

# Análise de Sistemas

## 1. O que é a análise?

- Processo de decomposição do todo em partes e estudo de cada uma das partes.
- Processo de estudo de uma situação com o intuito de melhorá-la através da utilização de melhores procedimentos ou métodos.

(Definições Genéricas)

## 2. O que não é análise:

- Estudar uma sistema para ver que processos existentes devem ser tratados informaticamente e quais devem ser tratados manualmente. Este é um problema secundário que a análise poderá desvendar, mas o mais importante é o conhecimento aprofundado do sistema.
- Determinar que alterações devem ser feitas. Uma alteração poderá ser um resultado não uma necessidade. Uma alteração pode não ser necessária ou possível.
- Determinar a melhor maneira de resolver um problema. Os problemas da empresa ou da área a analisar devem ser vistos como um todo. A resolução de um problema particular poderá advir da resolução global.

## 3. O que é um sistema?

- Conjunto de partes ou entidades que concorrem para um determinado fim ou resultado.
- Virtualmente todas as organizações são sistemas.

**Exemplos:** Sistema nervoso, sistema solar, sistema económico, sistema escolar.

### 3.1. Definição de sistema:

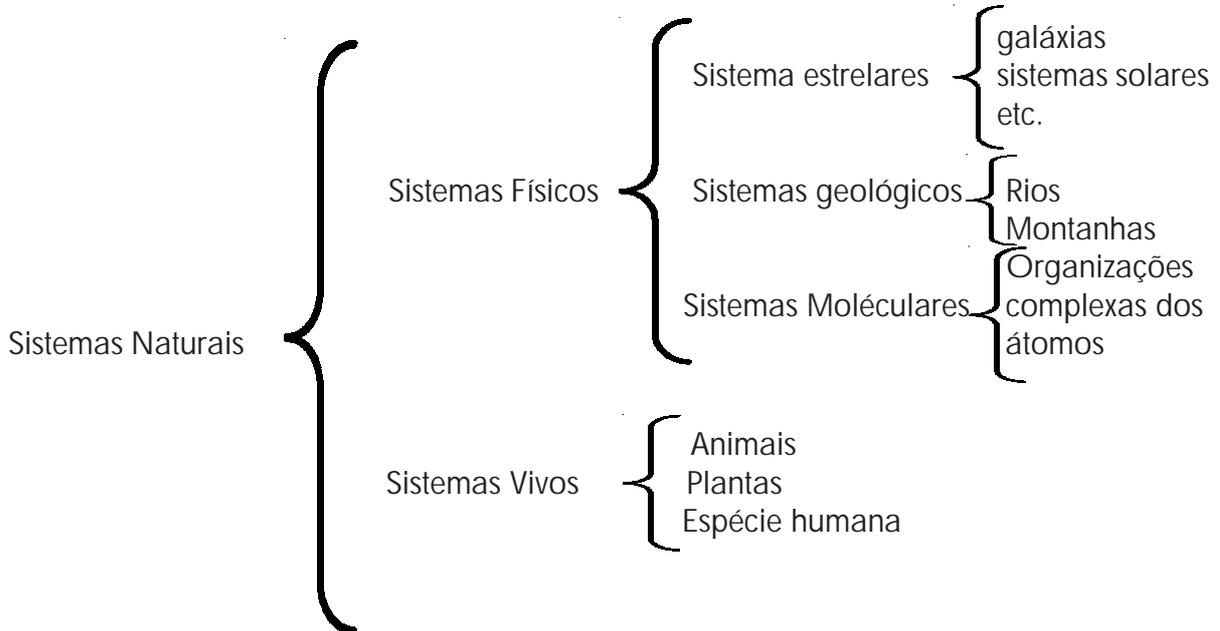
Um conjunto organizado de doutrinas, ideias ou princípios, habitualmente previstos para explicar a organização ou funcionamento de um conjunto sistemático.

### 3.2. Tipos comuns de sistemas

- Sistemas Naturais
- Sistemas feitos pelo homem

### 3.2.1. Sistemas Naturais

A maioria dos sistemas não é feito por pessoas. Eles encontram-se na natureza e servem os seus próprios propósitos.



### 3.2.2. Sistemas Feitos pelo Homem



#### ***Não automatizados***

- Estes sistemas são construídos, organizados e mantidos por seres humanos:

- \* Sistemas Sociais - organizações de leis, doutrinas, costumes, etc.
- \* Sistemas de transporte - redes rodoviárias, canais, linhas aéreas, etc.
- \* Sistemas de comunicação - telefone, fax, etc.
- \* Sistemas financeiros - contabilidade, inventários, controle de stocks, etc.

Hoje, muitos destes sistemas já são semi-automatizados e até nalguns casos seria impensável que o não fossem (ex: contabilidade).

#### ***Automatizados***

Sistemas feitos pelo homem, que interagem com ele, ou são controlados por um ou mais computadores. Existem muitos tipos diferentes de sistemas automatizados, mas todos têm componentes comuns. ( Hardware, software, pessoas, dados, procedimentos).

Sistemas automatizados

- Sistema On-Line
- Sistemas tempo real
- Sistemas de Apoio à decisão
- Sistemas baseados no conhecimento

### Sistemas On-Line

Sistemas On-line, são os que recebem entradas directamente, e também os sistemas em que as saídas, ou os resultados do processamento, são dirigidas para onde sejam necessárias. (ex: internet, correio electrónico).

### Sistemas de Tempo-Real

É um sistema de processamento, que controla um ambiente pelo recebimento de dados, e o seu processamento e apresentação de resultados é feito com a rapidez suficiente para afectar o ambiente naquele momento. (ex: sistemas de caixas automáticas, etc).

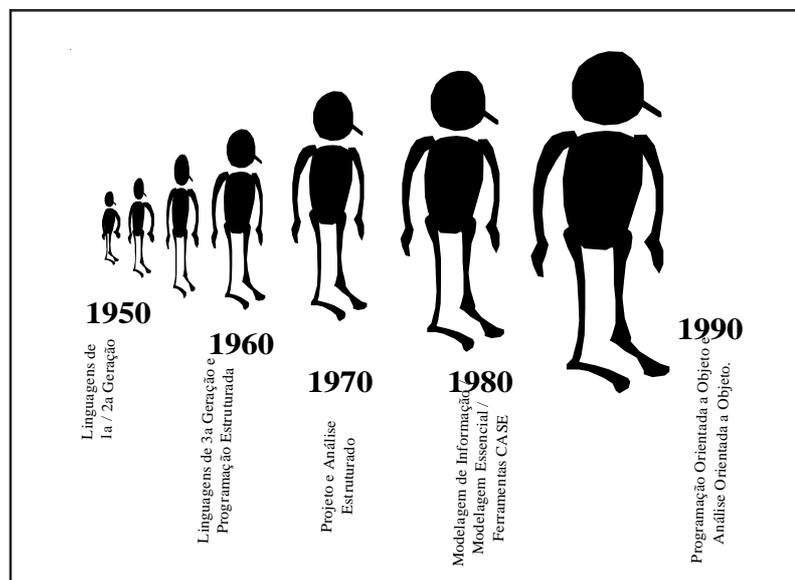
### Sistemas de Apoio à decisão

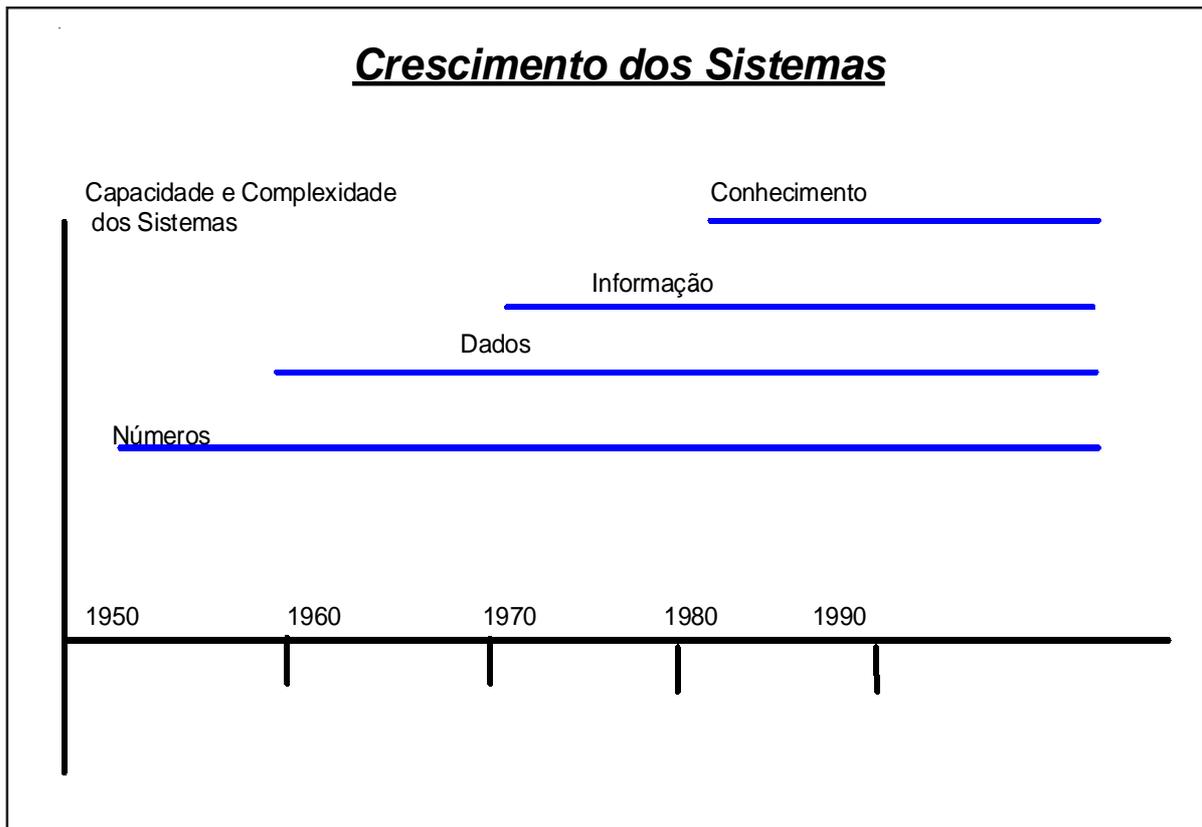
Estes sistemas de processamento não tomam decisões por eles mesmos, mas auxiliam os gerentes e outros profissionais. São sistemas que só funcionam quando para isso são solicitados. (ex: word, excel, sistemas de análise estatística).

### Sistemas baseados no conhecimento

Estes sistemas estão associados ao campo da Inteligência artificial, também chamados por "Sistemas Especialistas". Os cientistas de computação que trabalham no campo da I.A. , têm como meta a produção de programas que imitam o desempenho humano em uma ampla variedade de tarefas.

Análise de Sistemas de Processamento de Dados





### 3.3. Princípios Gerais de Sistemas

- Quanto mais especializado é um sistema, menos capaz ele é de se adaptar a circunstâncias diferentes.
- Quanto maior for um sistema, maior o número de seus recursos que serão destinados á sua manutenção diária.
- Os sistemas sempre fazem parte de sistemas maiores, e sempre podem ser divididos em sistemas menores.
- Os sistemas crescem. (não se pode presumir que um sistema que se tenha construído vá permanecer estático. O custo da sua expansão pelo tempo deve ser incluído nos cálculos do "custo-benefício".)

## 4. Sistema de desenvolvimento informático

Um **sistema de desenvolvimento informático** tem duas grandes componentes:

- **Análise** ( analysis)
- **Desenho** (Design)

O **desenho de Sistemas** é um processo de planear um novo sistema quer alterando por completo o anterior quer complementando-o.

## 5. Análise de sistema

A **análise de sistema** é o processo de conhecer (obter) e interpretar factos, diagnosticar problemas e através deles melhorar o sistema.

## 6. Sistema de desenho e análise

O sistema de desenho e análise é um processo de estudo da situação dum empresa para ver como ela funciona e que melhoramentos necessita.

A **Análise** especifica "o que" o sistema deve fazer;

O **Desenho** mostra "como" acompanhar esse objectivo.

Este é o trabalho do **Analista**.

O **Papel do Analista** não é, só, resolver problemas actuais do sistema, poderá ser chamado, inclusive, para estudar a expansão da empresa.

## 7. Participantes no desenvolvimento dum sistema

Os participantes no desenvolvimento dum sistema são:

- Utilizadores
- Analistas de Sistemas
- Programadores
- Pessoal Operativo

## 7.1. Utilizadores - *Tipos de Utilizadores*

**Utilizadores Directos** - são os que actuam interactivamente com o equipamento. Alimentam o sistema através do input de dados ou recebem as respostas através do output.

**Utilizadores Indirectos** - beneficiam dos resultados ou relatórios produzidos pelo sistema mas não interagem directamente nem com o hardware nem com o software (exemplo: Gestor de Marketing responsável pela análise de custos).

**Administradores** - responsáveis pela gestão de grandes sistemas de informação. Responsáveis pela aprovação de investimentos no desenvolvimento de aplicações.

## 7.2. Analistas de Sistemas

- Membro essencial de qualquer projecto de desenvolvimento de sistemas.
- O analista conduz o estudo do sistema, a fim de conhecer os factos relevantes da actividade da empresa sempre em estrita relação com os vários níveis utilizadores.
- Tem a responsabilidade acrescentada de desenhar o novo sistema.
- Conduzir a Investigação, desenvolvendo as especificações de processos e o desenho do projecto.

## 7.3. Programadores

Depois do projecto estar pronto (da parte do analista), é entregue ao programador, que vai transformá-lo em código.

## 7.4. Pessoal Operativo

Responsável pelo centro de processamento, rede de telecomunicações, pela segurança do Hardware e dos dados do computador.

## 8. Relações Utilizador-Analista

- 8.1. Cada tipo de utilizador tem informações essenciais acerca da organização (como deve funcionar e como está a funcionar).
- 8.2. O analista deve ser um dos elementos capazes de oferecer ideias acerca da melhor utilização da informática.
- 8.3. As informações captadas pelo analista são as bases para o desenho do novo ou modificado sistema.
- 8.4. O analista tem dificuldades em aprender o funcionamento da empresa para determinar os requisitos do sistema através das palavras do utilizador.
- 8.5. O analista poderá ficar sobrecarregado de detalhes rapidamente.
- 8.6. O analista deve combinar o que é tecnicamente possível com o que vale a pena ser feito para a empresa.
- 8.7. A especialização do sistema feita pelo analista e, que traduz um contrato entre o utilizador e o analista, torna-se, por vezes, de difícil entendimento para o utilizador.
- 8.8. Essa mesma especificação quando construída a pensar só no utilizador, poderá trazer problemas ao desenhador ou ao programador.
- 8.9. O utilizador não conhece o suficiente sobre o processamento de dados para saber o que é, ou não, viável.

Em resumo, dificuldades técnicas e políticas; problemas de comunicação e de tempo são em suma as principais contrariedades na relação utilizador-analista.

# Sistema de Informação & Sistema Informático

## 1. Sistema de Informação

**Informação** – é qualquer forma de comunicação, formal ou informal, sendo um conceito chave para um bom funcionamento de um sistema.

**Sistema de Informação** - Em qualquer organização existem múltiplas actividades de processamento de *Informação*:

- Registo e armazenamento de informação
- Pesquisa/Consulta de informação
- Comunicação de Informação ( Verbal, escrita,...)
- "Produção" de Informação
- ...

ou seja, um **Sistema de Informação** – é o mecanismo que permite armazenar, gerar e distribuir informação, com a finalidade de suportar as funções operacionais e de gestão de uma organização. Este sistema pode ser automático ou manual.

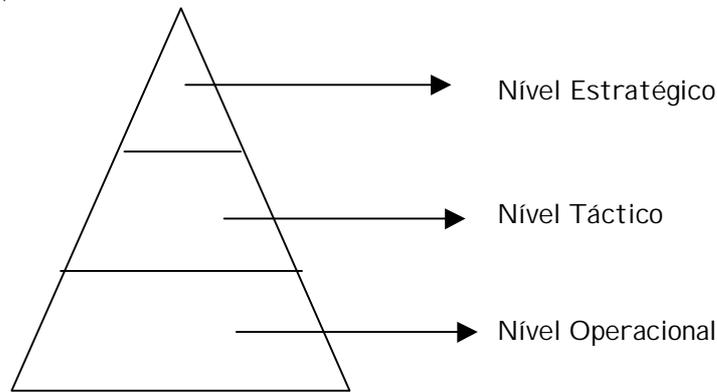
As Actividades de processamento de Informação numa organização geralmente estão interrelacionadas, isto é, cada posto de trabalho recebe informação proveniente doutros postos de trabalho e, por sua vez, fornece informação a outros postos de trabalho.

Essas actividades fazem parte do **Sistema de Informação da Organização**.

**Organização** – é um sistema onde normalmente existe o elemento humano (Homem); é uma estrutura hierárquica muito rígida.

## As diferentes necessidades de informação

No nível de gestão de uma organização há sempre uma estrutura *caracterizada por* três níveis que são o *Estratégico, Tático, Operacional ou Controlo*.



No **nível Estratégico** deve-se fazer um estudo e delinear-se objectivos com base nas previsões que foram feitas, ou seja, definem-se políticas estratégicas da empresa a médio e longo prazo, são elaborados planos estratégicos, são tomadas as decisões e é dado um grande relevo à informação externa. (A estratégia a seguir para obter maior lucro).

No **nível Tático** (Controlo de Gestão) são elaborados planos táticos utilizando modelos de gestão num curto prazo. Vão ser elaborados planos, que vão dar origem a um conjunto de instruções passando ao nível operacional, elaborando ordens de operação num horizonte a muito curto prazo. Visa garantir que os objectivos vão sendo cumpridos.

**Nível Operacional** – Execução de tarefas; Controlo das actividades do dia-a-dia.

## Diferentes Tipos de Sistemas de Informação

**Sistemas de apoio à decisão:** são sistemas constituídos com base em regras e modelos de decisão, estão vocacionados para os gestores de topo, permitem a simulação e a tomada de decisões.

Paradigmas típicos usados são:

Linguagens de inteligência artificial;

Folha de cálculo electrónica.

**Sistemas de informação de Gestão:** são constituídos por várias aplicações integradas, estão vocacionados para os utilizadores, são responsáveis pelo tratamento de informação com o objectivo da produção de mapas.

**Sistemas de processamento de transações:** são constituídos por aplicações independentes, vocacionados exclusivamente para o pessoal operacional. Estas aplicações são típicas de paradigmas de linguagens funcionais (Cobol, Pascal).

**Objectivos:**

- Apoiar a tomada de decisões
- Assegurar o Controlo da utilização dos recursos da organização
- Coordenar a actividade dos diversos elementos que fazem parte da organização
- Fornecer informação para o exterior.

**Componentes:**

- Informação manipulada (ficheiros; base de dados)
- Meios utilizados para processar a informação (pessoas, máquinas, suportes, ...)
- Regras de processamento da informação (procedimentos administrativos, circuitos de informação, programas de computador).

***Ao desenvolver um Sistema de Informação é necessário:***

- 1- Interactividade entre o construtor e o utilizador do SI.
- 2- Normalizar relações de forma a existir um padrão para se construir o **modelo**.

**Modelo** – é a representação de um sistema.

*Características do Modelo:*

- é mais barato de construir e mais simples de modificar (porque ainda não é o programa final)
- representa o sistema ressaltando determinadas características.

### **Decomposição do Sistema de informação:**

A decomposição do sistema de informação em módulos é essencial por razões de:

- Tempo de implementação
- Custo
- Rentabilidade
- Flexibilidade

**Cada Módulo** deve ser rigorosamente caracterizado no que diz respeito a:

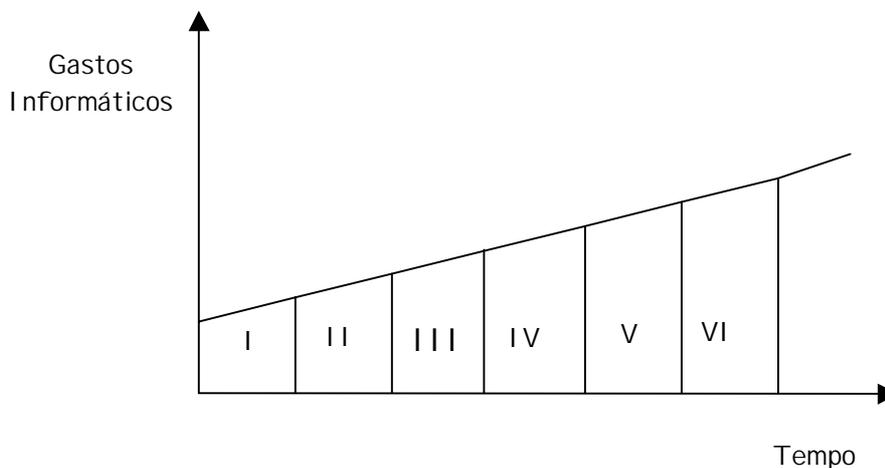
- Conteúdo Funcional (que actividades abrange)
- Conteúdo informacional (que informações abrange)
- Interfaces (interligações) com os outros módulos (que informação recebe, fornece ou partilha).

