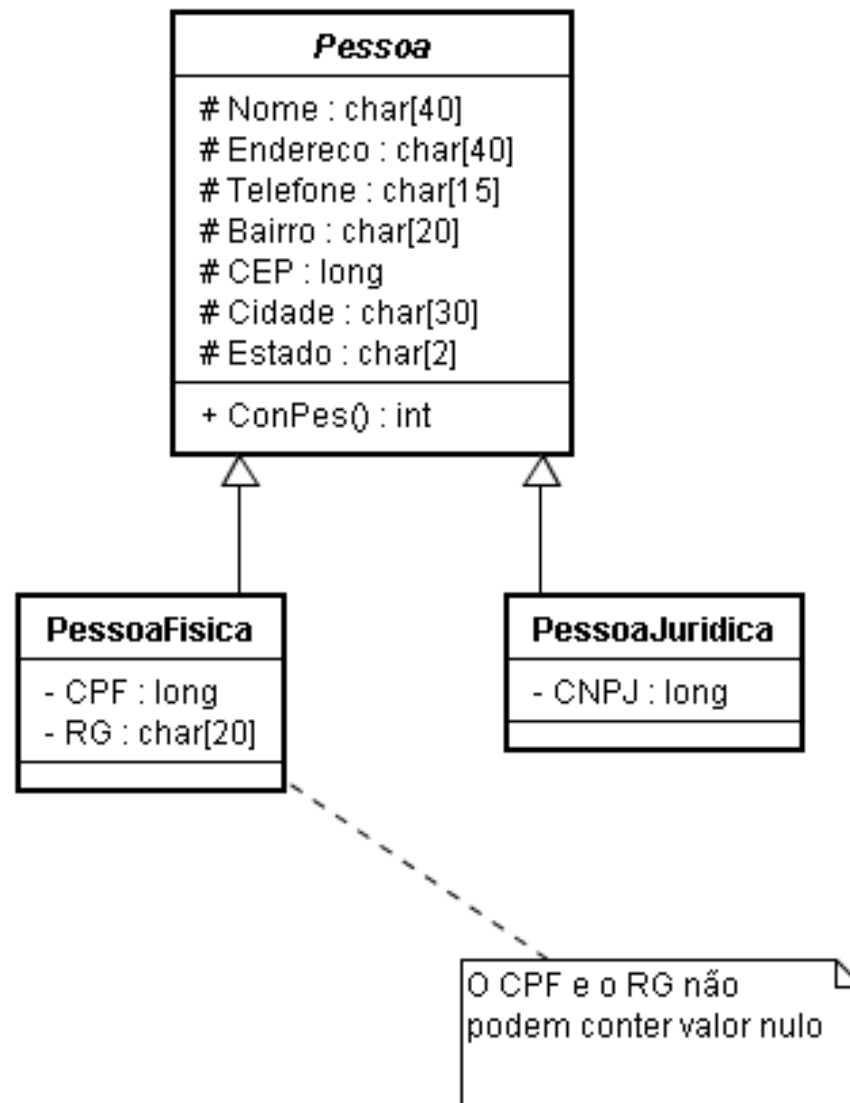
# Classe Associativa



# Notas

- São importantes para informar algum comentário necessário a classe, método ou atributo, fazendo com que, todos tomem conhecimento de forma imediata a observação feita, seja essa observação feita para validar ou simplesmente informar como o objeto notificado se comporta.

# Notas





# Exercício 1

- Desenvolva um Diagrama de Classes para um sistema de vídeo locadora equivalente ao módulo de locação de DVD's, de acordo com as informações abaixo:
  - É necessário um controle dos filmes existentes na locadora;
  - Um Sócio pode realizar muitas ou nenhuma locações enquanto permanecer sócio da locadora, mas uma locação estará vinculada unicamente a um determinado sócio.
  - Cada locação deve obrigatoriamente conter ao menos um filme, podendo conter vários filmes, no entanto uma mesma cópia pode ter sido locada diversas vezes, em épocas diferente obvimanete.

# Exercício 2

- Desenvolva o Diagrama de Classes para um sistema de cursos de informática equivalente ao módulo de matrícula, de acordo com as informações abaixo:
  - Um curso pode ter muitas turmas, mas uma turma se relaciona exclusivamente com um único curso.
  - Uma turma pode possuir diversos alunos matriculado, no entanto uma matrícula refere-se exclusivamente a uma determinada turma.
    - Cada turma tem um número mínimo de alunos para poder ser iniciada.
  - Um aluno poderá realizar muitas matrículas, mas cada matrícula refere-se exclusivamente a uma aluno.

# Exercício 3

- Desenvolva o Diagrama de Classes para um sistema de Controle de Apólice de seguros, de acordo com as informações abaixo:
  - Um cliente para ser cliente, necessita possuir no mínimo uma apólice em seu nome, podendo possuir diversas, no entanto, uma apólice será atribuída a um único cliente.
  - Da mesma forma que uma apólice pode possuir de uma até quatro parcelas, mas uma parcela estará vinculada a uma única apólice.

# Exercício 3

- ❑ Um veículo segurado poderá ou não possuir sinistro. Cada sinistro possuirá um tipo.
  - Acidente, roubo, incêndio, etc.
- ❑ Será notificado também os danos no veículo, sabendo-se que um sinistro poderá causar danos ou não ao veículo.
- ❑ Cada veículo segurado possuirá uma modelo, e cada modelo estará vinculado exclusivamente com uma marca.

# Diagrama de Seqüência

Os eventos em ordem

# Diagrama de Seqüência

- Procura determinar a seqüência de eventos que ocorrem em um determinado processo, ou seja, quais condições devem ser satisfeitas e quais métodos devem ser disparados entre os objetos envolvidos e em que ordem durante um processo específico.

# Diagrama de Seqüência

- Assim, Determinar a ordem em que os eventos acontecem, as mensagens que são enviadas, os métodos que são chamados e como os objetos interagem entre si dentro de um determinado processo é o objetivo principal deste diagrama.

# Diagrama de Seqüência

- Geralmente baseia-se em um caso de uso:
  - Isso acontece porque geralmente um Caso de Uso é um processo disparado pelo o usuário
  - Um diagrama de Casos de Uso pode gerar vários Diagramas de Seqüência.
  - Nem sempre um Caso de Uso gera um Diagrama de Seqüência, isso acontece por exemplo com Casos de Uso do tipo <<include>>.



# Atores

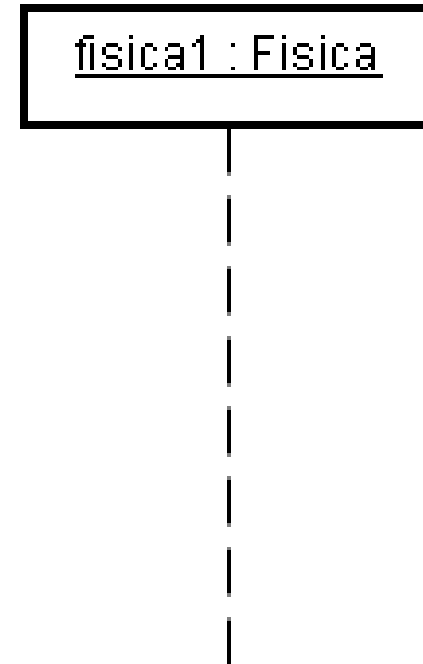
- São exatamente os mesmos descritos no Diagrama de Casos de Uso.
  - Entidade externas que interagem com o sistema e que solicitam serviços.

