

Engenharia de Softwares

Prof.ª Kassya

Prof. Ismar Frango – ismarfrango@uol.com.br – <http://ismar.cjb.net>

Conteúdo

- Análise e Projeto Estruturados
 - Histórico e Conceitos Básicos
 - Análise de Requisitos
 - Projeto de Software
 - Implementação e integração
 - Métricas de software
 - Teste e validação
 - Qualidade de software
 - Manutenção e Documentação
- Análise e Projeto Orientado a Objetos
- UML (padrão para os diagramas)

Bibliografia

RUMBOWGHI, James; BOOCH, Grady - *UML – Guia do Usuário* – Ed. Campus

SOMMERVILLE, I. & WESLEY, Adison – *Software Engineering*

PRESSMAN, R. S. – *Software Engineering* – McGraw Hill

YOURDON, E. – *Análise Estruturada Moderna* – Ed. Campus

UML Brasil - <http://www.uml.com.br>

Conceitos Básicos

Sistema

É um conjunto de elementos que se relacionam logicamente entre si, para atingir um objetivo.

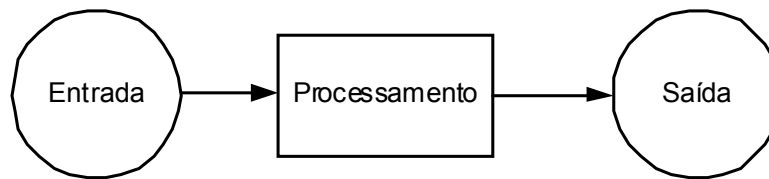
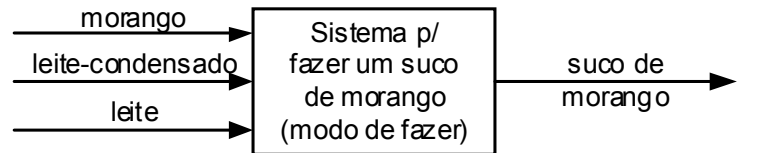
Características

- **Objetivo:** é a razão pela qual o sistema existe.
- **Elementos:** são as partes que compõem o sistema.
- **Estrutura:** é a forma como os elementos se relacionam entre si.

- **Comportamento:** é determinado pelos procedimentos atribuídos aos elementos.
- **Ciclo de vida:** envolve as fases de criação, desenvolvimento e decadência.

Modelo de um Sistema

Exemplo:



Modelo de Qualquer Sistema

Organização

É a estrutura de autoridades, responsabilidades, atividades e relações criadas a fim de coordenar esforços de pessoas que visam atingir objetivos comuns.

Exemplo: Escola, igreja, exército, empresa.

Sistema Empresa

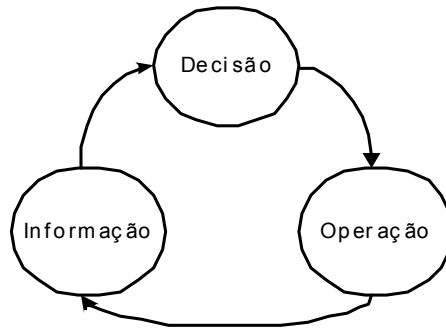
Características

- **Objetivo:** lucro como objetivo maior e outros como aumento de vendas, criação de novos produtos, construção de filiais, ...
- **Elementos:** capital, mão-de-obra, equipamentos, ...
- **Estrutura:** é a hierarquia existente entre os elementos.
- **Comportamento:** é a forma como os elementos reagem, de acordo com suas funções.
- **Ciclo de vida:** quando um sistema atinge seu objetivo, ele não precisa mais existir.

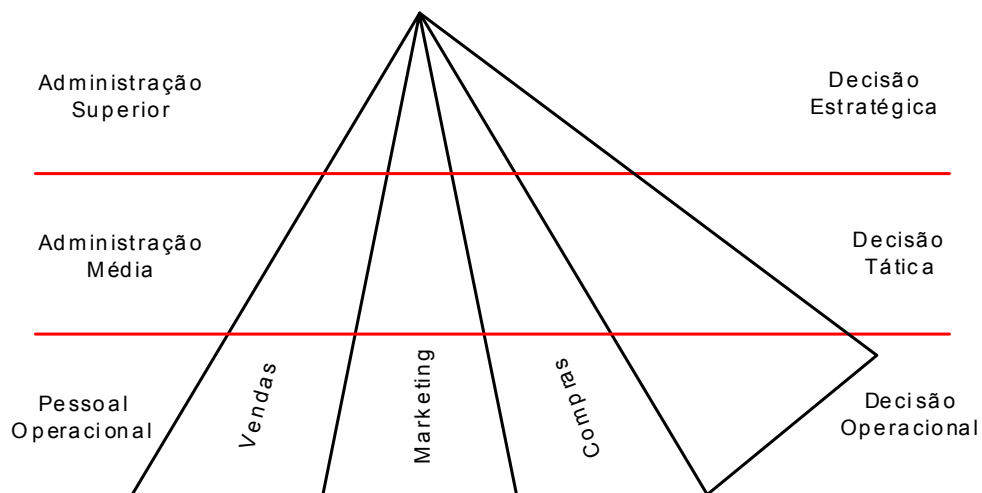
Subsistemas da Empresa

- **Subsistema de decisão:** faz parte deste subsistema os proprietários, acionistas, diretoria que *elaboram os planos* e estratégias.

- **Subsistema de operação:** é o pessoal operacional que executa os planos estabelecidos pelo subsistema de decisão.
- **Subsistema de informação:** faz parte o pessoal encarregado de obter informação sobre a execução dos planos, elaborar estas informações e levá-las até o subsistema de decisão.



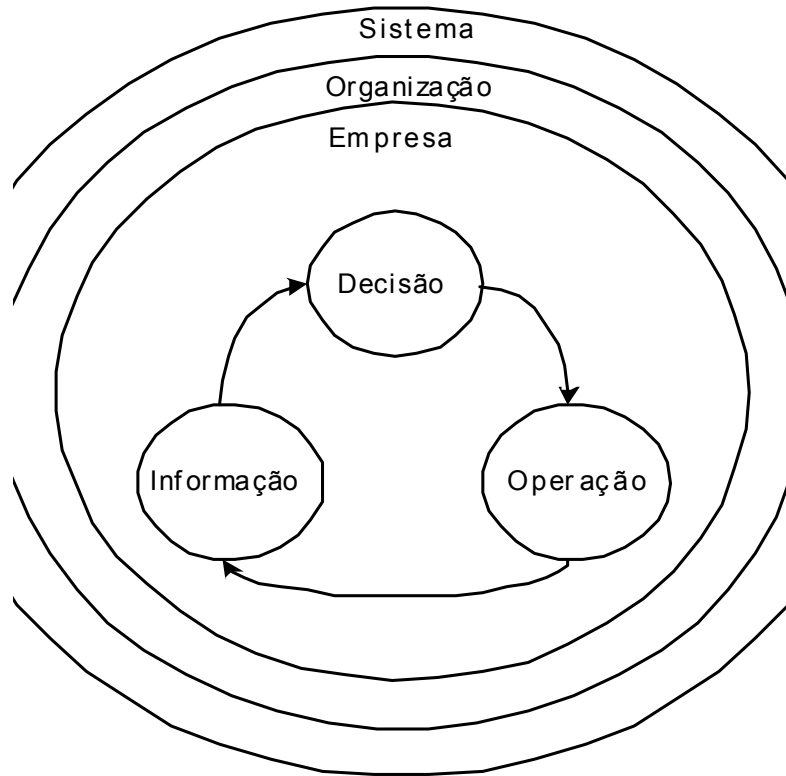
Níveis de decisão



- **Nível estratégico:** são decisões para um período maior que um ano.
- **Nível tático:** são decisões para um período menor que um ano
- **Nível operacional:** são decisões tomadas no dia-a-dia

Exemplos: Informações sobre um colégio

- Quinhentos alunos serão graduados neste ano (tático).
- O número de alunos matriculados em outros colégios está diminuindo (estratégico).
- Jane está matriculada na 2.^a série (operacional).



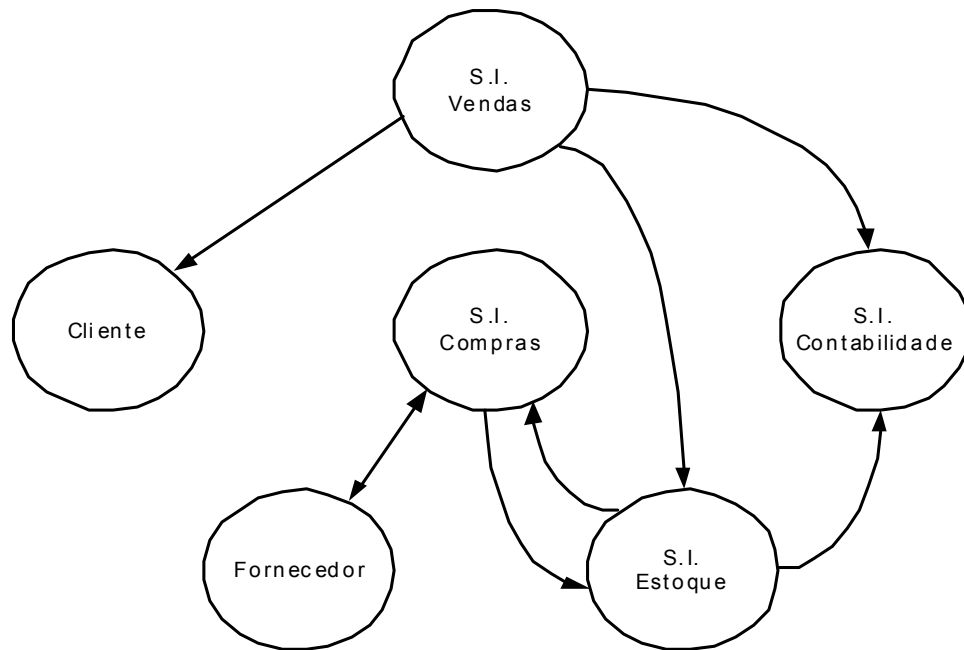
Sistema de Informação

É um conjunto de elementos que integram logicamente entre si, com a finalidade de gerar informação para a tomada de decisão.