### **INSERÇÃO DA INFORMAÇÃO NAS ORGANIZAÇÕES**

Inicialmente a informática introduziu-se nas organizações como um pequeno órgão que automatiza os processos administrativos.

Com o decorrer dos tempos observou-se que os proveitos retirados por esta secção pelo uso da informática eram enormes e resolveu-se, então ampliar o âmbito de acção da informática a todas as áreas informais da empresa. Neste sentido e para que a informática possa prestar um serviço independente optou-se colocando nas estruturas como órgão de STAFF.

### **EVOLUÇÃO DOS SISTEMAS DE INFORMAÇÃO**



**Modelo Richard Nolan** – modelo que representa em seis estádios esta evolução.

**I – Inicializado:** Introdução dos computadores nas empresas e novas tecnologias.

**II – Contágio:** Adesão às novas tecnologias de informação em massa onde se investe em demasiado e por outro lado encorajar a sua utilização.

**III – Controle:** Fase estacionária onde o planeamento tem em vista a diminuição dos gastos. O planeamento de sistemas de informação leva a uma interligação dos sistemas (multipostos, redes, todas as formas que rentabilizam os sistemas – controlar os custos).

**IV – Integração:** Fase dos multipostos em que a informática passa a assumir-se como órgão se Staff, necessidade de obter sistemas de informação integrados.

**V – Gestão de Dados:** Necessidade que o gestor tem na organização e na adesão de dados, visto tirar partido deles ou seja, necessidade de eficiência na gestão de dados. Sistemas de Informação e Gestão.

**VI – Maturidade:** Surge a gestão integrada que se trata de colher uma só vez informação que nos permita várias utilizações, Equipamento adequado.

### *DESENVOLVIMENTO DE SISTEMAS DE INFORMAÇÃO*

#### **Dificuldades:**

- Natureza do produto;
- Especificação explícitas de requisitos;
- Actividades complexas e de difícil apresentação;
- Participação de vários técnicos cada um com linguagens próprias e diferentes;
- Hardware, única parcela palpável do sistema de informação.

#### **Características de um bom sistema:**

- Oportunidade: informação deve ter oportunidade de sair.
- Selectividade: as informações devem ser escolhidas premonitivamente.
- Simplicidade: a informação deve ser mais simples quanto possível.
- Eficiência: um sistema de informação deve ser eficaz e eficiente, ser rápido e fazer bem.

## 2. Sistema Informático

O Sistema formado pelos componentes informáticos (Hardware, Software, Base de Dados) que fazem parte do sistema de informação da organização.

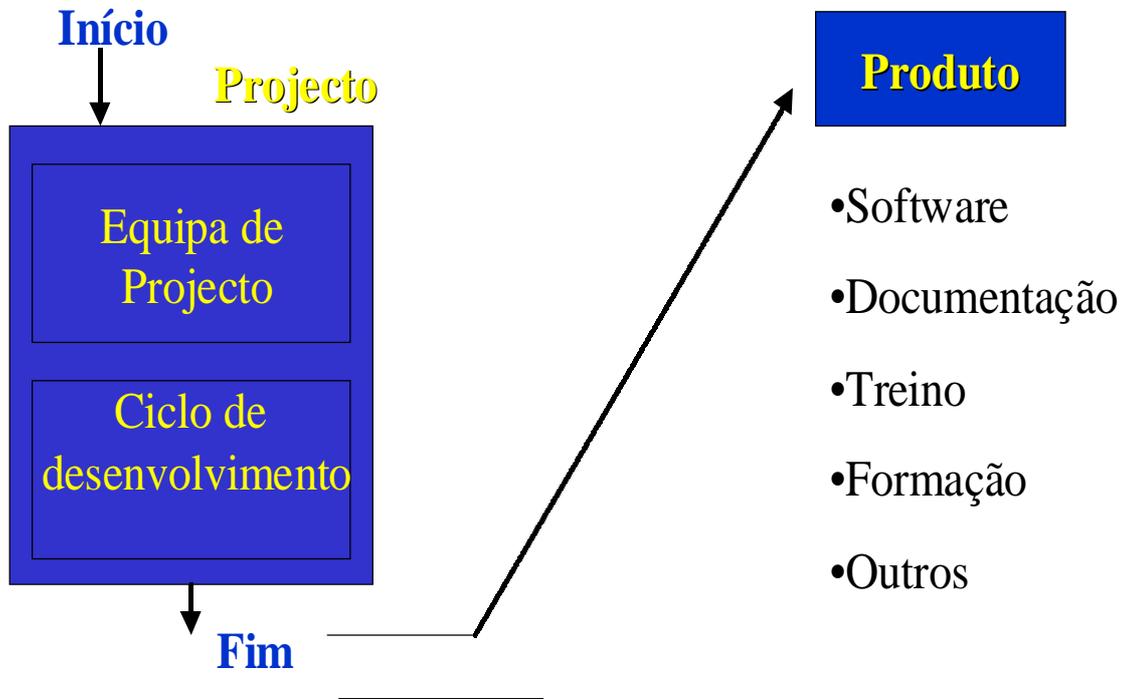
O sistema informático é um subsistema do sistema de informação.

A concepção do sistema informático deve ser baseada num sistema de informação adequado à organização.

**DIFERENÇA ENTRE UM SISTEMA DE INFORMAÇÃO E UM SISTEMA INFORMÁTICO:** O sistema de informação é abrangente enquanto o sistema informático é mais específico.

# Gestão de Projectos

As actividades efectuadas no âmbito do projecto destinam-se a produzir um “produto” o qual modela o funcionamento do processo correspondente.



## 1. Características dos Projectos:

- \* Um ponto de início
- \* Um ponto Final
- \* Um produto Final:
  - Software
  - Documentação
  - Treino
  - Outros produtos
- \* Uma equipa de projecto
- \* Um ciclo de Desenvolvimento

## 2. Tipos de Projectos

- Informatização de uma (ou mais) aplicação(s) ou serviço(s)
- Modificação substancial de uma aplicação (actualização, conversão,...)
- Integração ( e modificação) de aplicações
- Introdução de métodos de Trabalho ( projecto piloto)

## 3. Gestão de Projectos

Porque é que os Projectos de Software estão sempre atrasados?

Ou

Porque é tão difícil dirigir projectos com Sucesso?

**Dirigir um projecto com Sucesso é ... Satisfazer as expectativas de:**

- *Utilizadores*
- Director Financeiro
- Chefe da Informática
- Equipa de Projecto

### O que significa dirigir um projecto?

Dois tipos de actividades no projecto:

- as conducentes à realização do produto
- as respeitantes à Gestão e Direcção do Projecto

### Quais são as Tarefas que competem ao gestor de projecto?

- antes do projecto começar
- no decurso do projecto
- no fim do projecto

## 4. Tarefas do Gestor de Projectos

### Antes do projecto começar

- 1• Participar na definição do projecto
- 2• Definir claramente os objectivos
- 3• Seleccionar a equipa ... ou aceitá-la
- 4• Relembrar ou estudar as técnicas de estimativa, planificação, controlo, ...e seleccioná-las
- 5• Estimativa de tempos e custos
- 6• Planeamento:
  - Tarefas e suas prioridades
  - Relação Homem-tarefa
  - Tempo
  - Definição de Pontos de controlo e revisão
- 7• Definir Standards, Técnicas de DP, Linguagens, Programas-produto.... A utilizar... Ou definir quem os vai seleccionar
- 8• Definir procedimentos de controlo de alterações

### Durante o projecto

- 1• Controlo do projecto
- 2• Revisões
- 3• Documentação do projecto (não do produto)
- 4• Direcção do Pessoal
- 5• Assegurar ligação com os departamentos utilizadores
- 6• Re-estimar
- 7• Re-planificar
- 8• Estudar e aceitar (ou rejeitar) as alterações.

### Fim do projecto

- 1• Assegurar-se de que o projecto está completo, isto é :
  - Produto final ( software, documentação,..)
  - Todas as cláusulas do contrato estão cumpridas
  - O produto é fácil de alterar e manter
  - Todos os interessados estão satisfeitos com o produto/projecto
- 2• "Vender" o Produto Final ( ao utilizador, à direcção, à exploração,...)
- 3• Fazer um relatório que permita refinar as técnicas e métodos e colher experiência
- 4• Preparar-se para o próximo projecto.

## Quais são as responsabilidades do gestor do projecto?

- Estimar
- Planificar
- Controlar
- Dirigir a Equipa

## Quais são as responsabilidades que não são do gestor do projecto?

- As responsabilidades da equipa
- As responsabilidades do utilizador
- As responsabilidades do seu chefe

## Que conhecimentos e capacidades deve ter o gestor de projecto?

- Conhecimento de Técnicas de Gestão:  
*Estimativa, Controlo, Revisão...*
- Conhecimento de Técnicas Informáticas:  
*Conhecimentos Gerais de Hardware, Software,...*
- Características Pessoais:  
*Comunicação oral e escrita*  
*Negociação e relações Humanas*  
*Direcção Pessoal*
- Experiência:  
*De projectos ou actividades similares ... que permitem maior probabilidade de sucesso*

# Ciclo de Vida de um Projecto

1. Ciclo de vida de um projecto desenvolvimento clássico
2. Ciclo de vida moderno de um projecto

## 1. Ciclo de vida de um projecto desenvolvimento clássico

**Fases do Projecto** - Tarefas a realizar e produto final a obter em cada fase.

**Fase 0** - Estudo das Necessidades ( requirements)

**Fase 1** - Desenho do sistema (Design)

**Fase 2** - Criação ( desenvolvimento) do sistema (Implementation)

**Fase 3** - Instalação ( Arranque) ( Cut-Over)

Pós-projecto - manutenção e exploração

### Fase 0 - Estudo das Necessidades

#### Tarefas

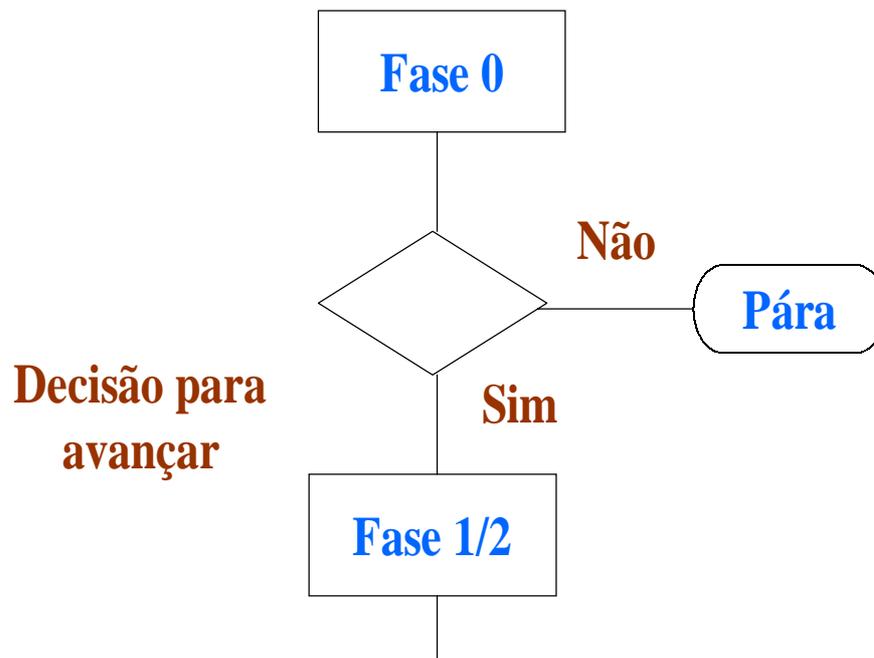
- 1 - Definição do Projecto
  - Objectivos
  - Grupo de Trabalho
  - Chefe do projecto
  - Suporte Directivo
- 2 - Análise da situação actual
  - Operações actuais
  - Fluxo de dados
  - Operações de rotina
  - Papel do computador
  - Relações com outros sistemas
- 3 - Vantagens e inconvenientes da situação actual
- 4 - Previsão de crescimento dos volumes (documentos,...)
- 5 - Definição do Novo Sistema
  - Especificações do sistema
  - Pessoal
  - Fluxo de Dados
  - Custos e Benefícios
  - Como resolver as necessidades
- 6 - Preparação da Proposta/caderno de encargos da fase 1 ou 1/2.

**Responsabilidades:**

Utilizador ou informática (com assistência activa da outra parte)

**Aprovação:**

Utilizador

**Como Termina?****Produtos Finais:**

- Relatório de estudo de necessidades ( ou proposta de caderno de encargos)
- (Introdução)
- Necessidades do sistema
- Descrição geral do novo sistema
- Resumo de Custos e mão de obra
- Benefícios e lucros
- (Conclusão)
- (Resumo das recomendações)

**As actividades do Gestor de Projectos**

O gestor de projectos deve assegurar:

- a obtenção de necessidades e documentos do cliente
- que os requisitos do sistema são definidos e estão dentro do âmbito dos serviços fornecidos.

O Gestor de Projectos deve assegurar que os seus pedidos de assistência ou notificação de problemas são rapidamente resolvidos pelo seu chefe.

## Fase 1 - Desenho do Sistema

### Tarefas

- Revisão do Relatório do estudo de Necessidades
- Planificação/ Estimativa
- Desenho geral do sistema
- Necessidades administrativas
- Plano de Conversão
- Desenho detalhado
- E/S , Programas, Testes, ...

### Responsabilidades

- Informática (com participação dos utilizadores).

### Aprovação

Utilizador.

### Produto Final

Especificações dos sistema (ou caderno de Sistema).

### As actividades do Gestor de Projectos

- Deve assegurar-se de que o utilizador e a chefia conhecem o formato e conteúdo dos produtos finais que receberão
- Definir critérios de complemento com o utilizador
- Verificar se o utilizador e o centro fornecem os recursos previstos
- Verificar se o utilizador e a manutenção estão preparados para ficar com o sistema no fim do projecto.
- Acordar com o utilizador o tempo e as condições de apoio pós-projecto
- Não dar garantia de "performances" impossíveis, nem aceitar outras condições que conduzam a um projecto sem fim.
- Ir obtendo aprovação parciais formais do utilizador e da chefia.

## Fase 2 - Desenvolvimento do Sistema

### Tarefas

- Desenvolvimento dos programas: Codificação/ testes / integração/ documentação
- Documentação: manual de operação; manual do utilizador; documentação de manutenção, ...
- Formação de pessoal

### Responsabilidades

- Informática

### Produto Final

- Sistema operacional
- Documentação do sistema
- o que estiver escrito no contrato (ex: formação de pessoal, etc)

## Fase 3 - Arranque

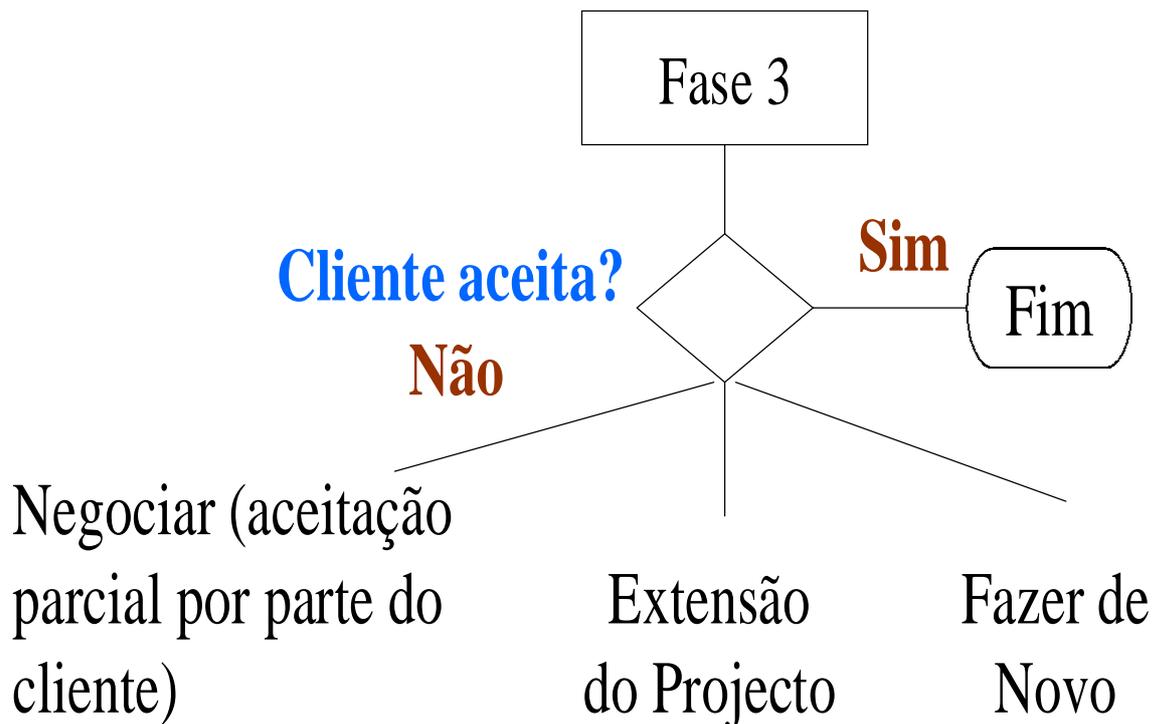
### Tarefas

- Criar Ficheiros
- Testes do Sistema ( Final ou de Aceitação)
- Formar operadores e utilizadores
- Verificar procedimentos de operação

### Responsabilidades

- Informática (... Instalação do sistema)
- e do utilizador (... Aceitação final e instalação )

### Como Termina?



# 1. Ciclo de vida moderno de um projecto

## Fases do Projecto

- Fase 1** - Investigação preliminar
- Fase 2** - Determinação dos requisitos
- Fase 3** - Desenvolvimento do sistema - protótipo
- Fase 4** - Desenho do sistema
- Fase 5** - Desenvolvimento do Software
- Fase 6** - Teste do Sistema
- Fase 7** - Implementação

## Fase 1 - Investigação preliminar

Começa por um pedido de assistência sobre um determinado sistema de Informação. Esta actividade envolve e alíneas possíveis:

- Pedido de Clarificação
- Estudo de Viabilidade
- Pedido de Aprovação

### 1.1 - Pedido de Clarificação

Muitos pedidos de empregados e utilizadores numa organização não são claramente relatados.

Antes de qualquer investigação, um pedido de projecto, deve ser examinado a fim de determinar precisamente o que a empresa quer.

Por vezes nem o próprio utilizador sabe qual é o problema nem o que está errado.

### 1.2 - Estudo de Viabilidade

#### *O sistema requerido será viável?*

O estudo de viabilidade deverá abranger 3 grandes áreas:

- **Viabilidade Técnica** - pode o projecto ser executado com o equipamento, software, tecnologia e pessoal existente?

Se é requerida nova tecnologia, qual é a probabilidade de que ela pode ser desenvolvida?

- **Viabilidade Económica** - os benefícios compensarão os custos do novo projecto?

Ou inversamente, serão os custos da nova criação do novo sistema tão grandes que o projecto deve ser executado?

- **Viabilidade Operacional** - será o sistema utilizado se for desenvolvido e implementado? Será que a resistência dos utilizadores prejudicará os possíveis benefícios?

### 1.3 - Pedido de Aprovação

Nem todos os projectos requeridos são desejáveis ou viáveis. Algumas organizações recebem tantos pedidos de projecto que só alguns poderão prosseguir. A gestão decide que projectos são mais importantes e lista-os.

Depois de um pedido de projecto ser aprovado, o seu custo, prioridade, duração e requisitos de pessoal são estimados e usados para determinar onde acrescentar este projecto a uma lista de Projectos.

Depois dos projectos precedentes estarem completos, o desenvolvimento da aplicação proposta pode ser lançada.

### Fase 2 - Determinação dos Requisitos

O analista deverá trabalhar afincadamente com empregados e Gestores, e estudar os processos empresariais para responder às seguintes questões-chave:

- O que está a ser feito?
- Como está a ser feito?
- Com que frequência ocorre?
- Qual o volume das transações ou decisões?
- O serviço está a ser bem executado?
- Existe algum problema?
- Se existe algum problema, qual a sua gravidade?
- Se existe algum problema, qual a causa principal?

- Para responder a estas questões o analista de sistema fala com inúmeras pessoas, utiliza questionários, estuda manuais e relatórios, observa actividades laborais e junta amostras de modelos e documentos para a compreensão total do processo.

