

**UNIP – Ciência da Computação**  
**AES – Análise Essencial de Sistemas**

**O modelo Estruturado – Uma síntese**

*“Todos os pedaços de vidas postos juntos, todo esse mosaico de experiências acaba compondo um quadro coerente. É esse quadro coerente que se torna objeto de análise.”*

*Paulo Freire, educador (Freire et al., 1983)*

Conforme visto, uma inspeção histórica, envolvendo o desenvolvimento de software, nos leva a encontrar uma progressão de métodos e técnicas que passam a ser aplicados no transcorrer dos anos. Na tabela abaixo, que mostra o resumo histórico dos métodos de desenvolvimento de software, tem-se de forma sintética uma visão dos formalismos propostos.

Tabela – Resumo histórico dos métodos de desenvolvimento de software.

<b>Modelo</b>	<b>Abordagem</b>	<b>Ferramentas</b>
<i>Balbúrdia (Informal)</i> Desde o início dos anos 50 (até hoje em muitos casos); porém, sem outra alternativa até o início da década de 70.	Funcionalidades (com maior enfoque) e dados.	Textos Fluxogramas
<i>Estruturado</i> O método começou efetivamente a ser empregado a partir de 1975 e deverá ainda continuar sendo utilizado mais alguns anos pelas empresas que possuem estruturas de sistemas concebidas a partir desse modelo, que ainda se encontra ativos.	Inicialmente a funcionalidade e depois os dados. Posteriormente, com a maturidade das ferramentas de SGBD, os dados ganham mais ênfase.	Diagrama de Fluxo de Dados (DFD) Especificação dos processos Diagrama de Entidade Relacionamento (DER) Normalização Diagrama de Estrutura de Dados (DED) Dicionário de dados
Alguns metodologistas: Chris Gane Edward Yourdon Trish Sarson Tom de Marco		
<i>Essencial</i> Trata-se de um aprimoramento do modelo estruturado que teve início em 1984.	Essência Funcional e Dados Integração Funcional e Dados	DFD de Contexto Lista de eventos DFD particionado por eventos Diagrama Entidade Relacionamentos Diagrama de Estrutura de Dados Normalização Dicionário de dados
Alguns metodologistas: Sthepehn McMenamim John Palmer		
Orientado a objetos Decorrentes dos conceitos já existentes nas linguagens de programação, especialmente na Simula(67) e Smaltalk(70). A aplicação na análise de sistemas teve início na década de 90.	Funcionalidades Objeto = Encapsulamento de Funções e Dados Contempla o estado de um objeto Visão estática e dinâmica	Diagrama de Casos de Uso Diagrama de Classes e Objetos Diagrama de Seqüência Diagrama de Colaboração Diagrama de Componentes Diagrama de Distribuição
Alguns metodologistas: Grady Booch		

# UNIP – Ciência da Computação

## AES – Análise Essencial de Sistemas

James Rumbaugh  
Ivar Jacobson  
Peter Coad  
Sally Shlaer  
Stephen J. Mellor  
Scott W. Ambler

Como primeiro método formal com ampla divulgação e alguma aceitação, tem-se a Análise estruturada.

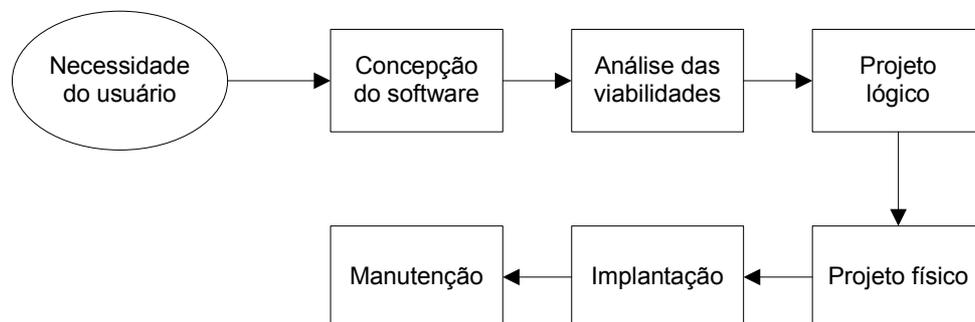
O conceito fundamental embasado na Análise Estrutura de Sistemas foi a construção de um *modelo lógico* (não físico) de um sistema. Para tanto, buscou facilitar a comunicabilidade e a clareza dos objetivos do sistema, empregando técnicas gráficas, uma vez que até então não havia como se visualizar graficamente as funções lógicas básicas e os requisitos do sistema, pois logo de início se adentrava aos detalhes de implementação física do software.