Exemplos: TIA, Sistema de Folha de Pagamento, Sistema de Controle de Estoque, Jornal, Televisão.

Características de um Sistema de Informação

- **Objetivo:** gerar informação em nível estratégico, tático e operacional.
- **Elementos:** dados, sistema de processamento de dados e meios de comunicação.
- **Estrutura:** maneira pela qual, diversos sistemas de informação se relacionam entre si.



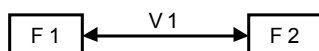
- **Comportamento:** deseja-se que o Sistema de Informação armazene, forneça informação em formato, tempo e custo apropriado.
- **Ciclo de vida:** quando se percebe que não há mais necessidade da informação, o Sistema de Informação pode deixar de existir.

Características ideais de uma informação

- Clara
- Objetiva
- Rápida
- Dirigida a quem necessita dela

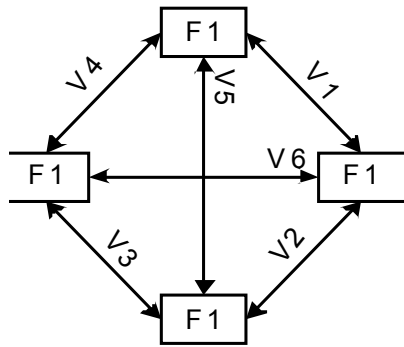
Qualidade na informação

Situação A:



F: fonte de informação
V: veículo de informação

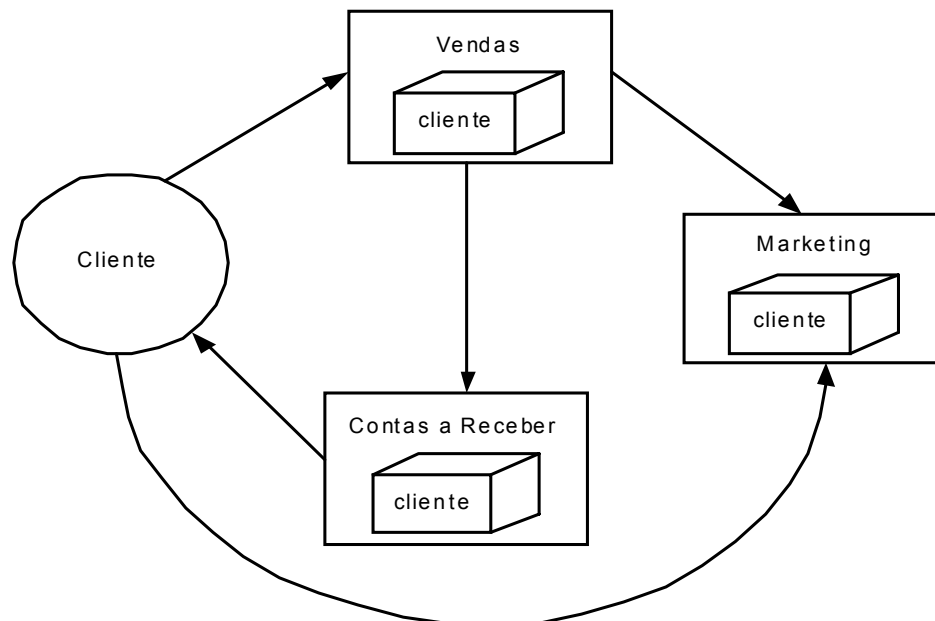
Situação B:



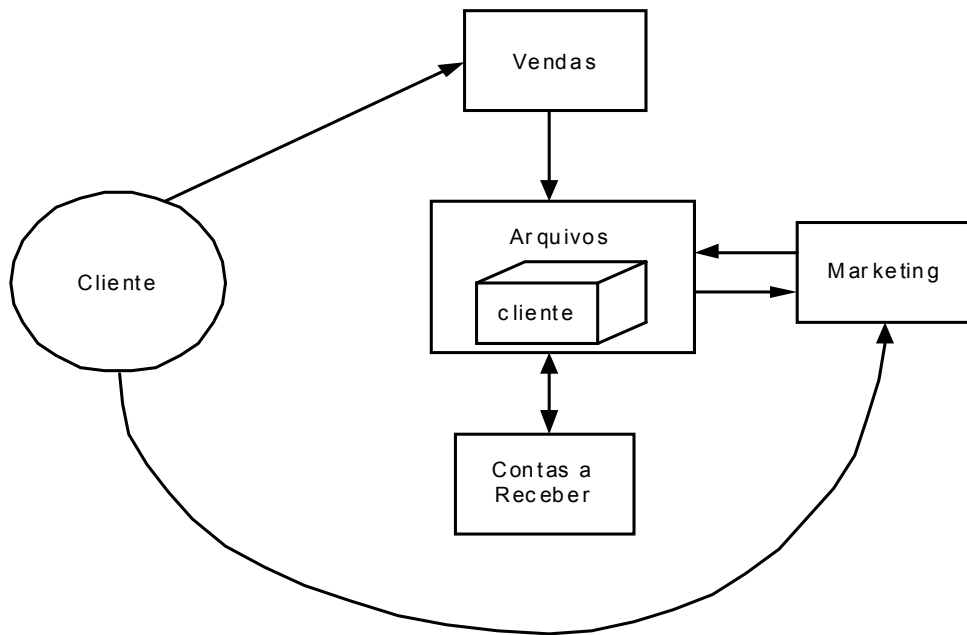
F: fonte de informação
V: veículo de informação

Estágios do Sistema de Informação

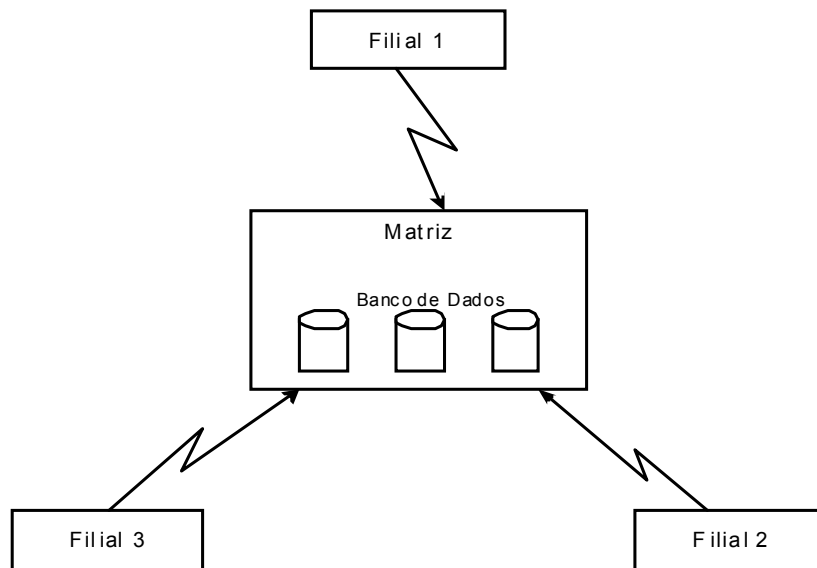
1. Sistemas Descentralizados



2. Sistemas Integrados



3. Sistemas Informatizados e Integrados

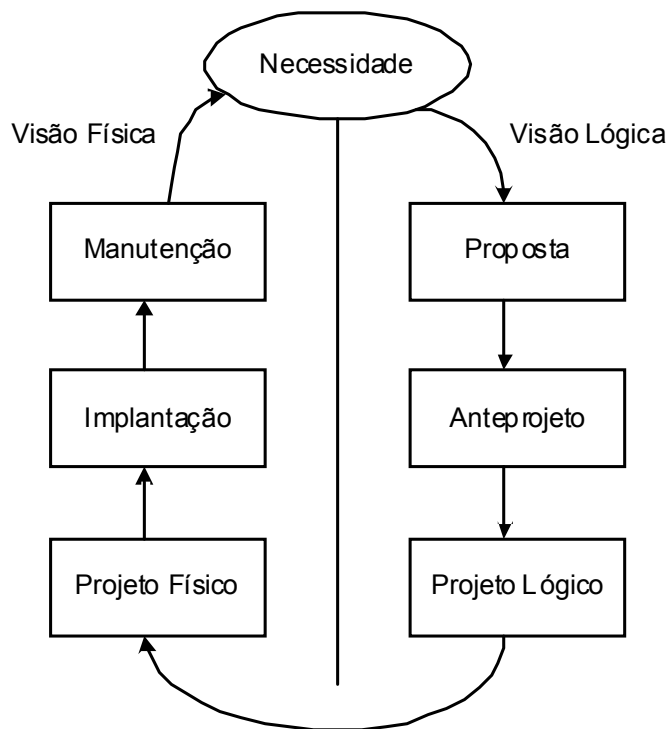


A Metodologia Estruturada para Análise e Projeto de Sistemas

Fases da Metodologia

1. **Proposta:** é o 1.º contato com o usuário e visa estabelecer as premissas que regerão o contrato de trabalho. É a integração entre ambas as partes.
2. **Anteprojeto:** é feito o levantamento de dados para obter os requisitos do sistema. São feitas a análise do sistema e o rascunho da proposta para o novo sistema.
3. **Projeto Lógico:** é a fase de definição detalhada do sistema. Define-se a lógica do sistema, o hardware, o software... (tudo ainda está no papel).
4. **Projeto Físico:** é a produção real do novo sistema. Os programas são codificados, são feitos os testes, é efetuada a compra de hardware e software...
5. **Implantação:** o sistema é colocado em prática.
6. **Manutenção:** acompanhamento do sistema, onde são adicionados novos módulos, são feitas alterações.

Ciclo de vida da Metodologia Estruturada



- **Visão Lógica:** (proposta, anteprojeto, projeto lógico) – o que fazer.
- **Visão Física:** (projeto físico, implantação, manutenção) – como fazer.