- À medida que os detalhes vão sendo obtidos o analista estuda os requisitos que identificam o novo sistema, incluindo a informação que o sistema deve produzir e alguns traços operacionais como o controlo de processos, tempos de resposta e métodos de entrada e saída.

### Fase 3 - Desenvolvimento do Sistema- Protótipo

- Não é fácil definir á partida os traços do novo Sistema.
- Situações únicas em que não há informação nem experiência, são frequentemente seleccionadas para prototipagem.
- O protótipo fornecerá informação preliminar acerca da exequibilidade do conceito.
- O protótipo não é um produto final, e, naturalmente, sofrerá algumas alterações ao longo das várias iterações.
- O protótipo é um sistema de trabalho que deve ser facilmente alterável.
- A informação ganha através do seu uso é aplicada para modificar o desenho
- A versão modificada pode ser novamente usada como protótipo a fim de revelar nova informação válida.
- O processo é repetido tantas vezes quantas as necessárias para revelar os requisitos para o desenho.

## Fase 4 - Desenho do Sistema

- O desenho de um sistema de informação produz os detalhes que mostram como o sistema encontrará os requisitos identificados durante o sistema de análise.
- Há quem considere este passo como um “Desenho lógico” em contraste com o desenvolvimento do programa referenciado como “Desenho Físico”.
- O analista de sistemas começa por identificar mapas e outros Outputs que os sistema produzirá. Os dados especificados em cada são referenciados com grande pormenor, incluindo a sua exacta localização no papel, no écran, ou em outro meio. Os Desenhadores esboçam a forma ou o aspecto do output final.
- Os desenhadores também descrevem os dados de input, de cálculo ou de armazenamento. Os itens de dados e os procedimentos de cálculo são escritos em detalhe.
- Criam estruturas de ficheiros e seleccionam dispositivos de armazenamento como Discos, Bandas ou mesmo papel.
- Os procedimentos escritos dizem-nos como processar os dados e produzir o output.
- Os documentos contendo as especificações do desenho podem ser de variado tipo: cartas, tabelas, símbolos especiais.
- O desenho da informação detalhada passará, então, para a fase de programação
- Os desenhadores são responsáveis por entregar aos programadores especificações claras e completas sobre o que os programas devem fazer e, mostrarem-se disponíveis para responder a questões e clarificarem áreas imprecisas.

## Fase 5 - Desenvolvimento do Software

- O software pode ser adquirido comercialmente ao exterior ou pode ser desenvolvido na própria empresa através de programadores existentes.
- A decisão acerca de qual a hipótese a optar depende de 3 grandes factores:
  - Custo
  - Tempo
  - Pessoal
- Os programadores são responsáveis pela documentação dos programas, incluindo a explicação de “como” e “porque” um procedimento foi codificado de uma dada forma.
- A documentação é essencial para testar o programa e fazer a manutenção da aplicação instalada.

## Fase 6- Teste do Sistema

- Nesta fase, o sistema é usado experimentalmente para verificar que o software não falha , isto é, que o sistema funciona de acordo com as especificações e segundo o que os utilizadores esperavam.
- Podem ser, inclusivé, convidados alguns utilizadores para operarem o sistema de forma a que os analistas possam ver se os caminhos usados são caminhos esperados.
- É preferível encontrar estas possíveis surpresas antes da implementação do sistema.
- Em algumas empresas os testes são executados por pessoas diferentes das que escreveram os programas.

## Fase 7- Implementação

- Um sistema diz-se implementado quando o pessoal foi verificado e treinado, o novo equipamento está instalado, a nova aplicação está pronta e testada e os ficheiros estão constituídos.
- Por vezes e durante algum tempo correm em paralelo os dois sistemas ( Novo/antigo).
- Uma vez instalada, a aplicação poderá ser utilizada durante largos anos, contudo, quer a organização quer os utilizadores poderão querer modificá-la.
- A partir deste momento a manutenção assume particular importância.

# Elaboração de protótipos

**Protótipo dum sistema informático:** é uma versão provisória que apresenta as funcionalidades essenciais do sistema pretendido.

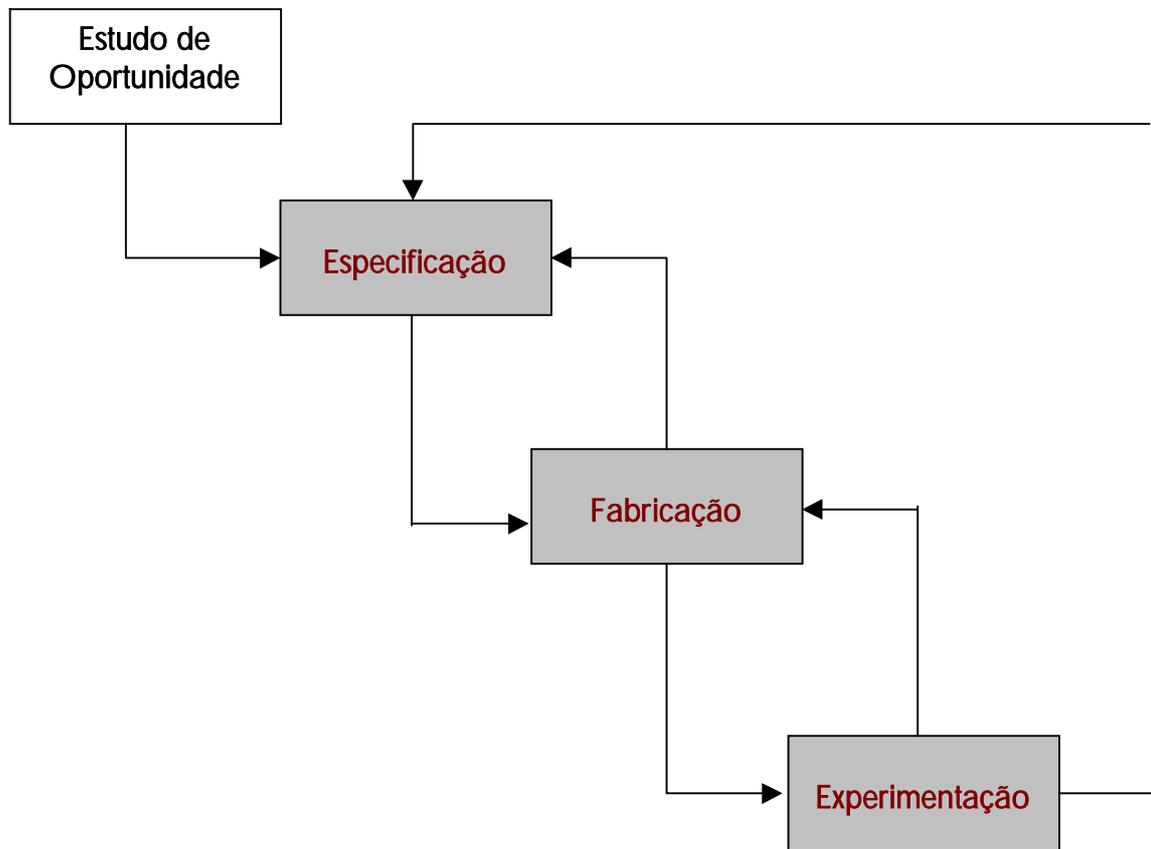
## Características dum protótipo:

1. um protótipo é essencialmente incompleto. É construído com a intenção de ser modificado e melhorado.
2. Um protótipo é um sistema utilizável que é avaliado em ambiente operacional.
3. O objectivo do protótipo é melhorar a compreensão do sistema a servir de veículo para a experimentação.
4. O protótipo deve ser desenvolvido o mais rápido e economicamente possível.

## Pressupostas da metodologia

1. As necessidades a satisfazer por um sistema informático não podem ser exaustivamente especificadas antes do desenvolvimento do sistema.
2. Os utilizadores tem dificuldade em compreender e validar os documentos habitualmente utilizados pelos informáticos para descreverem os sistemas a desenvolver (fluxogramas, etc).
3. A concepção dum sistema deve ser iterativa. Desenvolve-se rapidamente uma primeira versão dum utilização experimental. Dessa avaliação resultam alterações ao sistema que dão origem a uma nova versão. O processo repete-se até que se obtenha uma versão satisfatória.
4. O processo de aprendizagem e experimentação utilizado na elaboração do protótipo evita alterações que, de outro modo, seriam solicitadas mais tarde.
5. O utilizador deve ter uma participação activa no processo:
  - análise das necessidades
  - concepção das soluções
  - avaliação do protótipoe eventualmente,
  - fabricação do protótipo

## Fases do processo



## Especificação

### Objectivos:

Definir as características de um sistema informático que se considere “a priori” capaz de satisfazer as necessidades detectadas.

### Acções:

\* Especificação Externa

- enquadramento na organização (donde provém os inputs, a quem se destinam os outputs)
- Principais funções: input de dados; armazenamento; processamento; outputs.
- Organização lógica dos dados

\* Especificação Interna.

- estruturas de organização interna dos dados
- organização das funções (menus e transacções)
- especificação das transacções (input, output, tratamento)

**Resultado:**

Colecção de drafts (menus específicos) de especificação do sistema.

## Fabricação do Protótipo

**Objectivos:**

“Fabricar” um protótipo que concretize as especificações.

**Acções:**

- criação de menus e seu encadeamento
- criação das estruturas de dados
- implementação dos tratamentos (transacções)

**Resultados:**

Versão operacional do protótipo

## Experimentação

**Objectivos:**

Verificar em que medida o protótipo satisfaz as necessidades. Detectar desvios em relação às expectativas ou melhoramentos a introduzir.

**Acções:**

Experimentação do protótipo em ambiente tão próximo do real quanto possível.

**Resultados:**

Alteração às especificações do protótipo ou aprovação do protótipo.

# Caderno de Encargos

É um documento que contém as especificações técnicas de um projecto:

1. Descrição dos objectivos e tarefas;
2. Condições financeiras e prazos;
3. Penalizações por (in)cumprimento defeituoso;
4. Especificações técnicas e volumes de aquisição;
5. Cláusulas de contratação;
6. Plano de ensaio e aceitação;
7. Possibilidade de alternativa e evolução;
8. Especificações de instalação e de formação.

# Ferramentas de Análise

As principais ferramentas de análise são:

- Diagrama de Fluxo de Dados (DFD)
- Diagrama de Entidade-Associações (DEA)
- Ciclo de Vida das Entidades (CVE)

As ferramentas complementares são:

- Dicionário de dados
- Especificação de Processos
  - Árvores de Decisão
  - Tabelas de Decisão
  - Português Estruturado

**Diagrama de Fluxo de dados** - representação de fluxos de informação através de um sistema, e entre este e o mundo exterior.

Podem ser utilizados para:

- representar os sistemas físicos existentes
- colher destes os fluxos de informação que suportam os requisitos dos utilizadores
- construir uma representação lógica do sistema pretendido.

**Diagrama de Entidade-Associações** - um modelo de entidade e a estrutura utilizada para o desenvolvimento do sistema que suportará os requisitos em termos de processamento. Os modelos de entidade são criados em pontos diferentes do desenvolvimento do sistema; desde as tentativas experimentais do início até ao detalhado diagrama de entidades e associações.

**Ciclo de vida das Entidades** - os diagramas representativos do ciclo de vida das entidades fornecem uma terceira visão dos dados de forma a evidenciar os efeitos do factor "Tempo" no sistema.

## Relação entre as 3 representações do sistema:

Os DFD's representam o dia-a-dia do funcionamento do sistema e evidenciam a forma como os processos, transacções e os dados mantidos nos sistemas se conjugam e também como o sistema se inter-relaciona com o mundo exterior.

O modelo de entidades evidencia as associações e a maneira como os dados podem ser acedidos. Cada entidade do modelo deverá ser identificada como um arquivo de dados ou parte de um arquivo, no diagrama de fluxo de dados.

O ciclo de vida das entidades evidencia as mudanças de uma entidade ao longo do tempo, quais os eventos que o sistema terá de manipular e em que ordem estes ocorrem. Cada evento no ciclo de vida de uma entidade pode ser relacionada com o fluxo no diagrama de fluxos de dados. Em conjunto, as três fornecem uma representação complementar da forma como o sistema usa e organiza os seus próprios dados, constituindo uma representação funcional completa.

# Diagrama de Fluxo de Dados

## Definição:

**Diagrama de Fluxo de Dados (DFD)** – é uma representação em rede de um sistema. O sistema pode ser automatizado, manual ou misto. O Diagrama de Fluxo de Dados retrata o sistema em termos de suas componentes, com todos os interfaces entre as partes indicadas

*DeMarco, 1979*

## O Diagrama de Fluxo de Dados:

- Representa o fluxo de dados num sistema de informação, assim como as sucessivas transformações que estes sofrem;
- Mostra, para o sistema em estudo, as funções e como são armazenados e transferidos os dados;
- Na fase de análise de processos e principalmente com o recurso a diagramas de fluxo de dados (DFD) são identificados os processos em que se divide a funcionalidade de cada módulo, bem como a interacção desses processos em termos de partilha e fluxos de dados.
- É uma ferramenta gráfica que transcreve, de forma não técnica, a lógica do procedimento do sistema em estudo.

## Características dos Diagramas de Fluxos de Dados (DFD):

- Mostra a essência lógica do SI em estudo;
- Não é um diagrama técnico – permite ser facilmente entendido por pessoas que não sabem nada de computadores;
- Para ser construído tem-se que definir a área do sistema em estudo – tudo que estiver fora desta área não pertence ao SI a desenvolver;
- Informação temporal não é representada;
- Fluxo de controlo não é representado

## Objectivos dos Diagramas de Fluxos de Dados (DFD):

A técnica dos DFDs visa criar uma representação do funcionamento de um sistema, promovendo:

- \* A clareza da descrição
- \* A economia da representação
- \* A compreensão pelos utilizadores
- \* A comunicação:
  - com os utilizadores
  - com a equipa de desenvolvimento
- \* A documentação do trabalho desenvolvido

O diagrama de fluxo de dados é uma das ferramentas de modelagem do sistema mais utilizada, devido à facilidade da linguagem e às poucas alternativas da simbologia. Estes factores tornam-no numa ferramenta fácil de utilizar, de elaborar, de ser lida e compreendida.

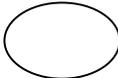
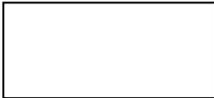
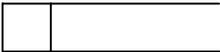
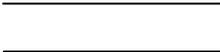
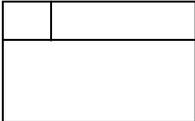
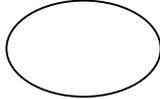
Assim, a comunicação entre o analista e o utilizador estabelece-se numa forma livre e valiosa, dando oportunidades a este último de intervir numa forma positiva no sistema, mesmo que não possua nenhum conhecimento desta ferramenta. Todos estes factores favorecem um desenvolvimento próspero para o sistema.

Existem duas principais notações de construção de Diagrama de Fluxo de Dados (DFD):

- Gane&Sarso, 1979
- DeMarco, 1979

O DFD abrange quatro elementos gráficos: **os processos, o fluxo de dados ou informação, os arquivos e a entidade de origem/destino.**

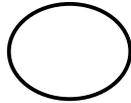
**Símbolos (objectos) a usar na construção de um DFD:**

	Chris Gane	Tom DeMarco
<b>Entidade</b> Origem/Destino Dados		
<b>Fluxos de dados</b>		
<b>Arquivos de dados</b>		
<b>Processos</b>		

Seguidamente são apresentadas as convenções simbólicas, bem como a descrição de cada um deles, tendo sido descritas por **Tom DeMarco**.

## PROCESSO

**Simbologia utilizada**

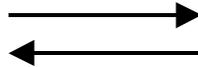


O processo é representado graficamente por um círculo

*Descrição* – O processo mostra uma parte do sistema que transforma entradas em saídas – isto é, mostra como uma ou mais entradas são convertidas em saídas. O nome do processo deve ser único e é constituído por um verbo e um substantivo e deve figurar dentro do círculo, descrevendo o que este faz. A adequada atribuição dos nomes é muito importante, uma vez que tem que reflectir, de uma forma clara, a actividade que o processo engloba. A identificação normalmente é um número, em ordem crescente, colocado no meio da parte superior; esse número, na verdade, não possui outra função a não ser identificar o processo no momento em que ele ocorre. Além disso, um mesmo processo pode ser desdobrado em vários (novos números serão incluídos). No entanto, os números do processo inicial não podem ser modificados, já que são usados como referencial para o fluxo de dados na decomposição dos níveis superiores.

## FLUXO DE DADOS

**Simbologia utilizada**



O fluxo é representado por setas

*Descrição* – Um fluxo é graficamente representada por uma seta que entra ou sai de um processo e é utilizado para mostrar o movimento de pacotes de informação de um ponto a outro do sistema. Deste modo, o fluxo representa dados em movimento. Cada fluxo de dados tem de ter um nome apropriado e o mais preciso possível.

## ARQUIVOS

**Simbologia utilizada**



O arquivo é representado graficamente por duas linhas paralelas

*Descrição* – Cada arquivo possui um nome exclusivo que se encontra no plural. O arquivo contém um conjunto de informações que pode ser acedido e/ou actualizado por um processo.

As operações que se podem fazer com um arquivo estão representadas em baixo:



➤ Ler o arquivo; isto é, extrair informação



➤ Gravar o arquivo; isto é, colocar--lhe informação

➤ Actualizar o arquivo; isto é, adicionar--lhe informação

A origem e o destino mostram, respectivamente, de onde o dado requerido pelo sistema vem e para onde o dado produzido pelo sistema vai. Uma origem é um provedor de fluxos de dados para o sistema, e o destino é um receptor de fluxos de dados do sistema.

## ENTIDADE DE ORIGEM/DESTINO

**Simbologia utilizada**



A entidade de origem/destino é representada por um rectângulo

*Descrição* – São frequentemente categorias lógicas de coisas ou pessoas que representam a origem ou o destino para as transações. A identificação e definição das entidades externas definirá as fronteiras ou limites do sistema.

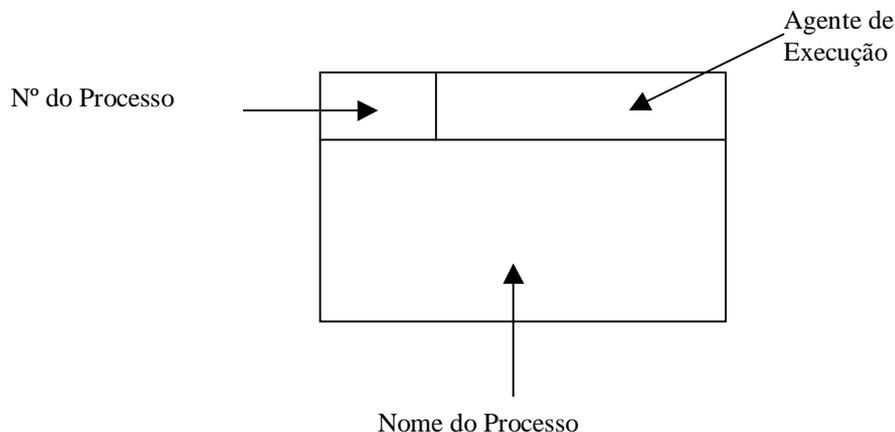
A forma como estes quatro elementos podem/ou não estar ligados tem que respeitar determinadas regras que são aqui esquematizados:

Relacionamento	Entidades	Processos	Arquivos
Entidades	Não	Sim	Não
Processos	Sim	Sim	Sim
Arquivos	Não	Sim	Não

As convenções simbólicas segundo [Chris Gane](#).

## O PROCESSO

Actividades de transformação de informação de fluxos de dados de entrada em fluxos de dados de saída



Nome para o Processo:

Verbo + Substantivo

Exemplo:



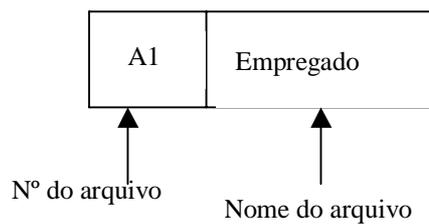
**Regras:**

- tem que ter um nome
- tem de ser significativo
- deve ser único e representativo do objecto

**O ARQUIVO**

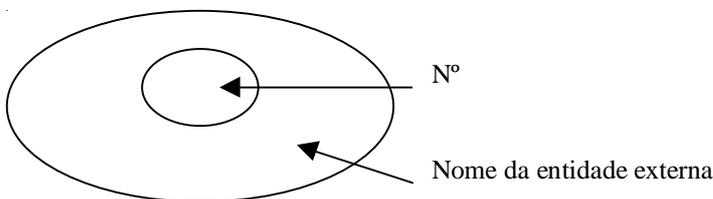
Repositório de Informação ( permanente ou temporário) cuja estrutura é conhecida. Os repositórios de dados podem dividir-se em dois grandes grupos:

Permanentes - D  
Temporários - T



**A ENTIDADE EXTERNA**

Entidade fora do âmbito do estudo receptora e/ou produtora de informação de/para o sistema em estudo



Pode ser um organismo, um indivíduo processador, uma área funcional ou um serviço.

