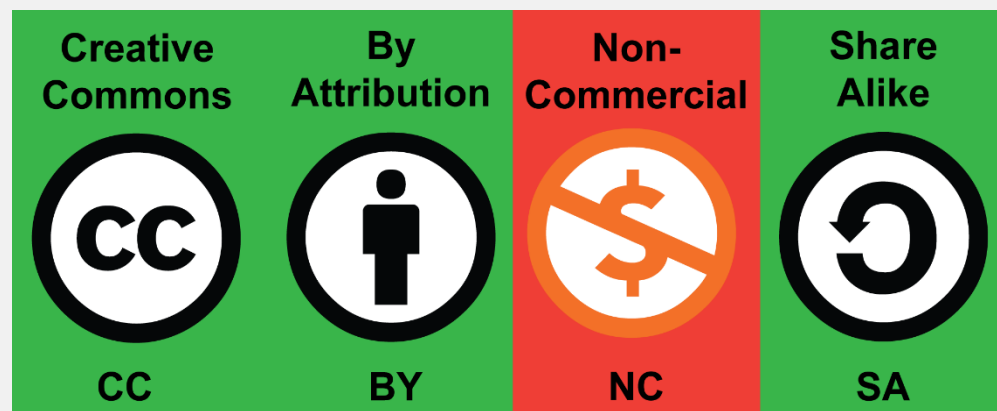


Análise Essencial

Prof. MSc. Jackson Antonio do Prado Lima
jacksonpradolima at gmail.com / jackson.lima at udesc.br

Departamento de Sistemas de Informação – DSI

Licença



Este trabalho é licenciado sob os termos da Licença Internacional Creative Commons Atribuição-NãoComercial-CompartilhaIgual 4.0 Internacional (**CC BY-NC-SA 4.0**)

Para ver uma cópia desta licença, visite
<http://creativecommons.org/licenses/by-nc-sa/4.0/>

Histórico de Modificação

- Esta apresentação possui contribuição dos seguintes professores:
 - Fabio Fernando Kobs (UDESC)
 - Jackson Antonio do Prado Lima (UDESC)

Agenda

- Introdução
- MiniMundo
 - Exemplos 1, 2 e 3
- Análise Essencial
 - Principais Ferramentas de Modelagem
 - Diagrama de Contexto (DC), Diagrama de Fluxo de Dados (DFD), Diagrama de Entidade e Relacionamento (DER), Diagrama de Transições de Estado (DTE) e Diagrama de Estrutura (DE)
 - Ferramentas complementares
 - Dicionário de Dados (DD)
 - Especificação de Processo
 - Árvore de Decisão, Tabela de Decisão e Português Estruturado.
 - Etapas do Projeto de Sistemas

Introdução

Tipos de Metodologias:

- Análise Clássica;
- Análise Semi-Estruturada;
- Análise Estruturada;
- **Análise Essencial;**
- Análise Orientada a Objetos.

Introdução

- Análise Essencial é uma **evolução/refinamento** da Análise Estruturada, substituindo-a como metodologia de análise, podendo ser utilizada até os dias de hoje.
- O problema existente (ou situação que requer a informatização) é **estudado, porém não é modelado**: os esforços são concentrados na identificação das funcionalidades **lógicas** requeridas para o software que será criado e, a partir daí, cria-se um **modelo essencial** do software que será desenvolvido, não se incorporando as **exigências físicas**.

Introdução

- Na Análise Essencial, a premissa básica é descrever o sistema de maneira **independente de restrições tecnológicas**, o que permitirá uma solução ideal ao problema, sem deixar influenciar-se por questões decorrentes das restrições, o que poderia antecipadamente impor alguma restrição à solução pensada.

Introdução

- Isso implica dizer que devemos considerar na confecção do modelo essencial a existência de uma tecnologia perfeita. Deve-se entender esse aspecto como uma abstração em que se supõe uma tecnologia ideal, sem limitações, onde:

Introdução

- Os custos, o consumo e o desgaste dos equipamentos são zero.
- A capacidade de armazenamento de dados do sistema é infinita.
- A velocidade dos processadores é infinita.
- O tempo de acesso a dados é instantâneo.
- Zero erro (não ocorrem falhas).

Introdução

- Antecedendo a aplicação do método da análise essencial faz-se um exame do domínio do problema (levantamento de requisitos, buscando-se funcionalidades e dados exigidos ao sistema que será desenvolvido) inicialmente focando os aspectos mais essenciais pertinentes ao problema.
- De posse do conhecimento sobre aquilo que se deseja resolver com o desenvolvimento do software, aplica-se o método da análise essencial.

Introdução

- Na análise essencial um sistema de informação é visto como um sistema de resposta planejado.
- Atividades planejadas com resposta para eventos do ambiente são projetadas em um sistema.
- Os eventos no ambiente geram fluxos de dados (estímulos) para o sistema, os quais acionam ações (ativa-se processos que são alimentados com os dados), que podem, por sua vez, gerar respostas internas (persistência de dados) ou respostas que retornam ao ambiente (relatórios, e-mails etc.).

Introdução

- Também há possibilidade de ocorrência de eventos internos ao sistema, os quais geram fluxos temporais, que também acionam ações no sistema.

Introdução

Características da Análise Essencial

- Continuou com o levantamento através de “minimundo”;
- Manteve os Modelos de Projeto da Análise Estruturada (DFD, DER, DTE, DD, e Especificação Lógica de Processos);
- Criou regras para gerar **Atividades** mais resumidas e independentes, para melhorar a compreensão do Sistema.

Introdução

Características da Análise Essencial

- Emprega uma metodologia orientada a **Eventos**
 - Um **evento** pode ser definido informalmente como um acontecimento do mundo externo, que requer uma resposta do sistema;
 - Objetiva compreender melhor o sistema, e desvincula a tecnologia usada do que realmente é importante que sejam as rotinas e os dados;
 - Um **estímulo** é um ativador de uma função do sistema. E a forma como o evento age sobre o sistema;
 - Uma **resposta** é o resultado gerado pelo sistema devido à ocorrência de um evento.

MINIMUNDO

MiniMundo

- Descrição do problema que o sistema irá tratar;
- Apresentação formal para definição de propostas de sistemas;
- Um sistema de informação modela um MiniMundo;
- Este mundo é uma versão **abstraída** do mundo normal, onde os únicos detalhes que importam são os relacionados com a aplicação, com o que o sistema faz.

MiniMundo

1. Objetivo do sistema
2. Procedimentos atuais
3. Necessidades do usuário (requisitos)
4. Funcionalidades do sistema
5. Identificação de usuários
6. Interface com outros sistemas

MiniMundo

- 7. Benefícios
- 8. Fatores críticos (pontos de atenção)
- 9. Limitações
- 10. Prazos
- 11. Custos

MiniMundo

1. **Objetivo do sistema:** objetivo, em linhas gerais, do sistema a ser construído
2. **Procedimentos atuais:** se necessária, uma breve descrição de como o processo a ser automatizado é realizado no ambiente onde o sistema será inserido
3. **Necessidades levantadas pelo usuário:** lista de requisitos levantados pelo usuário que ele considera relevantes para o sistema

MiniMundo

4. **Descrição geral das funcionalidades do sistema:** lista das funções gerais do sistema
5. **Identificação dos usuários:** identificação dos papéis e responsabilidades de cada usuário do sistema
6. **Interface com outros sistemas:** relação de outros sistemas que podem se relacionar ao sistema em construção, indicando as informações que fluem entre eles

MiniMundo

7. **Benefícios esperados com o Sistema:** lista de melhorias em qualidade e produtividade para o processo sendo automatizado, em relação ao procedimento atual
8. **Fatores críticos de sucesso:** fatores considerados críticos para que o sistema seja bem aceito e utilizado com eficiência
9. **Limitações:** previsão de limitações ao uso do sistema

MiniMundo

- 10. **Prazos para desenvolvimento:** definição da duração do desenvolvimento do Sistema
- 11. **Custos:** inclui pessoal, soft, hard, etc...
- Pode ser utilizado como proposta comercial do projeto.

MiniMundo

Exemplo de Objetivo

- Um atendente realiza o cadastro do cliente através do sistema, mediante a apresentação de seus principais documentos e de um informe de residência.
- Uma vez cadastrado, o cliente poderá consultar, solicitar a reserva ou mesmo alugar produtos (filmes e jogos eletrônicos), num total de até três produtos por aluguel registrado no sistema pelo atendente.
- Em caso de reserva, o cliente poderá realizar via telefone ou presencialmente, entrando numa fila de espera mantida pelo atendente que a controla através do sistema.

MiniMundo

Exemplo de Objetivo

- Cabe ao atendente via telefone ou através de mensagem pelo próprio sistema enviar ao cliente o aviso de que o produto solicitado está disponível para ser alugado.
- Ao entregar os produtos e realizar o pagamento, o cliente tem sua baixa de aluguel realizada pelo atendente, que registra a devolução no sistema e libera os mesmos produtos para que outro cliente os alugue.
- Ao final de cada dia, o atendente gera o relatório de aluguéis realizados, e o envia para seu gerente.

MiniMundo

Exemplo de Funcionalidades

Módulo Pessoas

1. O Sistema deverá ser capaz de gerenciar e arquivar dados relativos aos cadastros dos seguintes tipos de pessoas físicas:

1.1. Clientes

1.2. Funcionários.

Módulo Locação

2. O Sistema deverá ser capaz de gerenciar o agendamento de Locações que se dará *in-loco* (balcão), via telefone ou internet.

EXEMPLOS DE MINIMUNDO

MiniMundo

- Uma descrição de MiniMundo deve ser basicamente o **mínimo** necessário para descrever sua aplicação de forma objetiva, através de levantamento de requisitos como entrevistas, questionários, workshops de requisitos, observação, investigação de documentos, prototipagem, cenários, abordagens baseadas em objetivos e reutilização de requisitos, para então conseguirmos enxergar melhor o cenário que nos espera.

EXEMPLO 1 DE MINIMUNDO

Exemplo 1 de MiniMundo

- Uma loja de venda de Eletrodomésticos quer automatizar o seu **controle de compra e troca de aparelhos** por parte de seus clientes.
- Todo aparelho vendido possui garantia de 1 ano, a partir da data de venda.
- Isto significa que qualquer troca só poderá ser realizada dentro deste período, mesmo que já tenha havido várias trocas em função desta compra.

Exemplo 1 de MiniMundo

- No termo de garantia é anotado a data da compra, marca, modelo e número de série do aparelho vendido juntamente com o nome e endereço do cliente que o comprou.
- A cada troca de aparelho, relativo a primeira compra, é verificado se ainda está no prazo de garantia, e é registrado o cliente que realizou a troca.
- Qualquer cliente pode realizar uma troca, mesmo que não tenha sido o comprador.
- Os aparelhos defeituosos são devolvidos para a fábrica e não mais retornam para a loja.

Exemplo 1 de MiniMundo

- A loja quer saber:
 - a) Relação de aparelhos disponíveis na loja
 - b) Relação de aparelhos que apresentaram defeitos contendo quem realizou a troca, a data e o defeito apresentado
 - c) Relação de clientes cujas compras nunca apresentaram defeito

EXEMPLO 2 DE MINIMUNDO

Exemplo 2 de MiniMundo

- Uma firma que utiliza equipamentos de informática necessita de um sistema que **gerencie a sua rede de microcomputadores** (ponto-a-ponto, não existe servidor da rede), **controlando usuários, máquinas e impressoras**.
- A rede é composta de servidores de impressão, estações e impressoras.
- O sistema também irá **controlar a partir de qual estação o usuário está conectado a rede, e os seus arquivos enviados para impressão**.

Exemplo 2 de MiniMundo

- Para todos os micros deseja-se cadastrar: código do patrimônio, descrição, capacidade do disco rígido, quantidade de memória e, sendo uma estação a sua localização, sendo um servidor o tamanho máximo do buffer e a quantidade máxima de buffers de impressão e que ele suporta, e as impressoras ligadas a ele (no máximo 3), caso existam.
- Para impressoras deseja-se cadastrar: código do patrimônio, descrição, velocidade (CPS) e, conseqüentemente, o servidor a que está ligada. Nesta firma todas as impressoras estão ligadas a algum servidor, não sendo compartilhada por mais de um servidor.

Exemplo 2 de MiniMundo

- Para controlar os usuários, o sistema só precisa do nome de guerra e senha de cada um.
- Como os usuários não possuem máquina fixa, a sua conexão à rede pode ocorrer a partir de qualquer estação.
- Tendo o usuário uma conexão ativa, o sistema não permitirá que ele se conecte a partir de outra estação.
- Há interesse em controlar apenas as conexões ativas (as conexões desfeitas são irrelevantes).

Exemplo 2 de MiniMundo

- No caso de impressão, o sistema deverá saber qual o arquivo, de quem ele é, e em qual impressora será impresso (atenção: somente usuários com conexão ativa e que possuem condição de enviar arquivos para impressão).
- É o usuário que escolhe a impressora onde ele quer que o seu arquivo seja impresso.
- Nada impede que usuários diferentes enviem arquivos de mesmo nome para impressão, porém (nesta firma) para o mesmo usuário isso não é possível, mesmo em impressoras diferentes.
- Neste caso o sistema permite alterar o número de cópias a serem impressas. Só deve ser mantido registro dos arquivos que ainda estão na fila de impressão.

Exemplo 2 de MiniMundo

- O sistema deverá listar, para cada impressora, os arquivos que estão aguardando impressão, com o respectivo usuário que a enviou, mesmo que o usuário não esteja mais ativo na rede.
- Sempre que solicitado o sistema exibirá, para cada estação, o seu código e, caso exista, o nome do usuário conectado, a data e hora início desta conexão e, se houver, nome e quantidade de cópias dos arquivos que ele enviou e que ainda estão aguardando impressão.

EXEMPLO 3 DE MINIMUNDO

Exemplo 3 de MiniMundo

- Um Vídeo Clube deseja controlar o empréstimo de fitas a seus sócios.
- Atualmente, o controle de empréstimo é feito utilizando dois conjuntos de fichas: o das informações dos sócios e o das informações das cópias, no qual se controla também os empréstimos.