### 4.1 Análise Estruturada

O método da Análise Estruturada envolve a construção de um sistema de forma topdown (do geral para o particular, do topo para as partes) considerando-se os refinamentos sucessivos, produzindo-se em um primeiro momento uma fotografia global do sistema, por meio do emprego do DFD (Diagrama de Fluxo de Dados), em um nível macro, chamado de DFD zero ou de contexto (Fournier, 1994), (Gane & Sarson, 1990).

Com base nesse primeiro diagrama, faz-se sua decomposição funcional, criando-se outros fluxos que são um detalhamento do fluxo macro. Esses detalhamentos começam a dar pistas sobre os dados requeridos, os quais, posteriormente, são objetos de uma estruturação empregando-se o Diagrama de Entidades Relacionamentos.

O modelo estruturado apresenta o caminho exposto na figura abaixo, para que se possa construir um software. No modelo estruturado (como em todos os outros modelos existentes), tudo começa pela necessidade do usuário em resolver determinado problema (levantamento de requisitos) ou ainda pela oportunidade de antecipar a um problema, proporcionando funcionalidades que minimizarão os agravantes futuros.



Desenvolvimento de software pelo método da Análise estrutura.

Segundo William S. Davis (Davis, 1987), a seqüência das atividades pertinentes ao modelo estruturado pode ser expressa de forma tabular, segundo as questões-chave e atividades principais das fases, conforme mostra a Tabela abaixo.

**UNIP – Ciência da Computação**  
**AES – Análise Essencial de Sistemas**

<b>ETAPA</b>	<b>QUESTÃO-CHAVE</b>	<b>ATIVIDADES PRINCIPAIS</b>
Concepção do software	Quais são os problemas?	Especificação da delimitação e objetivos. Identificação de mais de uma alternativa.
Estudo de viabilidade	Há uma solução viável?	Análise geral custo/benefício das alternativas.
Projeto lógico	Em geral, como os problemas devem ser resolvidos?	DFD DER Especificação dos processos
Projeto físico	Especificamente, como o projeto deve ser implementado?	Codificação dos programas Testes
Implantação	Os usuários e a organização estão prontos para utilizar o sistema?	Treinamento Acompanhamento
Manutenção	Existem alterações requeridas	Modificar o sistema conforme necessário. Ajustar documentação. Apoio ao usuário.

### 4.1.1 Concepção do software

Uma vez identificados os problemas (realizado o levantamento de requisitos), o analista de sistemas passa a avaliar quais procedimentos comportam a adoção de alguma funcionalidade que deva ser contemplada por uma solução informatizada. Deve delimitar o problema e levantar, *a priori*, eventuais alternativas diante da situação inicial verificada. Trata-se da fase de concepção do software. Nesta fase desenvolve-se as etapas de levantamento das necessidades, a coleta de documentos, as entrevistas e as observações. Chris Gane e Trish Sarson (Gane & Sarson, 1990) sugerem que sejam empregados três questões para a fase de concepção, as quais irão nortear a atuação do analista:

- Que há de errado com a situação atual?
- Que melhoramento é possível
- Quem será afetado pelo novo sistema?

Há melhorias contínuas sendo solicitadas em empresas, qualquer que seja sua dimensão e, muito embora realmente se possa contribuir via sistema com o aumento de funcionalidades ou maior performance, nem sempre tais solicitações se revestem de uma real necessidade. Não raro, *algumas* solicitações podem ser derivadas de uma das seguintes situações:

- Estratégias gerenciais de desvios de foco diante de situações de turbulência dentro das organizações, visto que a solicitação para a área de informática passa a ser uma válvula de escape. Por exemplo, em uma indústria, onde frequentemente ocorram faltas de matérias-primas gerando com isso atraso no processo de fabricação (turbulência), pode ser solicitado para a informática determinado relatório ‘mais adequado’ para o departamento de Compras, justificando-se que os recursos de informação existentes não são suficientes.
- Aspectos misteriosos do tipo ‘deu um problema no sistema’. Todos os envolvidos no contexto do problema conseguem uma camuflagem... menos a organização (que detém, a partir daí, *dois* problemas). É de se esperar que os sistemas liberados para uso estejam livres de erros (é uma pré-condição de uso). Ocorre que em muitos lugares ainda existem os famosos sistemas *Frankenstein* (monstruosidade cheia de remendos e sem documentação) que realmente trabalham como *bem entendem*, pois,