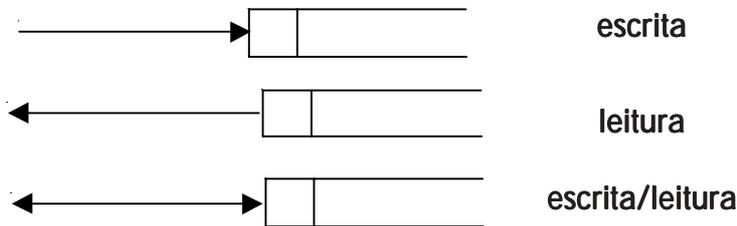
**A ENTIDADE EXTERNA**

Instrumento de particionamento inicial do âmbito do estudo.

## FLUXOS DE DADOS

Contentor de informação em movimento

*Fluxos de dados de e para arquivos:*



Processos:

Quando o fluxo sai de um processo deve ter um nome diferente do fluxo que entra no processo, uma vez que entrou no processo, em princípio sofreu alterações.

### Passos a seguir para o desenho de um DFD

- 1- Elaborar uma lista de documentos principais usados e produzidos pelo sistema.  
Deve-se representar apenas os primeiros 10 a 12 documentos. Se no sistema houver mais do que 12 documentos deve fazer-se a tiragem dos principais.
- 2 - Produzir um diagrama de fluxo de documentos evidenciando a origem e o destino de cada um desses principais documentos.
- 3 - Acordar com o utilizador qual a fronteira do sistema em análise e elaborar um diagrama onde estejam representadas várias entidades externas fora da fronteira.
- 4 - Diagrama de contexto
- 5 - DFD

Na execução do DFD não são considerados detalhes como o hardware a ser utilizado, os aplicativos, a forma como os dados são arquivados, ou qualquer outra implicação física.

Cada informação só é armazenada uma única vez, não tendo o analista de se preocupar com funções de manutenção básica como seja abrir/fechar arquivos. A sua única preocupação é descrever o que acontece sem se preocupar em como acontece.

Os diagramas de fluxo de dados podem ser desenvolvidos através de uma análise *Top-down* (partimos do geral para o particular, originando um detalhe cada vez maior na decomposição dos processos) ou *Bottom-Up* (partimos dos processos- módulos para o geral).

De acordo com a teoria da análise estruturada, o primeiro diagrama do conjunto de DFDs corrente é o diagrama de contexto. Ele contém uma única bolha de processo que é nomeada para o projecto como um todo.

O **diagrama de contexto** define todo o projecto estruturado quando mostra quais são os dados físicos que o sistema recebe e envia. Desta forma ele mostra de maneira não-ambígua as fronteiras do projecto estruturado. Quando desenhamos o diagrama de contexto ainda não conhecemos os detalhes, estes apenas surgirão quando estivermos a desenhar os diagramas subsequentes, todavia este é valioso para iniciar o trabalho na visão geral dos processos do sistema.

Quando queremos aumentar o grau de detalhe é necessário ter em atenção dois aspectos importantes que são eles:

**Levelling** – Cada processo do DFD pode ser decomposto num outro mais detalhado.

**Balancing** – Quando se constrói um DFD todos os dados que fluem para e de um determinado processo têm que aparecer em qualquer DFD que seja uma decomposição deste processo.

Os diagramas de nível 0 podem ser esboços ainda sem muito detalhe e precisão, dependendo sobretudo da complexidade do sistema. Contudo quando o detalhe é definido, ele deve ser reflectido de forma que fique coerente e minimize a complexidade do diagrama.

Número de níveis que deve ter um DFD

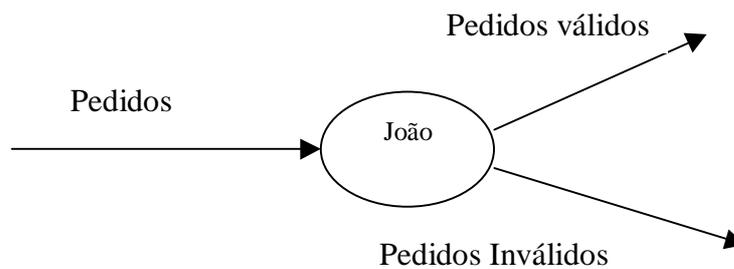
- Num sistema simples encontram-se provavelmente dois a três níveis
- Num sistema médio três a seis níveis
- Um sistema grande apresenta 5 a 8 níveis.

## Directrizes para a elaboração do DFD

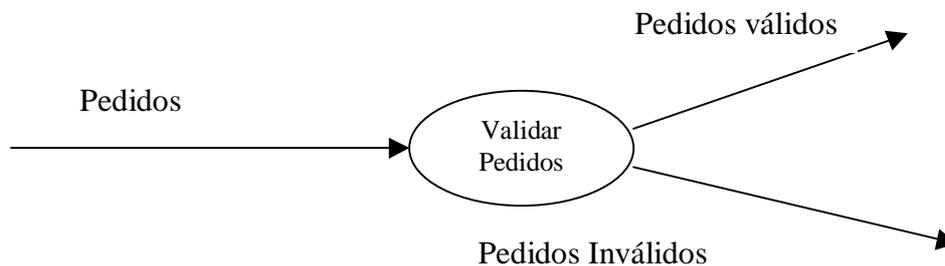
1. Escolher nomes significativos para os processos, fluxos, arquivos e entidades externas.
2. Numerar os processos
3. Refazer o DFD tantas vezes quantas forem necessárias até obter uma boa estética.
4. Evitar DFDs complexos demais.
5. Certificar-se de que o DFD seja internamente consistente.

### 1. Escrever nomes significativos

- Muitas vezes acontece que determinada pessoa executa um Processo e cai-se no erro de dar o nome dessa mesma pessoa, ao processo.



Devemos evitar o uso de nomes de pessoas, grupos e de funções políticas, mas rotular os processos de modo a identificar as funções que o sistema executa.



## 2. Numerar os processos

- Método prático de referenciar os processos
- O esquema de numeração, pode não implicar uma determinada sequência de execução, porque o DFD é uma rede de processos assíncronos que se intercomunicam, dando uma representação do modo como funciona a maioria dos sistemas.

## 3. Refazer o DFD tantas vezes quantas necessárias:

Um projecto de análise de sistemas, para que esteja tecnicamente correcto, aceitável pelo utilizador e tão bem desenhado que seja perceptível por todos, terá de ser feito, refeito e novamente refeito, por dez ou mais vezes.

## 4. Evitar DFD's Complexos

- Deve-se modelar as funções que o sistema executa e as interacções entre elas, de modo a que ao ser lido seja compreendido, não só pelo analista que o elaborou o modelo, mas também pelos utilizadores.
- Deve-se modelar em níveis, de forma a que cada nível ofereça sucessivamente mais detalhes.

## 5. Consistência do DFD

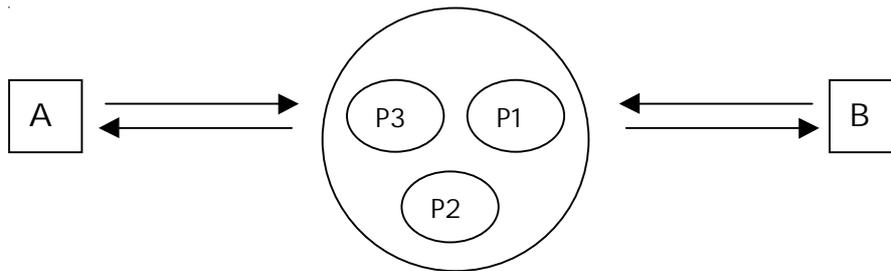
- Além do relacionamento com outros modelos do sistema, existem algumas directrizes consideradas principais:
- Evitar os processos que tenham entradas mas que não tenham saída.
- Evitar processos que tenham saídas, mas que não tenham entradas.

**Resumo:**

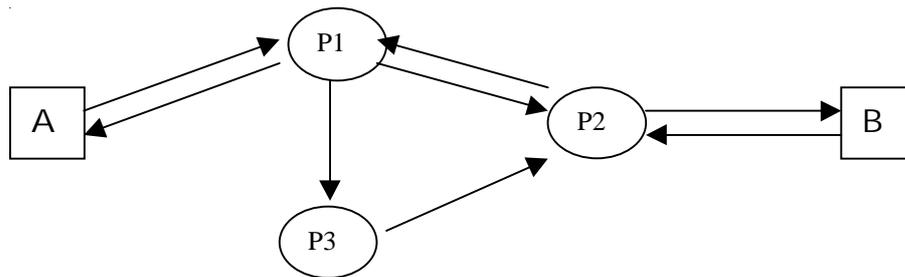
DIAGRAMA DE FLUXO DE DADOS – É uma primeira representação do sistema. Pode ser feita de um forma muito simples, mostrando somente os principais inputs e output's do sistema em estudo – *Diagrama de contexto*.

- O *Diagrama de Contexto* é então “explorado” no DFD o qual mostra mais detalhes sobre o sistema.
- O que se passa nas origens/destinos de dados não é representado no DFD.
- Cada processo do DFD pode por sua vez ser decomposto num outro DFD mais detalhado. (Levelling)
- Quando se constrói um DFD todos os dados que fluem para e de um determinado processo têm que aparecer em qualquer DFD que seja uma decomposição desse processo (Balancing).
- Quando os processos estão numa forma que não podem ser mais decompostos, diz-se que se encontram sob a forma de “primitiva funcional”.

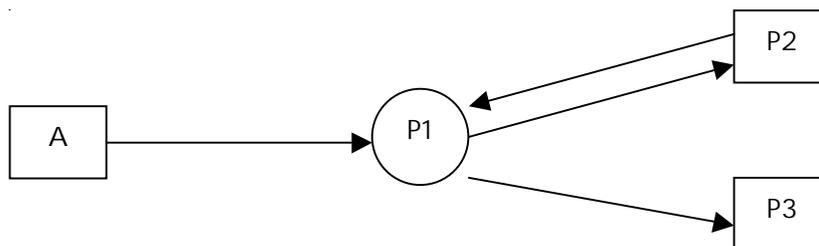
Diagrama de Contexto



DFD nível 1



DFD P1



## Regras de Nivelamento

As regras que gerem o processo de nivelamento dos Diagramas de Fluxos de dados são essencialmente indicativas, mas que se devem tentar cumprir.

Pode-se enumerar as seguintes como sendo as mais importantes e de maior obrigatoriedade:

1. Referenciar os processos com uma numeração hierárquica, para que qualquer processo seja rapidamente identificado na sequência descendente de diagramas que originou;
2. Manter o DFD equilibrado ou seja, manter as mesmas entradas e saídas a quando de cada decomposição;
3. Caso seja possível não representar fluxos cruzados, alterando as posições, se for caso disso, dos elementos ligados no diagrama.

### Diagrama de Contexto

- Identificar as origens/destinos de dados envolvidos;
- Identificar os principais dados – Entradas e Saídas;
- Um só processo representa o sistema.

### DFD nível 1

- Identificar as principais funções;
- Identificar os principais arquivos de dados;
- Identificar os fluxos de dados.

### DFD nível 2

- Subdivisão dos processos (“O que fazer”);
- Subdivisão dos fluxos de dados (dados de uso localizado);
- Arquivos de dados de utilização local;
- Aparecimento das primitivas funcionais.

# Diagrama de Entidades e Associações (DEA)

## CONCEITOS BÁSICOS

### ENTIDADE

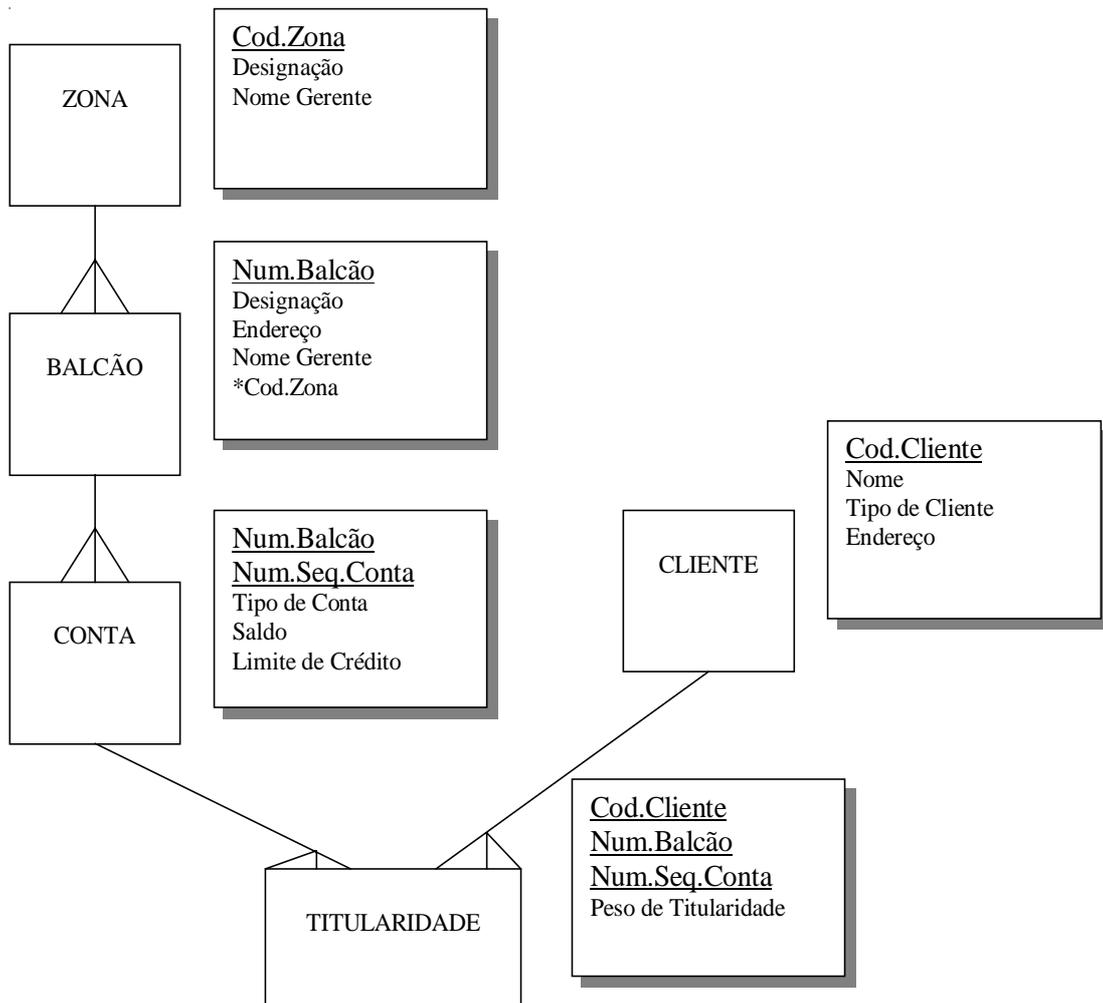
“Objecto ou conceito relevante na organização, acerca do qual se armazena informação”

*Exemplos:* Produto; Cliente; Armazém; Fornecedor

### MODELO DE ENTIDADES

Diagrama que evidencia as diversas Entidades patentes na organização e as respectivas Associações relevantes no sistema.

## EXEMPLO DE MODELO DE DADOS



## DEA – FASEAMENTO DO PROCESSO

### 1 – PREPARAÇÃO

- Investigar os principais processos decorrentes dos requisitos de funcionamento do sistema.
- Investigar os tipos de dados de suporte ao sistema, bem como os respectivos volumes
- Esclarecer o significado dos nomes de dados e a sua utilização

### 2 – CONSTRUÇÃO DA ESTRUTURA

- Seleccionar um conjunto inicial de Entidades-tipo, com base no conhecimento do ambiente de dados do sistema.
- Identificar e registar as correspondentes Associações-tipo directas, recorrendo a uma Grelha de Entidades/Associações
- Construir o respectivo Diagrama de Entidades/Associações, previamente tipificadas
- Validar e racionalizar a estrutura.

### *Passos para a elaboração de um DEA*

- 1º - Selecção das Entidades
- 2º - Construção da grelha de entidades/associações
- 3º - Identificação das Associações (verificar a existência de uma associação directa com qualquer uma das restantes sem intervenção de uma terceira entidade)
- 4º - Tipificação das associações (N:1/1:N/N:N)
- 5º - Montagem da estrutura (DEA)
- 6º - Validação da estrutura
- 7º - Necessidade de existência de Chaves

## Associações

Existem três tipos de associações:

- associações unárias
- associações binárias
- associações complexas

### **Grau das associações e forma de participação das entidades nas associações**

Os tipos de associações distinguem-se entre si pelo seu grau. Este conceito é independente do tipo de associação e caracteriza-se da seguinte forma:

- Associação 1:1 (um pra um)
- Associação 1: N ( um para muitos)
- Associação N:N (muitos para muitos)

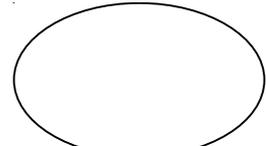
### **Obrigatoriedade ou opcionalidade da participações entidades**

- 1- obrigatória em ambas as entidades;
- 2- obrigatória em uma entidade e não obrigatória na outra;
- 3 - não obrigatória em nenhuma das duas entidades.

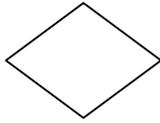
Notações:



Representa as entidades



Atributos



Representa os relacionamentos

Os tipos de relacionamentos:

Quanto ao grau:



um para um



um para muitos



Muitos para muitos

Quanto à obrigatoriedade:



Opcional

ou



obrigatória



Pelo menos uma



Exemplos:



## DEA – EXEMPLO

### AMBIENTE DE DADOS DO SISTEMA

- 1 – Existem aproximadamente 2500 clientes
- 2 - Os clientes encomendam produtos através de notas de encomendas
- 3 – Uma nota de encomenda de um cliente poderá envolver vários produtos
- 4 – Cada cliente está englobado numa das 60 áreas existentes
- 5 – Cada cliente é servido por um dos 20 armazéns existentes, de acordo com a área em que está inserido
- 6 – Uma área é servida por um único armazém
- 7 – Um armazém poderá guardar vários produtos, cada um dos quais poderá existir em vários armazéns.

### REQUISITOS DE ACESSO A DADOS

- 1 – Produzir uma lista de todas as notas de encomenda de um dado cliente, evidenciando não só os produtos encomendados como também o respectivo armazém de abastecimento.
- 2 – Identificar todos os clientes que tenham encomendado um dado produto
- 3 – Conhecer os níveis de “stockagem” de um dado produto em qualquer um dos armazéns
- 4 – Listar as existências de um dado armazém
- 5 – Listar os nomes e moradas dos clientes de um dado tipo
- 6 – Identificar os clientes com linhas de encomendas por satisfazer

### SELECÇÃO DAS ENTIDADES

Identificar no ambiente do sistema quais os objectos que necessitam de ser univocamente identificados.

No caso presente:

- Cliente
- Produto
- Área
- Armazém
- Nota de Encomenda

## CONSTRUÇÃO DA GRELHA DE ENTIDADES/ASSOCIAÇÕES

Representar cada entidade ao longo de eixos de uma matriz

	CLI	PROD	AREA	ARM	N. ENC
CLIENTE					
PRODUTO					
AREA					
ARMAZEM					
N.ENCOMENDA					

## IDENTIFICAÇÃO DAS ASSOCIAÇÕES

Para cada entidade, verificar se existe uma associação DIRECTA com qualquer uma das restantes (sem a intervenção de um terceira entidade).

