

## 1 Análise Essencial

A análise essencial pode ser considerada um refinamento da análise estruturada. O problema existente (ou situação que requer a informatização) é estudado, porém não é modelado: os esforços são concentrados na identificação das funcionalidades lógicas requeridas para o software que será criado e, a partir daí, cria-se um modelo essencial do software que será desenvolvido, não se incorporando as exigências físicas.

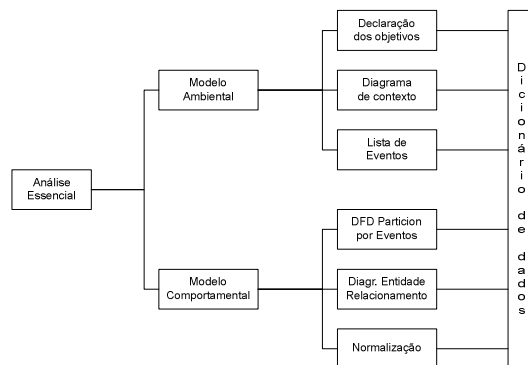
Na análise essencial, a premissa básica é descrever o sistema de maneira independente de restrições tecnológicas, o que permitirá uma solução ideal ao problema, sem deixar influenciar-se por questões decorrentes das restrições, o que poderia antecipadamente impor alguma restrição à solução pensada. Isso implica dizer que devemos considerar na confecção do modelo essencial a existência de uma tecnologia perfeita. Deve-se entender esse aspecto como uma abstração em que se supõe uma tecnologia ideal, sem limitações, onde:

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

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

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

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



Modelo essencial, adaptado de (Pompilho, 1995)

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

Vejam a composição do modelo essencial:

### 1.1 Modelo Essencial

Apresenta o sistema num grau de abstração completamente independente de restrições tecnológicas. Antes que um sistema seja implementado, é necessário conhecer a sua verdadeira essência, não importando saber se sua implementação vai ser manual ou autorizada, e nem mesmo que tipo de hardware ou software vai ser usado. É o que os autores da análise estruturada chamam de modelo lógico; no caso, o modelo essencial corresponde ao modelo lógico proposto, que é a denominação que a análise estruturada dá ao modelo lógico desejado para o sistema que ainda vai ser implementado.

#### 1.1.1 Modelo ambiental

No modelo ambiental tem-se a especificação macro do sistema (como se fosse uma caixa preta) inserida em um meio ambiente; busca-se representar a relação do sistema com o meio onde ele está. Eventos que ocorrem no meio ambiente são geradores de estímulos, os quais acionam procedimentos nos sistemas que, por sua vez, geram respostas. As respostas poderão ser internas ao sistema ou ainda serem enviadas para o meio ambiente (respostas externas). Três grandes atividades são elaboradas nesse modelo: declaração dos Objetivos do sistema, a elaboração do Diagrama de Contexto e a especificação da Lista de Eventos.

##### 1.1.1.1 Levantamento de Requisitos

Os problemas que os engenheiros de software têm para solucionar são, muitas vezes, imensamente complexos. Compreender a natureza dos problemas pode ser muito difícil, especialmente se o sistema for novo. Conseqüentemente, é difícil estabelecer com exatidão o que o sistema deve fazer. As descrições das funções e das restrições são os requisitos para o sistema [SOMMERVILLE03].

Os Requisitos (*Requirements*) podem ser definidos como as necessidades básicas do cliente, geralmente explicitadas como condição de negócio no contrato com o fornecedor. São características, tais como especificações técnicas, prazo de entrega, garantia, que o cliente “requer” do produto [MCT02].

“Uma condição ou capacidade necessitada por um usuário, para resolver um problema ou alcançar um objetivo” [IEEE90].

Segundo Tonsig, Os requisitos compõem o conjunto de necessidades estabelecido pelo cliente / usuário do sistema que definem a estrutura e comportamento do software que será desenvolvido [TONSIG03].

Segundo Pádua, o fluxo de requisitos reúne as atividades que visam a obter o enunciado completo, claro e preciso dos requisitos de um produto de software. Esses requisitos devem ser levantados pela equipe do projeto, em conjunto com representantes do cliente, usuários chaves e outros especialistas da área de aplicação [PADUA03].