## UNIP – Ciência da Computação AES – Análise Essencial de Sistemas

de tanto *remendo*, adendos e anexos, ninguém detém o conhecimento total de sua estrutura funcional; e, nesses casos, invariavelmente problemas acontecem. Para se precaver contra tais aspectos misteriosos que eventualmente possam vir à tona, e que *podem* ser empregados para acobertar falhas operacionais em decorrência do não cumprimento de procedimentos, sugere-se a criação de arquivos de *log* (registro de ocorrências). Determina-se no sistema corporativo quais são as operações mais estratégicas para a organização e a potencialmente problemática, passando-se a rastrear quem, quando, faz-se o quê.

Filtrando essas questões, uma solicitação para a área de informática realmente pode tratar-se de uma grande oportunidade para aumentar os rendimentos da organização, aperfeiçoar seus processos, melhorar qualidade de serviços e efetivamente colaborar com os objetivos globais da empresa.

Na fase de concepção, além de reunir-se as informações sobre os recursos que o novo software deverá apresentar, espera-se que o analista de sistema elabore algumas alternativas de desenvolvimento do software, que pode ser até um descritivo em forma de texto. Essas alternativas irão figurar como diferentes soluções para se fazer a mesma coisa. Por exemplo, mediante a solicitação da informatização de uma Faculdade, várias alternativas podem ser apresentadas, algumas mais simples (de menor custo), por exemplo, onde não seria contemplado a presença automática do aluno pela leitura de um *chip* subcutâneo e em outras isso estaria presente.

### 4.1.2 Estudo de viabilidade

Para efeito de comparação e escolha entre soluções previamente elaboradas (ainda de uma forma macro), baseadas nos levantamentos de requisitos aplica-se um estudo de viabilidade. Nesse estudo encontram-se três etapas distintas:

- **Análise da viabilidade técnica**  
Serão avaliados os recursos técnicos disponíveis, próprios ou não, que podem viabilizar as soluções que estão sendo analisadas. Esses recursos restringem-se ao hardware (configuração), software e pessoas qualificadas (*peopleware*). Se verificará pessoal técnico disponível, necessidade ou não de contratação ou treinamento.
- **Viabilidade econômica**  
Para as possíveis soluções, tecnicamente disponíveis e utilizáveis, se verificará o montante financeiro a ser empregado.
- **Viabilidade operacional**  
Aspectos das soluções que implicarão em mudanças de rotinas existentes ou a serem criadas e seus impactos na organização e sociedade, entram aqui discussões de ordem ética e moral também (apesar de tecnicamente funcional, seria uma solução adequada a implantação de *chip* subcutâneo para assinalar a presença de alunos?).

As soluções elaboradas e as respectivas análises de viabilidade devem ser apresentadas ao cliente. Naturalmente, quando da avaliação das soluções por parte do cliente, o analista de sistemas deverá ampará-lo quanto aos aspectos tecnológicos envolvidos.

## UNIP – Ciência da Computação AES – Análise Essencial de Sistemas

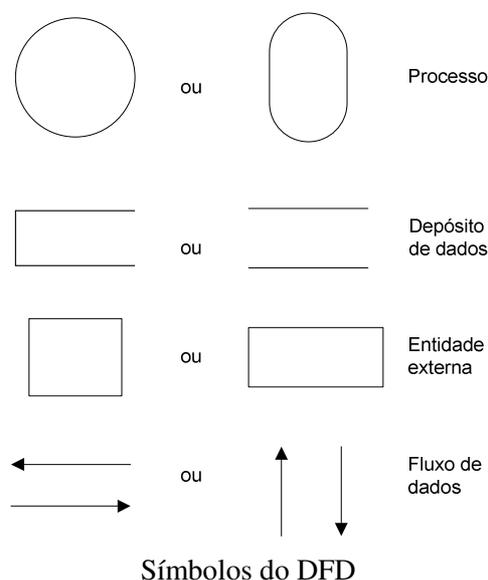
Uma vez que se tenha decidido por determinada solução, parte-se para a fase seguinte: chamado projeto lógico.