# Objetos

- Os Objetos representam as instâncias das classes envolvidas no processo ilustrado pelo Diagrama de Seqüência.
  - Os objetos são representados por um retângulo contendo um texto que identifica primeiramente o nome do Objeto, em minúsculo, e depois o nome da classe, com letras iniciais maiúsculas.
    - Essas informações são separadas por dois pontos (:).

# Objetos

- Logo abaixo do objeto surge uma linha vertical tracejada.
- O Diagrama de Seqüência não possui atributos



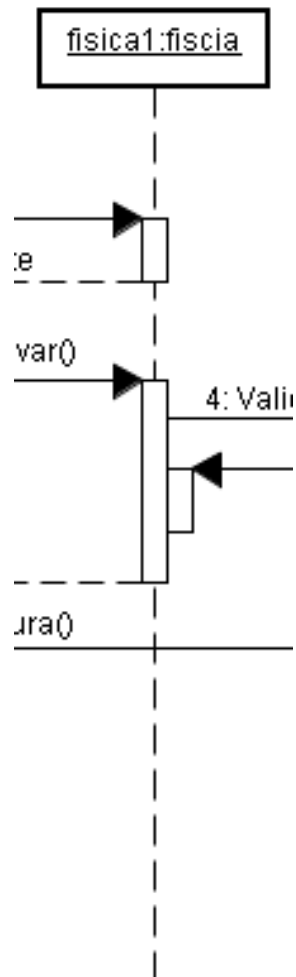
# Linha de Vida

- A Linha de Vida representa o tempo em que um Objeto existiu durante um processo.
- As Linhas de Vida são representadas por linhas finas verticais tracejadas partindo do retângulo que representa o Objeto.

# Foco de Controle ou Ativação

- Indica os períodos em que um determinado objeto está participando ativamente do processo.
- Os focos de controle são representados dentro da Linha de Vida de um Objeto, enquanto as Linhas de Vida são representadas por tracejados finos, o Foco de Controle é representado por uma linha mais grossa.

# Foco de Controle ou Ativação



# Mensagens ou Estímulos

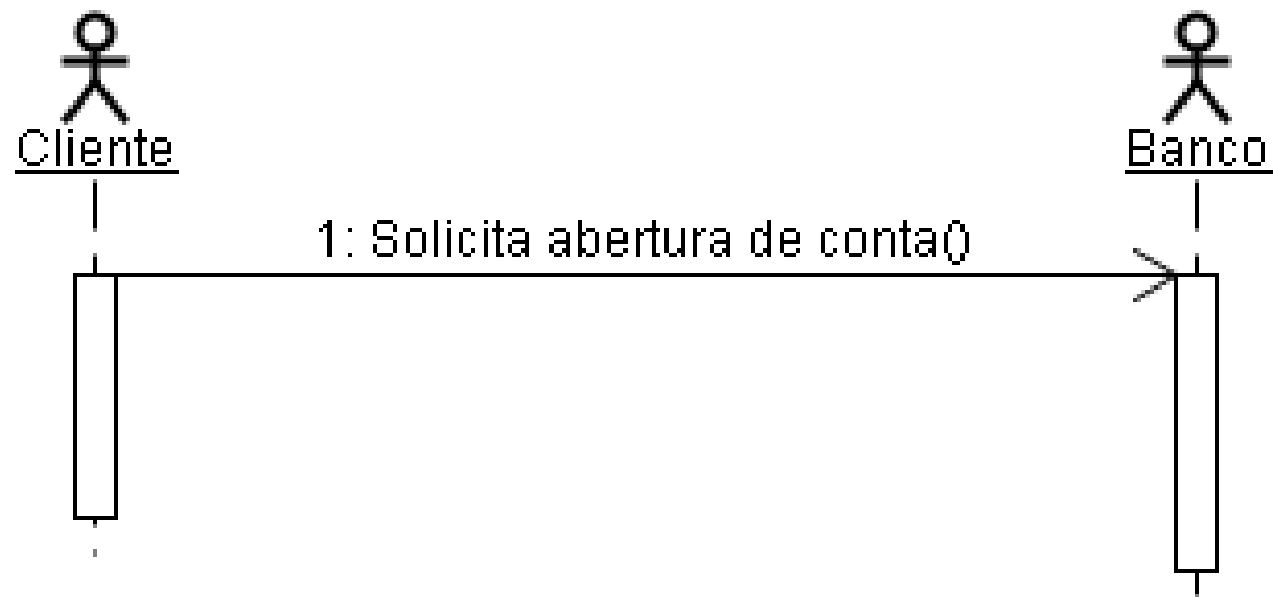
- As mensagens procura demonstrar a ocorrência de eventos, que normalmente forçam a chamada de um método em algum dos Objetos envolvidos no processo.
  - Pode ocorrer, no entanto, de uma mensagem representar simplesmente a comunicação entre dois atores, o que, neste caso, não dispara nenhum método.

# Mensagens ou Estímulos

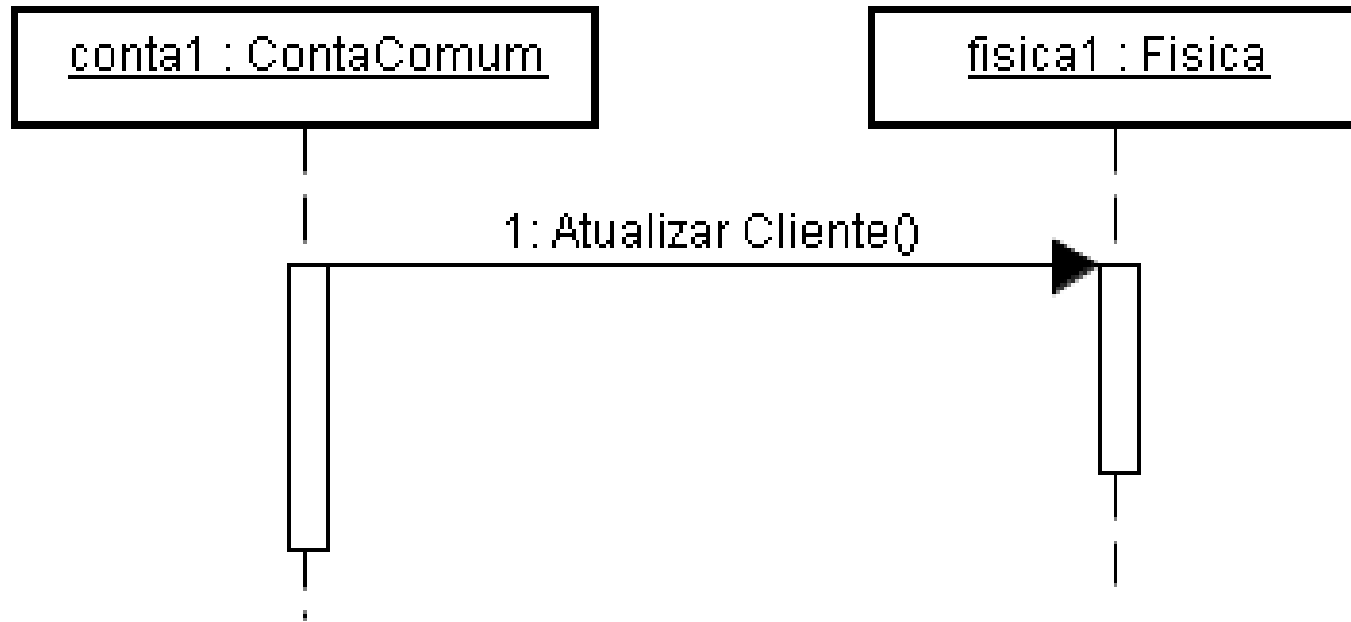
- As Mensagens podem ser disparada entre:
  - Um Ator e outro Ator. (Não é muito comum, mas facilita a compreensão do processo).
  - Um Ator e um Objeto. (O Ator produz um evento que força o disparo de um método).
  - Um Objeto e outro Objeto. (O mais comum, o objeto transmite uma mensagem para outro objeto, solicitando a execução de um método).
  - Um Objeto e um Ator. (Geralmente quando um objeto envia uma mensagem de retorno).