A Metodologia Estruturada para Análise e Projeto de Sistemas

Ferramentas da Metodologia Estruturada

- Levantamento de dados
- DFD (Diagrama de Fluxo de Dados)
- Técnicas de especificação de processos
 - Árvore de decisão
 - Tabela de decisão
 - Português estruturado
- Dicionário de dados
- DER (Diagrama Entidade-Relacionamento).
- MER (Método Entidade-Relacionamento)

Técnicas para levantamento de dados

- Entrevista
 - Formular perguntas antes da entrevista
 - Conquistar a confiança do entrevistado
 - Ser objetivo e impessoal
 - Conduzir a entrevista com segurança
- Questionário
 - Utilizar em situações onde há muitas pessoas para se obter informação e não há tempo para entrevistá-las e em situações onde podemos trabalhar as respostas
 - Formular questões do tipo “assinalar com um X” e que possam ser prontamente atendidas
- Pesquisas
 - Consultar manuais, documentos
- Observação local
 - Informar a todos que serão observados que trabalho está sendo executado
- Reuniões
 - Discutir sobre o sistema, com as pessoas envolvidas

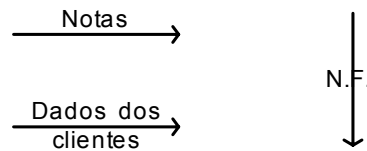
Diagrama de Fluxo de Dados (D.F.D.)

Simbologia

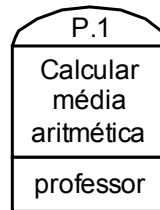
Entidade Externa: Representa a origem/destino das entradas e saídas do sistema. Não faz parte do processamento do sistema.



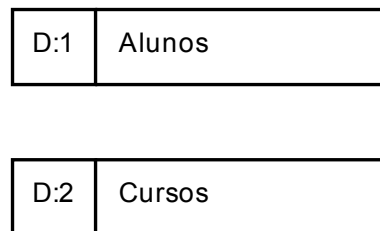
Fluxo de Dados: Representa as entradas e saídas do sistema



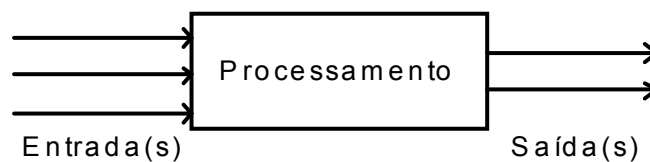
Processo: Representa o processamento, ou seja, o que irá ocorrer com a informação.

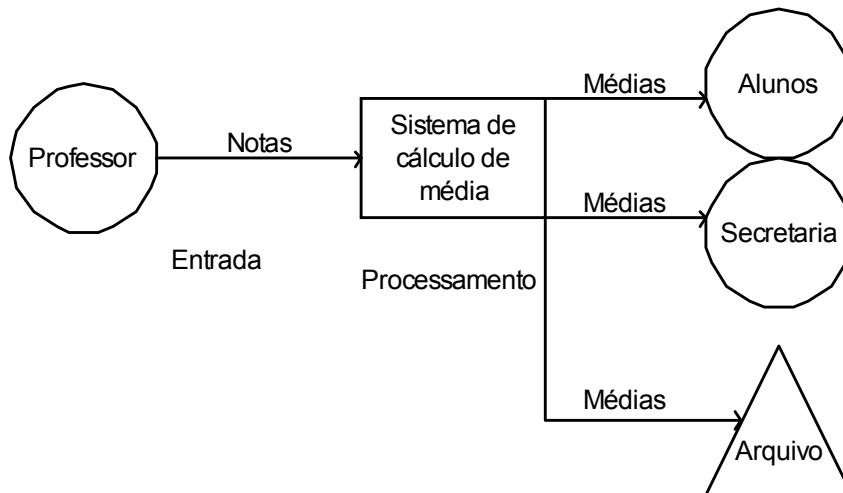


Depósito de Dados: Representa o armazenamento de informações dentro do sistema.

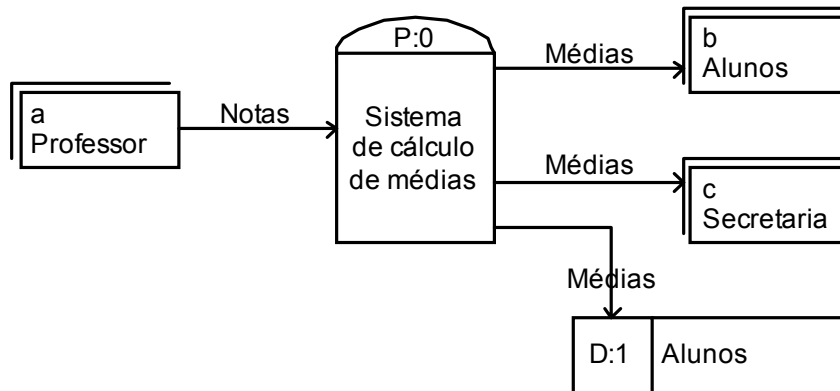


Modelo de um Sistema





Exemplo de D.F.D.:



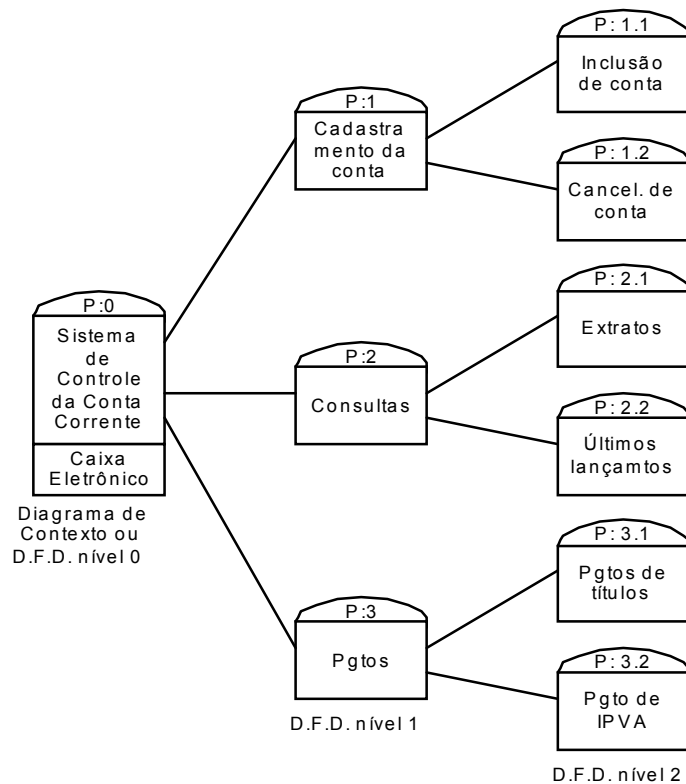
Níveis de Detalhamento no D.F.D.

O D.F.D. é representado em vários níveis onde:

1º nível) Diagrama de Contexto ou D.F.D. nível 0: possui só um processo e representa uma visão global do sistema, sem detalhes. Este nível define as fronteiras do sistema.

2º nível) D.F.D. nível 1: representa as macro-funções do sistema.

3º, 4º... níveis) D.F.D. de explosão de processos com níveis de 2 a n: representam o detalhamento do nível acima. Nestes níveis temos uma visão detalhada da lógica do sistema.



Exemplo de D.F.D. nível 0

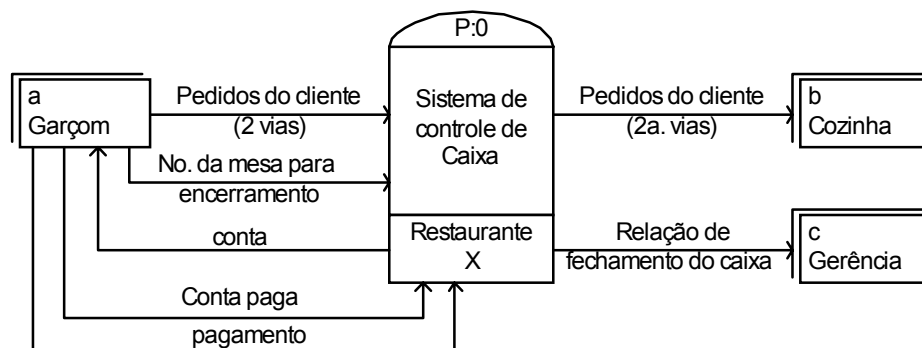
Sistema de controle de caixa de um restaurante (Sistema não informatizado)

O garçom entrega os pedidos dos clientes (2 vias) para o caixa que, registra os preços dos itens nos pedidos, consultando o cardápio e guarda a 1ª via no escaninho correspondente à mesa e encaminha a 2ª via para a cozinha.

O garçom informa ao caixa o nº da mesa para fechamento e o caixa obtém todos os pedidos da mesa para totalizar e emitir a conta para o garçom entregar ao cliente.

O garçom retorna com a conta e o pagamento para o caixa, que registra o pagamento e armazena as contas pagas.

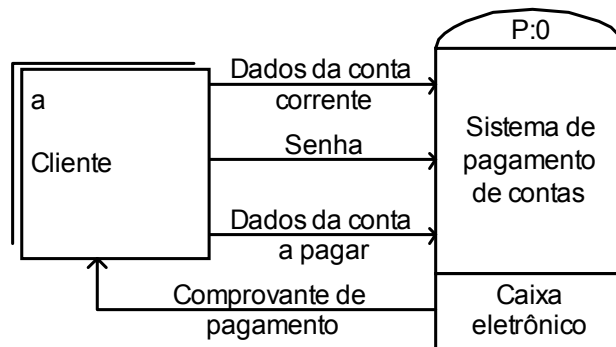
Diariamente, o caixa gera a relação de fechamento do caixa para a gerência.



Regras para representação do DFD nível 0

- Este nível só terá 1 processo com referência P:0 e o nome do sistema;
- Não representar depósitos de dados;
- Representar todas as entradas e saídas do sistema.