### 4.1.3 Projeto lógico

No projeto lógico pode-se ter novos contatos com os usuários envolvidos no processo de criação de software. A solução que foi escolhida começa a ser detalhada e vai sendo criteriosamente desenhada por meio do DFD (Diagrama de Fluxo de Dados). Ao iniciar este trabalho, que exige um detalhamento de todos os processos, os usuários poderão ser convocados para esclarecimentos sob aspectos obscuros ou ainda que não sejam do completo domínio do analista. Para Tom DeMarco (DeMarco, 89), há três tipos de usuários, cada qual tendo uma participação diferenciada no levantamento de requisitos inicial e nas fases posteriores em que se fazem necessárias, o que irá originar visões ou perspectivas diferentes de uma mesma informação:

- *Usuário que põe a mão na massa.* Essa pessoa é aquela que estará diretamente envolvida com as interfaces do sistema, alimentando-o ou extraíndo as informações. Normalmente está preocupado com as facilidades de uso e velocidade dos processos, além de detalhes operacionais que dizem respeito ao seu trabalho.
- *Usuário chefe.* Aquele que tem a incumbência de gerenciar o ambiente e o procedimento cotidianos, zelando pelo bom andamento dos processos e o atendimento a novas necessidades. Não está diretamente envolvido com os aspectos operacionais, porém depende deles para a obtenção de informações que ajudarão em processos de decisão.
- *Usuário dono.* Geralmente a diretoria ou o proprietário, dependendo da dimensão e da organização da empresa. Sua importância reflete-se nos sistemas com relação ao planejamento estratégico e às visões sistemáticas das informações.

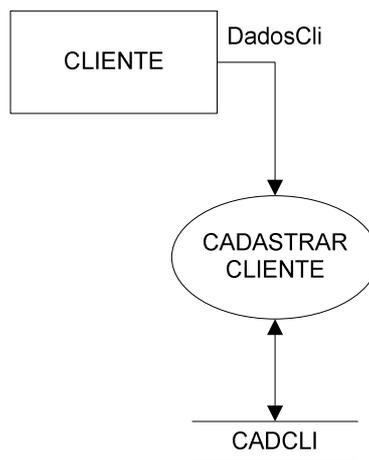
Ao desenhar um DFD pode-se retratar tanto os procedimentos manuais quanto os outros procedimentos que deverão ser informatizados e que fazem parte do fluxo normal de trabalho (*workflow*) para o sistema. O DFD possui uma convenção simbólica constituída de quatro representações gráficas que, juntas, permitem retratar uma abstração da realidade, particularmente mostrando *o que* existe, sem se preocupar em *como* tais coisas são feitas.



## UNIP – Ciência da Computação AES – Análise Essencial de Sistemas

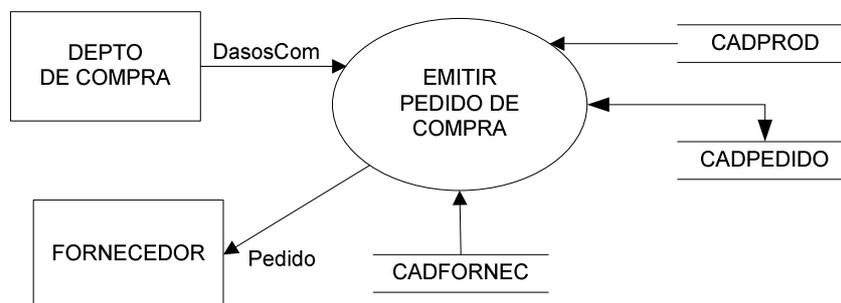
O símbolo de processo representa uma atividade de transformação. Ele é alimentado com dados, provenientes de um meio externo (via entidade externa), ou pela leitura de dados armazenados em algum depósito de dados (tabelas ou arquivos).

Todo processo tem um objeto claramente especificado, que, em geral, explicita o resultado oriundo de sua execução (mas isso não é regra). Seguem alguns exemplos de nomes de processos: Cadastrar Clientes, Emitir Pedido de Compra e Receber Materiais.



Processo 'Cadastrar cliente'

Na figura acima, tem-se o exemplo gráfico do processo 'Cadastrar cliente'. No referido DFD, observa-se que uma entidade externa (retângulo), chamada de 'Cliente', envia dados do cliente ao processo. O envio dos dados está representado pelo fluxo de dados (seta) chamado de 'DadosCli', que parte do cliente em direção ao processo, implicando, portanto, que o processo possui uma alimentação, um input de dados. Observa-se também que existe um fluxo de dados (seta) bidirecional entre o processo e um depósito de dados (traços paralelos) chamado de 'CADCLI', onde haverá o armazenamento dos dados do cliente. Portanto, o objetivo do processo 'Cadastrar cliente' é pegar os dados referentes aos clientes e depois armazenar esses dados no sistema (para que possam ser recuperados depois por algum outro processo).