# Mensagem entre Atores



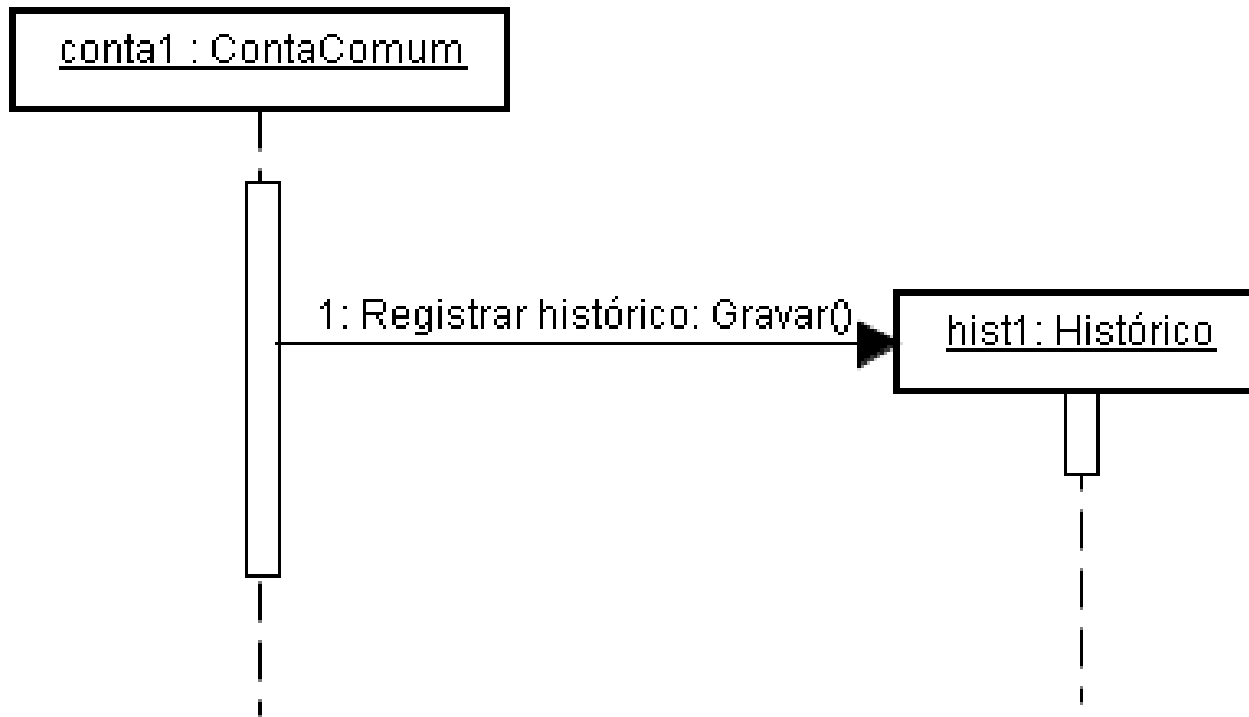
# Mensagem com disparo de Métodos entre Objetos



# Instanciando um novo objeto

- Quando a mensagem é dirigida a um objeto que já existia, a seta da mensagem atinge a Linha de Vida do objeto, engrossando-a, identificando que o Foco de Controle está sobre o objeto em questão.
- Quando a mensagem cria um novo objeto, no entanto, a seta atinge o retângulo que representa o objeto, indicando que a mensagem representa um método construtor e que o objeto passa a existir a partir daquele momento.