Exemplo 3 de MiniMundo

- Nas fichas dos sócios constam as seguintes informações: número de inscrição, nome, endereço e telefone.
- Nas fichas das cópias estão registrados: código de identificação da fita, título do filme, duração, ano, gênero, nome do diretor e de 2 (dois) dos artistas principais, data de aquisição e o estado da fita (bom ou ruim), além de uma lista de empréstimos com número de inscrição do sócio, data do empréstimo, data da devolução e valor pago. Estes dois últimos são preenchidos quando a cópia é devolvida.

Exemplo 3 de MiniMundo

- Nesta locadora existem vários filmes com várias cópias, e cada cópia recebe uma etiqueta com um código, por exemplo: 5.315 (como se fosse o número do CONCINE, único para cada cópia original).

Exemplo 3 de MiniMundo

- Um empréstimo é válido por 72 horas no máximo e possui preço fixo.
- A locadora pretende colocar terminais de consulta em sua loja, possibilitando seus sócios a escolha de filmes a partir do gênero, diretor ou dos artistas prediletos.
- Não é aceita reserva e todos os pagamentos são feitos no momento da devolução.
- Se o sócio atrasar a devolução deverá pagar multa.

Exemplo 3 de MiniMundo

- A gerência deseja que o sistema forneça um relatório de todos os filmes que possui, informando a quantidade de cópias por filme, além de relatórios de filmes por gênero, diretor e artista.
- Ela deseja também uma lista das cópias em mal estado e outra de sócios inadimplentes.
- Existe interesse em manter um controle sobre diretores e artistas com nome, país de origem e data de nascimento de cada um.

ANÁLISE ESSENCIAL

Técnicas	Abordagens	Ferramentas
Análise Tradicional	<ul style="list-style-type: none"> • Funcional 	<ul style="list-style-type: none"> • Textos • Fluxogramas
Análise Estruturada	<ul style="list-style-type: none"> • Funcional 	<ul style="list-style-type: none"> • Especificação Funcional • Normalização • Diagrama de Fluxos de Dados (DFD) • Diagrama de Estrutura de Dados (DE) • Dicionário de Dados (DD)
Análise Essencial	<ul style="list-style-type: none"> • Funcional • Dados • Controle 	<ul style="list-style-type: none"> • Especificação Funcional • Normalização • Diagrama de Fluxo de Dados (DFD) • Diagrama de Estrutura de Dados (DE) • Dicionário de Dados (DD) • Tabela de Eventos • Diagrama de Entidade e Relacionamento (DER) • Diagrama de Transição de Estados (DTE)

Técnicas	Abordagens	Ferramentas
Análise Orientada a Objeto	<ul style="list-style-type: none"> • Funcionalidade • Dados • Controle 	<ul style="list-style-type: none"> • Diagramas de Comportamento (UML) Diagramas de Casos de Uso Diagramas de Estados Diagramas de Atividades

Análise Essencial

- A análise essencial é constituída basicamente por duas fases ou modelos: **ambiental** e **comportamental**.

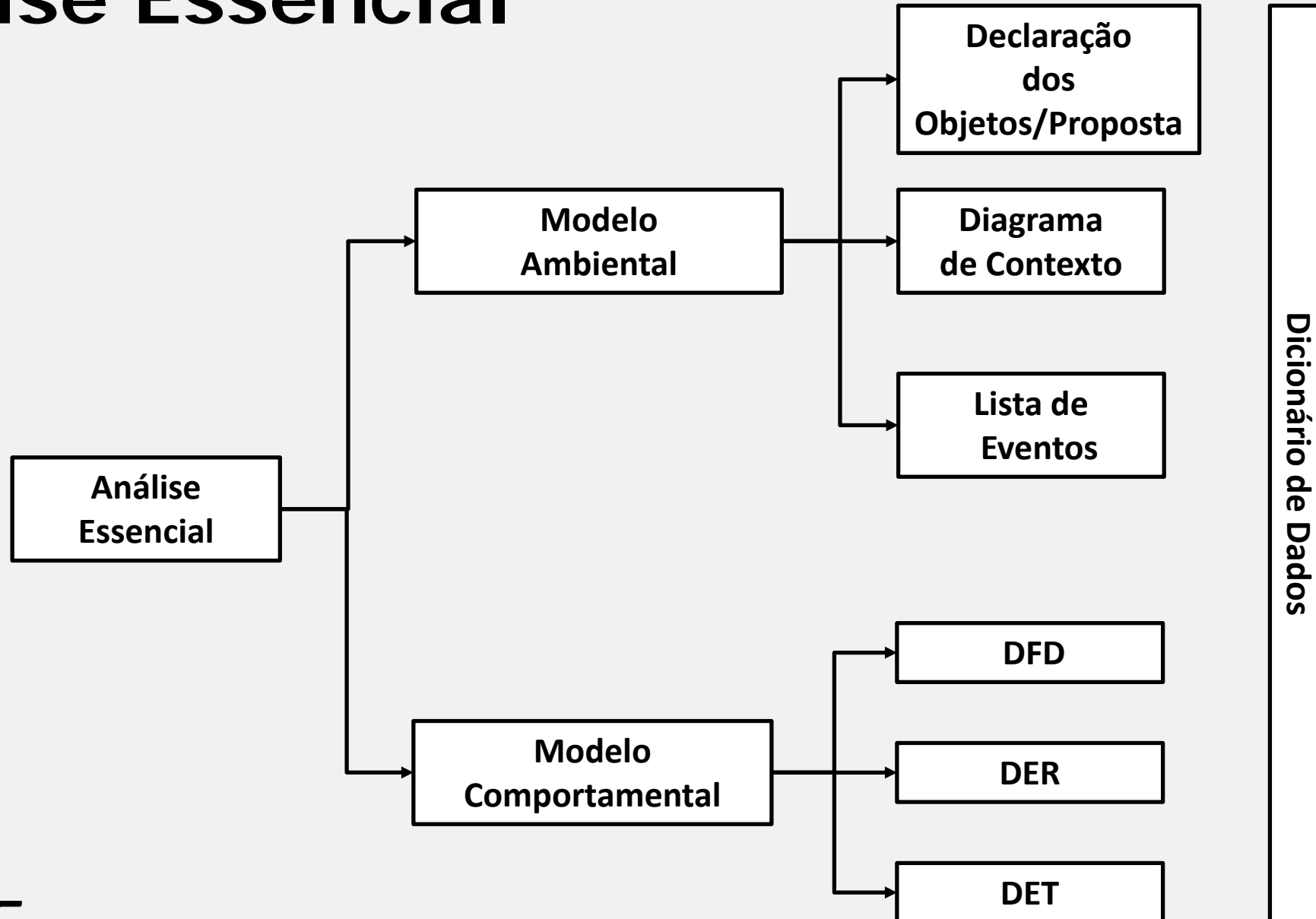
Modelo Ambiental

- Define a fronteira entre o sistema e o “resto do mundo”;
- Voltado para o ambiente em que o sistema está inserido;
- Definido do ponto de vista externo;
- Representa a interface do sistema (sua fronteira) com o mundo exterior;
- Mostra a interação do sistema com os elementos externos a ele.

Modelo Comportamental

- Descreve o comportamento do sistema e o que é necessário para interagir com o ambiente.
- Voltado para o sistema, para o comportamento de suas partes internas.
- Mostra como o sistema deve reagir internamente aos estímulos do ambiente.

Análise Essencial



Análise Essencial X Análise Estruturada

- **Análise Essencial:** indica o quê o sistema deve fazer para satisfazer os requisitos do usuário, sem mencionar como será a implementação.
- **Análise Estruturada:** o analista deve desenvolver 4 modelos: físico atual, lógico atual, novo modelo lógico e novo modelo físico.

Análise Essencial

Principais Ferramentas de Modelagem

- Diagrama de Contexto (DC)
- Diagrama de Fluxo de Dados (DFD)
- Diagrama de Entidade e Relacionamento (DER)
- Diagrama de Transições de Estado (DTE)
- Diagrama de Estrutura (DE)

Análise Essencial

Ferramentas complementares:

- Dicionário de Dados (DD)
- Especificação de Processo
 - Árvore de Decisão
 - Tabela de Decisão
 - Português Estruturado.

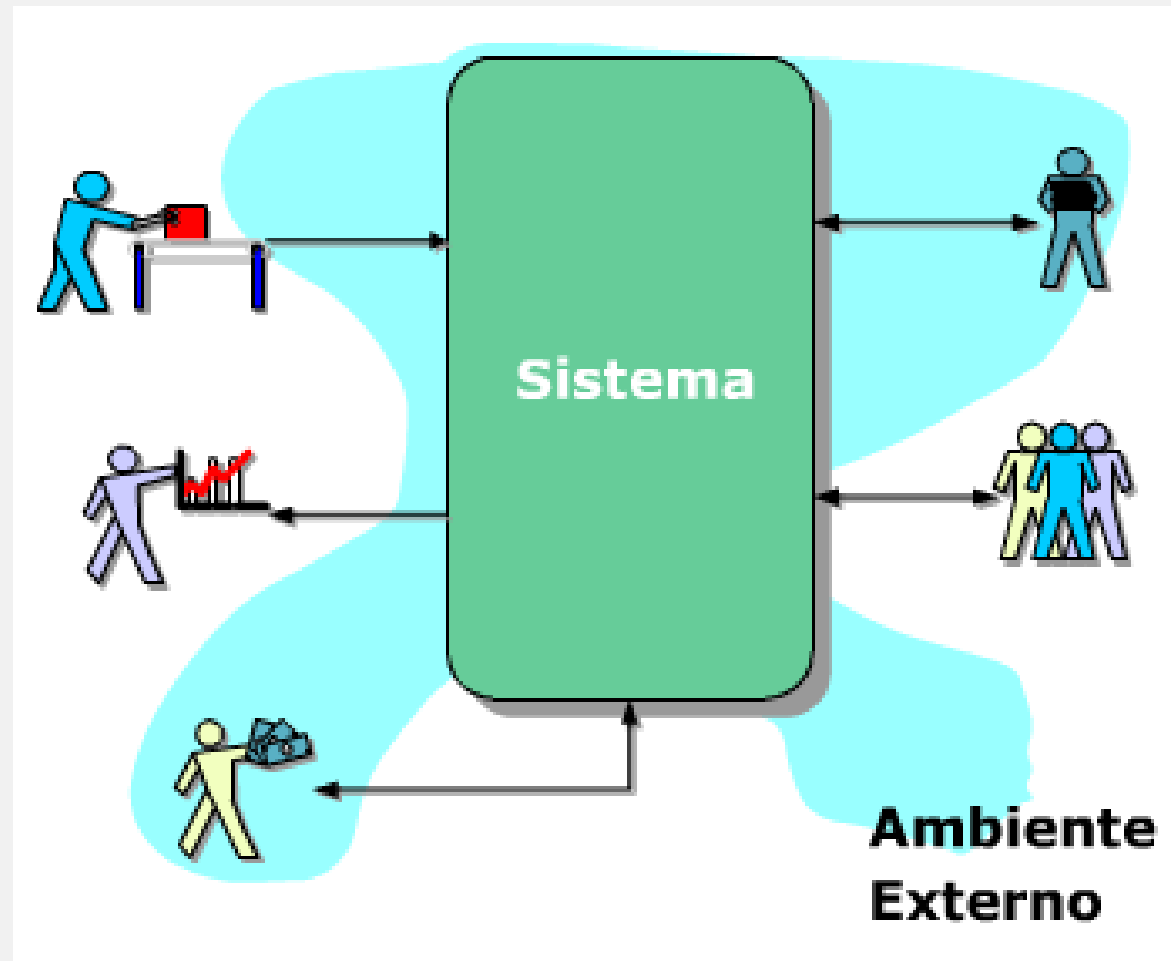
PRINCIPAIS FERRAMENTAS DE MODELAGEM

Modelo de Contexto

Serve para:

- Identificar os elementos externos que interagem com o sistema.
- Mostrar o **fluxo de informação** existente entre o **sistema** e seu **ambiente externo**.
- Estabelecer os limites do sistema.
- Identificar os eventos que ocorrem no ambiente externo e que provocam uma resposta do sistema.

Modelo de Contexto



Modelo de Contexto

- Deve ser desenvolvido em conjunto com os usuários. Se não for possível, ele pode ser desenvolvido pelos analistas e submetido à aprovação dos usuários.
- De qualquer forma, o modelo de contexto vai se constituir numa espécie de **contrato de prestação de serviços** entre analistas e usuários com relação àquilo que vai ser **incluído** no sistema.
- Por isso, todos os cuidados devem ser tomados para garantir uma correta determinação dos **limites** do sistema, do que vai ser e do que não vai ser objeto de análise

Modelo de Contexto

- Possui dois componentes:
- O **diagrama de contexto**, que é uma representação gráfica do sistema com seu ambiente externo.
- A **lista de eventos**, que relaciona os fatos do ambiente externo que o sistema obrigatoriamente deve reconhecer.

Modelo de Contexto

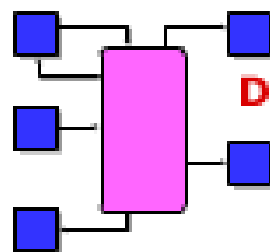


Diagrama de Contexto

Lista de Eventos

ESTÍMULO	RESPOSTA	EVENTO
Pagamento	Recibo	Cliente efetua pagamento
-	Extrato	Hora de emitir extrato
-	Relação de Inadimplentes	Hora de emitir relação de inadimplentes

Modelo de Contexto

- 1º Identificar as entidades externas
 - Podem ser pessoas, programas, organizações e outras entidades que interagem com o sistema, mas estão além de sua fronteira. Equivalente ao **ator** na UML
- 2º Identificar os fluxos de dados
- 3º Construir a lista de eventos
- 4º Verificar se os eventos estão todos contemplados

Exemplo de DC

Considere um sistema de cobrança de uma organização. Suponha que este sistema seja bastante simples.