Processo 'Emitir pedido de compra'

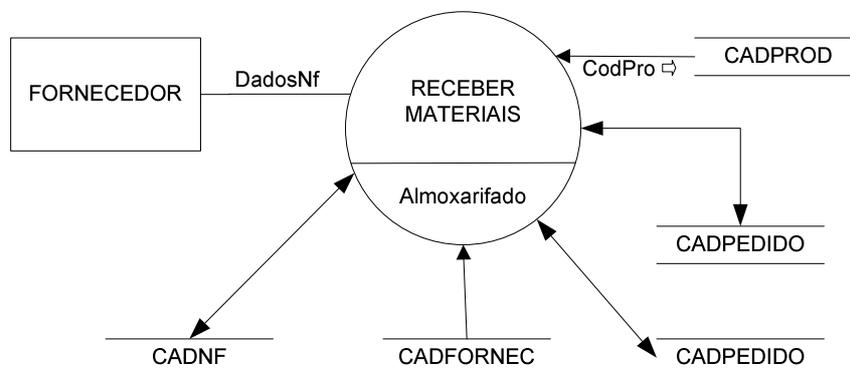
Na figura acima, tem-se outro exemplo de processo, neste caso 'Emitir pedido de compra'. Pela leitura que se faz no respectivo DFD, uma entidade externa chamada de 'Depto de Compra' alimenta o processo com dados da compra, representados pelo fluxo de dados chamado 'Dados Com'. Há três depósitos de dados envolvidos no

## UNIP – Ciência da Computação AES – Análise Essencial de Sistemas

processamento. Um deles chamado de ‘CADPROD’, onde se considera existir o cadastro dos produtos. Percebe-se que do cadastro de produtos parte um fluxo de dados em direção ao progresso, mostrando que há uma operação de leitura. No processo existe algum comando que requisita dados de um determinado produto que está armazenado em CADPROD; então, a seta indica a direção para a qual os dados estão fluindo. No caso desse depósito, só existe operação de leitura por parte do processo, pois não existe setas (fluxo de dados) partindo do processo em direção ao referido depósito de dados. A mesma operação de leitura é executada para o depósito ‘CADFORNEC’, porém em ‘CADPEDIDO’ existem duas operações: a de leitura e a de operação de gravação (setas bidirecionais indicam a operação de *Input-Output*, ou seja, podem ocorrer regravações – alterações em dados já gravados).

Partindo do processo de ‘Emitir pedido de compra’ em direção à entidade externa ‘Fornecedores’, há um fluxo de dados chamado ‘Pedido’, representando que há um resultado gerado pelo processo com destino a uma entidade externa. Esse resultado sempre será algo visível em algum meio, tal como um relatório ou uma consulta em vídeo; diferentemente do armazenamento de dados (fato não observável a olho nu), que normalmente ocorre internamente no hardware.

Uma operação de leitura de dados expressa em um DFD permite que se adicione a representação do chamado *argumento de pesquisa*. O argumento de pesquisa deve estar seguido de uma ‘ponta de seta’ indicando que o argumento de pesquisa é transmitido do processo ao depósito de dados. O argumento de pesquisa pode ser um dado, ou conjunto de dados, por meio dos quais se consiga encontrar um registro armazenado em um depósito (por intermédio da chave principal ou secundária).



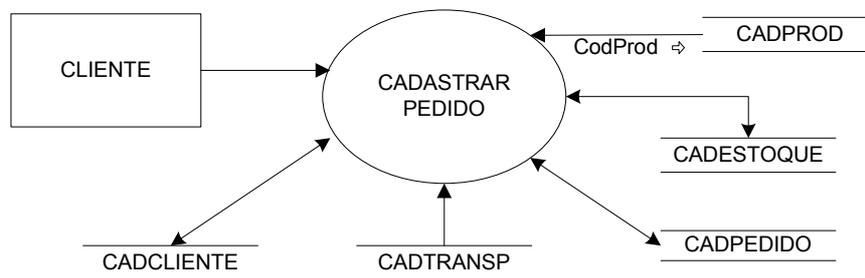
Processo ‘Receber materiais’

No DFD do processo ‘Receber materiais’, a entidade externa ‘Fornecedor’ alimenta o processo com os dados da nota fiscal (o fornecedor envia a nota, naturalmente com o respectivo material). Não está retratado no DFD a conferência física dos materiais que chegaram com a nota fiscal, o que certamente deve ser feito de forma manual. Contempla-se, contudo, pelo que se verifica expresso no DFD, a leitura do depósito de dados CADPEDIDO, o que nos leva a concluir que os dados da nota fiscal que chegou terá alguma confrontação com os dados do pedido que foi feito (produtos solicitados, condição de pagamento, preços, data de entrega combinada).