	CLI	PROD	AREA	ARM	N. ENC
CLIENTE		?	?	?	?
PRODUTO			?	?	?
AREA				?	?
ARMAZEM					?
N.ENCOMENDA					

## ASSOCIAÇÃO DIRECTA IDENTIFICADAS

- CLIENTE e ÁREA
- CLIENTE e NOTA DE ENCOMENDA
- PRODUTO e ARMAZÉM
- PRODUTO e NOTA DE ENCOMENDA
- ÁREA e ARMAZÉM

	CLI	PROD	AREA	ARM	N. ENC
CLIENTE			<b>X</b>		<b>X</b>
PRODUTO				<b>X</b>	<b>X</b>
AREA				<b>X</b>	
ARMAZEM					
N.ENCOMENDA					

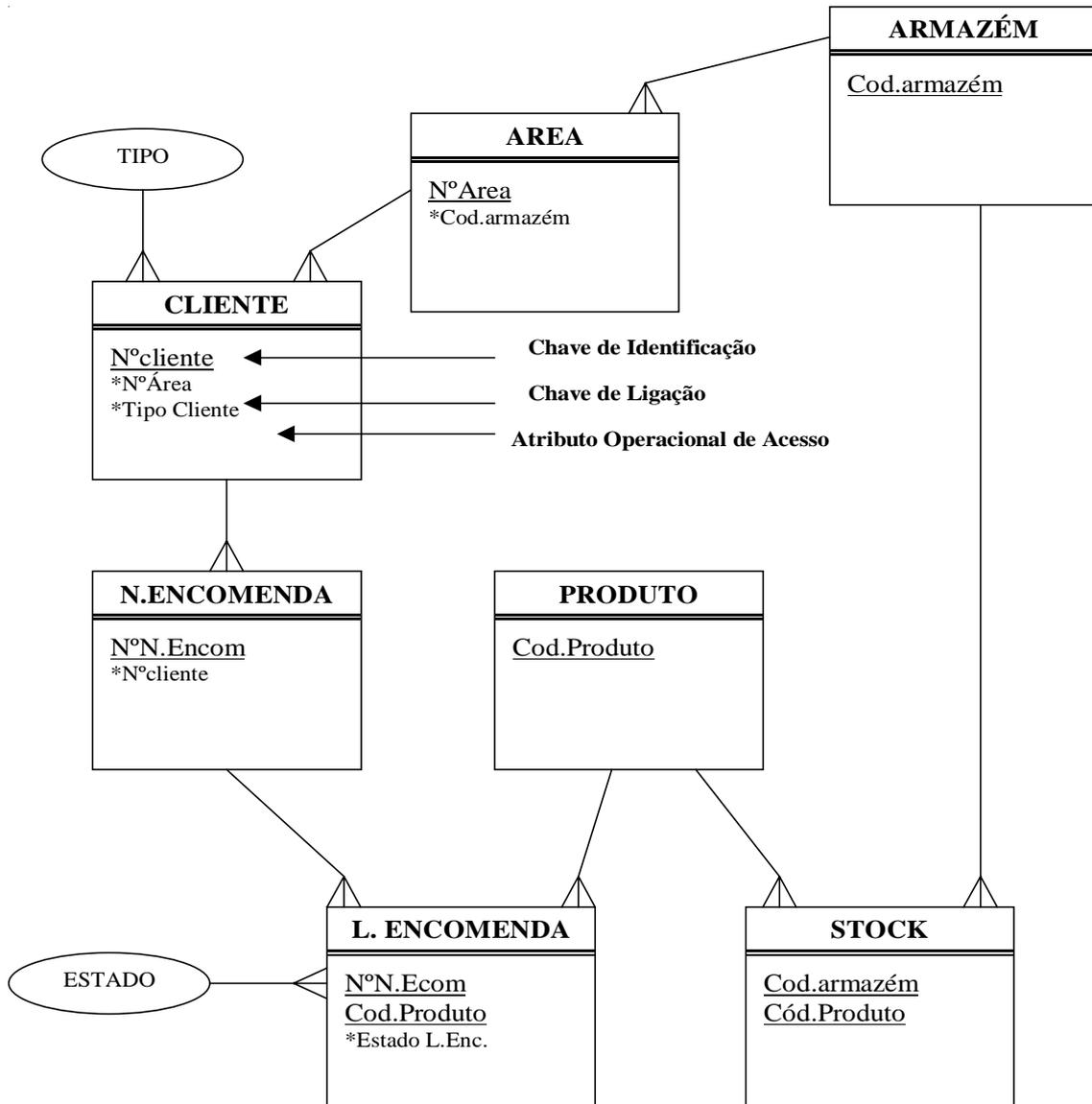
- CLIENTE e ÁREA.....: **N:1**
- CLIENTE e NOTA DE ENCOMENDA.....: **1:N**
- PRODUTO e ARMAZÉM.....: **M:N**
- PRODUTO e NOTA DE ENCOMENDA.....: **M:N**
- ÁREA e ARMAZÉM.....: **N:1**

**M:N** IDENTIFICAÇÃO DOS LINKS

PRODUTO e ARMAZÉM.....STOCK

PRODUTO e NOTA DE ENCOMENDA.....LINHA DE ENCOMENDA

# ESTRUTURA DO DEA



Nota:  
 L.Encomenda – Linha de Encomenda  
**Tipo** – Listar os nomes e moradas dos clientes de um dado tipo (requisitos de acesso)  
**Estado** – Identificar os clientes com linhas de encomendas por satisfazer (requisitos de acesso)

# Diagrama de Ciclo de Vida das Entidades (DCVE)

## DEFINIÇÃO:

É uma técnica de suporte à análise e concepção estruturada de sistemas que nos permite explicitar relativamente às entidades:

*O QUE ACONTECE*

*QUANDO ACONTECE*

*COMO ACONTECE*

Permite-nos a terceira visão do sistema, propondo-nos

**Integrar e Validar**

E dá-nos a noção de tempo em diferentes passos do método

## ***Diagrama de Ciclo de Vida da Entidade:***

- \* Mostra os eventos (acontecimentos) que cada entidade sofre ao longo da sua vida no sistema em estudo.
- \* Define-se Ciclo de Vida na modelação de eventos, como a série de estados que um objecto passa e que são de interesse para o sistema em estudo.

## ***Características:***

- Não é um diagrama técnico
- É um diagrama em árvore, em que o topo representa a entidade e as folhas os eventos que a entidade sofre ao longo da sua vida no sistema.

## ***Evento (acontecimento):***

Qualquer coisa instantânea à qual o sistema tem que responder.

- Os eventos têm um nome (verbo ou substantivo)  
Exemplo: Requisitar, Cancelamento, Criar, Anular, etc.
- Os eventos actuam sobre um objecto do mundo real, alterando o seu estado.
- Os eventos transportam dados.
- Os eventos são representados no DCVE segundo uma ordem cronológica.

### ***Regras de Construção:***

- Os eventos são representados cronologicamente no DCVE recorrendo às estruturas de sequência, selecção e iteração(repetição).
- Um evento só pode ser representado uma vez num DCVE, mas pode aparecer em mais do que um DCVE- eventos comuns.

## **Construção de C.V.E.**

### **Requisitos**

*Quais os requisitos para a construção de um CVE?*

- 1 – Conhecer as entidades
- 2 – Identificar os eventos e os efeitos destes em cada entidade.

### **Notação**

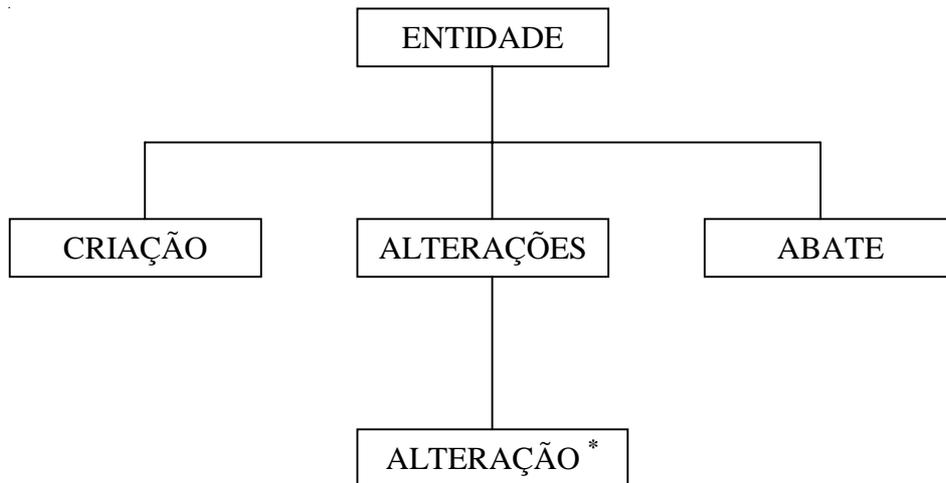
A notação proposta para a construção de um DCVE é do tipo “Jackson” e:

- 1- Mostra o que acontece cronologicamente a cada entidade
- 2- Não Mostra a estrutura dos processos
- 3- Mostra os eventos reconhecidos como necessários ao sistema
- 4- Não evidencia a ocorrência de operações individuais.

### **Convenções para o desenho do CVE**

Cada evento é representado por uma caixa

- A sequência é dada pela leitura
- A selecção é representada por ○
- A iteração é representada por \*

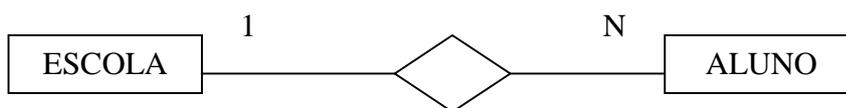
**Exemplo:****LEGENDA**

-Um sistema é alterado algures quando:

Se verifica a criação de uma ocorrência para a entidade x

Se verifica o abate de uma ocorrência para a entidade x

Ocorrem alterações que provocam a modificação do conjunto ou sub-conjunto da informação que está associada a ocorrência de uma entidade.

**Exemplo:****EVENTO**

Inscrição de um novo aluno

Desistência de um aluno

O aluno mudou de endereço

O aluno mudou de turma

**EFEITO**

Criação

Abate

Alteração

Alteração

## EVENTOS OU ACONTECIMENTO

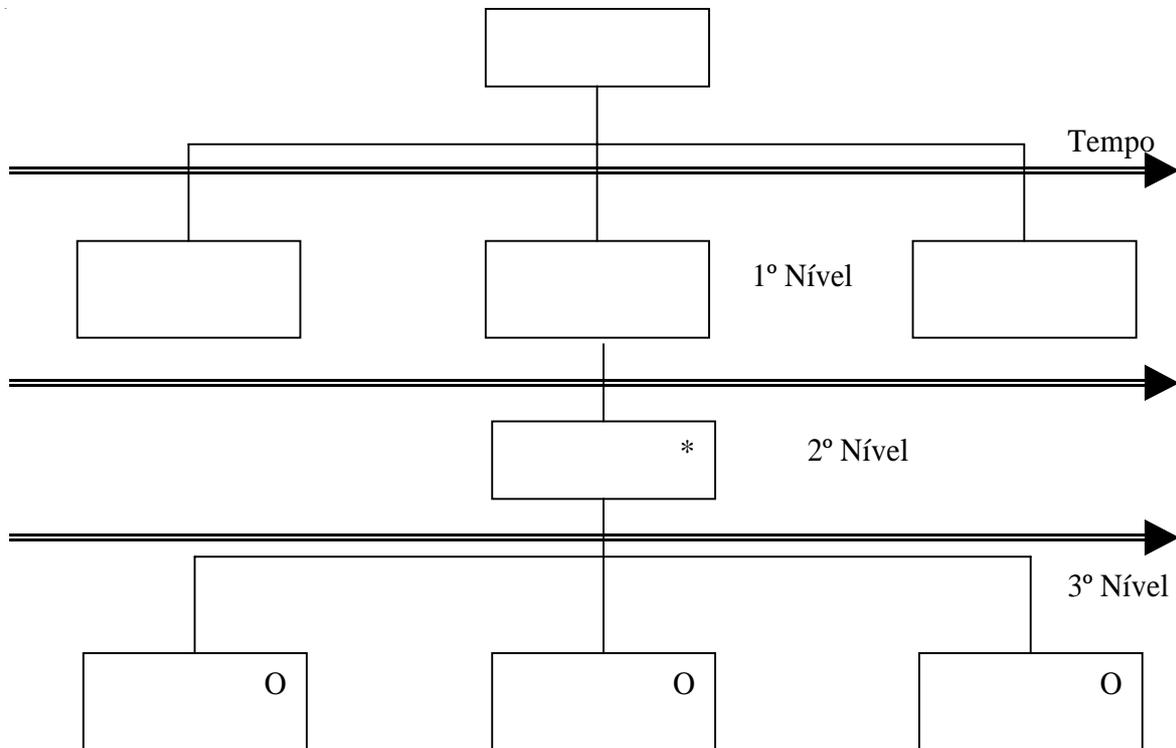
Quais os eventos que nos interessa (re)conhecer para construir o CVE?

Todos os que provoquem modificações no Modelo Informático do sistema, ou seja, os que implicam: Criações, alterações, abates.

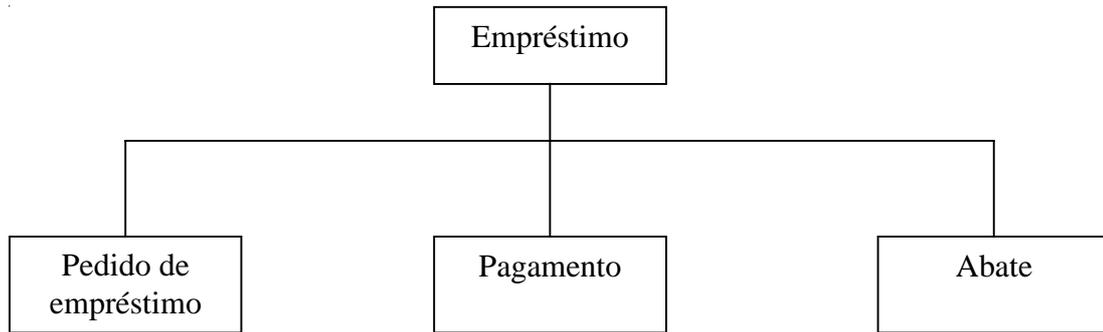
### Figuras de Representação

Sequências, selecções e iterações não podem estar ao mesmo nível (no mesmo ramo da árvores).

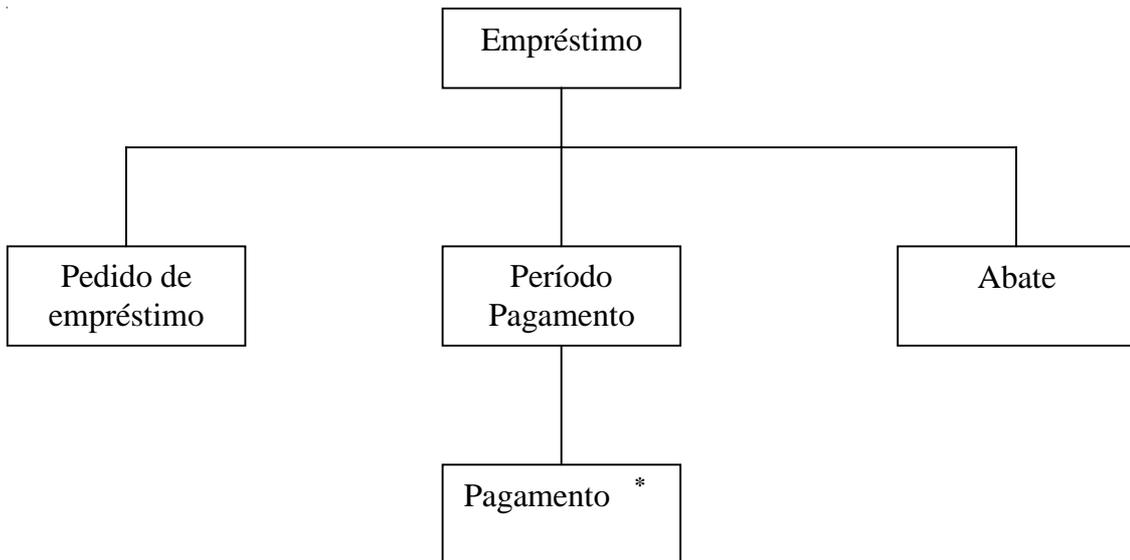
Exemplificando:



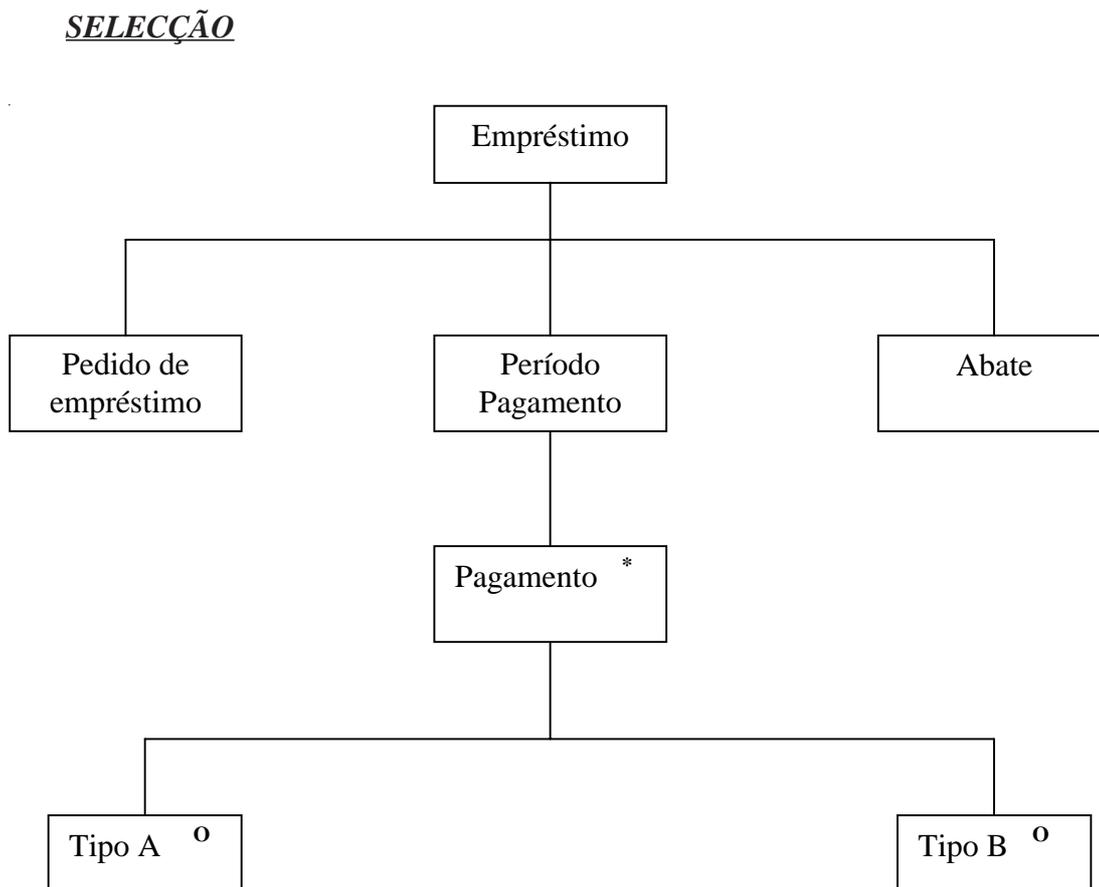
**SEQUÊNCIA**



**ITERAÇÃO**



\* - ocorre, 0 ou muitas vezes. Ex: alterar



○ – ou Evento é do tipo A ou é do tipo B

## Desenvolvimento de um CVE

### Passos do desenvolvimento

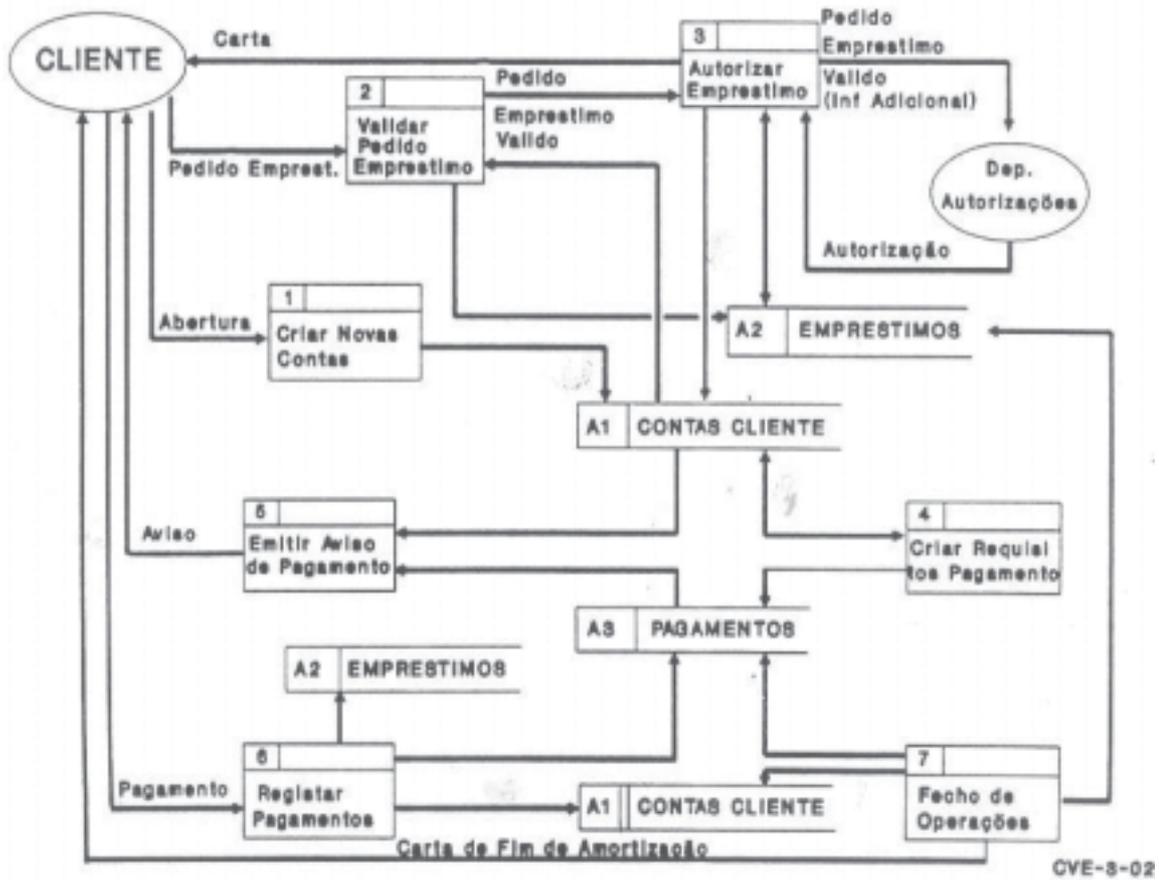
1- Criar a matriz eventos / entidades

- ✓ No DFD (re)conhecer os eventos
- ✓ No DEA reconhecer as entidades
- ✓ Para cada entidade verificar os eventos que a afectam
- ✓ Para cada evento rever a matriz