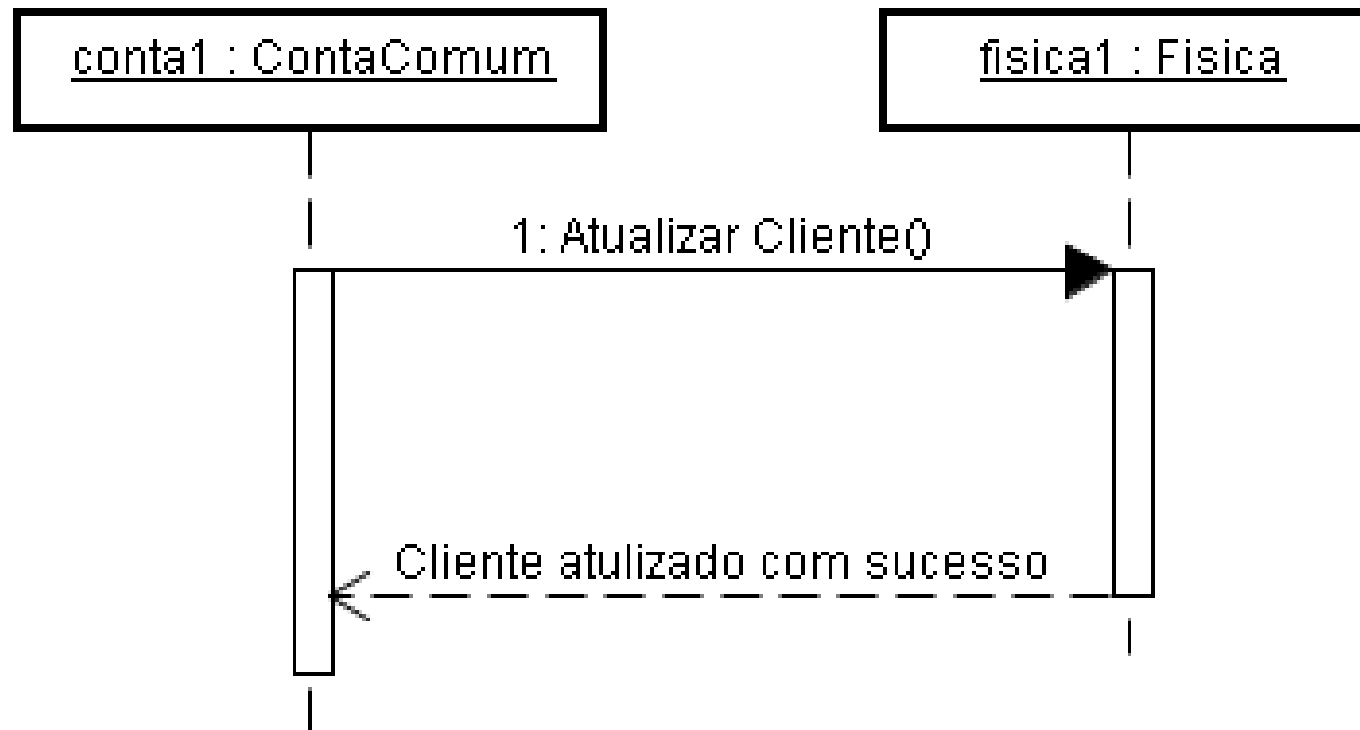
# Instanciando um novo objeto



# Mensagem de Retorno

- Este tipo de mensagem identifica a resposta a uma mensagem para o objeto ou ator que a chamou.
- Uma Mensagem de Retorno pode retornar informações específicas do Método chamado ou simplesmente um valor indicado se o método for executado com sucesso ou não.

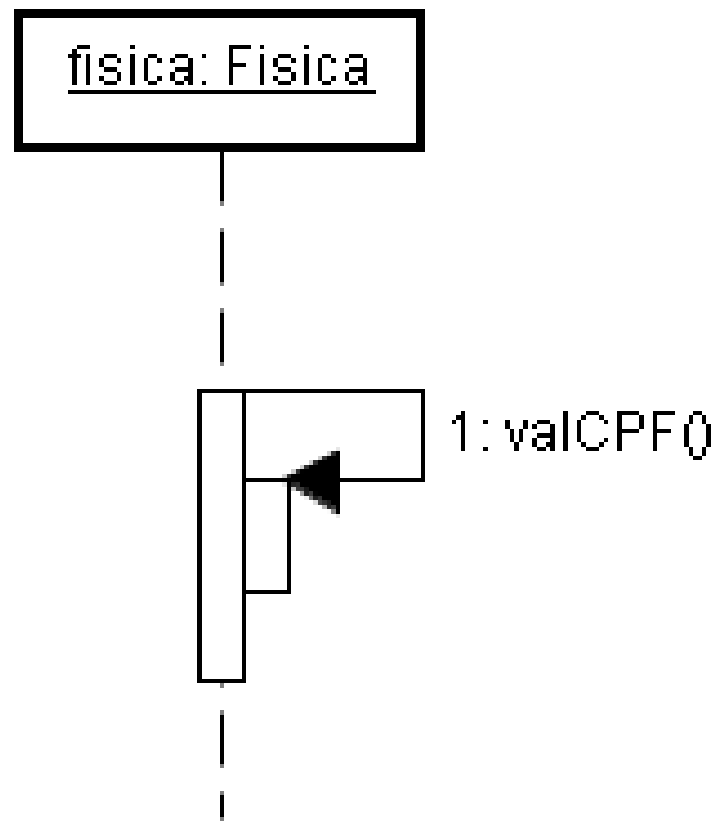
# Mensagem de Retorno



# Auto-chamadas

- São mensagens que um objeto envia para si mesmo. No caso de auto-chamadas uma mensagem parte do objeto e atinge o próprio objeto.

# Auto-chamadas





# Exercício 1

- Diagrama de Seqüência para abertura de conta comum
  - Inicialmente o Cliente solicita ao Funcionário a abertura de uma conta, então o Banco faz uma consulta do cliente pelo seu CPF (Método), na classe Física, se o cliente se encontra cadastrado, a consulta retorna com os Dados do Cliente, se não o cadastro do cliente deverá ser realizado.
  - No cadastro do cliente (Física), deverá conter um método para validar o CPF, evitando assim, o cadastro de clientes com CPF inexistente.
  - Após o cadastro do cliente o funcionário receberá uma resposta do Sistema informando que o cliente está atualizado, da mesma forma que o funcionário comunica ao cliente que seu cadastro foi aprovado.

# Exercício 1

- ❑ Ao receber a resposta do funcionário, o cliente deve informar valor do depósito a ser feito e sua senha. Essa mensagem irá disparar um método para abertura de uma nova conta comum, que por sua vez, irá registrar esse histórico.
- ❑ O Cliente deverá ser informado sobre o status de sua conta, ou seja, que a abertura da conta foi concluída.

# Exercício 2

- Diagrama de Seqüência para encerramento de uma conta comum.
  - Nesta caso o Cliente solicita ao Funcionário o encerramento de sua conta, o Funcionário por sua vez deve verificar a conta, neste momento, é necessário a senha do cliente e em seguida se existe Saldo.
  - Se o Funcionário receber a resposta de que o saldo é positivo, deve haver o saque do valor.
  - Assim como qualquer movimentação, havendo o saque deve-se registrar o histórico referente ao Saque.
  - Após a confirmação do saque, deve ser disparado o método de encerramento de Conta. Em seguida avisar ao cliente.

## Exercício 3

- Diagrama referente a solicitação de Extrato de uma conta comum através de um caixa eletrônico.

# Diagrama de Colaboração

A organização estrutural do objetos

# Diagrama de Colaboração

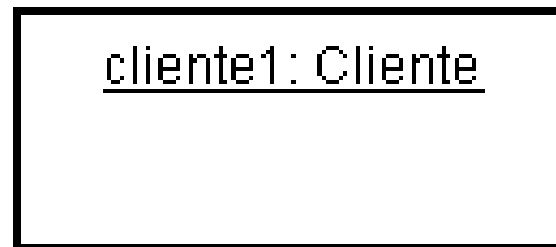
- O Diagrama de Colaboração é muito semelhante ao Diagrama de Seqüência.
- A diferença está principalmente porque o Diagrama de Seqüência concentra-se na seqüência temporal em que os eventos ocorrem e as mensagens que são trocadas, enquanto o Diagrama de Colaboração preocupa-se com a organização estrutural dos objetos, em como os objetos estão vinculados e as mensagens que estes trocam entre si.

# Diagrama de Colaboração

- A semelhança entre os dois são tantas, que são conhecidos como Diagramas de Interação.
- Na verdade o Diagrama de Colaboração mostra praticamente as mesmas informações que o Diagrama de Seqüência, apenas com uma outra visão, de maneiras diferentes.

# Objetos

- Os Objetos do Diagrama de Colaboração diferem-se dos objetos do Diagramas de Seqüências, apenas pelo fato de não possuir Linha de Vida e Foco de Controle.