Ainda no DFD referente ao processo ‘Receber materiais’, observa-se na parte inferior da bolha um traço (característica essa da representação do processo), abaixo do qual tem-se o nome do departamento responsável pela execução do processo. Um erro muito

## UNIP – Ciência da Computação AES – Análise Essencial de Sistemas

comum visto em DFD é imaginar que o executor do processo é a entidade externa, quando na verdade ele representa apenas a *origem da informação*, e não quem executa o processo.



Processo ‘Cadastrar pedidos’

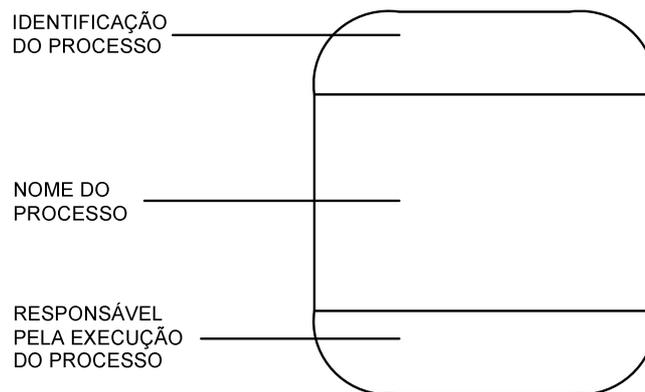
Em geral, ao construir DFD constata-se que não é documentado todo trâmite de uma informação antes da mesma ser inserida no sistema, embora isso seja possível e desejável, uma vez que o DFD pode expressar processos manuais e aqueles que serão objetos da informatização, de maneira que se vislumbre a totalidade das atividades. Na figura acima, pode-se observar a documentação lógica referente a um cadastro de pedido, cuja origem da informação foi o cliente, mas não está documentado como o pedido chegou até o sistema; por exemplo, todas as narrativas a seguir são situações possíveis de trâmite do pedido, desde sua origem até o momento do armazenamento dos dados no sistema de informação da organização:

- O cliente liga para a organização e conversa com o departamento de Vendas. Algum vendedor toma nota dos dados do pedido, solicita que o cliente envie um fax confirmando os dados e aí, de posse do fax, um auxiliar de vendas digita o pedido no sistema.
- O cliente recebe em sua loja um representante comercial, o qual preenche manualmente um pedido, que depois é colocado em um malote e segue para a organização, onde alguém do departamento de Vendas irá digita-lo no sistema.
- O cliente comparece no departamento de Vendas da organização e faz um pedido. Uma pessoa do departamento de Vendas digita os dados no sistema.
- O cliente utilizando a Web entra no sistema de vendas da organização e digita os dados do pedido.

O último tópico das narrativas anteriores revela que o cliente foi, além da origem da informação, aquele que também inseriu os dados no sistema, diferentemente dos casos anteriores.

Ainda com relação a simbologia de representação do processo, pode-se encontrar retângulos ‘em pé’ com os vértices arredondados, opcionalmente divididos em três áreas (Gane & Sarson, 1990), conforme mostra a figura abaixo.

## UNIP – Ciência da Computação AES – Análise Essencial de Sistemas

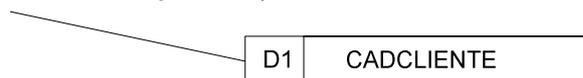


Notação do símbolo de processo

Com relação aos depósitos de dados, pode-se empregar na sua representação um par de linhas paralelas horizontais, podendo estar ou não ligadas em sua extremidade esquerda e devem ter um espaço suficiente para que se possa escrever dentro das linhas o nome de identificação do depósito. O nome, tanto quanto possível, deve ser auto-explicativo, significativo sob a ótica dos dados que armazena. O depósito de dados designa um local onde os dados são persistidos, isto é, há um armazenamento duradouro (ainda que temporário) e não momentâneo, entende-se que os dados permanecem armazenados mesmo que o equipamento onde estejam venha a ser desligado. Em geral, os depósitos de dados designam arquivos ou tabelas em banco de dados.