Exemplo:



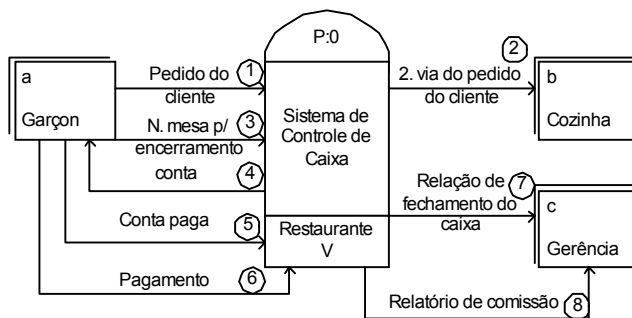
DFD nível 1

Exemplo: Sistema de Controle de Caixa Restaurante

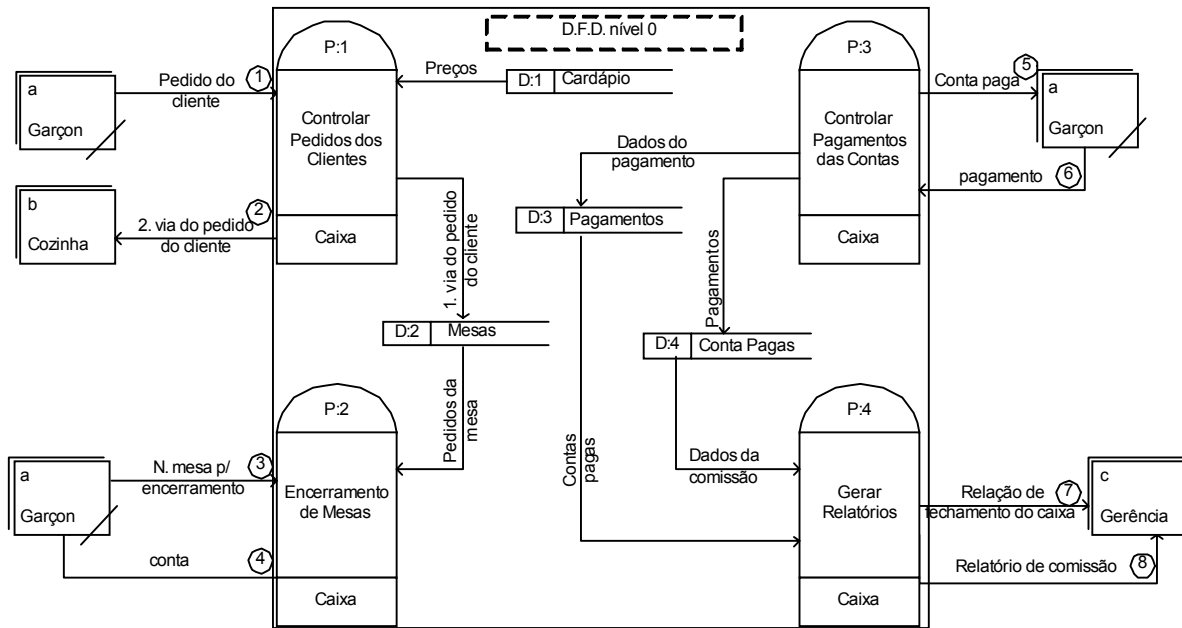
Macro-Funções:

- P:1 – Controlar Pedidos dos Clientes
- P:2 – Encerramento de Mesas
- P:3 – Controlar pagamentos das contas
- P:4 – Gerar Relatórios

D.F.D. nível 0:



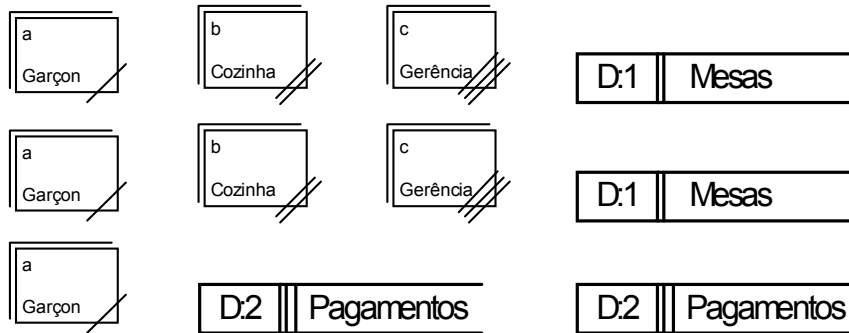
D.F.D. nível 1:



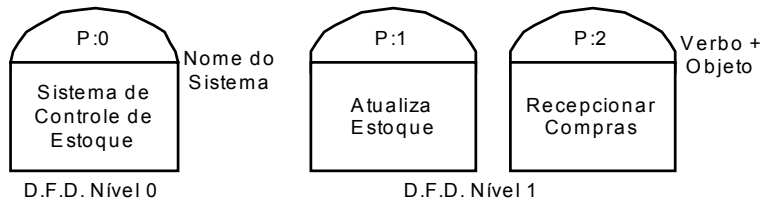
Regras para representação de DFD nível 1

- a) Repetição de símbolos: para melhor visualização do DFD, poderemos repetir entidades externas e depósitos de dados.

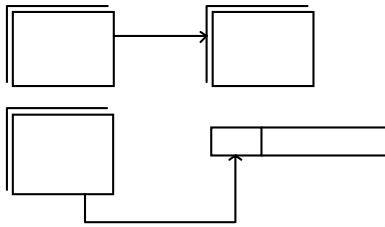
Exemplo:



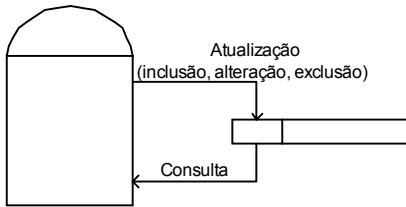
- b) Os fluxos de dados (entradas e saídas do sistema) e as entidades externas, dever ser compatíveis com o nível 0;
- c) No processo de nível 0, escrevemos o nome do sistema e no processo do nível 1, escrevemos um verbo + objeto referente à função do processo;



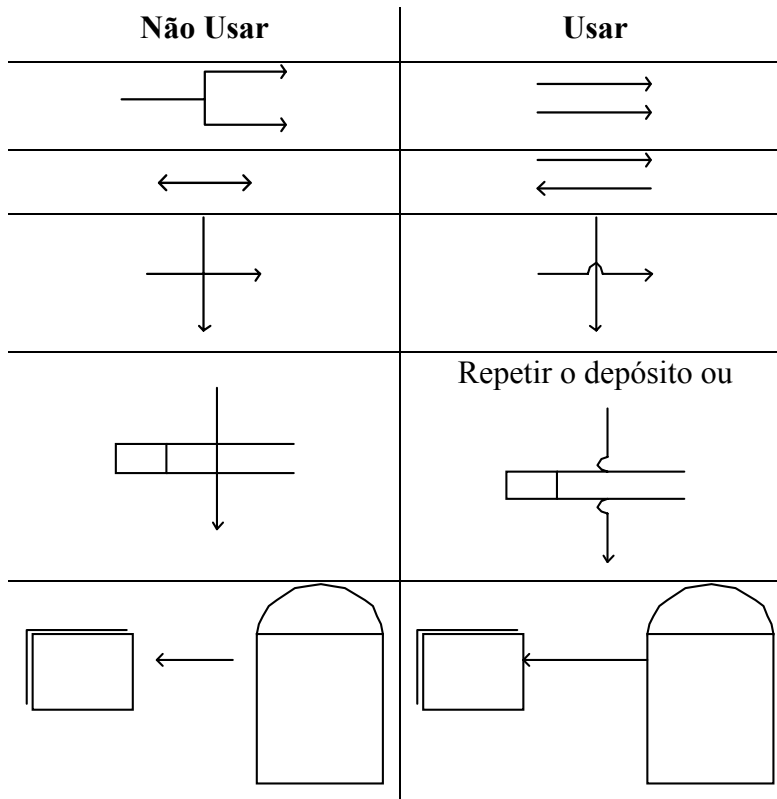
- d) No D.F.D., nunca haverá ligação do tipo:



e) Somente os processos têm acesso aos depósitos de dados e os processos só podem realizar duas ações no depósito:



f)



Exemplo de DFD nível 0 e 1

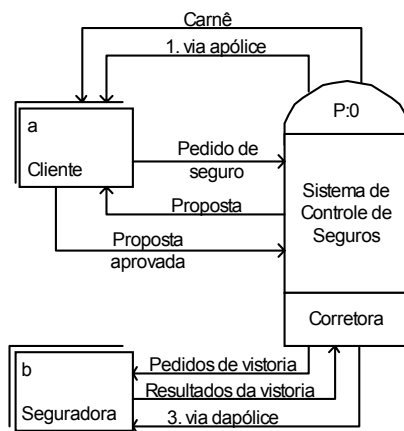
Sistema de Controle de Seguros

A corretora executa as seguintes funções:

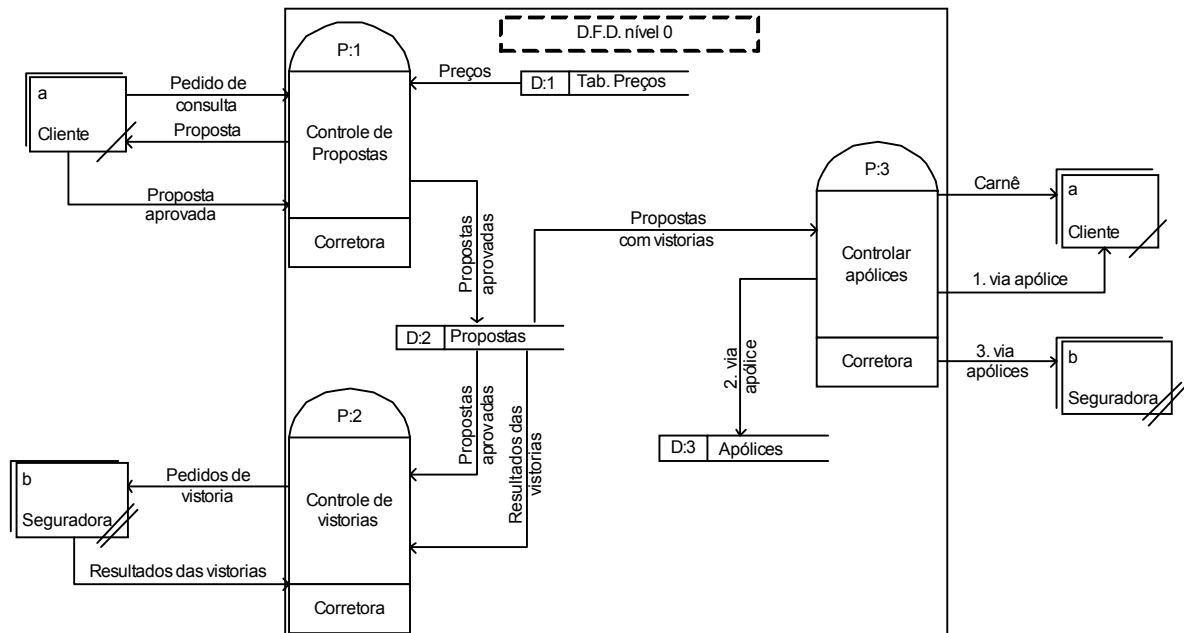
- Recebe o pedido de seguro, do cliente e consulta uma tabela de preços para fazer os cálculos e emitir uma proposta para o cliente;

- O cliente devolve a proposta aprovada, que é arquivada;
- Diariamente, a corretora obtém as propostas aprovadas e emite um pedido de vistoria, para cada proposta, que será enviado para a Seguradora;
- A Seguradora devolve os resultados das vistorias realizadas, que são registrados junto com as propostas;
- Diariamente, a Corretora emite, para as propostas com as vistorias realizadas, um carnê e uma apólice em 3 vias;
- A 1ª via da apólice e o carnê são enviados para o cliente, a 2ª via da apólice é arquivada e a 3ª via é enviada para a seguradora.