2 - Desenhar um CVE por entidade

- ✓ Criar a vida normal
- ✓ Adicionar os indicadores de estado
- ✓ Criar a vida alternativa

## DESENVOLVIMENTO DE UM C.V.E.



1 – (Re)conhecer os eventos

Quais os arquivos de dados e quais as setas que entram neles

- Arquivo Conta Cliente

### Estímulo

### Evento

1- Abertura de Conta/Cliente

Nova Conta

3- Resposta ao Pedido de empréstimo

Emp. Autorizado  
Emp. Não Autorizado

4 – Tempo

Ordens de Pagamento

6 – Pagamento

Pagamento

7 – Tempo

Abate

- Arquivo Conta Empréstimo

**Estímulos**

2- Pedido de Empréstimo

3 – Resposta ao Pedido de empréstimo

6 – Pagamento

7 – Tempo

**Evento**

Nova Conta

Emp. Autorizado  
Emp. Não Autorizado

Pagamento

Abate

- Arquivo Conta Pagamento

**Estímulos**

4 - Tempo

6 – Pagamento

7 – Tempo

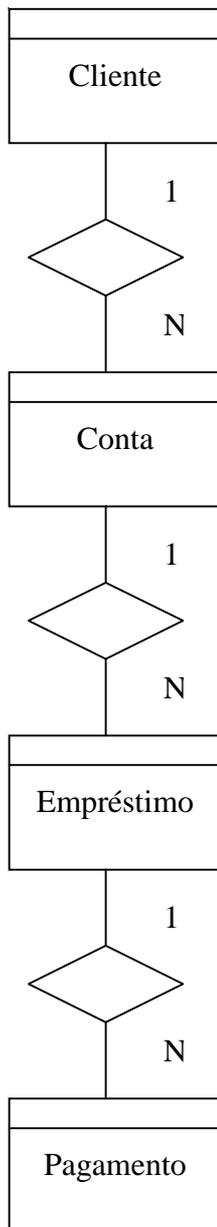
**Evento**

Ordens Pagamento

Pagamento

Abate

## Modelo Entidades - Associações



**DESENVOLVIMENTO DE UM C.V.E.  
MATRIZ EVENTOS - ENTIDADES**

	NOVA CONTA	EMPRES TIMO NÃO AUTORI ZADO	EMPRES TIMO AUTORI ZADO	ORDEN DE PAGA- MENTO	PAGA- MENTO	ABATE	NOVOS PEDIDOS EMPRES TIMO
CLIENTE	I						
CONTA	I	M	M	M	M	A	
EMPRESTIMO		M	M		M	A	I
PAGAMENTO				I	M	A	

### Algumas Sugestões para iniciar a construção do CVE

- 1- Quais os eventos que afectam a entidade?
- 2- Em que sequencia acontecem?
- 3- Alguns deles são alternativos? (Seleccção)
- 4- Alguns ocorrem repetidamente ? (Iteração)
- 5- Se há iteração, esta altera a seqüência?
- 6 - O sistema trata todas as iterações da mesma maneira?

### Respostas

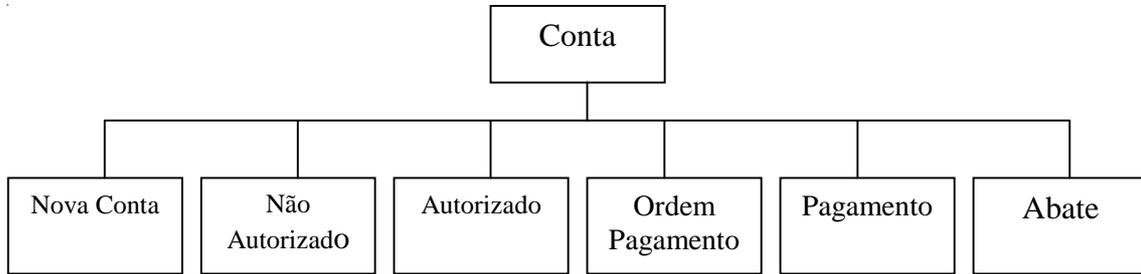
#### Resposta 1

Directamente da matriz eventos/entidades verificamos haver seis eventos:

- 1 – nova conta
- 2 – empréstimo autorizado
- 3 – empréstimo não autorizado
- 4 – ordens pagamento
- 5 – pagamento
- 6 – abate

**Resposta 2**

Entidade Conta

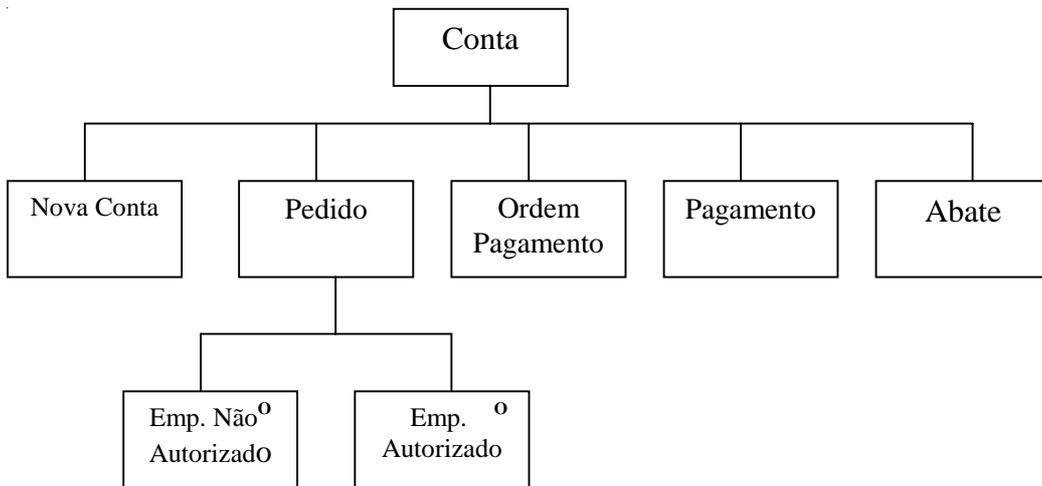


**Resposta 3**

Entidade Conta

Sim entre os Eventos:

Emp. Autorizado e Não Autorizado



**Resposta 4**

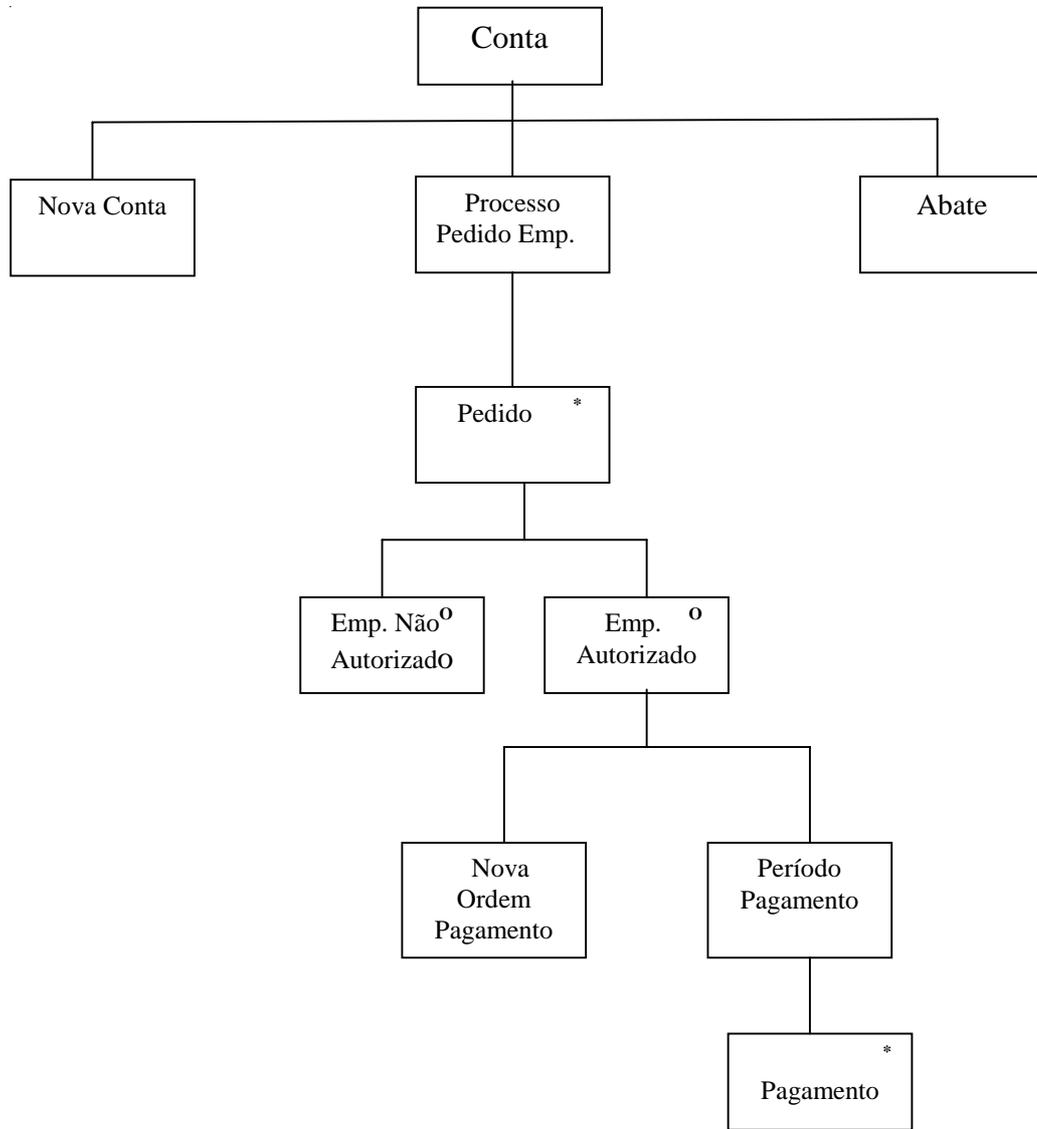
Eventos

Há Iteração?

Novas Contas	Não
Pedidos	Sim
Ordens de Pagamento	Sim
Pagamentos	Sim
Abate	Não

### Resposta 5

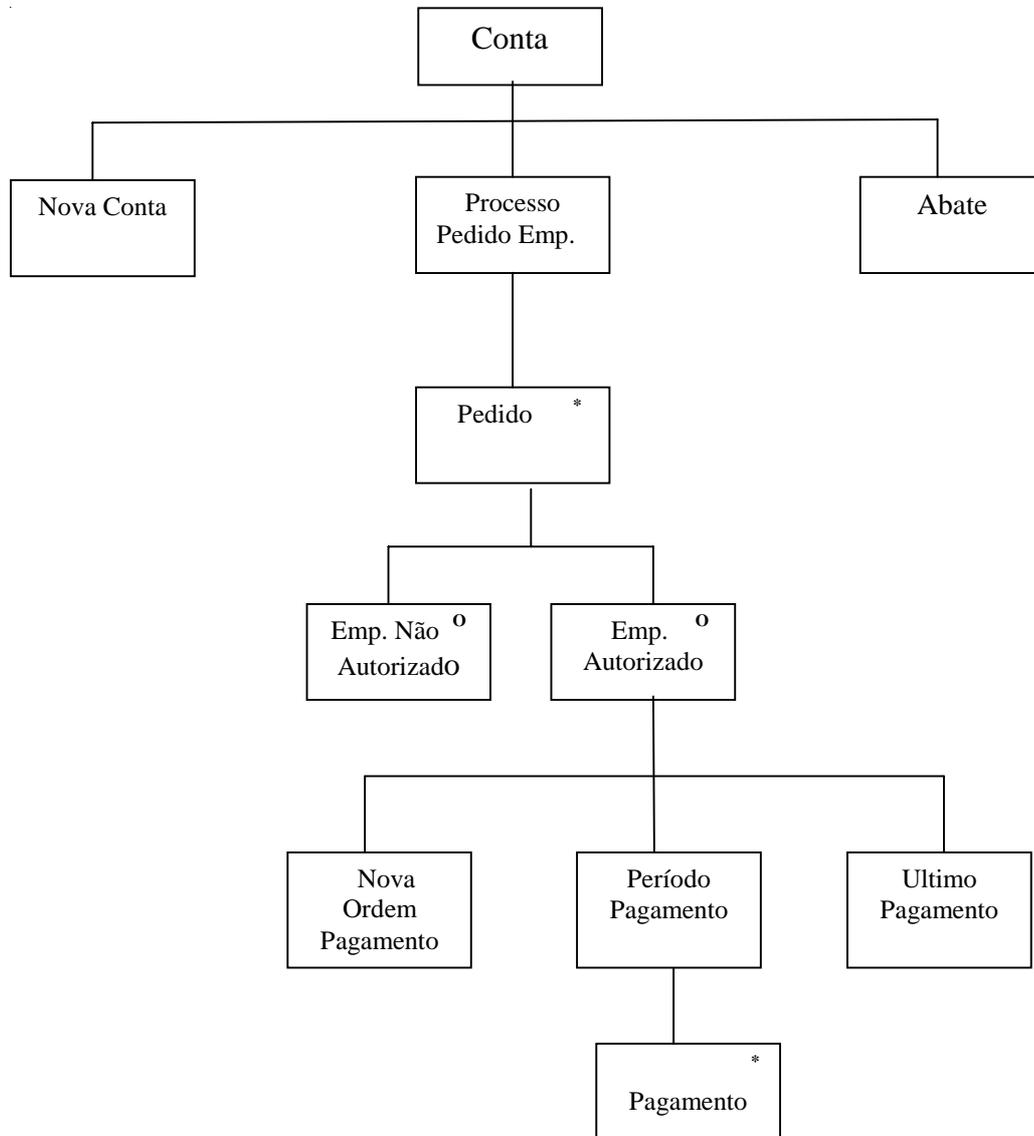
Entidade Conta



### Resposta 6

Entidade Conta

Não, Mas Como?



# Indicadores de Estado

Após a construção de um CVE, verificamos que os eventos correspondem ao acontecimentos/operações que é necessário executar sobre a informação "residente" na entidade para que esta tenha a sua vida representada no sistema em estudo.

Indicadores de Estado não são mais que:

- numeradores de acontecimentos

e

- indicadores de precedência

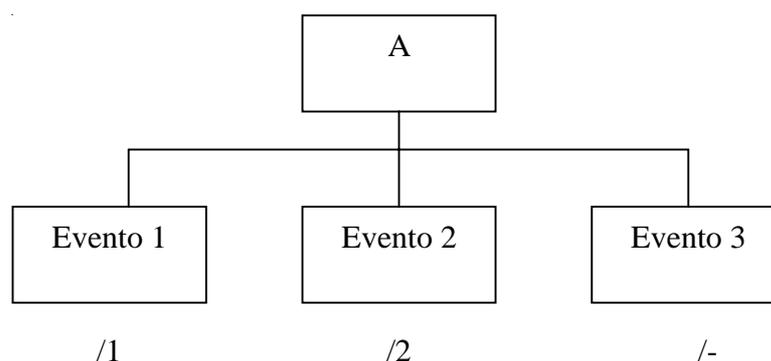
## Algumas regras a respeitar

### Numeradores de acontecimentos operações

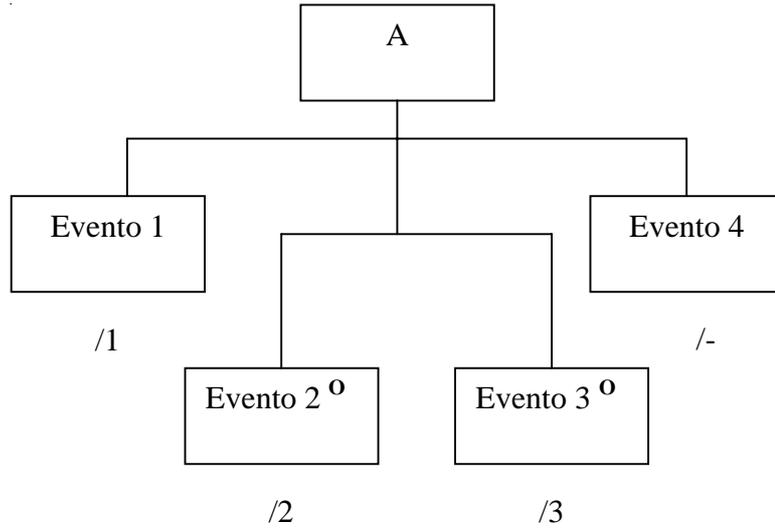
1- numerar a sequência das operações/acontecimentos elementares, ou seja, numerar os últimos ramos da árvore.

2 – A última operação não é numerada, porque obviamente não precede nenhuma outra.