A representação do depósito de dados também pode vir a ter uma identificação. Para que o recurso seja aplicado, é necessário uma pequena adequação em seu desenho gráfico, conforme mostra a figura abaixo.

Região de identificação do depósito de dados



Notação do símbolo de depósito de dados

Quando um processo armazena dados, *a seta do fluxo de dados aponta para o depósito de dados* (Gane & Sarson, 1990). Quando se fala em armazenar dados, significa que estão gravando dados no depósito. Por outro lado, quando o acesso a um depósito de dados é feito de forma a realizar-se apenas leitura, um fluxo de dados partirá do depósito em direção ao processo, para o qual a seta deve apontar. Operações de atualização de dados, que se caracterizam por serem bidirecionais, isto é, primeiro ocorre uma leitura dos dados armazenados em um depósito (seta aponta para o processo, porque é a direção que seguem os dados); depois no processo pode ou não ocorrer alguma alteração e, na seqüência, há uma instrução de regravar os dados, ou seja, os dados saem do processo e são sobrepostos onde estavam e, neste caso, representa-se a seta do fluxo apontando para o depósito. Assim, a operação de atualização exige uma representação I-O (*Input-Output*), para a qual pode ser empregado um único fluxo (com setas bidirecionais), ou dois fluxos, um representando a leitura e o outro o armazenamento.

## UNIP – Ciência da Computação AES – Análise Essencial de Sistemas



Representação de operação de input-output

Por último, os fluxos de dados são representados por setas horizontais ou verticais que indicam a direção do fluxo. Nos casos necessários, em que um *mesmo conjunto de dados* flua em dois sentidos, em vez de empregar-se duas setas, pode-se utilizar um mesmo fluxo com duas setas.

### 4.1.3.1 Como utilizar o DFD no projeto lógico

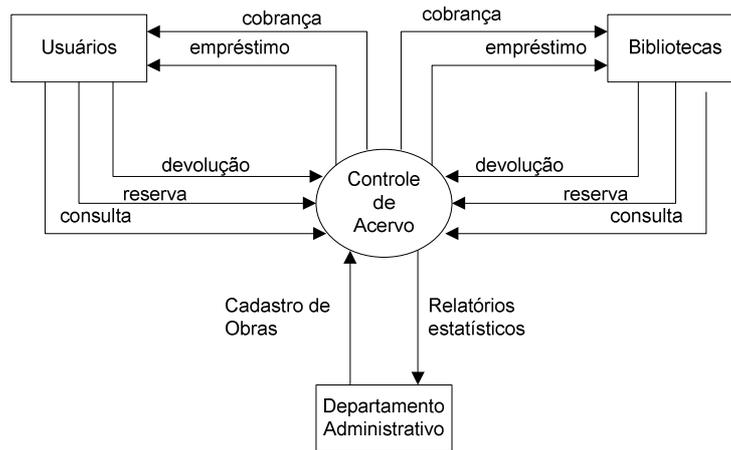
No modelo estruturado, o DFD é uma ferramenta aplicada para análise *top-down*, na qual a solução é retratada inicialmente de uma forma macro, obtendo-se a chamada visão de alto nível, ou visão contextual, ou, ainda, o chamado DFD zero. É a demarcação inicial em que se esboça os limites da abrangência do software. Posteriormente ocorre o detalhamento (explosão) dessa visão de alto nível, de forma a obter representações mais específicas dentro daquele contexto.

A seguir, temos um exemplo que retrata um sistema de controle de consulta, reserva e locação do acervo de uma biblioteca. O que exatamente se deseja informatizar por quê? Quais são os problemas existentes? Essas são algumas questões com as quais se poderia iniciar a investigação sobre a informatização da biblioteca. Observa-se que em uma biblioteca várias atividades podem ser objeto de informatização, como, por exemplo, a folha de pagamento dos funcionários. Assim, cabe a busca de um perfeito entendimento inicial sobre a abrangência que se espera do software que será projetado.