Nível 0



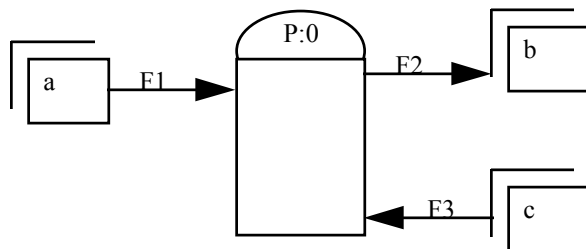
Nível 1



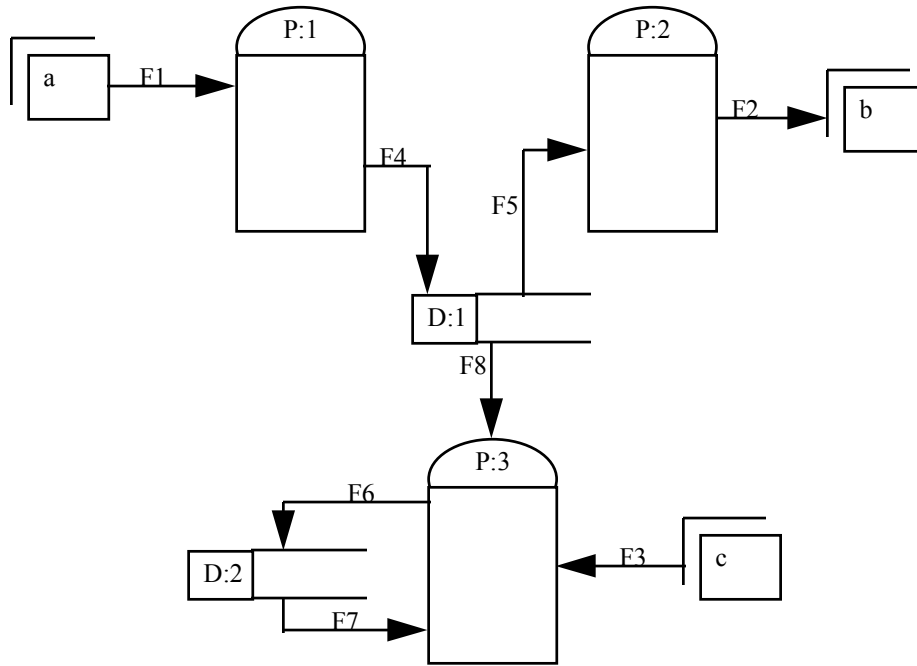
DFD Nível 2

Exemplo Hipotético:

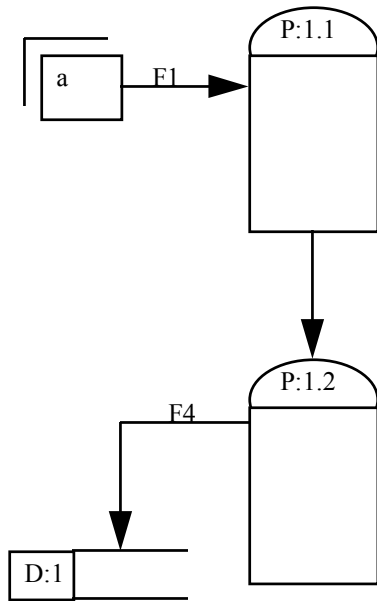
Nível 0:



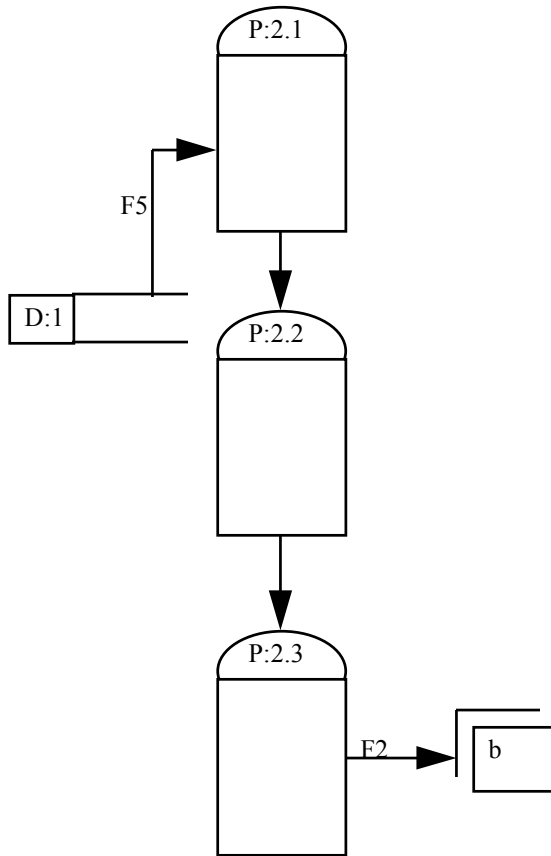
Nível 1:



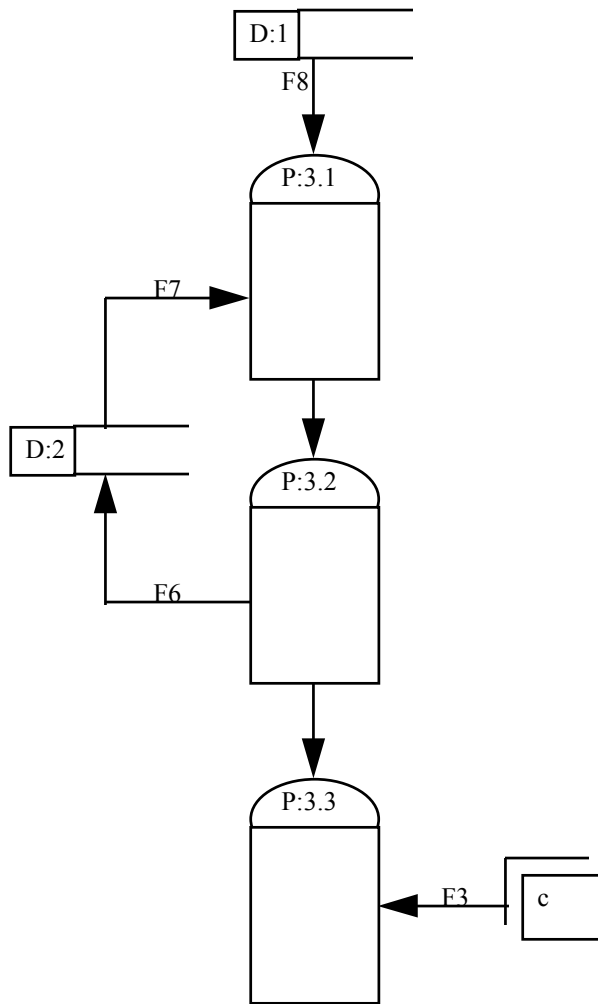
Nível 2 - P:1:



Nível 2 - P:2:



Nível 2 - P:3:



Sistema de Controle de Consultas de uma Clínica Médica

A recepção da clínica realiza as seguintes funções:

- Recebe o pedido de consulta, do cliente, e verifica na agenda a disponibilidade de datas e horários para informar ao cliente;
- O cliente informa a data e horário de preferência e a consulta é agendada;
- Quando o cliente chega para a consulta, ele informa seu nome/horário e médico para a recepção que dá baixa na agenda;
- Ainda na chegada, a recepção solicita os dados completos do cliente para cadastrá-lo e emitir uma ficha de acompanhamento que será entregue para o médico;
- Depois da consulta realizada, o médico devolve a ficha com o diagnóstico e uma lista de serviços e medicamentos utilizados para a recepção calcular o total da consulta, consultando uma tabela de preços;
- O cliente é informado do valor da consulta e efetua o pagamento;

- A recepção, emite um recibo para o cliente e registra o pagamento em um livro caixa;
- A ficha de acompanhamento é armazenada junto com o cadastro do cliente e a lista de serviços e medicamentos é arquivada também.

Controle de Pedidos

Representar o DFD nível 0 e nível 1 para o sistema de controle de pedidos da empresa CBE (Companhia Brasileira de Eletrônicos):

- O cliente faz o pedido de dispositivos eletrônicos ao Dpto de Pedidos da CBE e este faz a verificação dos itens pedidos no catálogo de produtos e atualiza o cadastro de clientes quando for feita a expedição dos produtos. Os pedidos são arquivados.
- Quando o lote de pedidos atinge uma quantidade onde a CBE possa ter grandes descontos com o fornecedor, o Dpto de Pedidos gera uma Requisição de Compras (RC), em 2 vias, a partir do lote de pedidos: a 1ª via da RC é enviada ao fornecedor e a 2ª via é arquivada.
- Os fornecedores enviam, ao Dpto de Pedidos, juntamente com os produtos, uma guia de remessa e uma nota fiscal que são conferidas com a 2ª via da RC. A nota fiscal conferida é enviada ao Setor de Contas a Pagar e a guia de remessa conferida é utilizada para marcar, nos pedidos dos clientes, os itens que foram atendidos. Depois a guia de remessa é arquivada.
- Todos os dias, o Dpto de Pedidos, consulta no arquivo de pedidos de clientes, quais os pedidos que já estão completos para a expedição e gera, para os pedidos completos, um aviso de cobrança (2 vias) consultando o cadastro de clientes; dá baixa nos pedidos completos, envia a 1ª via do aviso de cobrança ao cliente e a 2ª via para Setor de Contas a Receber.