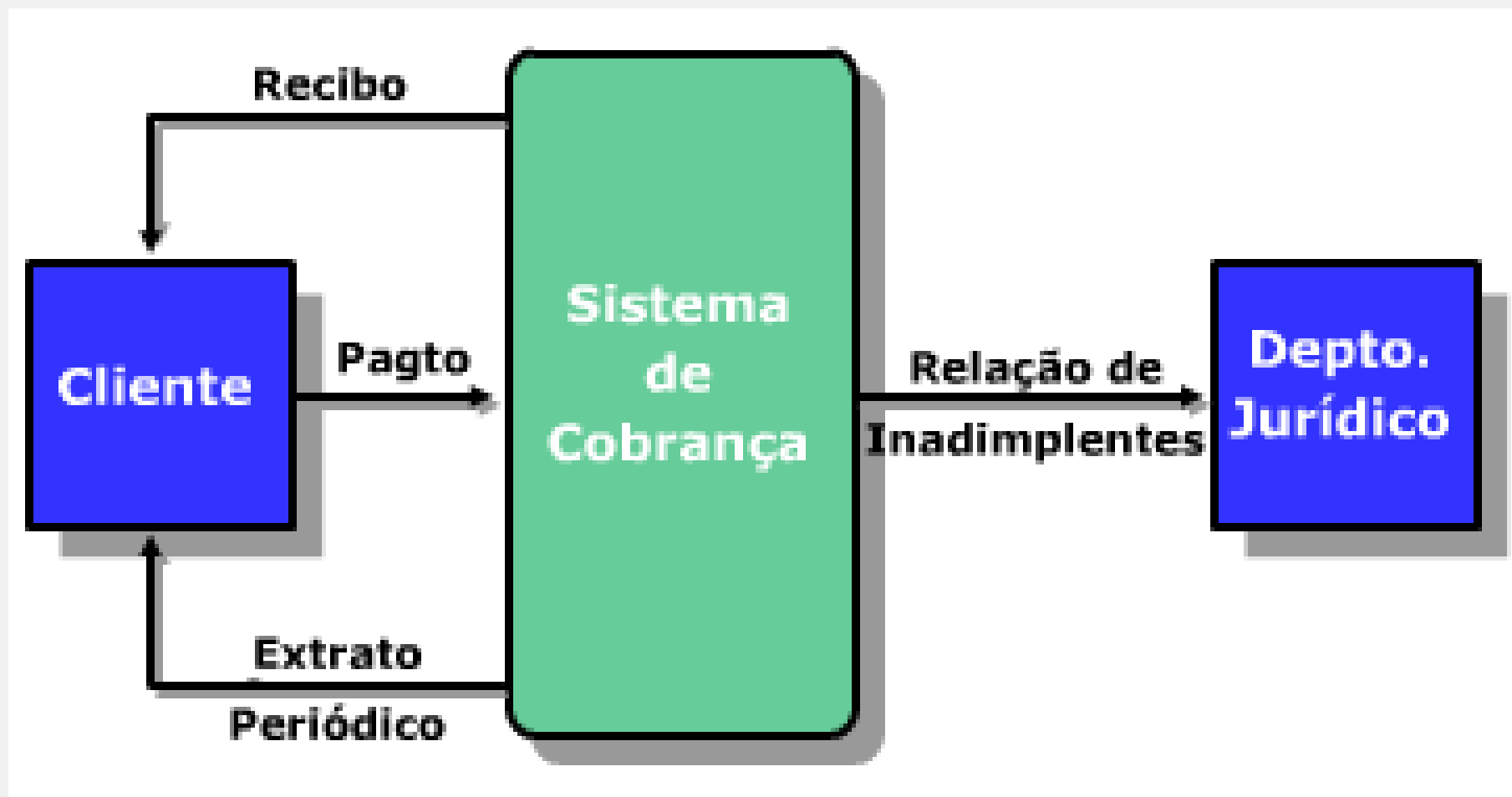
Ele deve ser capaz de:

- Receber os pagamentos feitos pelos clientes.
- Emitir os recibos dos pagamentos realizados.
- Emitir extratos periódicos para os clientes.
- Emitir relação de clientes inadimplentes para uso do Departamento Jurídico.

Exemplo de DC

- Neste exemplo, podemos destacar que:
 - O cliente é origem e destino das informações que o sistema recebe e gera.
 - O Departamento Jurídico é o destino das informações geradas pelo sistema.
 - O sistema recebe informações relativas aos pagamentos feitos pelo cliente.
 - O sistema emite os recibos e os extratos para os clientes.
 - O sistema emite a relação de inadimplentes para o Departamento Jurídico.

Exemplo de DC

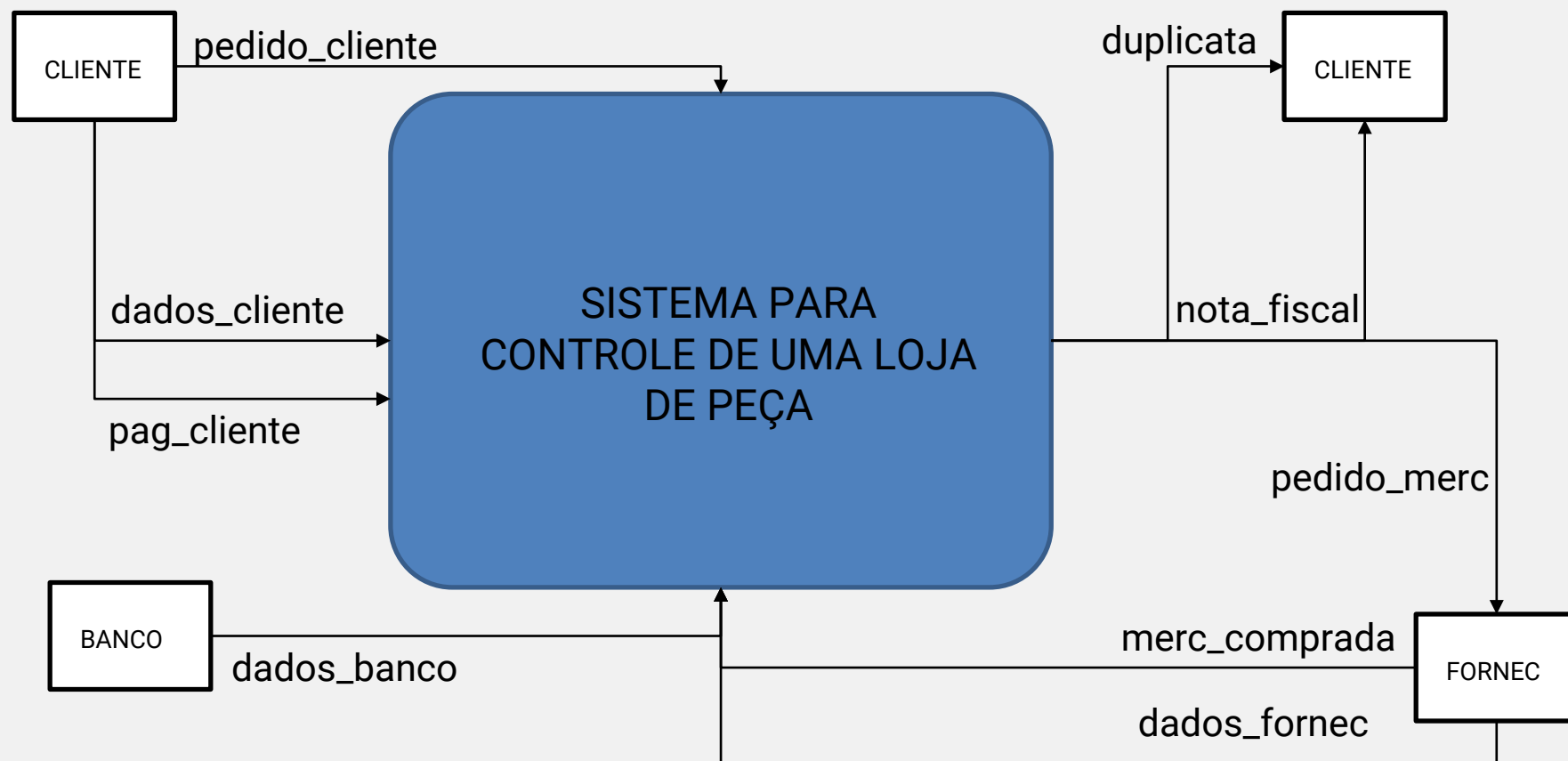


Exemplo de DC

Observamos que:

- O cliente e o Departamento Jurídico são as entidades externas, origem e destino das informações que o sistema recebe ou emite.
- Pagamento é um fluxo de dados recebido pelo sistema. Nós chamamos este tipo de fluxo de dados de "fluxo de entrada" ou de "estímulo".
- Recibo, extrato e relação de inadimplentes são fluxos de dados emitidos pelo sistema. São "fluxos de saída" ou "respostas" do sistema.

Exemplo de DC



Vantagens do DC

- Mostra o escopo e os limites de um sistema
- Nenhum conhecimento técnico
- Fácil de desenhar e alterar devido a sua notação limitada
- Fácil de expandir pela adição de diferentes níveis de Diagramas de Fluxos de Dados
- Pode beneficiar um público amplo

Diagrama de Fluxo de Dados (DFD)

- O diagrama de fluxos de dados (DFD) é uma ferramenta para a modelagem de sistemas.
- Fornece apenas uma visão do sistema, a visão estruturada das funções, ou seja, o fluxo dos dados.
- Produz uma indicação de como os dados são transformados e como se movimentam através do sistema, ou seja, descreve o **fluxo de informação** e as **transformações** que são aplicadas à medida que os dados se movimentam da entrada para a saída.

Diagrama de Fluxo de Dados (DFD)

- Se estivermos desenvolvendo um sistema no qual os relacionamentos entre os dados sejam mais importantes que as funções, podemos dar menos importância ao DFD e dedicar-nos aos diagramas de entidade-relacionamento (DER)

Diagrama de Fluxo de Dados (DFD)

Componentes

- **Processo:** Conjunto de atividade que produzem, modificam ou atribuem qualidade às informações.
- **Depósito de dados:** Conjunto de informações armazenadas pelo processo para serem utilizadas por algum processo, a qualquer momento.
- **Entidade Externa:** É algo situado fora do escopo do sistema, que é fonte ou destino das suas informações.
- **Fluxo de dados:** O nome deve expressar o significado do conjunto de informações que está fluindo.

Diagrama de Fluxo de Dados (DFD)

Componentes - Processo

- **Processo** ou bolha ou função: um transformador de informação que reside dentro dos limites do sistema a ser modelado.
 - o nome da função deve esclarecer **o que** será feito, de forma objetiva: **verbo + objeto**;
 - identificador de processo: n ($n=1, 2, 3, \dots$);
 - identificador de processos filhos: $n.m$ ($m=1, 2, 3, \dots$);
 - o identificador de processo não mostra ordem de execução;

Diagrama de Fluxo de Dados (DFD)

Componentes - Processo

- **Evitar:** verbos muito físicos, genéricos e técnicos, como:
 - imprimir, gravar, digitar;
 - processar, avaliar, analisar;
 - deletar, computar.
- Notação:

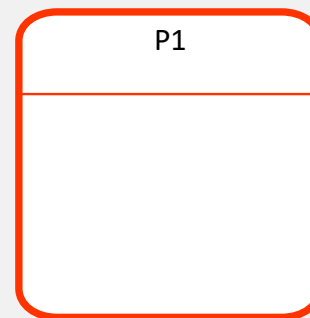
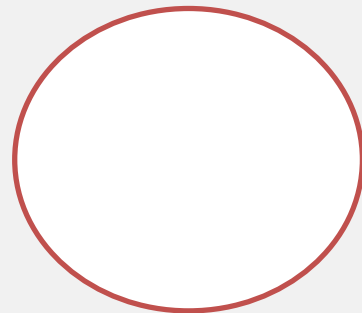


Diagrama de Fluxo de Dados (DFD)

Componentes – Depósito de Dados

- Repositório de dados.
- Representado por substantivos no plural. Exemplos: Produtos, Clientes, Contas_a_receber, Fornecedores;
- O nome do depósito de dados deve sugerir o seu conteúdo da forma mais sucinta possível;

Diagrama de Fluxo de Dados (DFD)

Componentes – Depósito de Dados

- Podem ser representados mais de uma vez para evitar o cruzamento de linhas de fluxos de dados.
- Notação:

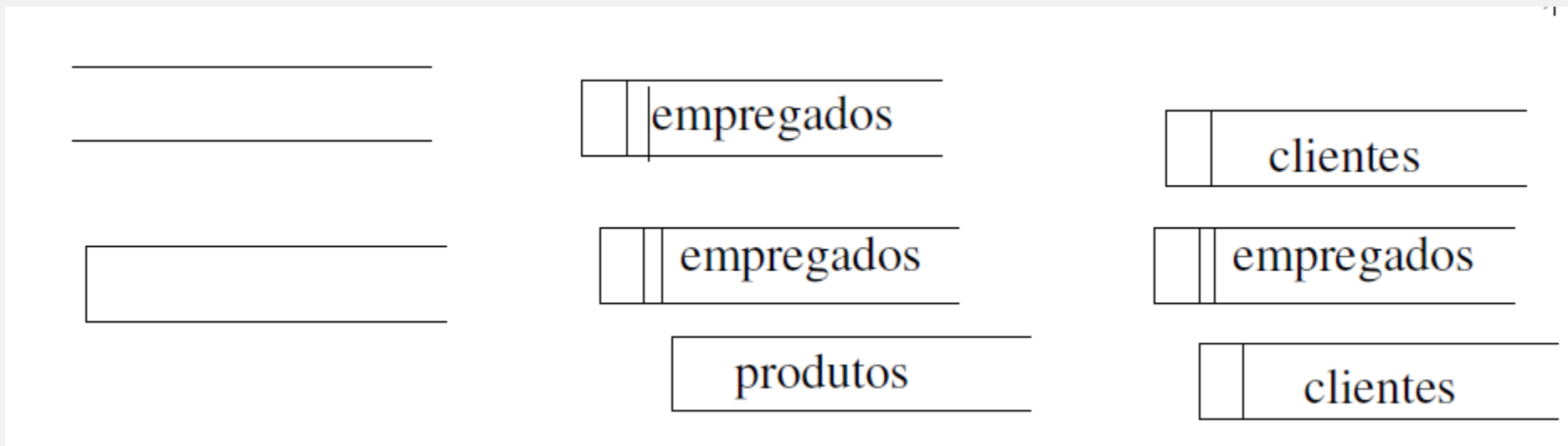


Diagrama de Fluxo de Dados (DFD)