### Exemplo 1



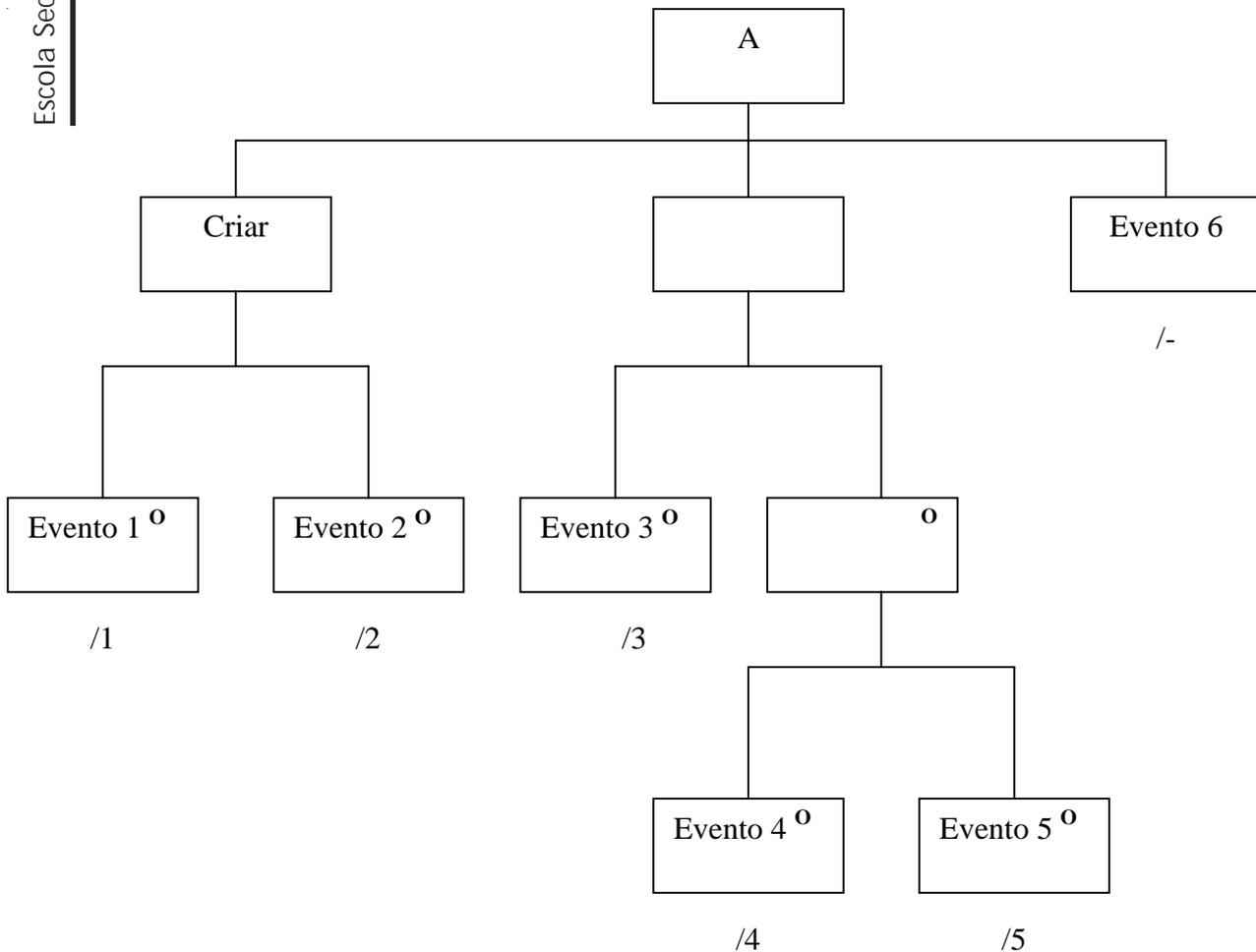
**Exemplo 2**



A figura "ou" implica que para a mesma ocorrência da entidade no instante t a sequência é:

Evento 1; Evento 2; Evento 4  
 Ou  
 Evento 1; Evento 3; Evento 4

**Exemplo 3**



**Legenda:**

Aqui a sequência pode ser:

**Hipótese 1**

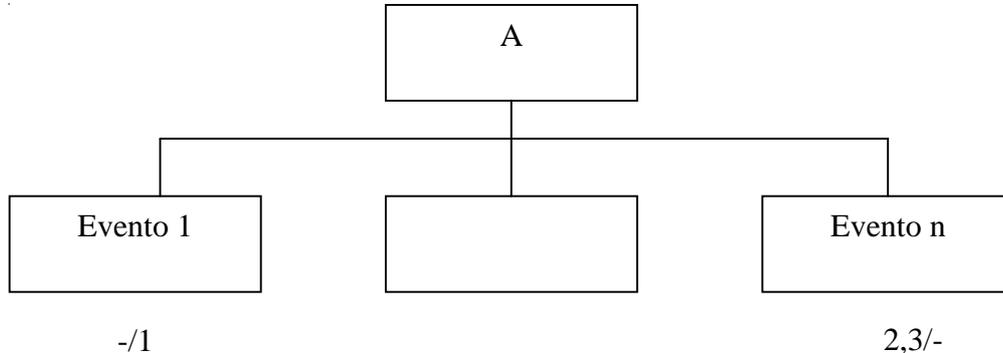
Evento 1; Evento 3; Evento 6      Ou      Evento 1; Evento 4; Evento 6      Ou  
 Evento 1; Evento 5; Evento 6

**Hipótese 2**

Evento 2; Evento 3; Evento 6      Ou      Evento 2; Evento 4; Evento 6      Ou  
 Evento 2; Evento 5; Evento 6

**Algumas regras a respeitar****Indicadores de precedência**

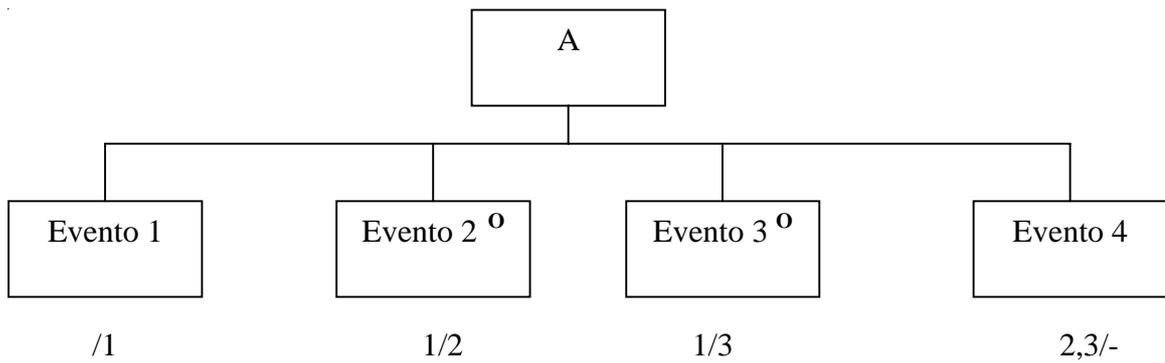
- 1 – A primeira operação/acontecimento é a única que não tem precedência
- 2 – As precedências são explicitadas usando o numerador de acontecimentos



O que quer dizer:

Para que o Evento n possa ocorrer, para além de ter de se verificar a regra subjacente a este evento, a entidade A tem que obrigatoriamente ter passado pelo estado 2 ou 3 (2,3), ou seja pelo acontecimento 2 ou acontecimento 3.

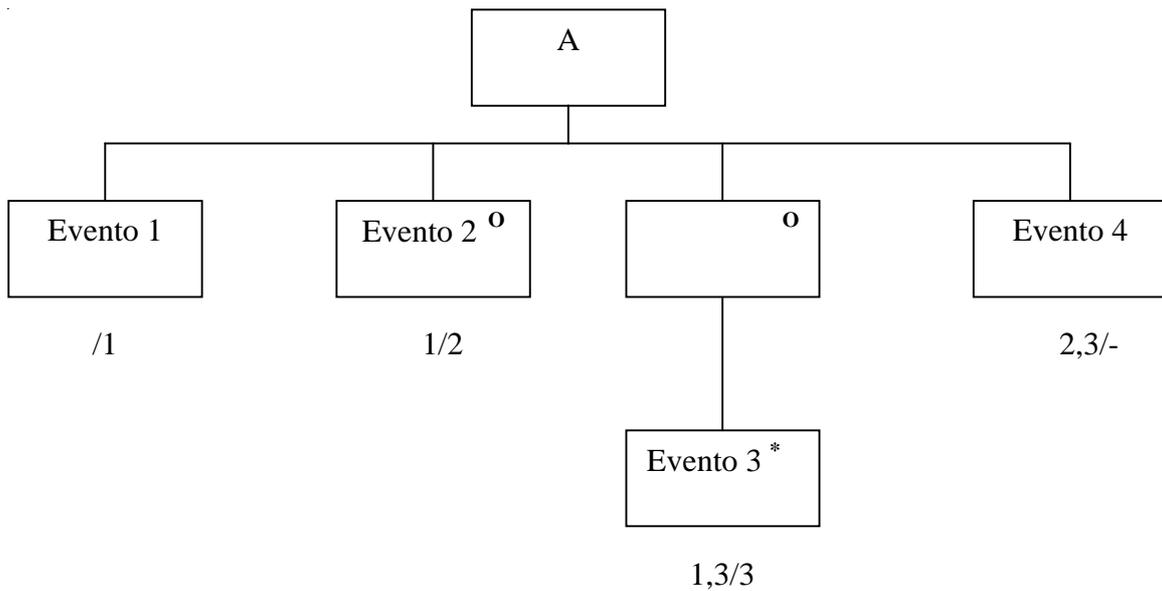
A mesma ocorrência passou obrigatoriamente pelo Evento1, mas em termos de precedências só representamos as imediatamente anteriores.

**Exemplo 1**

Lê-se:

O evento 1 é o primeiro a ocorrer. O evento 2 só pode ocorrer depois do evento 1. O evento 3 só pode ocorrer depois do evento 1.

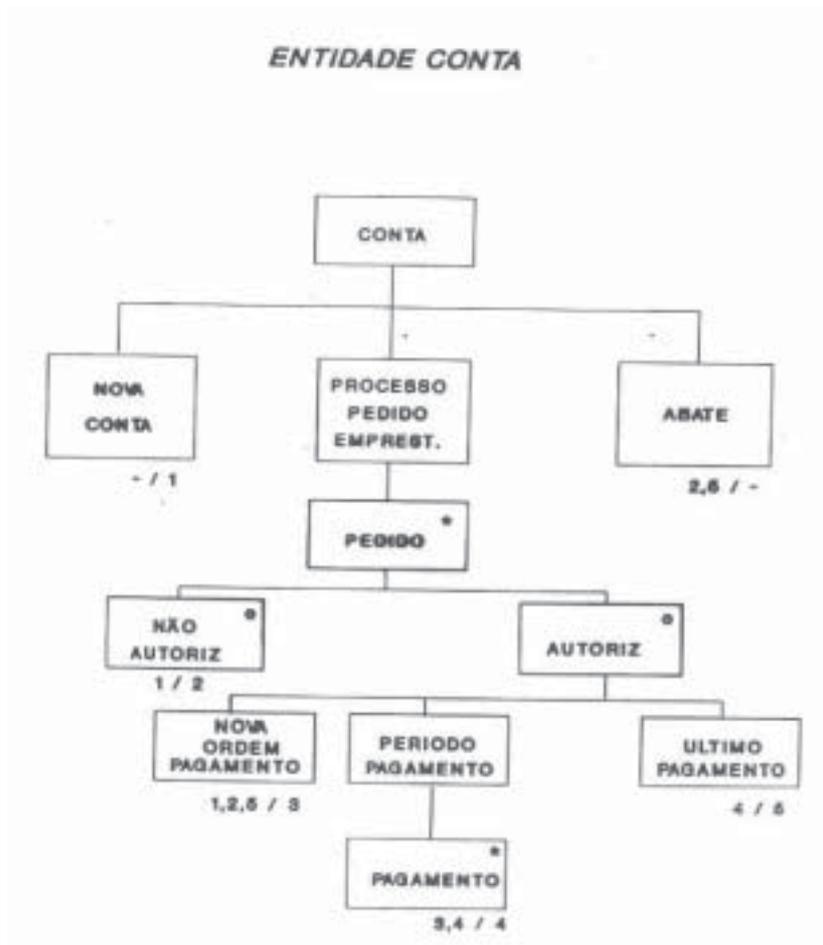
O evento 4 só pode ocorrer depois do evento 2 ou do evento 3.

**Exemplo 2**

Lê-se:

O evento 3 só pode ocorrer se tiver ocorrido o evento 1 ou o próprio evento3 (Iteração)

### Exemplo 3



# Dicionário de Dados

- É uma Listagem organizada de todos os elementos de dados pertinentes ao sistema, com definições precisas e rigorosas para que o utilizador e o analista de sistema possam conhecer todas as entradas, saídas, componentes de depósitos e cálculos intermediários.
- Colecção organizada de dados sobre dados.
- “Um Dicionário de Dados é um depósito de dados sobre dados” (Martin-1976).
- O DD é uma colecção de definições de tipos de dados simples ou compostos, associadas aos identificadores introduzidos nos diagramas de fluxos de dados, nos diagramas de entidade/relacionamentos, nas ferramentas de especificação funcional de processos e mesmo no próprio DD.

## Tipos de dados compostos

São definidos em função dos seus componentes e da forma de os associar.

## Tipos Simples (Dados elementares que não se podem substituir)

São definidos especificando o seu domínio (conjunto admissível de valores) e o significado atribuído a cada um dos valores admissíveis.

### *No DD encontramos definições sobre:*

Fluxos de Dados

Arquivos de Dados

Elementos Dados - É um tipo especial de fluxo de dados, que não pode ser decomposto em outros fluxos de dados subordinados.

Processos

Cada definição no Dicionário de Dados (DD) deve ser:

Concisa  
Precisa  
Não conter redundância

### ***Porque são importantes os DD?***

- Manusear o detalhe dos sistemas.
  - Comunicar o significado de cada elemento do sistema
  - Documentar características dos sistemas.
  - Avaliar características dos sistemas.
  - Detectar erros e omissões nos sistemas
- Os DD podem ser manuais ou automáticos.
  - Os DD devem ser de fácil acesso e actualização.

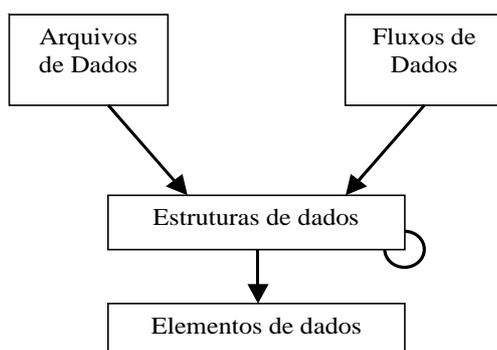
## **A necessidade da Notação de Dicionários de Dados**

Na maioria dos sistemas do mundo real em que vamos trabalhar os elementos de dados serão suficientemente complexos.

Os elementos de dados complexos são definidos em termos de elementos de dados mais simples, e elementos de dados simples são definidos em termos das unidades válidas.

## **Considerações**

### **Conceito de estrutura de dados**



**Elemento de Dados** - Dados quando na forma mais simples; podem tomar valores discretos ou contínuos e têm um determinado tamanho. O seu nome permite identificá-los na totalidade.

#### **Exemplo 1:**

- Altura = \* Unidades: Intervalo: 1-400\*
- Sexo = \* Valores [M|F]\*

## Notação do Dicionário de Dados

No DD a composição de uma estrutura de dados é representada recorrendo á seguinte simbologia:

	DeMarco	Gane
<b>Definição</b>	= (é composto de)	
<b>Sequência</b>	+ (e)	
<b>Seleccção</b>	[] (selecciona uma das opções alternativas)	{}
<b>Repetição</b>	() <i>interacção</i> - é usada para indicar a ocorrência repetida de um componente de um elemento de dados	*
<b>Opcional</b>	() (é o que pode estar presente ou ausente na composição de um elementos de dados completo)	[]

- Existem 3 formas básicas de obter tipos de dados compostos a partir de tipos mais simples não obrigatoriamente elementares). Essas formas são:

### Repetição

Onde um mesmo tipo de dados é repetido um número finito de vezes para definir a composição de um tipo composto.

### Sequência

Onde se concatenam diferentes tipos de dados para formar um tipo composto.

### Seleccção

Onde para se definir um tipo composto é opcional a consideração de um tipo de dados de entre um subconjunto de tipos mais simples.

- Assim,

#### Repetição

Pedido = nome-do-cliente + endereço-remessa + {Item}  
Stock-Local-de-peças = {Peça-Local}

#### Sequência

Peça-Local = cota + modelo + estado - peça

#### Seleccção

Requisição-de-livro = [Pedido-de-livro | Pedido-revista]  
Sexo = [Masculino | Feminino]

**Exemplo 1:***Segundo DeMarco*

*Requisição de Livro* = Nome do membro + Código do membro + (Curso)  
+ Data Requisição + { Nome + Editora + Ano Edição + {Nome do Autor}}

**Exemplo 2:**

Definir o nome de uma pessoa

Nome = Título-Cortesia + Primeiro-Nome + último-Nome

Título-Cortesia = [Sr. | Sra. | Dr. | Prof.]

Primeiro-Nome = {Caracter-válido}

Último-Nome = {Caracter-válido}

Caracter-válido = [A-Z | a-z | 0-9 | - | | ]

**CONCEITO DE ALIAS (Aliases são sinónimos)**

O sinónimo (alias), como o nome indica, alternativo para um elemento de dados é um nome. Vários nomes significando a mesma coisa.

**Exemplo:**

Cliente = \*Cliente da Empresa\*

= Nome + morada + ...

Freguês = \*Sinónimo de Cliente\*

**Especificação de um Fluxo de Dados no DD****Nome do Fluxo de Dados**

Pedido-de-Reserva

Pedido-Cliente

**Alias(es)****Composição**

Numero -de- quarto + Tipo -de- serviço + [Quantidade]

**Descrição****Origem****Destino****Comentários (Notas)**

*Tempo de resposta ao pedido, 30 segundos*

## Especificação de um Arquivo de Dados no DD

**Nome do Arquivo de Dados (Ficheiro)**

Reservas

**Alias(es)****Composição**Numero -de- reserva + nome -do- cliente + numero -de- pessoas +  
{numero -de- quarto + data -de- chegada + data -de- saída}**Descrição****Fluxo de input****Fluxo de output****Número identificador****Comentários (Notas)**

## Especificação de um Elemento de Dados no DD

**Nome do Elemento de Dados**

Estado-de-Quarto

**Alias(es)**

Situação-Quarto

**Tipo**

Constante

**Valores e significado**

P – Reservado

O – Ocupado

L – Livre

**Descrição****Tamanho**

1 carácter

**Comentários (Notas)**

## Descrição de Processos

Como fazer a Descrição Lógica dos Processos ?

- Português Estruturado
- Tabelas de Decisão
- Árvores de Decisão

Outras ferramentas

- Grafos e Diagramas
- Linguagem Narrativa
- Fluxogramas
- Diagrama de Nassi- Shneiderman
- ...

- Deve-se usar aquele que melhor transcreva cada caso particular.
- Cada descrição do processo deve mostrar as directrizes de transformação dos dados de entrada em dados de saída, e não a maneira de as implementar.

**A descrição do que ocorre no "interior" de cada processo de mais baixo nível deve ser feita de uma forma clara, completa e sem ambiguidades.**

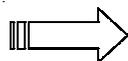
A descrição do que ocorre no "interior" de cada processo de mais baixo nível deve ser feita de uma forma clara, completa e sem ambiguidades.

Exemplo 1:

1. Somar A e B a menos que A seja menor que B onde, neste caso, subtrair A de B
2. Somar A e B. Mas se A for menor que B, a resposta será a diferença entre B e A.
3. Somar A e B, mas subtrair A de B quando A for menor que B.
4. O total é a soma de A e B; somente quando A for menor que B é que a diferença deve ser usada como total.

- Qual a diferença entre estas 4 afirmações?

**NENHUMA**



O analista deve reduzir esta variedade de afirmações a uma afirmação formato-padrão (standard).

SE A é menor que B

ENTÃO subtrair A de B

SENÃO somar A com B

## Especificação de um Processo no DD

### Nome do Processo

Criação de reservas

### Descrição Lógica (Português Estruturado / Tabelas de Decisão / Árvores de Decisão)

1. Receber pedido de reserva de cliente
2. Procurar quarto disponível no ficheiro quartos
  - 2.1 se quarto existe
    - então
      - 2.1.1 efectuar reserva
    - senão
      - 2.1.2 consultar ficheiro hotéis
      - 2.1.3 Preparar lista hotéis
  3. Preparar e enviar resposta ao pedido de reserva a cliente

### Número do Processo

3.1.2

### Fluxo Input

### Fluxo Output

### Comentários (Notas)

## Equilíbrio do DFD em Relação ao DD

- As normas para equilibrar o diagrama de fluxo de dados em relação ao dicionário de dados são as seguintes:

Cada fluxo de dados (uma flecha no DFD) e cada arquivo de dados deve ser definido no dicionário de dados. Se não constar no dicionário de dados, o fluxo de dados ou o arquivo é considerado como indefinido;

Inversamente, cada elemento de dados e cada arquivo de dados definido no dicionário de dados deve aparecer em algum lugar de um DFD. Caso negativo, o elemento ou arquivo de dados em questão é um "fantasma X" - algo definido mas não utilizado" no sistema;

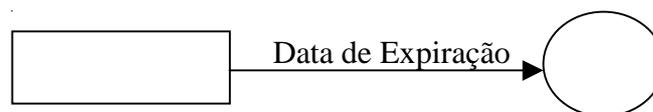
Isso pode acontecer se os elementos de dados forem definidos para corresponderem a uma versão anterior do DFD; o perigo está em que o DFD pode ser alterado (isto é, um fluxo ou um arquivo de dados pode ser suprimido) sem uma correspondente alteração no dicionário de dados;

Isso, naturalmente, significa que o analista de sistemas deve rever meticulosamente os DFDs e o dicionário de dados para se certificar que estejam equilibrados.

Não importa que modelo seja examinado em primeiro lugar, embora a maioria dos analistas comece pelo DFD para garantir que todos os elementos estejam definidos no dicionário de dados.