Análise & Projeto Orientados a Objetos

Conceitos de Orientação a Objetos

Diagrama de Fluxo de Dados (DFD): Ferramenta para modelagem estruturada. O sistema é modelado de acordo com suas funcionalidades (o que tem de ser feito).

Processos: indicam ações a serem feitas.

Análise e Projeto Orientados a Objetos: O sistema é modelado de acordo com os elementos observáveis no domínio de aplicação.

Classes: idéia genérica extraída a partir dos atributos comuns dos objetos observáveis, um molde de objetos.

Objeto: é uma abstração (modelo) de um objeto (qualquer elemento observável, tangível ou não) do mundo real. Um objeto é uma instância (exemplar) de uma classe.

Instanciação: a partir do modelo, criar uma representação do objeto do mundo real.

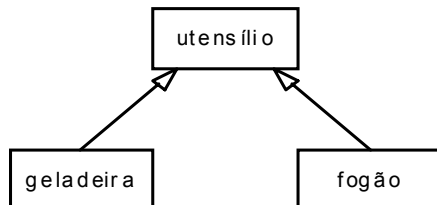
Definições

- **Domínio da aplicação:** onde o sistema irá funcionar?
- **Limites do sistema:** o que o sistema irá fazer?

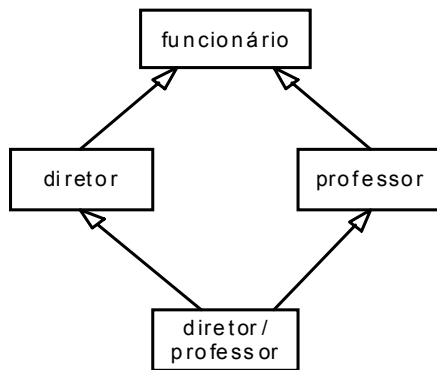
Exemplo: Funcionalidade do sistema: controlar folha de pagamento.

Relacionamento de herança

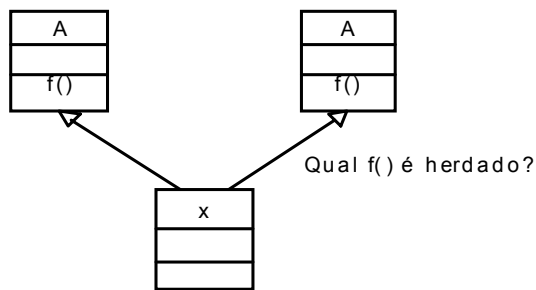
- Relação “é um”.
- Sentido de generalização (classe-mãe ou superclasse).
- Sentido de especialização (classes-filhas ou subclasses ou classes derivadas).
- Árvore de herança: herança simples



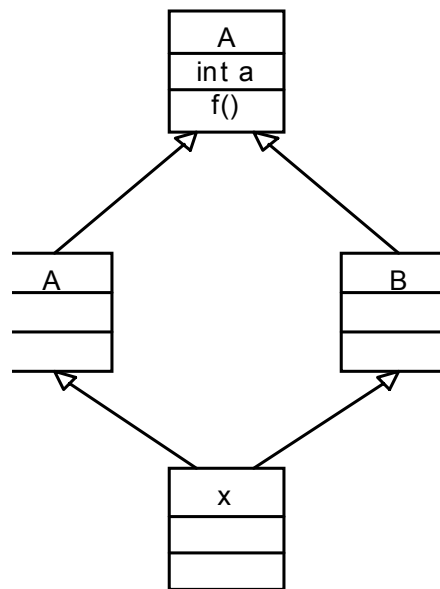
- Grafo de herança: herança múltipla



- Problemas:
- Nem toda linguagem suporta (ex. Java).
- Colisão de nomes



o Herança repetida



- A herança deve ser (bem) utilizada, sempre que possível, pois possibilita o *reuso de código*.
- Tomar cuidado para não “misturar” conceitos diferentes em uma só classe (ex. “prato”: herda de “utensílio” ou “comida”?)
- Usar herança múltipla cuidadosamente, tentando evitar que ela possa causar.

Pensar em casa:

Partindo de um domínio de aplicação, tentar obter classes para diferentes funcionalidades (limites do sistema).

Domínio de Aplicação	Funcionalidade 1	Funcionalidade 2
Futebol	Controle de times para um campeonato estadual.	Registro de informações de times para locutores de rádio e TV

Banco	Controle de correntistas com cheque especial	ATM (Caixa Automático)
-------	--	---------------------------

Exercícios

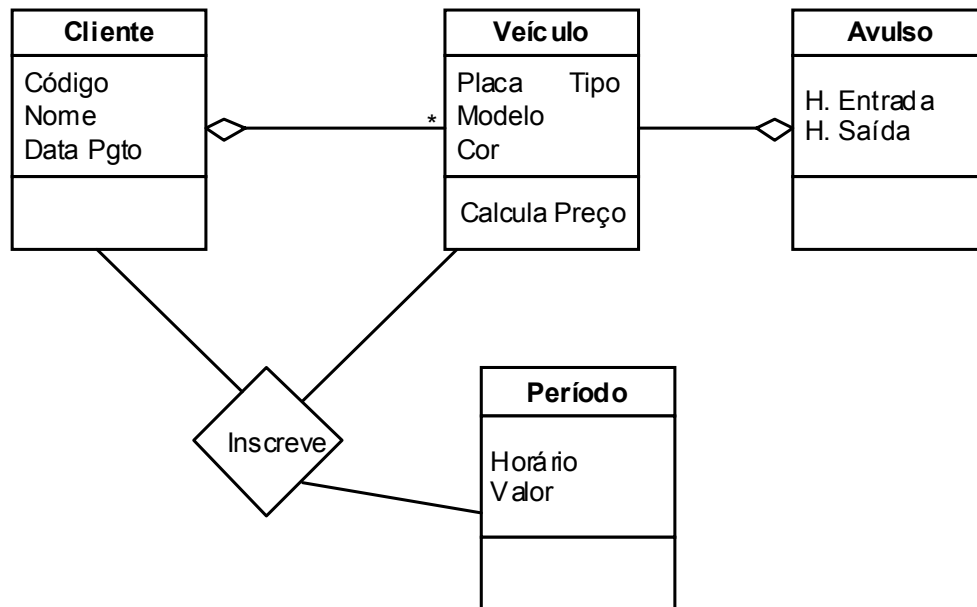
1. Determinar os relacionamentos possíveis entre as classes a seguir:

a) Domínio de aplicação: Estacionamento (genérico)

Funcionalidade: controle de veículos

Classes observáveis:

- Cliente
- Comprovante
- Ticket
- Carro
- Carro de Passeio
- Moto
- Veículo de Carga
- Seguro
- Pagamento
- Manobrista
- Mensal
- Avulso
- Diurno
- Noturno
- Integral



b) Domínio de aplicação: loja de porcelanas de fábrica

Classes Observáveis:

- Peça
- Preço
- Louça
- Fabricante
- Prato
- Xícara
- Pires
- Cor
- Jogo de Café
- Aparelho de jantar
- Cheque
- Pagamento
- Código-peça
- Nota Fiscal
- Cartão de crédito

c) Domínio de aplicação: montadora de veículos

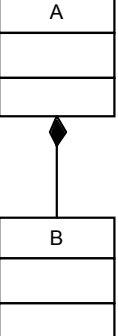
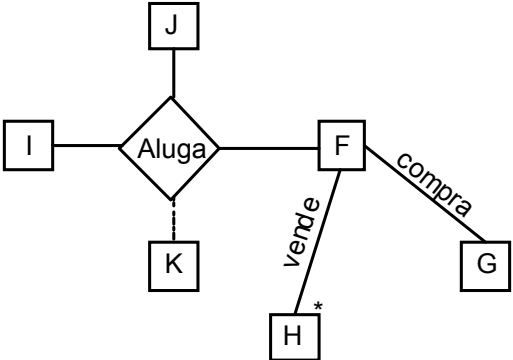
Classes Observáveis:

- Veículo

- Funcionário
- Greve
- Salário
- Bônus
- 13º salário
- Férias
- Motor
- Rodas
- Freio
- Pneu
- Capô
- Porta
- Volante
- Air-bag
- Suspensão

Relacionamentos

Nome	Símbolo UML	Implementação
Herança (é um)		Depende da linguagem ex.: extends – Java <pre>class A { }</pre> <pre>class B extends A</pre> : em C++ <pre>class B: public A</pre> <pre>{ }</pre>
Agregação (parte de)		Criam-se objetos de uma classe com atributos de outra classe <pre>class C</pre> <pre>{ D d;</pre> <pre>E e[] }</pre>

<p>Composição (parte de)</p> <p>Na composição, a existência dos objetos “contidos” depende da existência do objeto que contém</p>		
<p>Associação (outros relacionamentos mais genéricos)</p> <p>Advém do relacionamento entre outras classes</p> <p><i>K</i> é fruto do relacionamento de <i>I</i> e <i>J</i> com <i>F</i>.</p>		<p>Referências a objetos de uma classe são passados como parâmetros para métodos da outra classe</p> <pre>class F {public void comprar (G g){} public void vende (H h[]){} public K aluga (I i, J j){}}</pre>

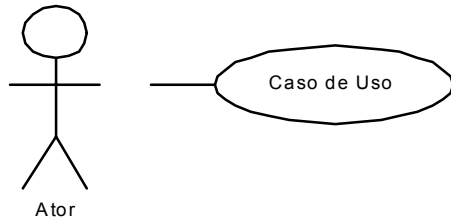
Roteiro:

- Declaração de requisitos
 - Análise de domínio de aplicação
 - Verificação da funcionalidade
 - Diagrama de casos de uso
- Levantamento das classes candidatas
- Estabelecimento de classes (atributos e métodos)
- Estabelecimento de relacionamentos entre classes
- Diagrama de classes

Diagramas de casos de uso (use cases)

Caso de uso: Descreve uma forma através da qual um dado agente (usuário) atua no sistema.

Sintaxe:



Ator: Os potenciais usuários do sistema.