Componentes – Entidade Externa

- ou **Terminal**: um produtor ou um consumidor de informações que reside fora dos limites do sistema a ser modelado.
 - refere-se à fonte ou destino dos fluxos de dados que chegam e saem do sistema;
 - representada por substantivos no singular. Exemplos: Cliente, Sistema de Crédito, Fornecedor, Diretoria, Aluno, Professor;

Diagrama de Fluxo de Dados (DFD)

Componentes – Entidade Externa

- Podem ser representados mais de uma vez para evitar o cruzamento de linhas de fluxos de dados.
- Notação:

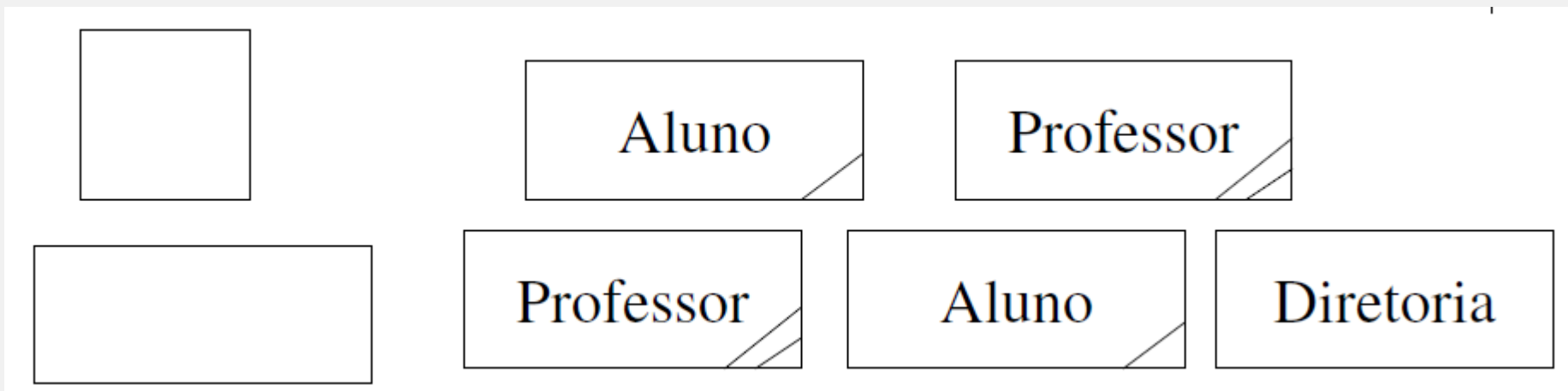


Diagrama de Fluxo de Dados (DFD)

Componentes – Fluxo de Dados

- Um item de dado ou coleção de itens de dados.
 - A seta indica a direção do fluxo de dados;
 - Representado por substantivos;
 - Exemplos: relatório_de_vendas, pedido_do_cliente;
 - Cada fluxo de dados será indicado por um nome, que deverá sugerir o teor da informação em deslocamento;
- É ilícita a existência de fluxos de dados entre:
 - duas entidades externas;
 - dois depósitos de dados;
 - uma entidade externa e um depósito de dados.

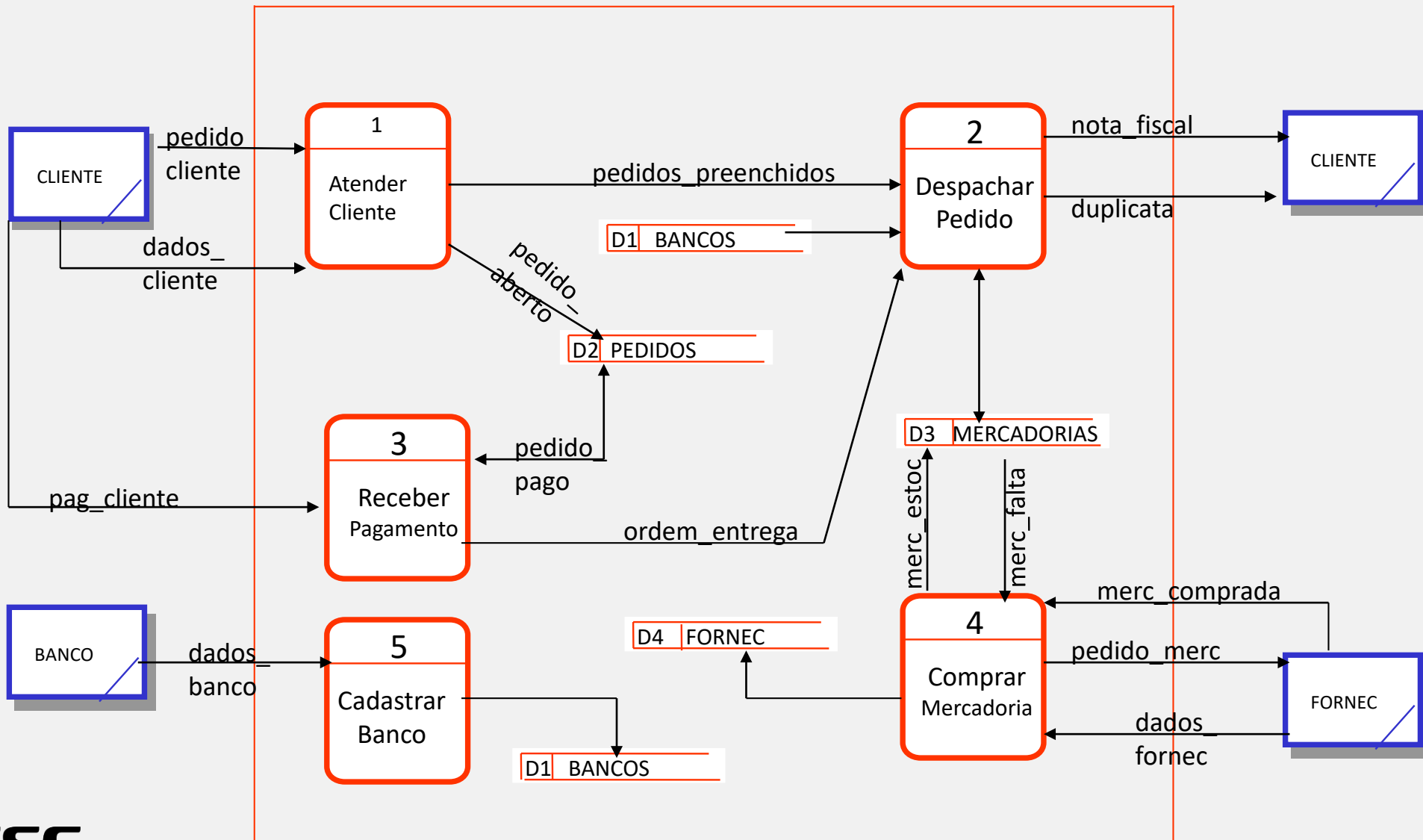
Diagrama de Fluxo de Dados (DFD)

Componentes – Fluxo de Dados

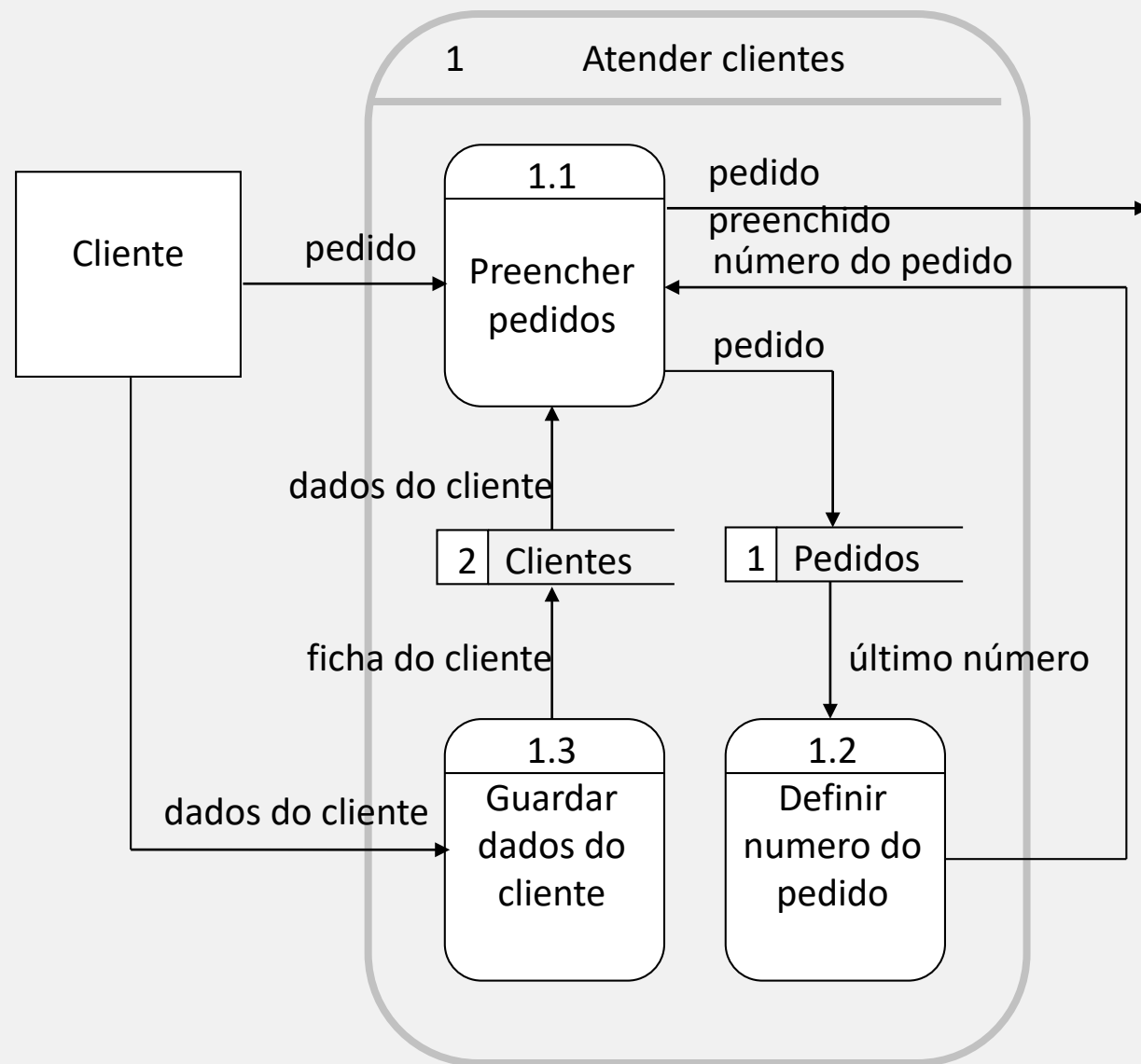
- Notação



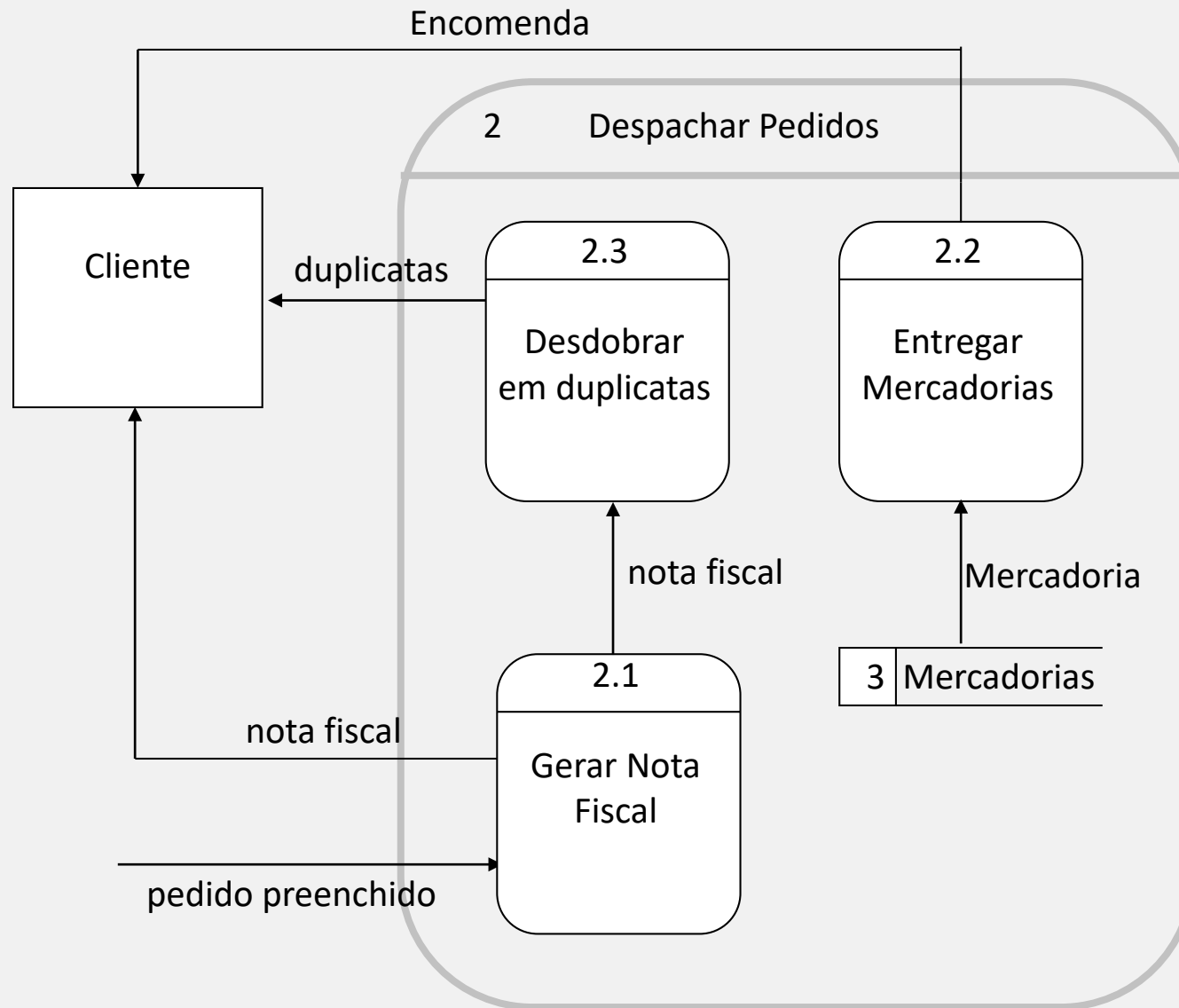
Exemplo DFD 0



Exemplo DFD 1



Exemplo DFD 2



Diretrizes para a Elaboração do DFD (Yourdon)