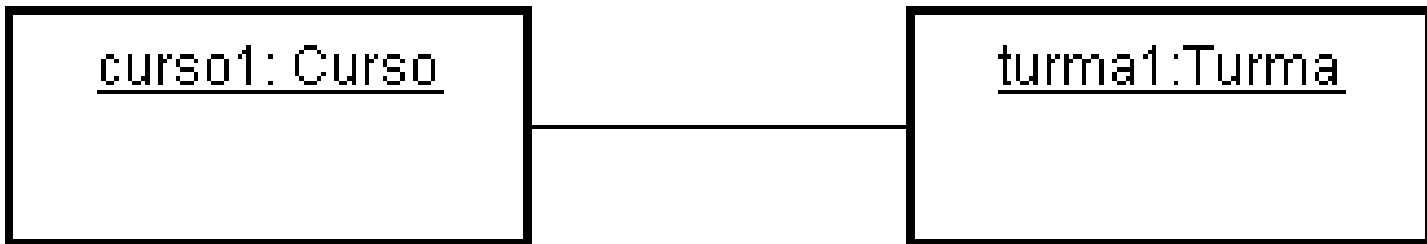


# Vínculos

- Está entre os principais objetivos do Diagrama de Colaboração, identificar os vínculos, ou seja, as ligações existentes entre os objetos envolvidos em um processo.
- Assim, fica caracterizado como vínculo sempre que dois objetos colaboram entre si dentro de um determinado processo.
  - Seja pelo envio ou pelo recebimento de mensagens ou ambos.

# Vínculos

- Um Vínculo é representado por uma linha unindo dois objetos



# Mensagens

- As mensagens usadas no Diagrama de Colaboração são as mesmas definidas no Diagrama de Seqüência, e quase sempre representam chamadas de métodos.
- No Diagrama de Colaboração não existe a preocupação com a ordem em que as coisas acontecem, o que de fato importa é que elas são disparadas

# Mensagens

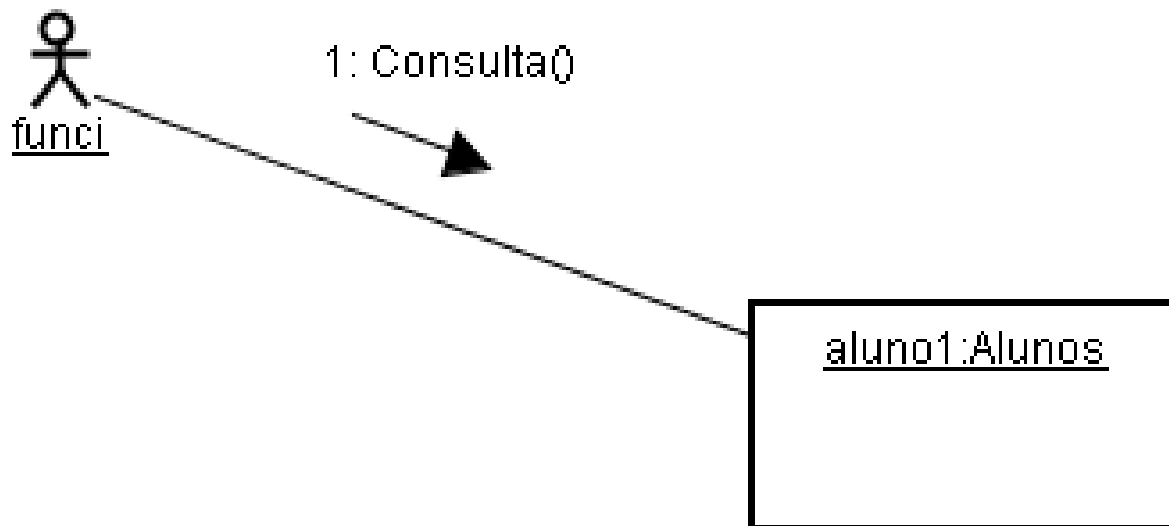
- No que diz respeito as Mensagens usadas no Diagrama de Colaboração, não existe Mensagem de retorno, como acontece no Diagrama de Seqüência.



# Atores

- Os Atores apresentados no Diagrama de Colaboração são os mesmo usados no Diagrama de Seqüência, conseqüentemente os mesmos do Diagrama de Casos de Uso.
- Este Ator possui vínculos com outros objetos ou outros Atores e envia e recebe mensagens através deste vínculo da mesma forma que os outros objetos.

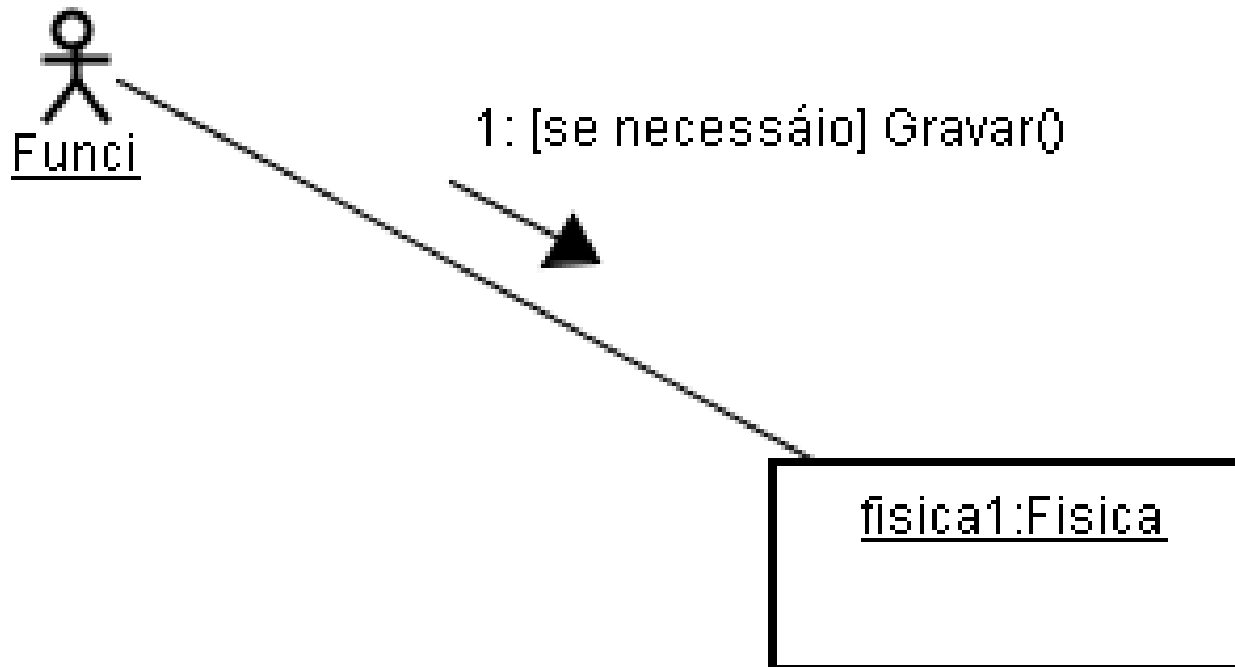
# Atores



# Condições

- São usadas para mostrar que uma mensagem só será enviada quando uma determinada condição for satisfeita.
  - As condições vêm entre colchetes antes da mensagem.