## Balanceamento Diagrama de Contexto vs DD

### Diagrama de Contexto



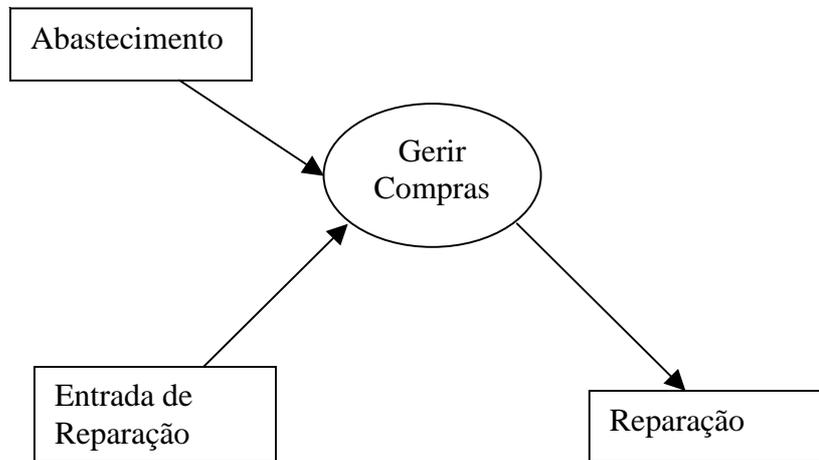
### Dicionário de Dados

Data de Expiração = \* Data impressa no cartão de crédito

\* Unidade = \*Data

*Exemplo :*

### Diagrama de Contexto



### *Dicionário de Dados*

Abastecimento-de-Peças = Fornecedor + LotePeça

Pedido-Peças = Reparação + Data + LotePeça

Peças-para-Reparação = Reparação + Cota

# Português Estruturado

## Definição:

O Português Estruturado é uma linguagem que combina um vocabulário limitado da língua portuguesa com a sintaxe das linguagens de programação estruturada.

## Português Estruturado – Convenções

O português estruturado é uma ferramenta de análise e apresentação da lógica de um processo recorrendo à escrita de sentenças em português e às seguintes convenções:

- 1- A lógica de todos os processos num sistema é expressa como uma combinação de estruturas de sequência, decisão, caso ou repetição.
- 2- As regras de português sem ambiguidades devem ser observadas.
- 3- As palavras SE, ENTÃO, SENÃO, LOGO, REPETIR e ATÉ devem ser escritas com letras maiúsculas e as estruturas devem ser deslocadas verticalmente (indentação) para mostrar a sua hierarquia lógica.
- 4- Blocos de instruções podem ser agrupados recebendo um nome significativo que descreva a função deles e que seja escrito também com letras maiúsculas.
- 5- Quando usarmos uma palavra ou frase que esteja definida em um dicionário de dados esta deve ser sublinhada.
- 6- Usa estrutura hierárquica recorrendo à indentação.

## Vocabulário:

- Verbos no imperativo  
Ex.:Determine, Acrescente, Copie, Some, etc...
- Termos definidos no Dicionário de Dados (em maiúscula)  
Ex.: COMISSÃO-VENDA, PEDIDO-COMPRA, etc...
- Palavras reservadas para a formulação lógica:
  - . conjunções: se, senão, enquanto, até que, para cada
  - . atributos relacionais: igual a, menor que, maior que, e, ou

Obs.: Palavras reservadas: minúsculas e sublinhadas.

## Português Estruturado – Características

- Apresenta muita precisão
- Não é um programa nem uma linguagem de programação
- Não apresenta especificações de leitura ou gravação de ficheiros físicos; de preparação de contadores ou chaves; ou de qualquer projecto físico.
- Apresenta-se sob uma forma de hierarquia de “blocos” de instruções.
- Apresenta itens sublinhados que traduzem termos devidamente especificados no Dicionário de Dados
- Apresenta uma forma muito completa ao registar passo a passo toda a lógica de um processo.
- A precisão e a integridade são os elevados preços de uma apresentação, por vezes, demasiado exaustiva e fora do comum para quem quer, apenas, uma panorâmica geral do processo.

## Vantagens / Desvantagens Português Estruturado

### ***Vantagens do Português Estruturado:***

- Fácil de ler.
- Fácil de alterar.
- Pode ser usado para documentar qualquer tipo de estrutura.
- Boa ferramenta para documentar processos em que haja a combinação de selecções e repetições.

### ***Desvantagens do Português Estruturado:***

- Como qualquer outra linguagem de especificação não há definição exaustiva da linguagem.
- Não é facilmente entendido quando usado para documentar uma estrutura de selecção muito complexa.

### Instruções Sequenciais

Esta estrutura cobre qualquer instrução, ou grupo de instruções, que não tenha repetição ou ramificação englobada nela própria

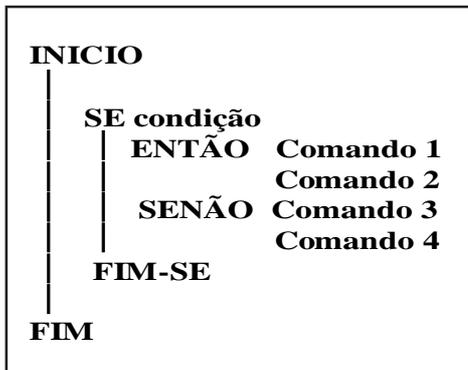
Ex:

“Multiplicar horas-trabalhadas por  
Salário-hora para obtenção do  
Salário-bruto”  
“Executar calcular-deduções”

### Instrução de Decisão

O formato padrão para este tipo de estrutura é:

#### **Estrutura Se**

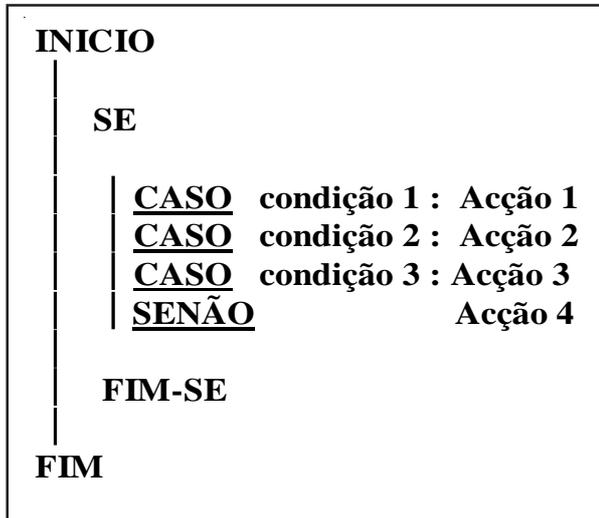


Cada acção aqui apresentada pode ser um conjunto de instruções sequenciais, ou um ciclo, ou outra decisão (SE's encadeados).

**Instruções de Decisão Tipo "CASO"**

Esta estrutura é um caso particular da estrutura anterior e aparece quando existem várias possibilidades de uma condição, que nunca ocorre em combinação; isto é, elas representam casos diferentes que são mutuamente exclusivos.

Formato padrão:

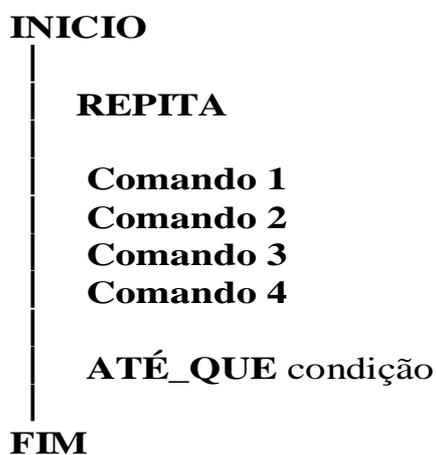
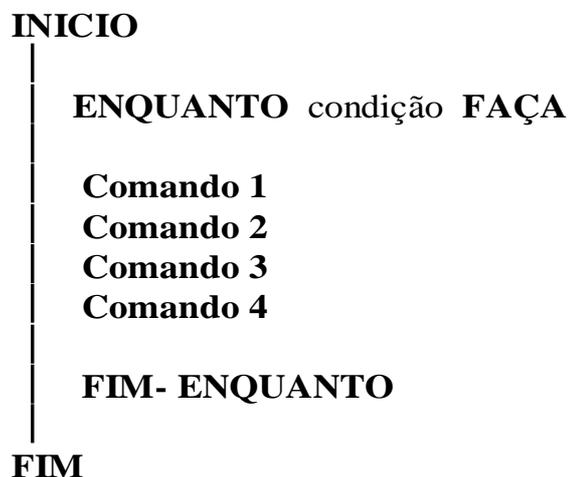
**Instruções de Repetição (loop)**

Esta estrutura é aplicada a qualquer situação em que uma instrução, ou grupo de instruções, é repetida até que o resultado seja obtido.

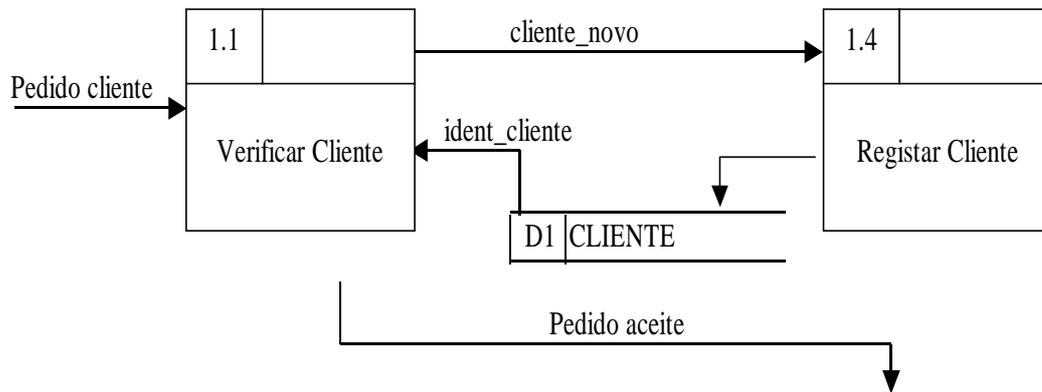
É preciso, no entanto, especificar dois aspectos para definir uma estrutura de repetição:

- O que deverá ser repetido
- O critério de paragem

Ex: REPETIR estender-item-linha ATÉ todas as linhas terem sido cobertas

**Estrutura Repita****Estrutura Enquanto**

Exemplo:



### Processo 1.1 - Verificar Cliente

**PARA** cada pedido\_cliente  
**ENCONTRAR** ident\_cliente em CLIENTE  
**SE** (ident\_cliente existe) **ENTÃO**  
**ENVIAR** pedido\_aceito para Elaborar Cartão  
**SENÃO**  
**ENVIAR** cliente\_novo para Registrar Cliente  
**FIM\_SE**

#### *Termos a serem evitados*

- Qualificadores vagos (adjetivos e advérbios)
- Verbos em tempos diferentes do imperativo
- Sinais de pontuação
- Notas de rodapé

#### Convenções para Português Estruturado de Chris Gane

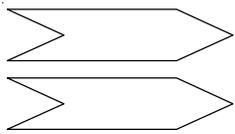
- Além das já vistas, as palavras reservadas "SE, ENTÃO, SENÃO, LOGO, REPETIR e ATÉ " devem ser escritas em letras maiúsculas.
- Permitido usar blocos de instruções com nome significativo (que descreva a função das mesmas) em letras maiúsculas.
- Palavra ou frase definida no Dicionário de Dados deve ser sublinhada.

## Pseudocódigo

É uma ferramenta que permite especificar a lógica do programa físico utilizando convenções do português estruturado, mas sem chegar à sintaxe detalhada de qualquer linguagem de programação em particular.

O pseudocódigo é, pois, uma notação de “quase-código” ou seja, de quase-programa em que o trabalho principal do projecto físico foi realizado e os ficheiros físicos já haviam sido definidos.

A especificação de um programa em pseudocódigo envolve para além da lógica do processamento, conceitos relacionados com:

- Inicialização
  - Finalização
  - Leitura
  - Gravação
  - Tratamento de fim-de-ficheiro
  - Especificações de contadores e indicadores
- 
- The diagram consists of two horizontal arrows pointing to the right. The top arrow starts at the level of 'Finalização' and points to 'Programa'. The bottom arrow starts at the level of 'Leitura' and points to 'Ficheiros'.

As estruturas que vamos utilizar, são iguais às utilizadas no português estruturado. Em pseudocódigo, diferenciamos a estrutura do ciclo REPETIR-ENQUANTO (DO-WHILE) da estrutura do ciclo REPETIR-ATÉ ( REPEAT UNTIL). Enquanto no ciclo REPETIR-ENQUANTO o teste de saída do ciclo é feito antes do conjunto de instruções que constituem o corpo do ciclo; no ciclo REPETIR-ATÉ o teste de saída do ciclo é feito depois de executado o conjunto de instruções que constituem o corpo do ciclo.

## Português Compacto

O português compacto é uma derivação do português estruturado. Tem a vantagem do português estruturado que é a de produzir uma lógica precisa e completa eliminando os problemas de ambiguidade e integridade do português corrente. Por outro lado suplanta o português estruturado, na medida em que, utiliza menos palavras e uma notação conhecida que favorece a apresentação aos utilizadores. Em resumo o português compacto apresenta uma aparência mais familiar mantendo a estrutura do português estruturado.

### Português Compacto - Convenções

- As operações sequenciais são apresentadas como instruções imperativas
- As estruturas SE-ENTÃO-SENÃO-LOGO são apresentadas com notação decimal e deslocamento vertical ou através de árvores de decisão
- As estruturas de caso são apresentadas como Tabelas

# Tabelas de Decisão

- Existem situações em que a linguagem estruturada (Português Estruturado) não é adequada para se elaborar uma especificação de processos.
- Isso verifica-se quando o processo deve produzir alguma saída ou executar acções com base em decisões complexas.
- Representação tabular de condições e acções, bem como, da indicação sob que condições uma acção é executada.

Componentes de uma tabela de decisão

**Condições** - Pode ser um acontecimento, um valor ou dimensão ou uma comparação entre dois itens.

**Acções** - É uma operação a ser executada (por meios manuais ou automáticos) por si ou numa série, sempre que um determinado conjunto de circunstâncias se tenha verificado. Qualquer tabela deve incluir pelo menos uma acção.

**QUALQUER TABELA É FORMADA POR 4 QUADRANTES:**

**1º Quadrante** - Listar todas as condições possíveis

**2º Quadrante** - Listar todas as acções possíveis.

**3º Quadrante** - Identificar todas as combinações possíveis das condições (Regras).

**4º Quadrante** - Identificar a acção que vai ser executada para cada combinação de condições.

Linhas Condições		Regras
	1º	3º
	2º	4º
Linhas Acções		Acção a ser tomada

Podemos dizer então, que a **ESTRUTURA BÁSICA** de uma tabela de decisão encontra-se esquematizada. Compõe-se de quatro quadrantes ou sectores normalmente separados por linhas duplas. Estes quatro quadrantes são respectivamente: o conjunto completo de condições aplicáveis (lista de condições), o conjunto completo de acções (lista de acções), as diferentes combinações daquelas condições (entrada de condições) e as combinações de acções (entrada de acções). Cada combinação de condições e acções associadas forma uma regra de decisão, que é lida verticalmente de cima para baixo e colocada paralelamente a todas as outras regras de decisão.

		Regra de Decisão 1	Regra de Decisão 2	Regra de Decisão 3	Regra de Decisão 4	Regra de Decisão 5	Regra de Decisão 6
SE	E	Lista de condições			Entrada de condições		
ENTÃO	E	Lista de acções			Entrada de acções		
			▼				

*Estrutura básica de uma tabela de decisão*

**Lista de (condições/acções)**

- Lista geral de (condições/acções)
- Parte curta de (condições/acções)
- Talão de (condições/acções)
- Matriz de (condições/acções)

**Entrada de (condições/acções)**

- (condições/acções)particulares
- Lista de (condições/acções) particulares

**Regra de decisão:**

- combinação de condições e acções associadas.

*Exemplo de Tabela de Decisão:*

	Regra de Decisão 1	Regra de Decisão 2	Regra de Decisão 3	Regra de Decisão 4	Regra de Decisão 5	Regra de Decisão 6
C1	S	S	S	S	N	N
C2	S	S	N	N	S	N
C3	S	N	S	N	S	S
A1	X	X		X		
A2				X		

**EXISTEM 3 TIPOS DE TABELAS DE DECISÃO:**

- Tabela de decisão de **Entrada Limitada**.
- Tabela de decisão de **Entrada Estendida** (alargada).
- Tabela de decisão de **Entrada Mista**.

**CARACTERÍSTICAS DAS TABELAS DE DECISÃO**

**Entrada Limitada**

- as condições só podem assumir valores **SIM / NÃO/ —**

- 'S' (Sim) - Significa que a respectiva condição deve ser satisfeita.
- 'N' (Não) - Significa que a respectiva condição não deve ser satisfeita.
- '-' (Indiferença) - Significa que é irrelevante que a condição seja ou não satisfeita.

-as acções são indicadas através do símbolo "**X**"

'X' - Significa que a respectiva acção deve ser executada se todas as condições da regra forem satisfeitas.

C1	S	S	N	N
C2	S	N	S	N
A1	X			
A2		X	X	
A3				X

2 CONDIÇÕES = Regras  
 $2^2 = 4$  combinações possíveis (4 regras)

**Entrada Estendida (alargada)**

- As **regras** > 16
- As **condições** assumem **valores específicos**
- As **acções** assumem **valores específicos**

C1	> = 10	]10,5]	< 5
C2	A	B	A
A1	10%	20%	20%

**Entrada Mista****- De Condições**

As **condições** assumem **valores específicos**.  
As **acções** são indicadas através **do símbolo "X"**.

C1	> = 10	]10,5]	< 5
C2	A	B	A
A1	X	X	
A2			X

**- De Acções**

As **condições** só assumem **valores Sim/Não**.  
As **acções** assumem **valores específicos**.

C1	S	S	N	N
C2	S	N	S	N
A1	10%	20%	20 %	5 %

***ELIMINAR REDUNDÂNCIAS, SITUAÇÕES IMPOSSÍVEIS E CONTRADIÇÕES.*****Eliminar Redundâncias**

Quando se identifica um par de regras com a mesma acção a ser executada (4ª quadrante) e só difere numa condição (3º quadrante).

C1	N	S	
C2	S	N	
A1	X	X	← Redundância
A2			

### Eliminar Situações Impossíveis

Salário > 12 500 Eur. / ano	S	
Salário < 7 500 Eur. / ano	S	
A1	X	← Situação Impossível

### Eliminar Contradições

C1	S	S
C2	S	S
C3	—	N
A1	X	
A2		X

Contradição

### SIMPLIFICAÇÃO DE TABELAS

Exemplo:

C1	S	S	N	N
C2	S	N	S	N
A1	X			
A2		X		
A3			X	X

Tabela Simplificada

C1	S	S	N
C2	S	N	--
A1	X		
A2		X	
A3			X

## Passos para elaborar uma Tabela de Decisão:

- 1 - Identificar todas as condições possíveis.
- 2 - Identificar todas as acções possíveis.
- 3 - Definir todas as combinações possíveis de condições.
- 4 - Identificar a(s) acções a levar a cabo para cada combinação identificada.
- 5 - Eliminar inconsistências (incertezas), redundâncias, situações impossíveis e Contradições.

### *VANTAGENS/ DESVANTAGENS DAS TABELAS DE DECISÃO*

#### **Vantagens:**

Conduz a uma especificação completa.  
Fácil de detectar situações impossíveis, contradições e redundâncias.  
Facilmente convertida em código.  
Boa para documentar estruturas de selecção complexa.

#### **Desvantagens:**

Só pode ser usada para exibir estrutura de selecção.  
Difícil de alterar.  
Só é fácil de ler e entender para utilizador treinado.

# Tabelas de decisão Versus Árvores de decisão

A tabela de decisão é mais completa, pois com ela teremos considerado todas as combinações possíveis de condição que possam ocorrer mas, em contrapartida, a árvore de decisão apresenta um quadro mais nítido da estrutura.

1. Utilizar uma árvore de decisão quando o número de decisões for pequeno e nem toda a combinação de condições for possível
2. Utilizar uma tabela de decisão quando o número de acções for grande e ocorram muitas combinações de condições.
3. Utilizar uma tabela de decisão se existirem dúvidas que a árvore de decisão mostra toda a complexidade do problema.

Mesmo que necessitemos de uma tabela de decisão para descobrir a lógica, procurar representá-la como uma árvore de decisão desde que o ponto 2 não seja violado.

**Numa árvore de decisão**, os ramos representam as condições a testar e as folhas as acções a levar a cabo.

## Vantagens e Desvantagens das Árvores de Decisão

### Vantagens:

1. Fácil de ler e entender.
2. A usar quando a sequência das condições é importante.
3. Boa ferramenta quando o número de combinações de condições e o número de acções não é grande.
4. Usar quando nem todas as condições são relevantes para todas as acções.

### Desvantagens:

1. Não é muito fácil de validar.
2. Boa ferramenta só para exibir a estrutura de selecção.

## O QUE USAR

**Árvores de Decisão** – para verificação lógica ou tratamento de decisões moderadamente complexas que dão origem a até 10-15 acções. Servem ainda para mostrar a lógica de uma tabela de decisão aos utilizadores.

**Tabelas de decisão** – Para combinações complexas de até 5-6 condições, com qualquer número de acções.

**Português estruturado** – Para problemas que envolvam combinações de sequências de acções com decisões ou ciclos.

**Português compacto** – Para apresentar lógica moderadamente complexa sem que surjam ambiguidades.