1. Escolher nomes significativos para os processos, fluxos de dados, depósitos de dados e entidades externas

- rotular o processo identificando o *papel* e não a *pessoa*; utilizar um *verbo* que represente a ação e um *objeto*;
- os nomes escolhidos devem provir de um vocabulário conhecido pelo usuário;
- o analista deve ter conhecimento mínimo sobre o domínio;

2. Numerar processos

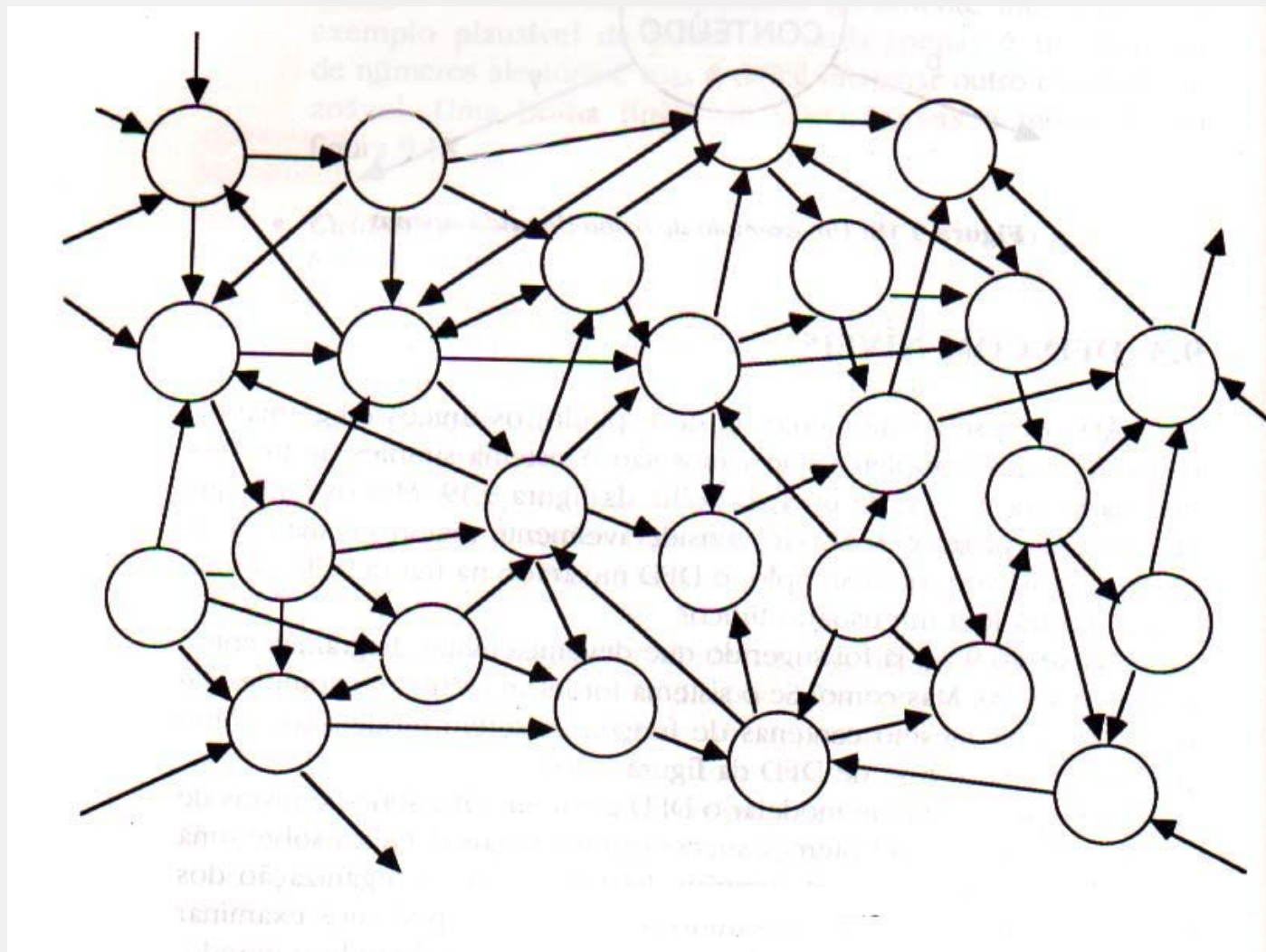
- um modo prático de referenciar os processos de um DFD é numerá-los;
 - isso não corresponde à uma sequência de execução;
- a numeração dos processos auxilia na identificação de processos de níveis inferiores;

Diretrizes para a Elaboração do DFD (Yourdon)

3. Evitar DFDs complexos demais

- o propósito do DFD é modelar corretamente as funções que o sistema deve executar e as interações entre elas;
- modelar o DFD em uma série de níveis => DFDs de nível inferior oferecem detalhes dos processos de nível imediatamente superior (cada nível deve ter até 6 funções; para outros autores de 5 a 9 funções);
- o refinamento de DFDs ocorre até que sejam alcançadas as *funções primitivas*;

Exemplo DFD Complexo



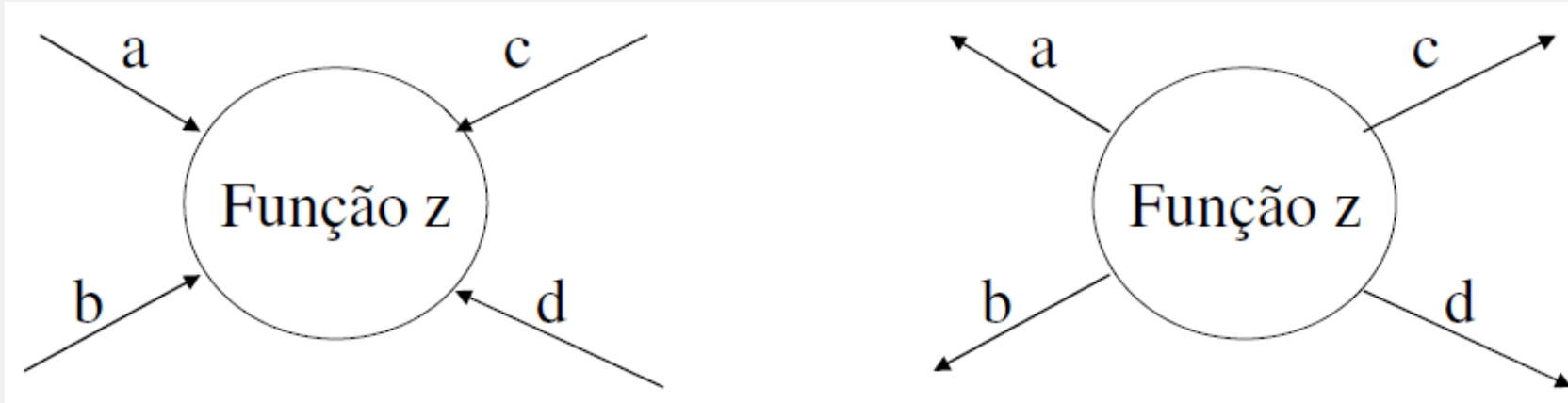
Diretrizes para a Elaboração do DFD (Yourdon)

- 4. Refazer os DFDs tantas vezes quantas forem necessárias até se obter uma boa estética, um DFD tecnicamente correto e aceitável pelo cliente;**

Diretrizes para a Elaboração do DFD (Yourdon)

5. Certificar-se de que o DFD seja logicamente consistente

- Evitar *poços sem fundos* (buracos negros) => processos que têm entradas mas não têm saídas;
- Evitar processos com *geração espontânea* => processos que têm saídas mas não têm entradas;



Diretrizes para a Elaboração do DFD (Yourdon)

5. Certificar-se de que o DFD seja logicamente consistente (cont.)

- Cuidado com fluxos e processos sem nome;
- Cuidado com depósitos de *apenas leitura* ou *apenas escrita* => um depósito típico deve ter entradas e saídas;
- A continuidade do fluxo de informação deve ser mantida, ou seja, a entrada e a saída em cada refinamento devem permanecer a mesma;
- É conveniente que, em cada nível de abstração, as funções estejam em um grau de detalhamento próximo;

Diagrama Entidade Relacionamento (DER)

- Diagrama entidade relacionamento é um modelo diagramático que descreve o modelo de dados de um sistema com alto nível de abstração.
- Ele é a principal representação do Modelo de Entidades e Relacionamentos.
- É usado para representar o modelo conceitual do negócio. Não confundir com modelo relacional, que representam as tabelas, atributos e relações materializadas no banco de dados.

Diagrama Entidade Relacionamento (DER)

- O DER é composto de poucos símbolos gráficos que representam os relacionamentos do banco, listados a seguir:
- **Retângulo**
 - Representa a entidade do banco de dados, o que muito provavelmente será uma tabela quando o banco for criado;
- **Losango**
 - Representa o relacionamento entre as entidades.
- **Triângulo**
 - Representam as especializações.
- **Bolinhas**
 - Representa os atributos.

Exemplo DER

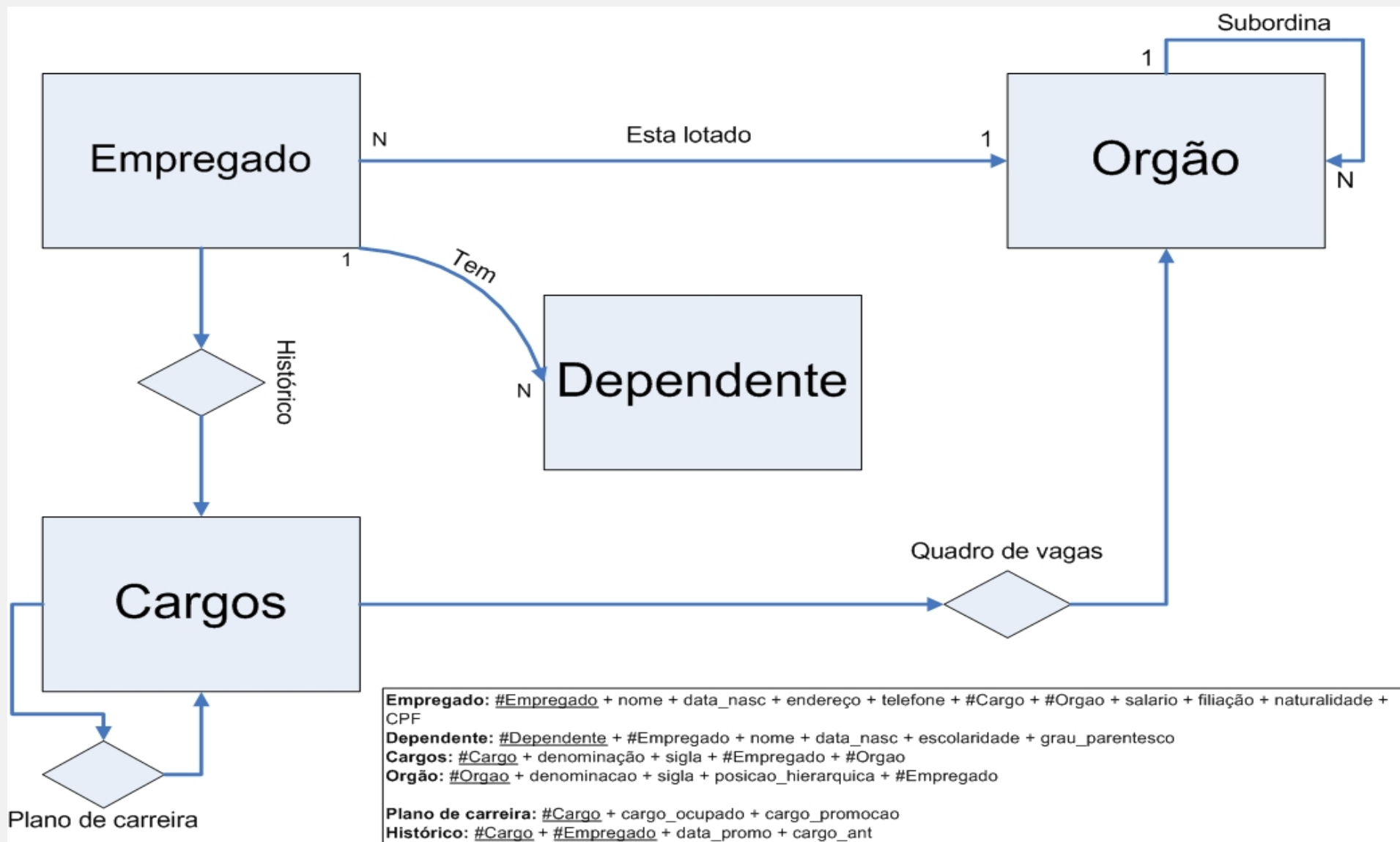


Diagrama de Transição de Estado (DTE)

- Em engenharia de software e eletrônica digital, um diagrama de transição de estados é uma representação do **estado ou situação** em que um objeto pode se encontrar no decorrer da execução de processos de um sistema. Com isso, o objeto pode passar de um estado **inicial para um estado final** através de uma **transição**.