# Condições

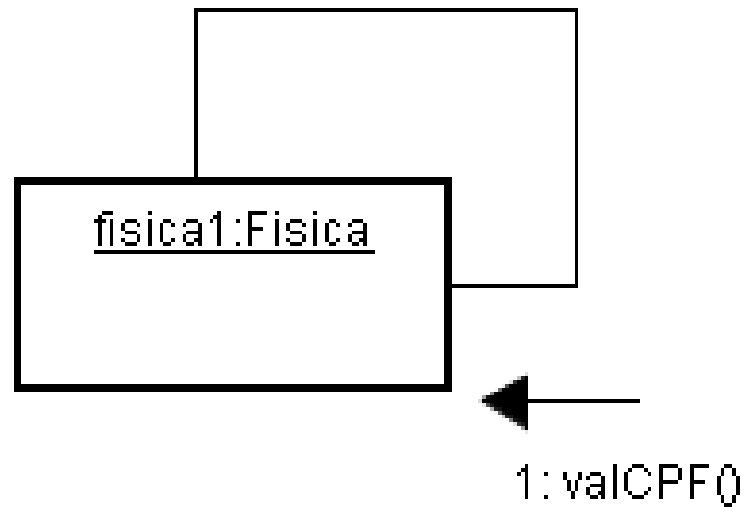




# Auto-chamadas

- Assim como usado no Diagrama de Seqüência, um objeto pode disparar uma mensagem para si mesmo, caracterizando assim uma Auto-chamada.
  - A mensagem parte do objeto para si próprio.

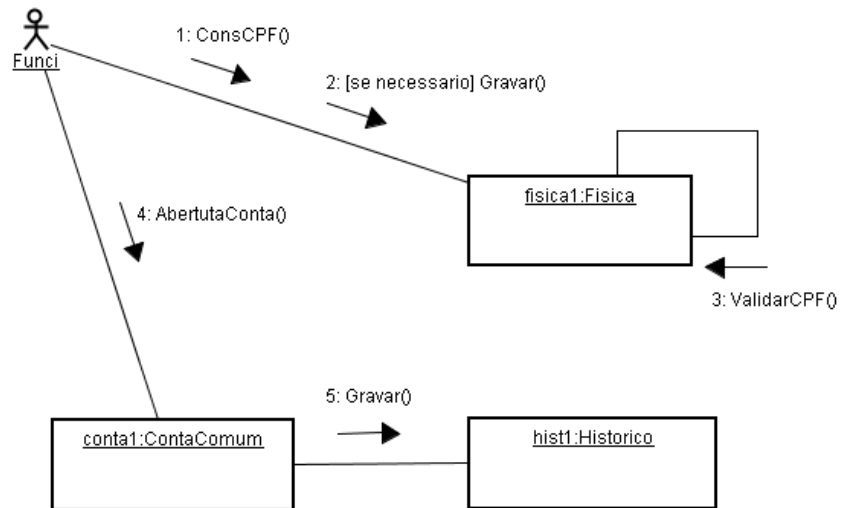
# Auto-chamadas



# Exemplo 1

- Abertura de uma conta comum
  - Primeiro o funcionário deve usar um método para consultar se já existe um CPF na classe (**física**), se necessário Atualizar / Gravar o cliente no objeto (**física1**) da classe (**física**), neste caso a classe irá disparar um método validação de CPF.
  - Assim o Funcionário poderá disparar o método Abertura de Conta no objeto **conta1** na classe **Conta Comum** que em seguida irá registrar o **histórico** dessa operação.

# Exemplo 1



# Exercício 1

- Crie um Diagrama de Colaboração para um sistema de Controle Bancário, referente ao módulo de Encerramento de Conta Comum.
  - Primeiro é necessário disparar o método **consultar conta** sobre o objeto **conta1** da classe **conta comum**, para analisar se a conta existe.
  - Em seguida é necessário **validar a senha e verificar o saldo**, se o saldo for positivo outro método será disparado (saque), e assim gerando outro método no objeto **historico1** da classe **histórico**.
  - Em seguida disparar o método **Encerramento**, se o cliente só possuir uma conta, será atualizado como inativo.

# Diagrama de Estados

Acompanhando as mudanças de um objeto

# Diagrama de Estados

- Procura acompanhar as mudanças de Estado sofridas por um objeto dentro de um determinado processo.
- O fato de o Diagrama de Estados estar geralmente associado a uma classe, ou mesmo aos objetos de uma classe envolvidos em um determinado processo, pode haver diversos Diagramas de Estados referentes ao mesmo processo.
  - É recomendado que só se construa Diagramas de Estados quando existir um certo grau de complexidade referente à Transição de Estados de um objeto envolvidos no processo.

# Estado

- Um estado representa a situação em que um objeto se encontra em um determinado momento durante o período em que este participa de um processo.
- Um objeto passa por diversos estados dentro de um mesmo processo. Um Estado pode representar:
  - ❑ A espera pela ocorrência de um evento;
  - ❑ A reação a um estímulo;
  - ❑ A execução de alguma atividade;
  - ❑ A satisfação de alguma condição.



# Estado

- Um estado é representado por um retângulo com os cantos arredondados com duas ou três divisões:
  - A primeira divisão armazena a descrição do estado;
  - A segunda, exibe as possíveis ações ou atividades, executadas por um objeto quando em um Estado.
  - A Terceira, com as possíveis transações internas.
    - Algumas ferramentas armazenam essas informações dentro da segunda parte.

# Estado

Atendendo Pedido

# Atividades / Ações

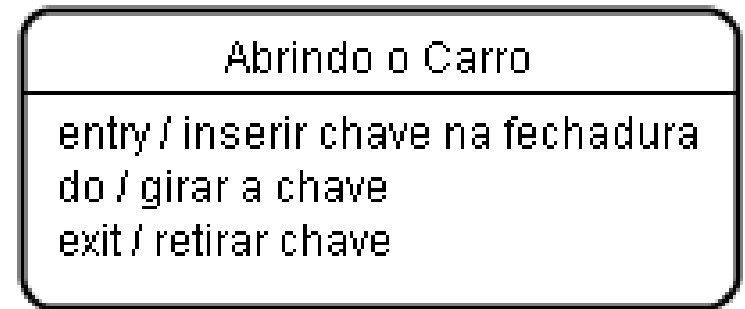
- Atividades e Ações são muito semelhantes, mas, apresenta diferenças com relação ao seu tempo de execução.
  - Atividades (tempo maior):
    - Geralmente são os métodos executados pelo o objeto.
  - Ações (tempo menor):
    - Geralmente representam a simples atribuição de um valor a um atributo.

# Atividades / Ações

- Tipos de Atividades ou Ações que podem estar na segunda parte do retângulo:
  - Entry → Representa as ações realizadas no momento em que o objeto assume o Estado em questão;
  - Exit → Identifica as ações executadas antes do objetos mudar de estado;
  - Do → Mostra as atividades executadas enquanto o objeto encontra-se em determinado Estado.