No levantamento inicial deve-se explorar alguns requisitos macros pertinentes aos limites da informatização, como no caso da biblioteca, que se supõe que as funcionalidades requeridas pelo usuário devam incluir a consulta de disponibilidade de obras, a reserva de obras, a locação, devolução, cobrança de obras em atraso e diversos relatórios ou consultas estatísticas, tais como: quais são as obras mais retiradas, quais são as obras solicitadas e que não existem no acervo, quais são os usuários em atraso, quais são as obras que estão emprestadas e quais são as obras reservadas. Vamos supor ainda que essa biblioteca forneça e possa solicitar obras de outras bibliotecas; para tanto, as funcionalidades exigidas para esse caso serão as mesmas que existirão para os usuários.

Todo esse conjunto de solicitações representaria um momento zero, no qual apenas se obteve uma visão genérica do que deve ser desenvolvido. Não se pode dizer que houve um levantamento de requisitos. É preciso explorar o escopo do software. Mesmo assim, o modelo estruturado permite que a partir de tais informações se desenhe o DFD de contexto, ou DFD zero.

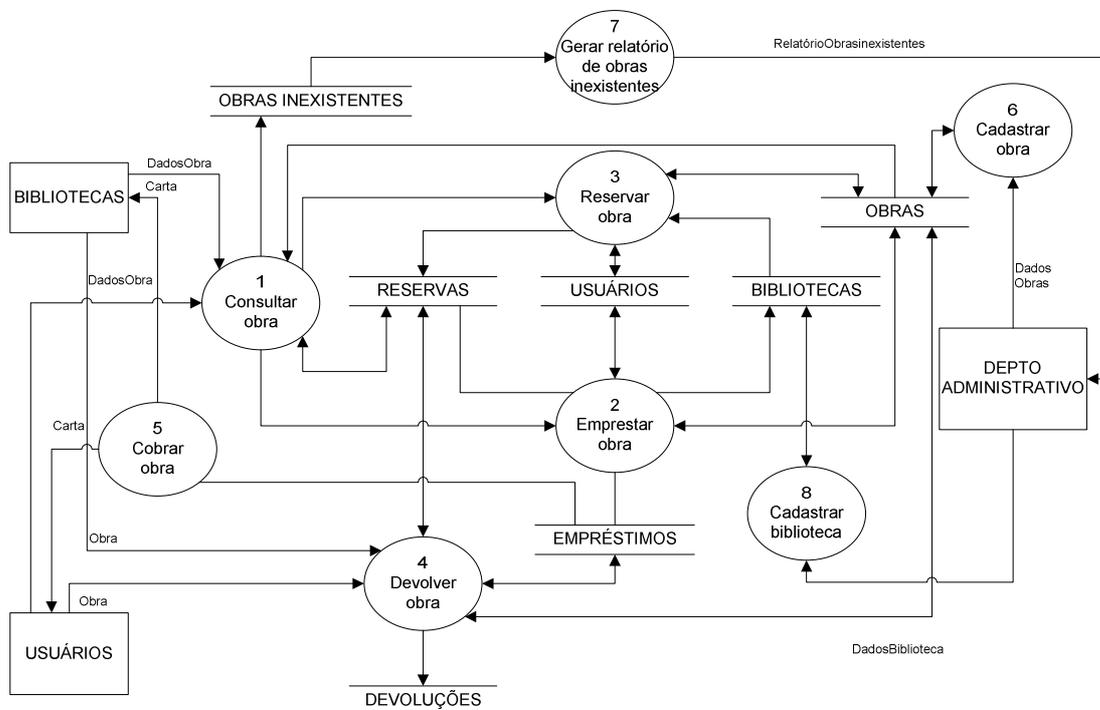
## UNIP – Ciência da Computação AES – Análise Essencial de Sistemas



DFD de contexto

Esse modelo inicial, esboçando uma síntese dos requisitos, objetiva a delimitação do problema a ser solucionado por meio de uma macro apresentação.

Posteriormente, por meio de um processo sucessivo de refinamento, começa-se a detalhar partes do modelo desenhado, obtendo-se expansões do primeiro esboço, como por exemplo, na figura abaixo, na qual temos uma explosão do DFD de contexto. Esse processo de detalhamento do DFD ocorre mediante a chamada *explosão* de uma bolha do modelo já desenhado. Isso nada mais é do que a expansão do significado de um processo. O DFD mostra o que deve ser previsto no software que será desenvolvido. A construção do DFD independe da plataforma de hardware e software que será utilizada na implementação do modelo; daí o nome desta fase: projeto lógico.



Explosão do DFD de contexto

Paralelamente à construção do DFD, pode estar sendo construído uma modelagem dos dados que são utilizados pelos processos.