Diagrama de Transição de Estado (DTE)

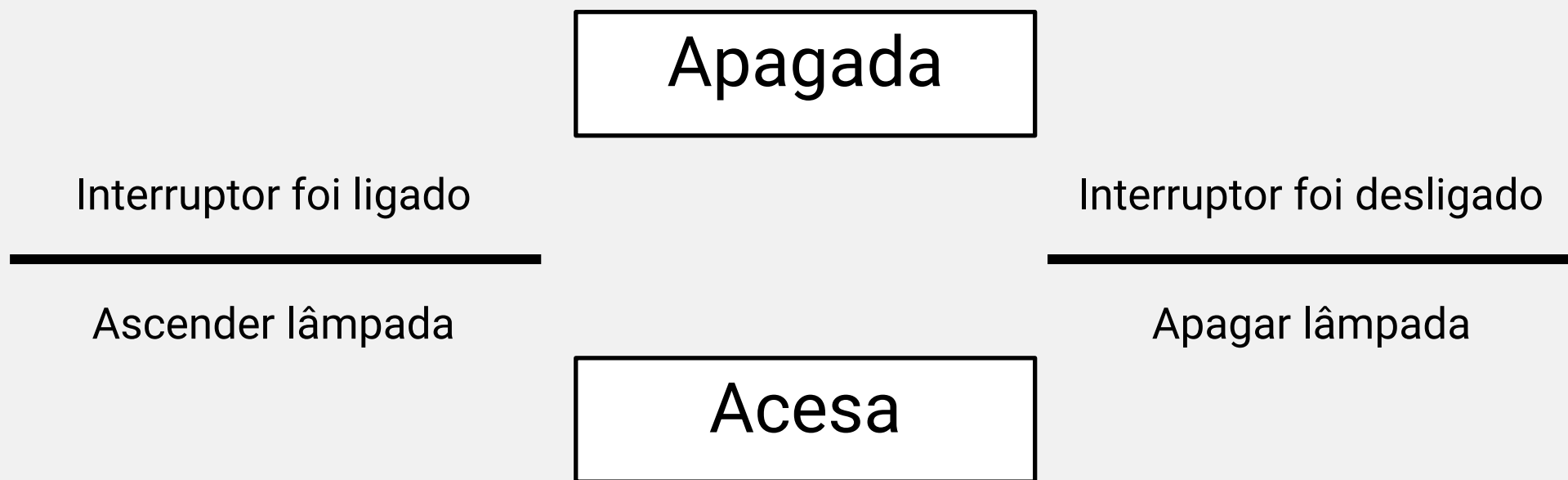


Diagrama de Transição de Estado (DTE)

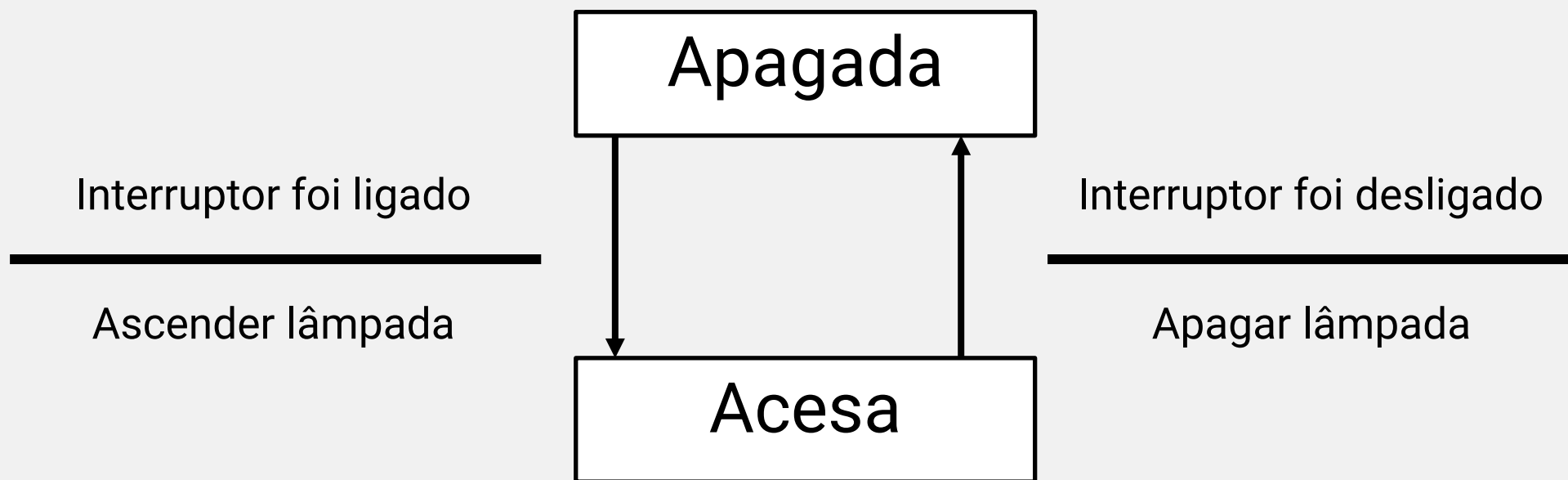
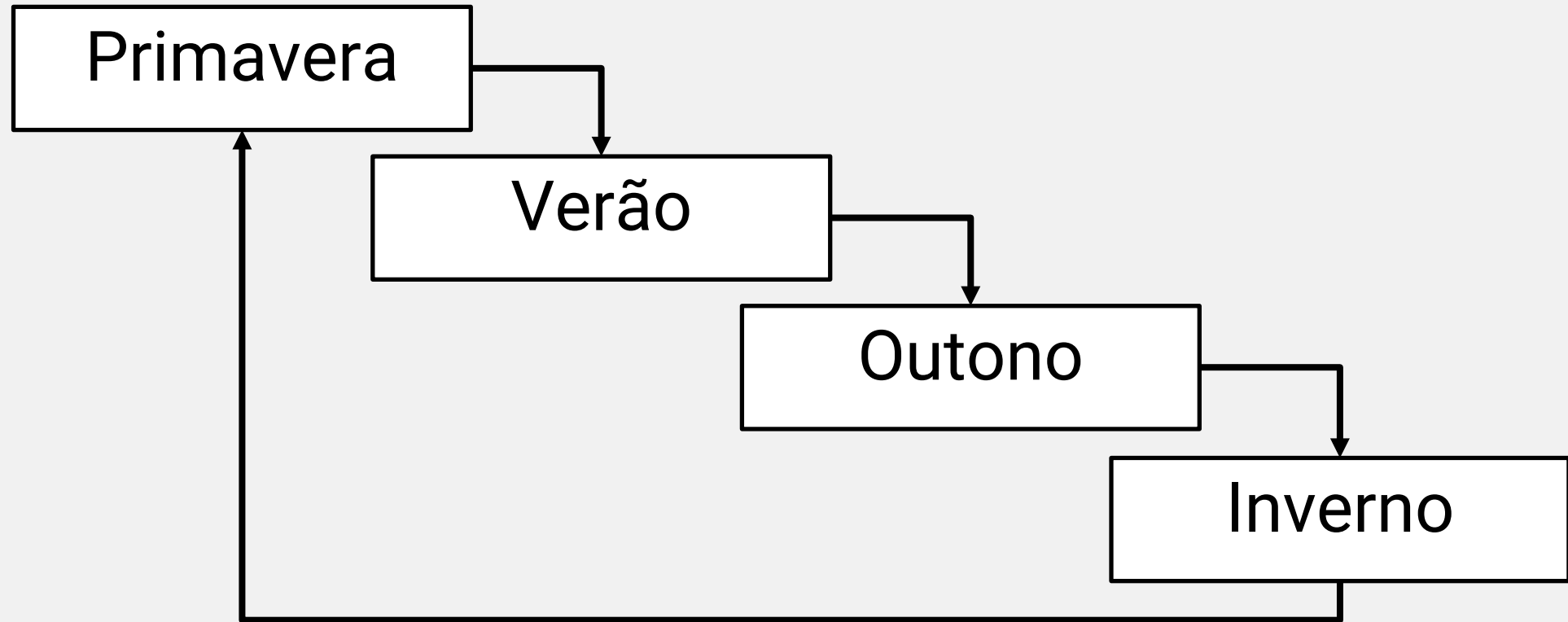


Diagrama de Transição de Estado (DTE)

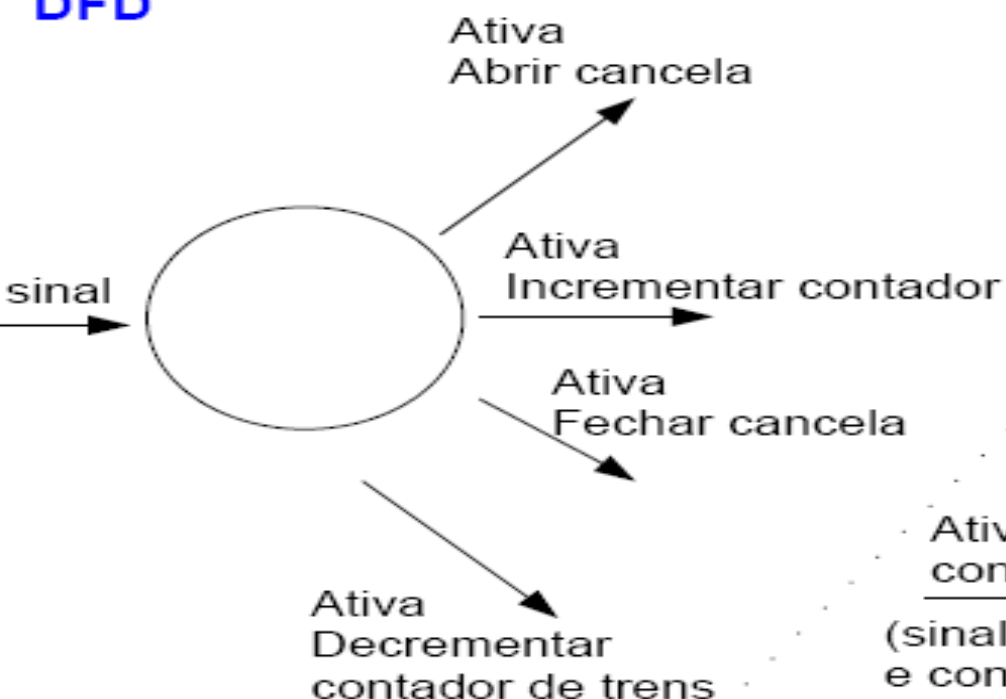


Exemplo DTE

Exemplo:

O sistema Controlar Cancela é um processo de controle de um sistema de tempo real que monitora os trens de uma estação de trens através de sensores. Nesse sistema, quando um trem se aproxima do cruzamento, a cancela deve fechar. Mais de um trem pode estar na área de cruzamento ao mesmo tempo e, quando o último trem for embora e a área estiver livre de trens, a cancela deve ser aberta. Um DFD corresponde deve conter os processos: **Abrir cancela**, **Fechar cancela**, **Incrementar contador de trens** e **Decrementar contador de trens**.

DFD



DTE

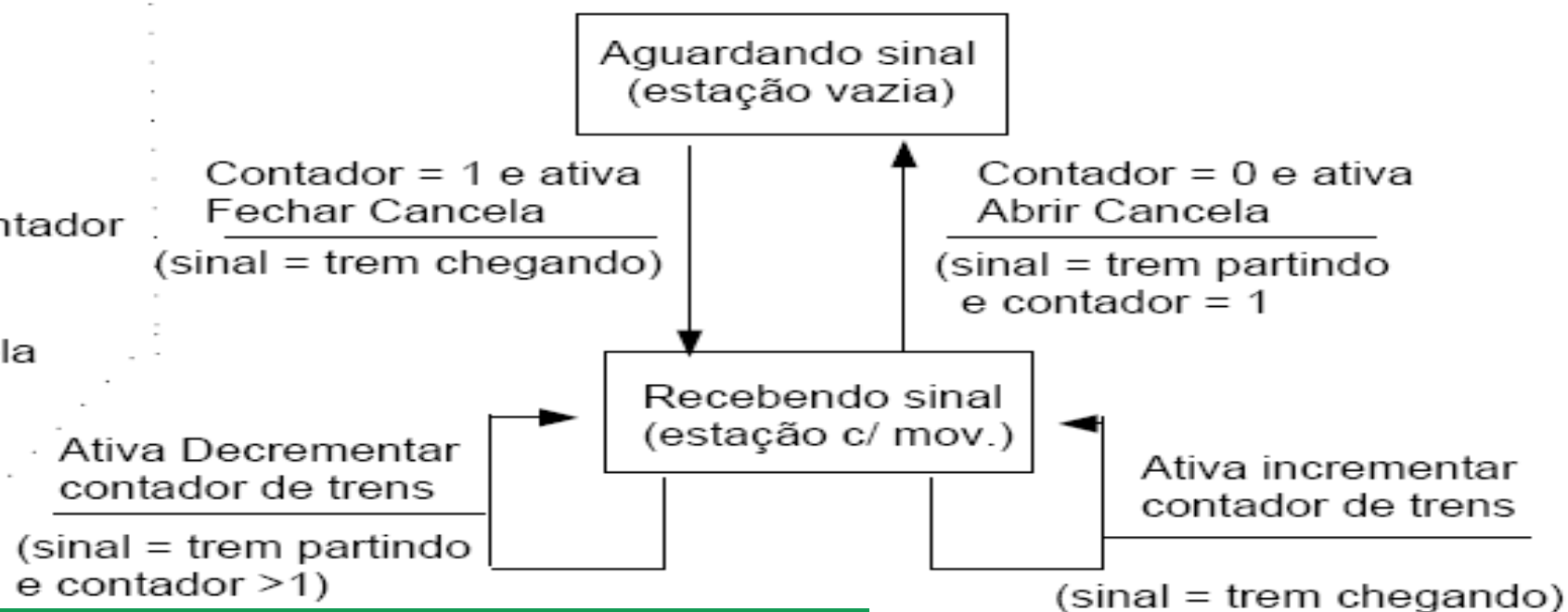


Diagrama de Estrutura (DE)

- O Diagrama de Estrutura é uma ferramenta de decomposição funcional, onde o programa é visto como uma hierarquia de componentes funcionais.
- O Diagrama de Estrutura é útil no projeto de programas no contexto do Projeto Estruturado, proposto por Constantine e Yourdon.
- O conceito básico do Diagrama de Estrutura é a modularização. Cada componente funcional é um **módulo do projeto**, que será codificado e gerará um **módulo do código fonte do programa**

Diagrama de Estrutura (DE)

- O Diagrama de Estrutura é um diagrama hierárquico, uma vez que cada módulo é decomposto em módulos menores, até que toda a árvore do projeto esteja completa.
- O programa é representado como um conjunto de módulos organizado hierarquicamente. A decomposição de um módulo em módulos menores é chamada de fatoração.
- A base para a construção do Diagrama de Estrutura é o Diagrama de Fluxo de Dados(DFD). O Projeto Estrutura propõe uma metodologia de derivação do Diagrama de Estrutura a partir do DFD