# Atividades / Ações

- Ao lado um exemplo dos tipos de atividades e ações de um Estado.



# Transição

- Representa um evento que causa uma mudança no Estado de um objeto, gerando um novo Estado. Este tipo de evento é conhecido como Evento de Ativação.
- Uma Transição é representada por uma reta ligando dois Estados, contendo uma seta em uma das extremidades. A ponta da seta indica o novo estado gerado pelo o evento.

# Transição

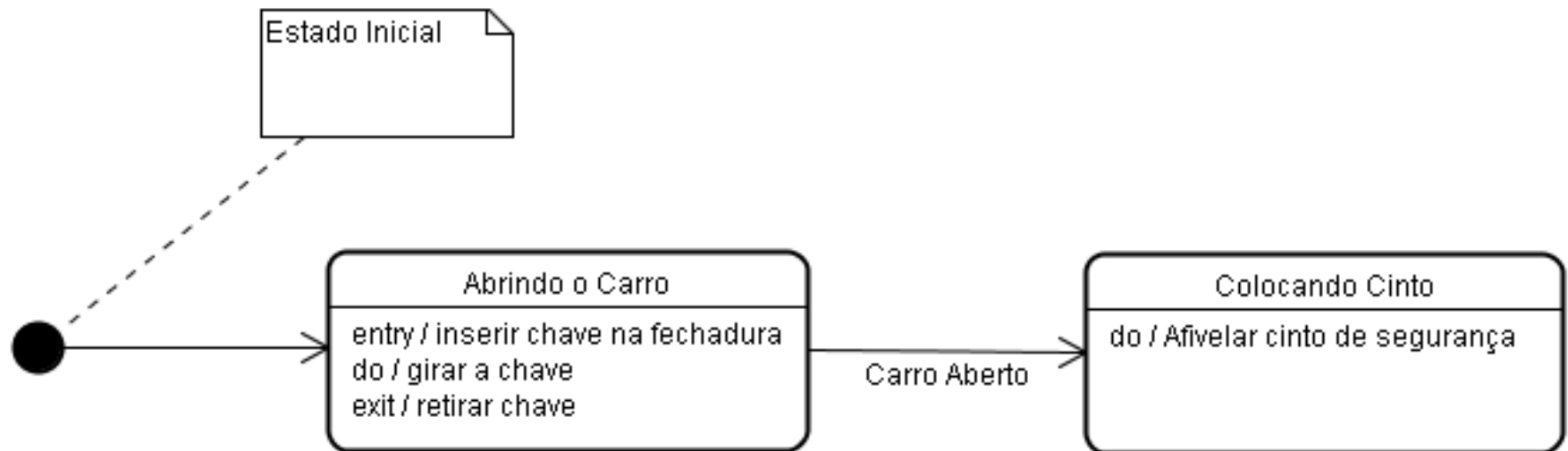


# Estado Inicial

- É um Estado abstrato cuja função é determinar o início de um Diagrama de Estados.
- O Estado Inicial é representado por círculo preenchido, a partir do qual é gerada uma transição que determina o início do processo.



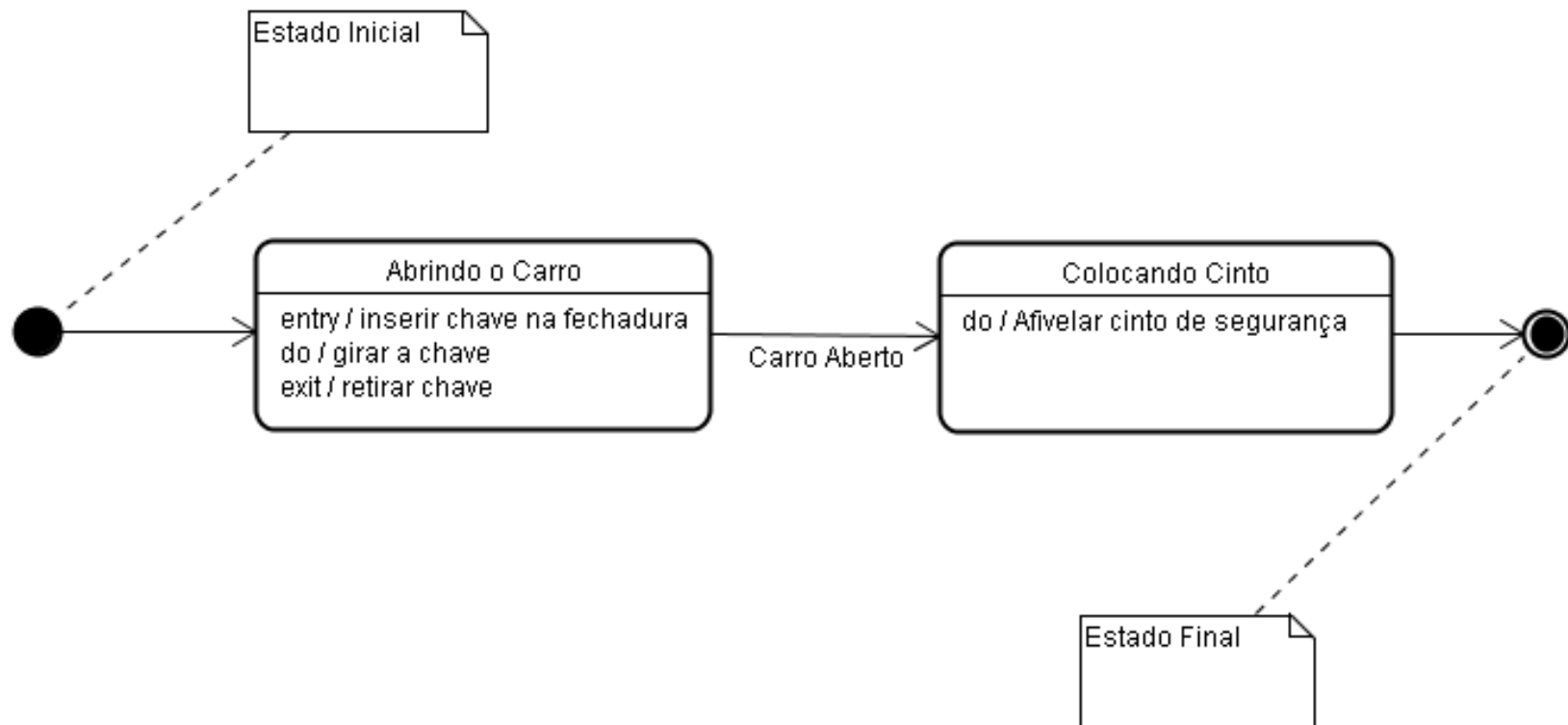
# Estado Inicial



# Estado Final

- Assim como o Estado Inicial, também é um estado abstrato cuja função é indicar o final do Diagrama de Estados.
- O Estado Final é representado por um círculo não preenchido envolvendo um outro círculo preenchido.