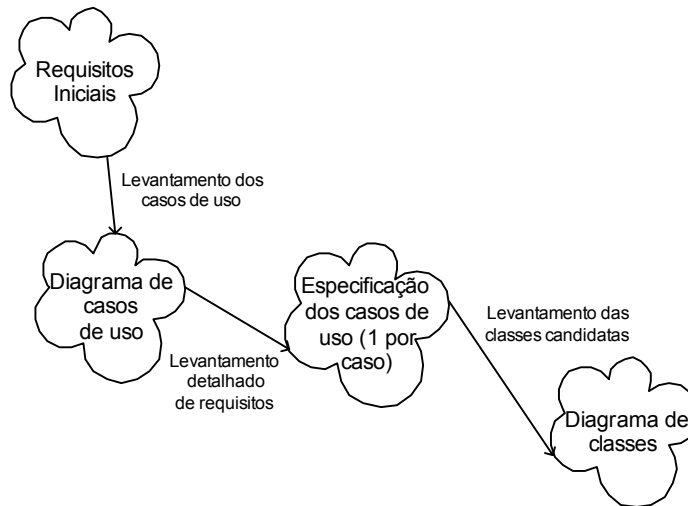
Casos de uso: formas através das quais o sistema é utilizado.

Um diagrama de casos de uso é diferente de um DFD, apesar de também mostrar as fronteiras do sistema, bem como os agentes que nele atuam.

Ao passo que processos em DFDs mostram as *funcionalidades* do sistema, casos de uso mostram como o sistema é usado.

Uso recomendado para facilitar o processo de “detecção” de classes.

“User-centered view”: visão do sistema tendo o usuário como o centro.



Exemplo:

Declaração de requisitos iniciais:

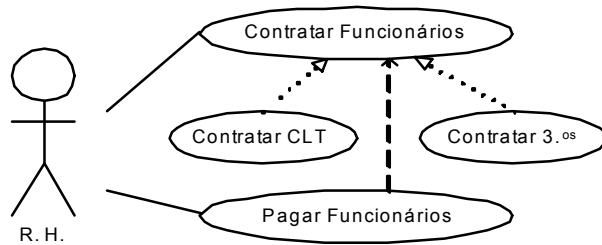
Elaborar um sistema de contratação e pagamento de funcionários, via CLT ou terceirização, para a Raposa’s Consultoria.

- Análise & Projeto Estruturados:
 - Funcionalidade
 - Contratar
 - Pagar
- Análise & Projeto Orientado a Objetos
 - Elementos observáveis – objetos – classes

Casos de uso:

Atores	Casos de Uso
Recursos Humanos	Contratação de funcionários Pagamento de funcionários

Diagrama de casos de uso:



Especificação de caso de uso “Contratar Funcionários”

Ator(es)¹: Recursos Humanos

Pré-Condições²: -----

Pós-Condições³: Funcionário contratado.

Descrição: Para contratar um funcionário, devem ser obtidos os dados relativos a: nome, endereço (rua, bairro, cidade, estado, CEP), telefone, e-mail, n.º CTPS*, RG e CPF. O funcionário, de acordo com a função, deve ser terceirizado ou contratado via CLT.

Se terceirizado...

Senão se CLT...⁴

Exceções⁵: (*)

Diagrama de Classes x Diagrama de Casos de Uso

- **Diagrama de Classes:** foco em estrutura de informação dentro do sistema.
- **Diagrama de casos de uso:** foco nas formas pelas quais o sistema é utilizado (dados + funcionalidades relevantes).

¹ Um mesmo caso de uso pode ser relacionado a mais de um ator.

² Requisitos necessários para a execução do caso de uso (Aponta dependência entre casos de uso).

³ “Estado” do sistema após a execução dos casos de uso.

⁴ Quando um caso de uso toma direções diferentes de acordo com uma certa condição, pode ser melhor “quebrá-lo” em casos de uso mais específicos.

⁵ Exceções devem ser indicadas no corpo da descrição, mas por motivos de clareza, devem ser tratados à parte.

Ator x Classe

Um ator não necessariamente é representado por uma classe no Diagrama de Classes, embora isso possa acontecer.

Caso de Uso x Método

Não necessariamente um Caso de Uso é implementado como um método de uma classe. O mais provável é que o Caso de Uso indique vários métodos para serem implementados em classes diversas.

Inclusão de Classes – Containeres

Para geração de listas de relatórios.

Verificar se as classes merecem ser classes

Valores de atributos não diferenciam classes!

Ponto de Vista do Usuário

- Diagrama de casos de uso
- Especificação dos casos de uso

Visão Estrutural

- Diagrama de classes

Visão Dinâmica

- Diagramas de Iteração
 - Diagrama de Seqüência
 - Diagrama de Colaboração
- Modelam a iteração entre objetos do sistema.

Diagrama de Seqüência

Ilustra a *ordem* das iterações entre atores e objetos do sistema, bem como as interações entre objetos.

Cada interação é representada por uma *mensagem* que é passada entre atores/objetos ou objetos.

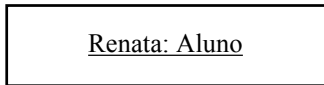
Em geral, uma mensagem é uma chamada de um método ou função.

Sintaxe para objetos

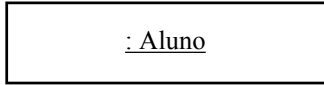
<u>nome obj: nome-classe</u>

Exemplo:

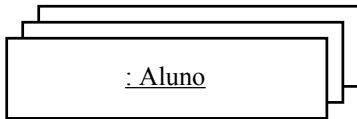
Objeto Renata da classe “Aluno”



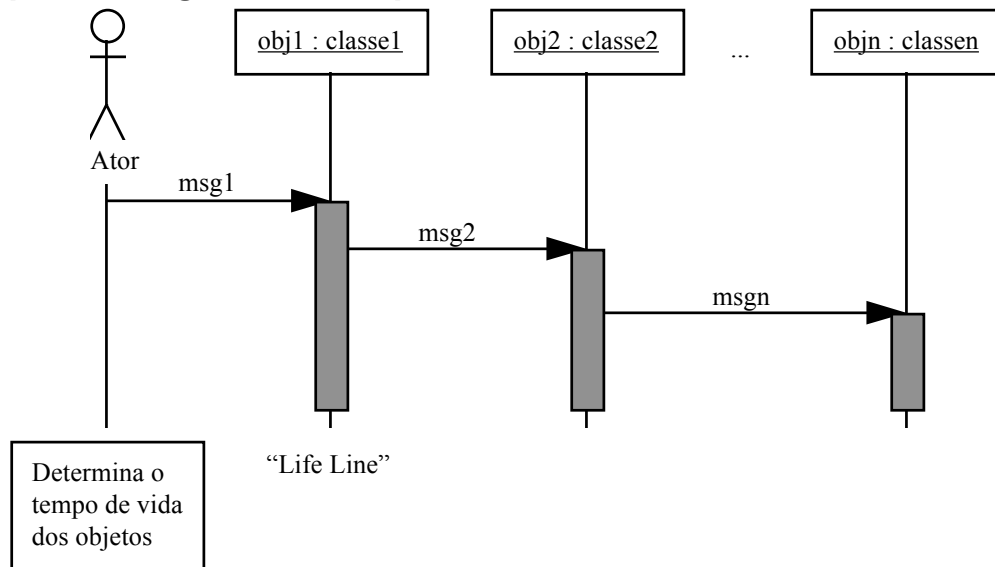
Objeto anônimo da classe “Aluno”



Vários objetos anônimos da classe “Aluno”

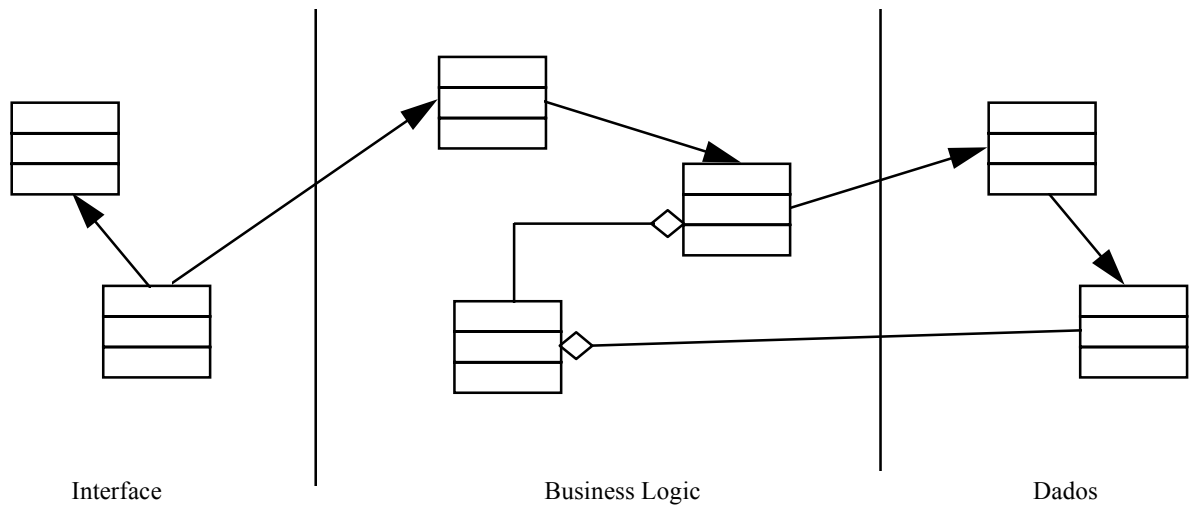


Exemplo de Diagrama de Seqüência



Alguns detalhes sobre a sintaxe do Diagrama de Seqüência

- Alguns objetos desempenham, claramente, um papel bastante específico em se considerando o modelo clássico de 3 camadas:

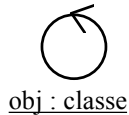


- Representações alternativas (esteótipos) para objetos:

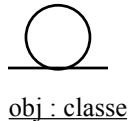
- Objetos da camada de interface:



- Objetos da camada de negócio:

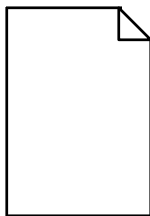


- Objetos da camada de dados:

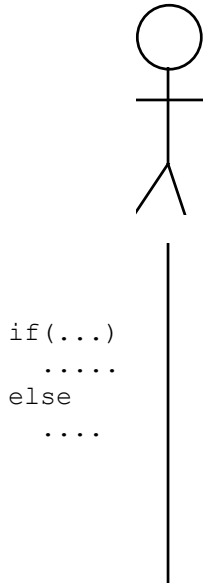


- Uma mensagem pode apresentar parâmetros.
- Não é usual representar *retornos* de mensagens (podem “poluir” o diagrama), embora não seja errado.
- Podem ser inseridas notas (comentários) como em qualquer diagrama UML