Diagrama de Estrutura (DE)

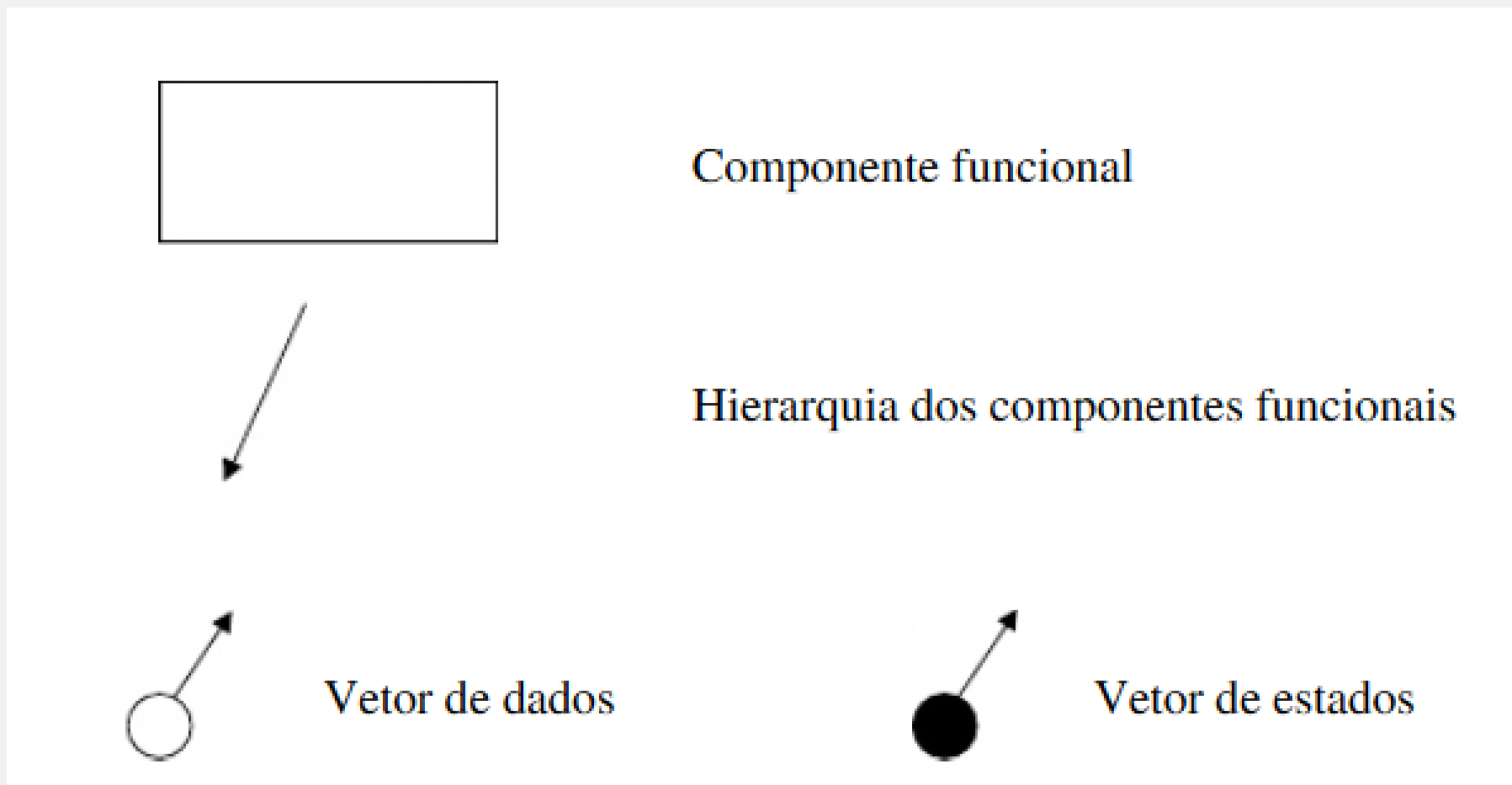


Diagrama de Estrutura (DE)

- O **componente funcional**, ou módulo, é um **conjunto de tarefas** desempenhada pelo programa com uma singularidade de propósito. Isto é, o módulo é responsável por executar uma função específica no programa, podendo ser chamado por outros módulos e podendo, também, chamar outros módulos para a consecução de seu objetivo.

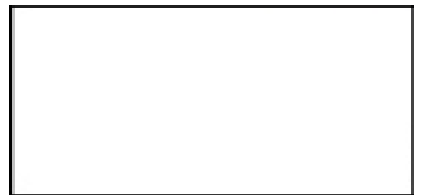


Diagrama de Estrutura (DE)

- Cada módulo, em um projeto, deve ser especificado, podendo-se utilizar o **Português Estruturado**, onde o vocabulário utilizado é o do projeto do software, ou seja, os nomes a serem usados como variáveis, arquivos e processos são os nomes usados no projeto de banco de dados (nomes de tabelas, arquivos e atributos) e no Diagrama de Estrutura (nomes dos módulos).

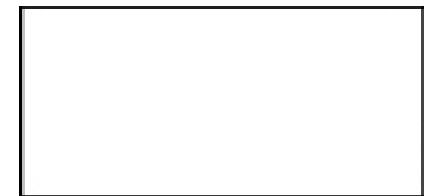


Diagrama de Estrutura (DE)

- A **hierarquia dos componentes funcionais** representa as chamadas entre os módulos. A seta indica qual módulo chama outro. Portanto, o Diagrama de Estrutura representa somente a **interface entre os módulos**.
- A lógica interna de cada módulo é dada pela especificação dos módulos em Português Estruturado (ou outra ferramenta de especificação)

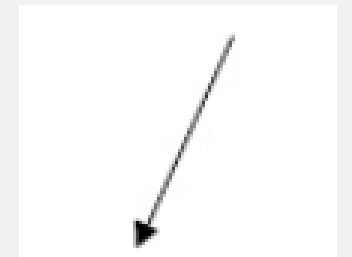


Diagrama de Estrutura (DE)

- O **vetor de dados** é um elemento de dados ou um conjunto de dados que um módulo passa para outro como parâmetro na chamada ou retorna como resultado de uma chamada.
- O **vetor de dados** contém informações com conteúdo relacionado ao domínio da aplicação. A seta do vetor de dados indica o sentido em que tal **dado é passado** (de qual módulo para qual)

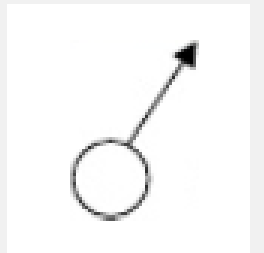
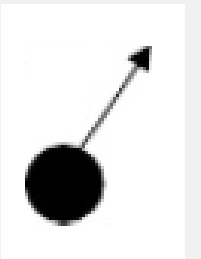


Diagrama de Estrutura (DE)

- O **vetor de estados** é semelhante ao vetor de dados, porém seu significado está relacionado ao **controle do programa** e não ao conteúdo de dados propriamente dito.
- Um **vetor de estados** é uma informação de controle que será usado para o teste de uma condição no módulo que o recebe. Geralmente é booleano e determina a execução ou não de outros módulos.



Exemplo DE

♦ COMUNICAÇÃO ENTRE MÓDULOS

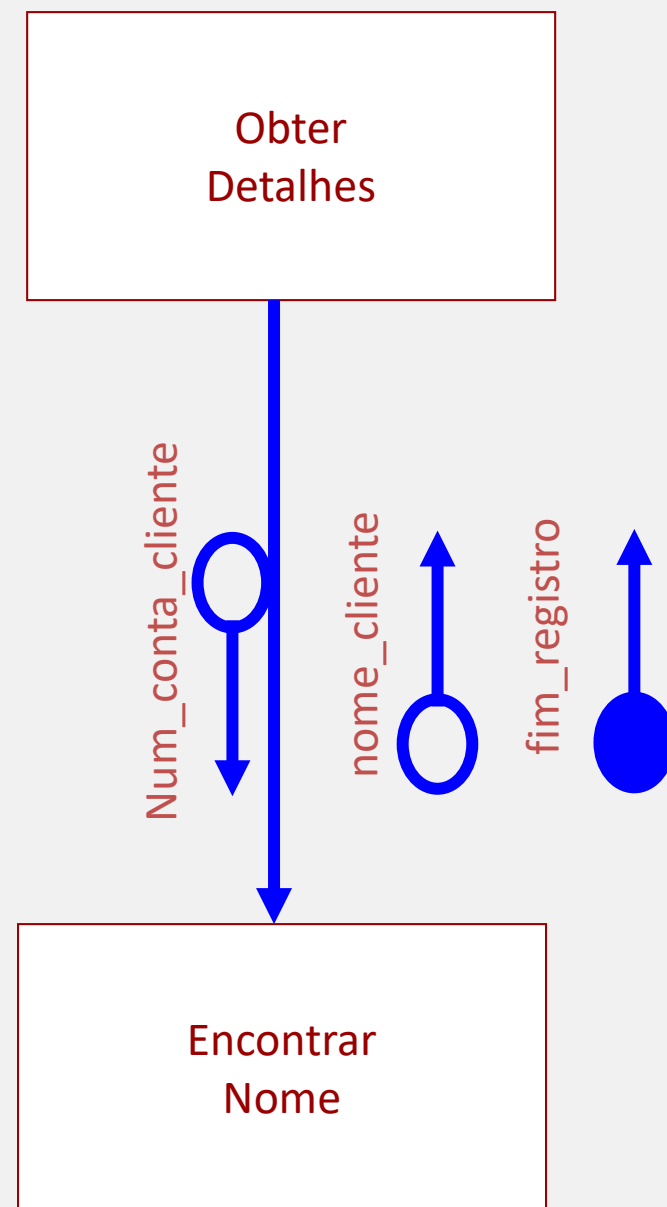
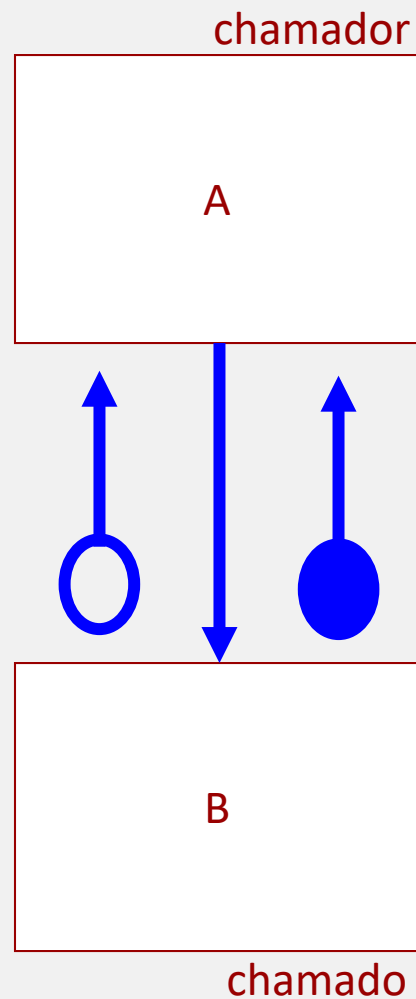
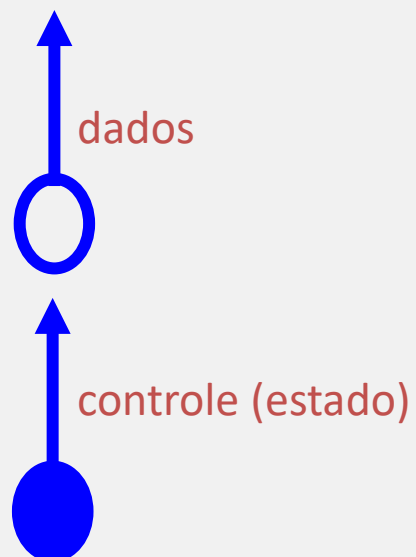


Diagrama de Estrutura (DE)

- Martin e McClure (1991) propõe algumas regras básicas de controle para um Diagrama de Estrutura:
 - Existe um módulo, e somente um, na parte superior (primeiro nível) do Diagrama de Estrutura, que é o ponto onde começa o controle (este módulo é chamado de raiz).
 - A partir da raiz, o controle é passado, nível por nível, para os outros módulos, retornando sempre ao módulo que fez a chamada.
 - Existe no máximo uma relação de controle entre quaisquer dois módulos no Diagrama de Estrutura, ou seja, um módulo que chama outro, não pode ser chamado por este último

FERRAMENTAS COMPLEMENTARES



UDESC
UNIVERSIDADE
DO ESTADO DE
SANTA CATARINA

Dicionário de Dados (DD)

- **Repositório de informações** sobre os componentes do modelo de sistemas: fluxos de dados, depósitos de dados e processos;
- **Define:**
 - O significado (apresentado por comentários) dos fluxos e depósitos do DFD;
 - A composição de pacotes de dados que se movimentam pelos fluxos;
 - A composição dos pacotes de dados nos depósitos;
 - Os valores e unidades relevantes de partes elementares dos fluxos e depósitos de dados;

Dicionário de Dados (DD)

- Um dicionário de dados é uma coleção de metadados que contêm definições e representações de elementos de dados.

Dicionário de Dados (DD)

- Dentro do contexto de SGBD, um dicionário de dados é um grupo de tabelas, habilitadas apenas para leitura ou consulta, ou seja, é uma base de dados, propriamente dita, que entre outras coisas, mantém as seguintes informações:

Dicionário de Dados (DD)

- Definição precisa sobre elementos de dados
- Perfis de usuários, papéis e privilégios
- Descrição de objetos
- Integridade de restrições
- *Stored* procedures (pequeno trecho de programa de computador, armazenado em um SGBD, que pode ser chamado frequentemente por um programa principal) e gatilhos (*triggers*)
- Estrutura geral da base de dados
- Informação de verificação
- Alocações de espaço

Dicionário de Dados (DD)

Simbologia

Símbolo	Significado
=	é composto de
()	opcional (pode estar presente ou ausente)
{ }	iteração
[]	escolha em uma das alternativas
**	comentário
@	identificador (chave) em um depósito
	separa opções alternativas na construção [].