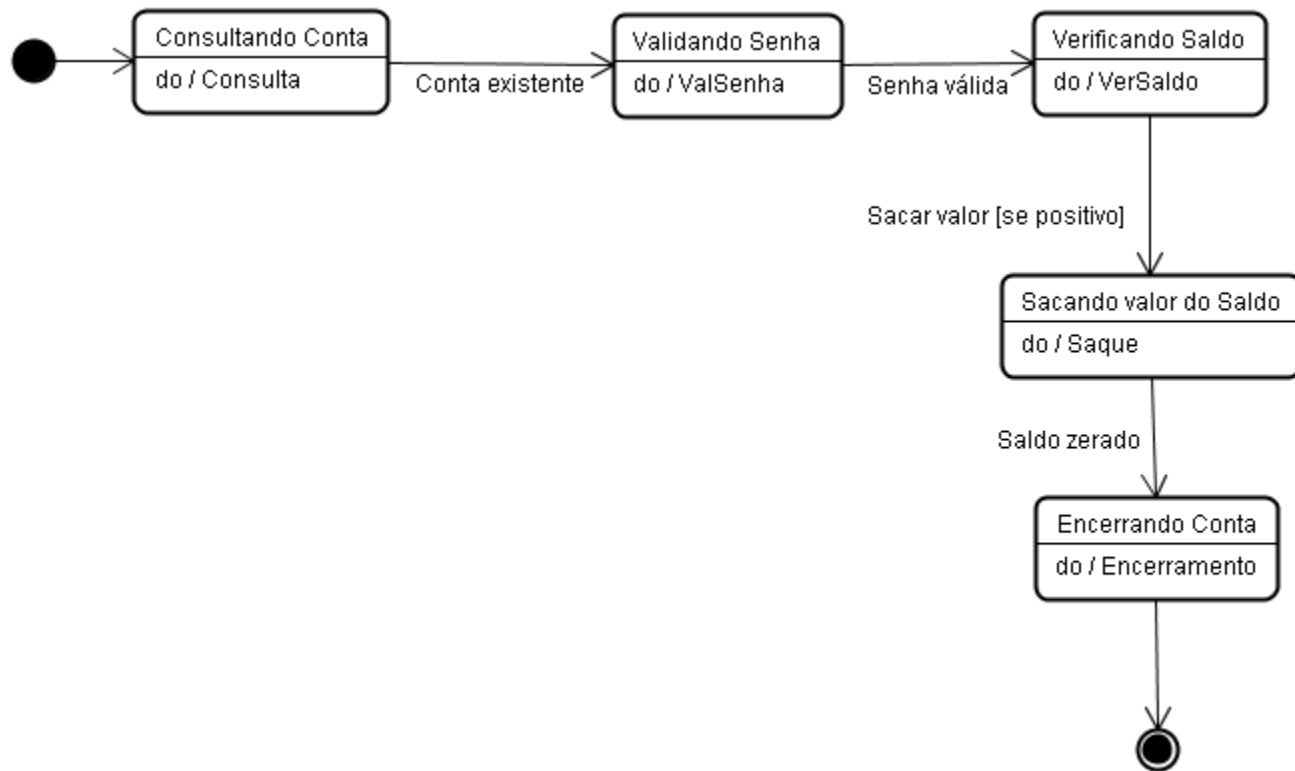
# Estado Final



# Exemplo 1

- Criar um Diagrama de Estados referente ao processo de encerramento de uma conta comum enfocando os estados assumidos pelo o objeto conta1.
  - ❑ Onde é necessário consultar a conta;
  - ❑ Validar senha;
  - ❑ Verificar saldo;
  - ❑ Realizar um saque [se saldo positivo];
  - ❑ Encerrar conta.

# Exemplo 1



# Exercício 1

- Desenvolva um Diagrama de Estados para um sistema de vídeo locadora equivalente ao módulo de locação de DVD's.
  - Durante o processo de locação de DVD's deve-se verificar se ele estar cadastrado;
  - Em seguida, deve-se verificar se não há locações pendentes (sem devolução ou pagamento).
  - Caso não haja pendências deve-se iniciar o registro de nova locação, bem como de cada item locado.
  - Após a seleção de cópias para a locação, esta deve ser finalizada.

## Exercício 2

- Desenvolva um Diagrama de Estados para um sistema de cursos de informática equivalente ao módulo de matrícula do aluno em uma turma de um determinado curso, enfocando os estados de um objeto da classe Matrícula, de acordo com os fatos já descritos anteriormente em outros diagramas desse mesmo sistema.

## Exercício 2

- ❑ Primeiramente, o usuário deve selecionar o qual a matrícula se refere;
- ❑ Em seguida, deve selecionar a turma da matrícula em questão;
- ❑ Finalmente o atendente irá selecionar o aluno que deseja realizar a matrícula e então registrar a matrícula.



# Diagrama de Atividades

Algoritmos em Diagramas

# Diagrama de Atividades

- É o diagrama com maior ênfase ao nível de algoritmo da UML e provavelmente um dos mais detalhistas. Apresenta muitas semelhanças com os antigos fluxogramas utilizados para desenvolver a lógica de programação e determinar o fluxo de controle de um algoritmo.

# Diagrama de Atividades

- Preocupa-se em descrever os passos a serem percorridos para a conclusão de um método ou algoritmo específico e não de um processo completo como acontece com o Diagrama de Seqüência e Colaboração.
- Concentra-se na representação do fluxo de controle de uma atividade.

# Componentes

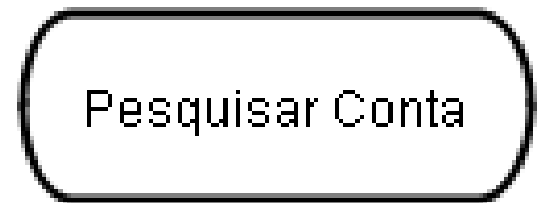
- Os componentes no Diagrama de Atividades são semelhantes aos utilizados no Diagrama de Estados, tais como:
  - Estado Inicial e Estado Final;
  - Transições;
  - Etc.

# Estado de Ação

- Representa a realização de uma ação dentro de um fluxo de controle.
  - Não pode ser decomposto em sub-estados.
- Uma atividade costuma possuir diversos estados de ação.
  - Um estado de ação pode conter uma descrição da ação que está sendo realizada, como a ação propriamente dita.

# Estado de Ação

- É representado por um retângulo arredondado sem divisões.



# Ponto de Decisão

- Um Ponto de Decisão representa um ponto do fluxo de controle onde deve ser realizado um teste, e uma tomada de decisão.
- De acordo com essa decisão, o fluxo optará por executar um determinado fluxo ou conjunto de Estados de Ação.

# Exemplo 1

- Diagrama de Atividades para consulta de uma conta:
  - Primeiramente o recebimento do número da conta;
  - Em seguida deve-se fazer uma pesquisa;
  - Se não encontrar emitir uma mensagem informando que o número da conta é inválido e sair do fluxo.
  - Se encontrar emitir uma mensagem informando que a conta é válida e sair.



# Exemplo 1