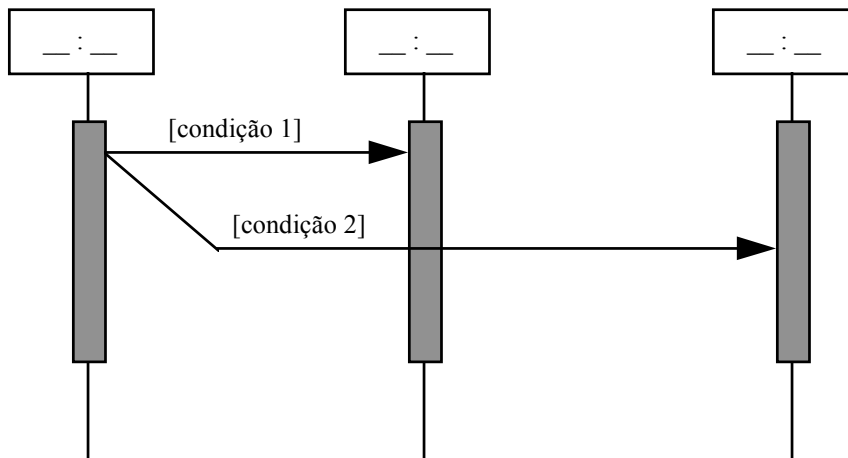
- Sintaxe:



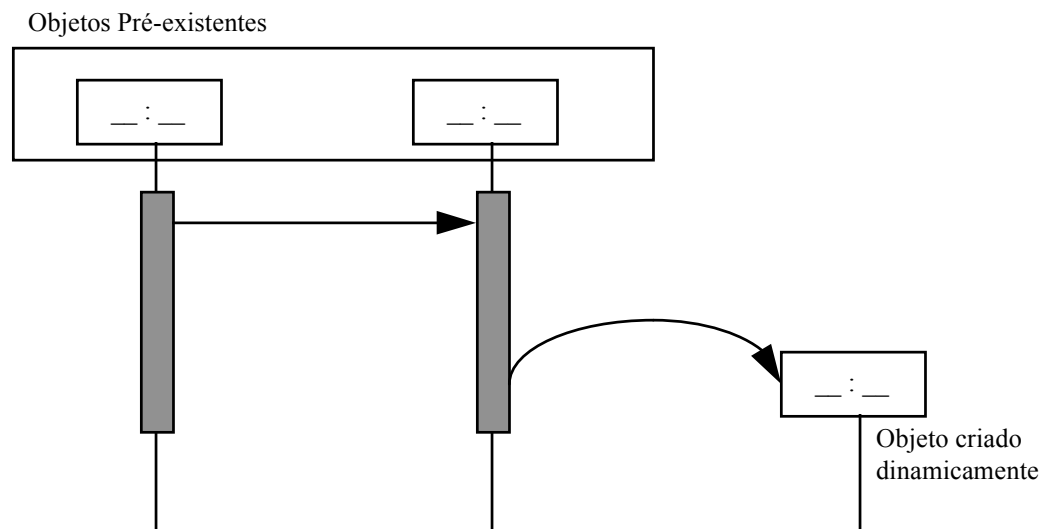
- Pode ser interessante representar trechos de código, caso isso ajude na clareza. Em geral, isso é feito próximo à “lifetime” do ator.



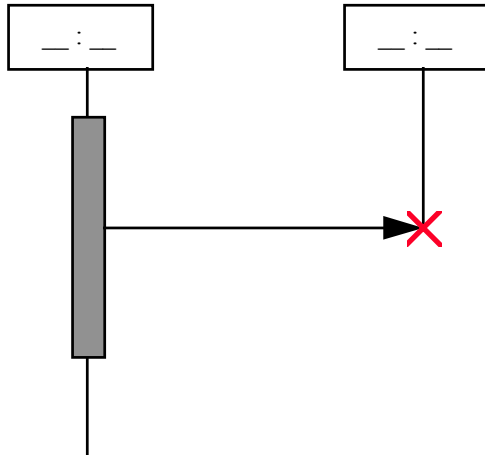
- Condições (if): representadas, em geral, por mensagens distintas partindo de um mesmo ponto.



- Criação (instanciação) dinâmica de objetos:



- Destruição de Objetos



A destruição depende de mecanismos explícitos (delete, dispose, ...) ou automáticos (garbage collection).

Quando e como usar o Diagrama de Seqüência?

Para cada caso de uso, detectar *cenários*. (Execuções específicas do sistema; situações determinadas dentro de um caso de uso, geralmente disparadas por eventos).

Cada caso de uso pode ter vários cenários; cada cenário corresponde a um diagrama de seqüência.

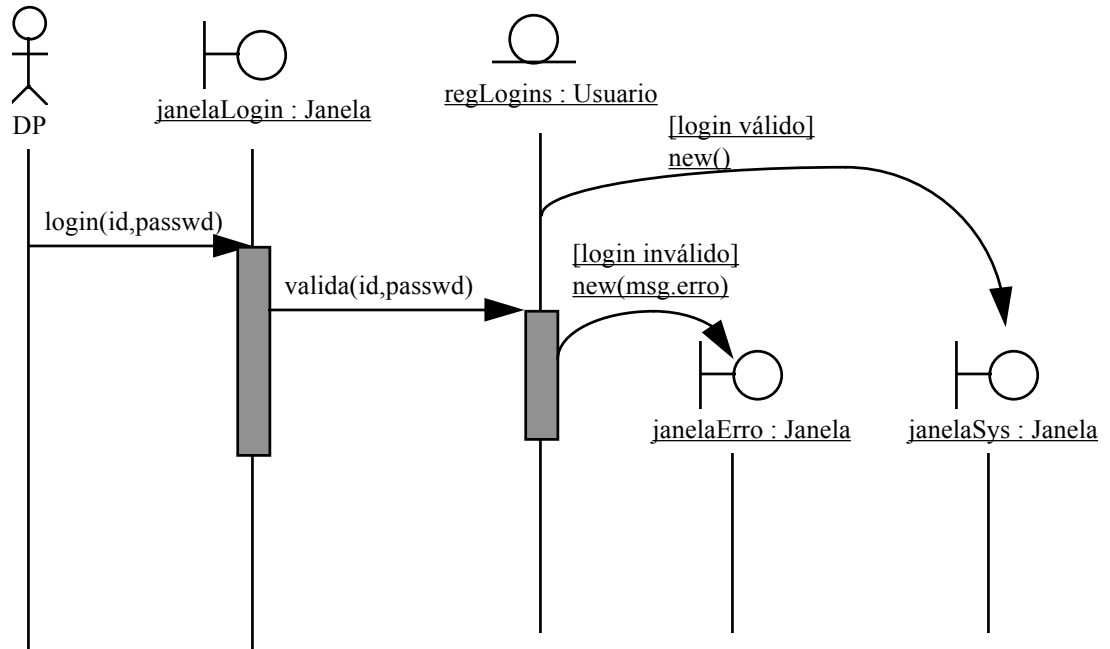
Exemplo:

Caso de uso “Controlar Funcionários” (Ator: DP)

Cenários:

DP	Faz login no sistema
DP	cadastra novo funcionário
DP	demite funcionário
DP	altera dados do funcionário
...	...
Alguém	Faz Algo

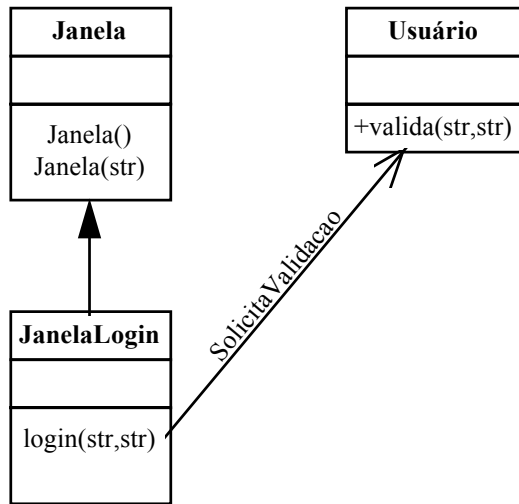
Diagrama de seqüência para o cenário “DP faz login no sistema”



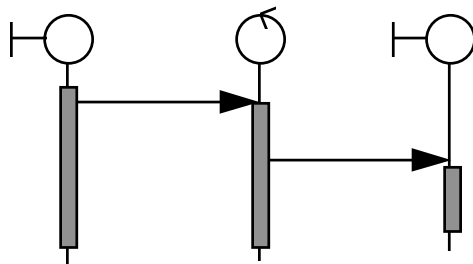
Algumas conclusões a partir do diagrama de seqüência:

- Devem existir as classes “Janela” e “Usuário” no diagrama de classes.
- Devem ser incluídos métodos relacionados às mensagens.

Exemplo:



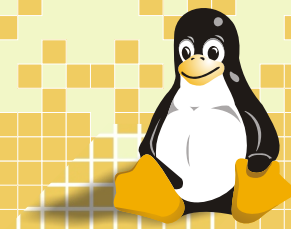
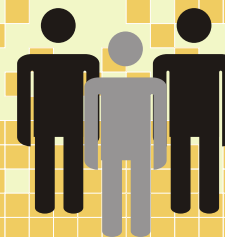
“Desconfiar” quando as camadas forem “ultrapassadas”



Se dois objetos interagem no diagrama de seqüência, as classes correspondentes devem se relacionar no diagrama de classes.

COMUNIDADE
COMUNIDADE
COMUNIDADE

tecnociência
tecnologia e ciência



A ciência e a prática já provaram: "Só evoluímos quando vivemos em comunidade!"

Alunos e professores, esta comunidade espera por vocês, venham participar!

Artigos, trabalhos, foruns, jornais. sistema de mensagem, tutoriais, debates com alunos de outras universidades, novos amigos, tirar dúvidas sobre as matérias, enfim...

INTERAGIR
INTERAGIR
INTERAGIR

Áreas de Bases Tecnológicas

Ciência da Computação

Eng. de Telecomunicações

Mecatrônica

Automação

Eng. Eletrônica

Eng. Elétrica

Robótica

Informática

Ciências Espaciais

acesse

www.tecnociencia.com.br



e faça parte desta comunidade

APOIO