Dicionário de Dados (DD)

Exemplos (Descrição dos dados)

pedido = nome-cliente + 1{ item }8

Endereço Cliente = @Código Cliente

Rua

Número

Complemento

[CEP]

Sigla-UF % Sigla da unidade da federação %

Telefone % Telefone para contato, com DDD %

Dicionário de Dados (DD)

Exemplos (Descrição dos dados)

- **Opcional**

- $endereço_cliente = (endereço_de_remessa) + (endereço_de_cobrança)$
 - poderia ser somente o de remessas, somente o de cobrança, os dois, ou nenhum dos dois;
- $endereço_cliente = endereço_de_remessa + (endereço_de_cobrança)$
 - endereço de remessa é obrigatório; endereço de cobrança é opcional;

- **Escolha de Alternativas**

- $sexo = [M \mid F]$
- $tipo_de_cliente = [Governo \mid Indústria \mid Universidade \mid Outro]$

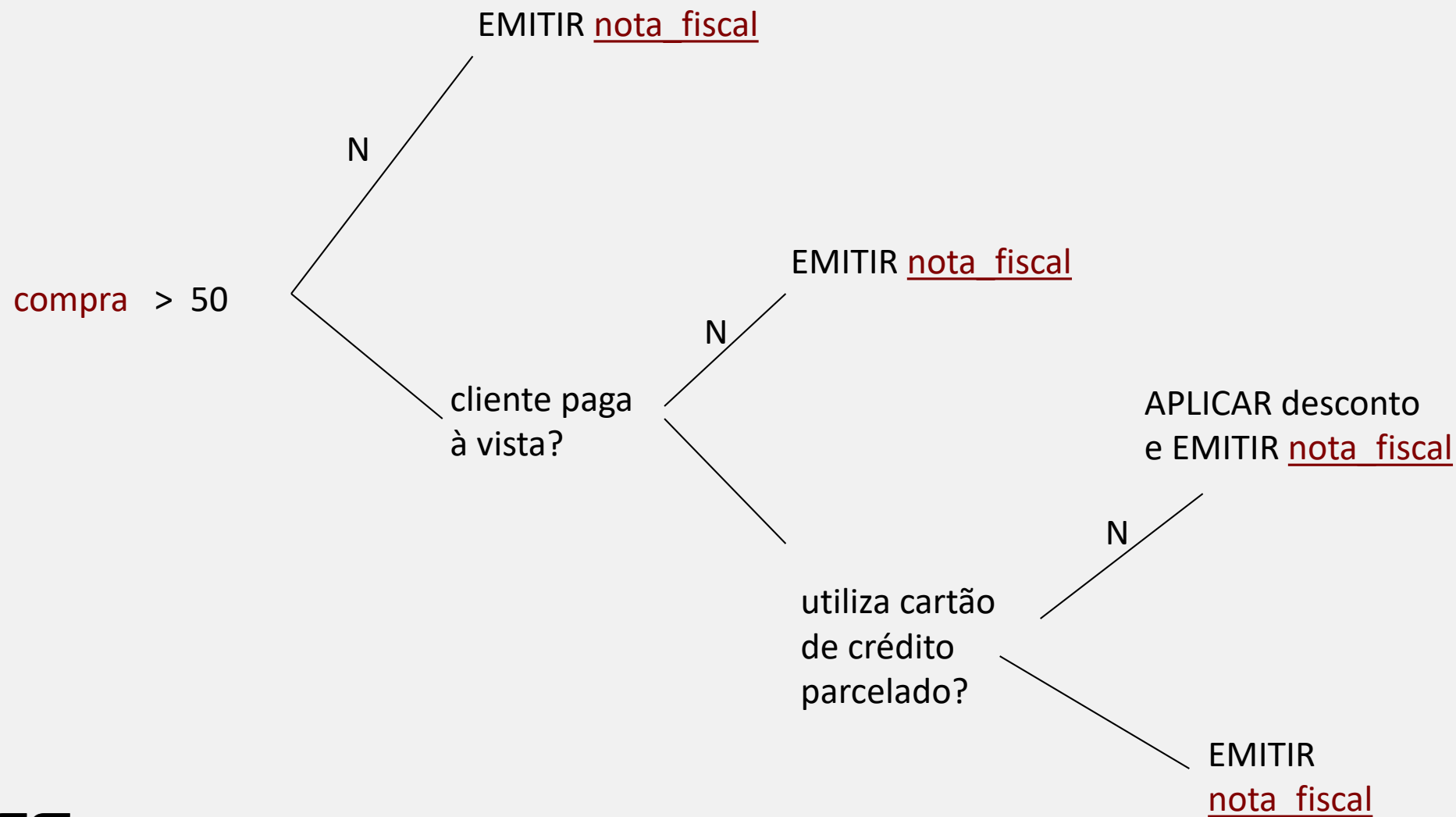
Dicionário de Dados (DD)

Exemplos (Descrição dos dados)

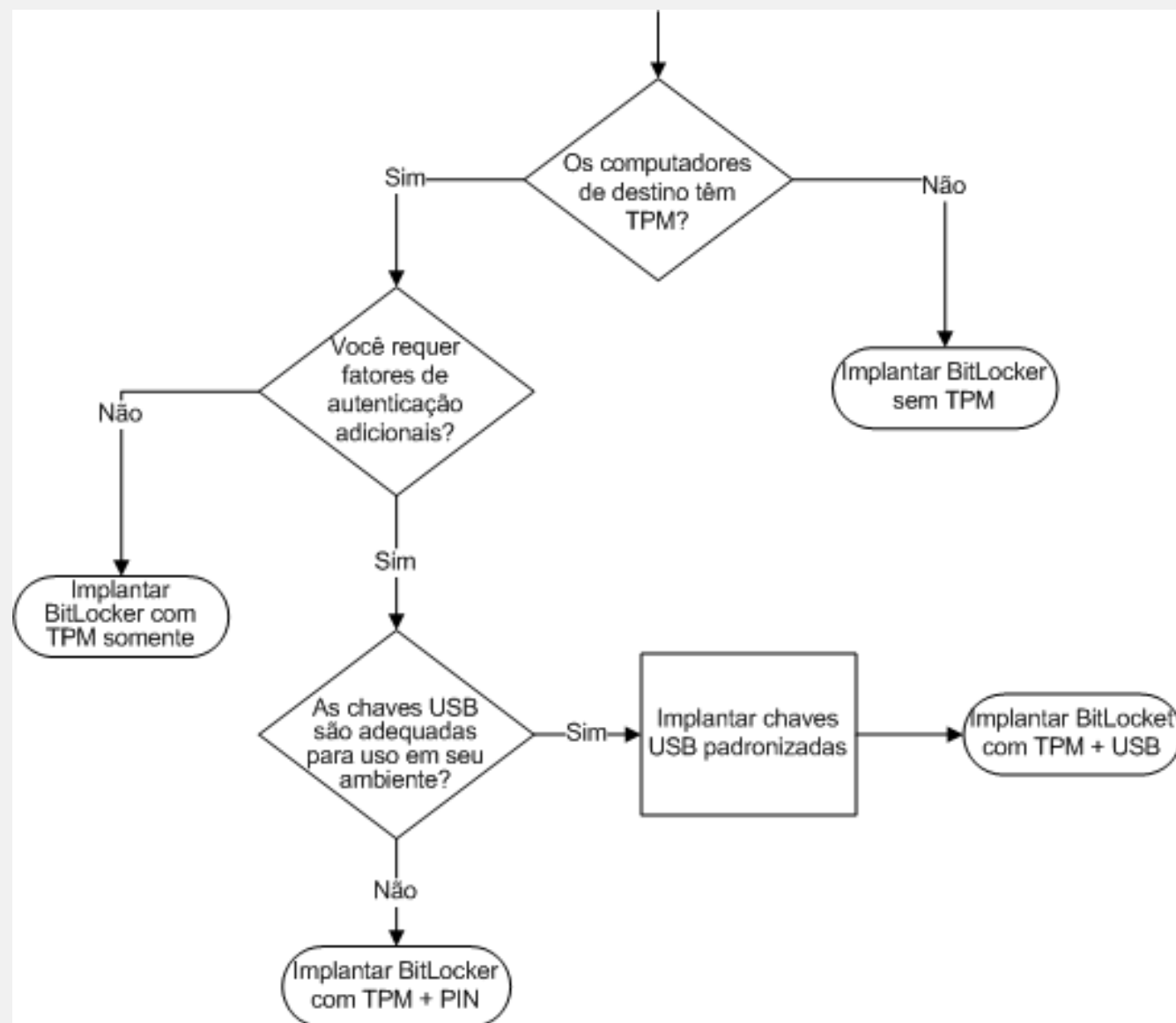
- **Iteração**

- $\text{pedido} = \text{nome_do_cliente} + \text{endereço_de_remessa} + \{\text{item}\}$
 - $\{\text{item}\}$: zero ou mais ocorrências de um item
- pode-se desejar especificar os **limites superior e inferior** da iteração;
 - exemplo: um pedido sem itens não faz sentido.
 - $\text{pedido} = \text{nome_do_cliente} + \text{endereço_de_remessa} + 1 \{\text{item}\} 10$
 - $1 \{\text{item}\} 10$, onde 1 é o limite inferior e 10 o superior

Árvore de Decisão



Exemplo Árvore de Decisão



Exemplo Tabela de Decisão

CONDIÇÕES	Regras 1	Regras 2	Regras 3	Regras 4	Regras 5	Regras 6	Regra 7	Regra 8
Ocupa cargo de chefia ?	S	S	S	S	N	N	N	N
Idade maior que 40 anos ?	S	S	N	N	S	S	N	N
Mais de 2 anos no cargo ?	S	N	S	N	S	N	S	N
AÇÕES								
Exame especial	X	X	X		X	X		
Exame normal				X			X	X

onde: S=sim; N=não; X=ação a ser executada.

Português Estruturado

- O Português Estruturado é, na verdade, uma simplificação extrema do Português, limitada a umas poucas palavras e estruturas que têm um significado muito bem definido.
- Portugal

Exemplo Português Estruturado

PARA cada pedido cliente
 ENCONTRAR ident cliente em CLIENTE
 SE (ident cliente existe) ENTÃO
 ENVIAR pedido aceito para Verificar Pgto
 SENÃO
 ENVIAR cliente novo para Cadastrar Cliente
 FIM_SE
FIM_PARA

ETAPAS DO PROJETO DE SISTEMAS

Etapas do Projeto de Sistemas

- 1) Levantamento
- 2) Projeto Lógico
- 3) Projeto Físico
- 4) Desenvolvimento
- 5) Testes / Homologação
- 6) Implantação
- 7) Consolidação

Etapas do Projeto de Sistemas

1. Levantamento:

Permite levantar as informações do cliente *in loco*, para compreender como o cliente trabalha, e permitirá a proposição de correções e melhorias, já obtendo os requisitos para o novo sistema.

Etapas do Projeto de Sistemas

2. Projeto Lógico:

Compreende as etapas de especificação do novo sistema, utilizando a metodologia e ferramentas da análise de sistemas.

3. Projeto Físico:

Etapa onde especifica-se a tecnologia, as limitações, os desempenhos, os usuários, o manuseio e a padronização do sistema.

Etapas do Projeto de Sistemas

4. Desenvolvimento:

- Nesta etapa as rotinas quer sejam manuais ou automáticas são passadas para o mundo real.
- As manuais serão os treinamentos e mudanças de forma de trabalho.
- Já nas automáticas ocorrerá a programação das rotinas e módulos que apoiará o trabalho dos usuários, bem como o banco de dados.

Etapas do Projeto de Sistemas

5. Testes / Homologação:

Parte desta etapa já foi feita na 4ª etapa, mas a fase de homologação – teste pelo cliente – deve ser efetuada quando estiver trabalhando em uma empresa desenvolvedora.

Etapas do Projeto de Sistemas

6. Implantação:

Nesta fase cria-se o cronograma para a implantação do sistema no cliente.

7. Consolidação:

Período de observação onde se estabelecerá que o sistema esteja implantado.

Fases

1) Escopo Básico - Levantamento dos Dados

- 1.1) Identificação do Cliente (*minimundo*)
- 1.2) Levantamento das Informações do Sistema do Cliente e DER Atual (*minimundo*)
- 1.3) Identificação dos Problemas no ambiente do Cliente
- 1.4) Requisitos para o Novo Sistema (Objetivos)
- 1.5) Alternativas de Solução (Plano de Desenvolvimento das Propostas)
- 1.6) Definição da Alternativa Recomendada
- 1.7) Justificativa da escolha
- 1.8) Documentação de Informações do Cliente

Fases

2) Projeto Lógico (modelo da essência do novo sistema)

2.1) Modelo Ambiental

2.1.1) Proposta Geral do Sistema

2.1.2) Lista de Eventos

2.1.3) Diagrama de Contexto

Fases

2.2) Modelo Comportamental

2.2.1) Modelo Esquemático

2.2.1.1) Modelo de Dados (DER - Diagrama de Entidade e Relacionamento)

2.2.1.2) DFD Particionado por Evento e DC

2.2.2) Modelo Detalhado

2.2.2.1) Dicionário de dados

2.2.2.2) Especificação dos Processos

Fases

3) Projeto Físico

3.1) Determinação das Fronteiras da Automação

3.2) Requisitos Tecnológicos do Sistema

3.3) Requisitos para Manutenção do Sistema

3.3.1) Nomenclatura

3.3.2) Interface com o Usuário

3.3.2.1) Padronização das Telas, Relatórios e Mensagens

3.3.3) Requisitos para operação do Sistema

3.3.4) Requisitos para o desenvolvimento/manutenção do sistema

3.3.5) Requisitos de desempenho

Fases

- 4) Desenvolvimento Físico das Rotinas e BD**
- 5) Teste Integrado e Homologação**
- 6) Implantação**
- 7) Consolidação**

EXERCÍCIOS

Exercícios

1. Baseado nos exemplos de MiniMundo fornecidos nesta apresentação, faça os diagramas de contexto de cada um.

Obrigado

*jacksonpradolima.github.io
github.com/ceplan*