No contexto de sistemas baseados em computador, o termo especificação significa coisas diferentes para pessoas diferentes. Uma especificação pode ser um documento

## **UNIP – Ciência da Computação**

### **AES – Análise Essencial de Sistemas**

escrito, um modelo gráfico, um modelo matemático formal, uma coleção de cenários de uso, um protótipo ou qualquer combinação desses elementos. Para sistemas grandes, um documento escrito, combinando descrições em linguagem natural e modelos gráficos podem ser a melhor abordagem. Cenários de uso podem, entretanto, ser tudo que é necessário para produtos ou sistemas pequenos que residem em ambientes técnicos bem-entendidos [PRESSMAN02].

#### **1.1.1.2 Declaração de Objetivos**

Trata-se da especificação daquilo que o sistema deverá propiciar, frente aos requisitos que foram identificados previamente. É uma descrição textual, sem um formato estabelecido pelo método. Deve também, tanto quanto possível, refletir os desejos do usuário no que diz respeito às solicitações que ele tenha apresentado como alternativas de solução dos problemas.

Naturalmente antes da elaboração dos objetivos do sistema, o analista deverá ter efetuado um minucioso e detalhado levantamento de requisitos, conhecendo profundamente o chamado domínio do problema. Se o sistema for referente a controle de uma biblioteca, o analista precisa saber ‘tudo’ sobre tal biblioteca, se interar dos processos de funcionamento da biblioteca, ou seja, regras gerais, linguagem utilizada, detalhes operacionais e exceções.

A declaração de objetivo do sistema deve estar resumida a um parágrafo, ser global, especificando o principal propósito da criação do software. Não cabe neste ponto estabelecer uma lista textual com todas as funcionalidades esperadas no sistema.

Ex: “Controlar o serviço de reservas, registros e cobrança de quartos de um sistema hoteleiro”.

#### **1.1.1.3 Diagrama de Contexto**

Após a especificação de requisitos e dos objetivos do sistema, o analista já estará em condições mais apropriadas para elaborar o diagrama de contexto. O diagrama de contexto reflete graficamente a relação do sistema com o meio ambiente onde está inserido. Essa relação dá-se por meio do recebimento de estímulos do meio ambiente, os quais ativam processos, e estes, por sua vez geram respostas, que podem vir a ser respostas externas ao sistema, ou seja, resposta ao meio ambiente. Para o diagrama de contexto na análise essencial, é aplicado o DFD.

O diagrama de contexto é um DFD onde aparece um único processo (bolha) que representa o sistema. Mostra a relação do sistema com entidades externas a ele. As entidades externas possuem alguma responsabilidade de interação, que seja para fornecer informações que seja como destino de informações geradas pelo sistema.

Outro aspecto importante a ser destacado é que não cabe no DFD de contexto a especificação de depósitos de dados. Considera-se que um depósito de dados seja algo interno ao sistema e portanto, não aparece no contexto, onde se mostra apenas os limites do sistema e sua relação com o mundo fora dele.

#### **1.1.1.4 Lista de eventos**

Trata-se da especificação das atividades (processos) essenciais que o sistema terá. Tais atividades (no sistema) são ativadas por estímulo (fluxo de dados, fluxo temporal ou de controle), executam processamento e geram respostas. Não há uma precedência estabelecida para a elaboração da lista de eventos e o diagrama de contexto: são atividades que podem estar acontecendo paralelamente, porém, devem estar consistentes.

A lista de eventos relaciona todas as atividades essenciais (fundamentais) do sistema que está sendo modelado. A lista de eventos é construída após, ou paralelamente, a construção do diagrama de contexto. A diretriz básica é que essas duas ferramentas devem apresentar dados coerentes entre si, ser consistidas a partir de um cruzamento das informações que apresentam; por exemplo, todos os estímulos que existem no DFD de contexto devem aparecer na coluna estímulo da lista de eventos (Cada um dá origem a um evento da lista), igualmente, todos os fluxos de dados existentes no diagrama de contexto que representam uma resposta do sistema, devem constar na respectiva coluna da lista. Só haverá resposta por parte de um sistema, se houver um estímulo que aciona a ação geradora da referida resposta. Pode-se concluir que a lista de eventos deverá ter no mínimo tantos eventos quantos estímulos existirem no diagrama de contexto; porém, nem toda ação executada a partir de um estímulo irá gerar uma resposta externa ao sistema.

#### **1.1.2 Modelo comportamental**

O modelo comportamental é a fase em que o Analista passa a olhar para dentro do sistema. Irá detalhar como cada atividade existente na lista de eventos se comporta (com ela deve funcionar). Também fará um modelo de dados sobre o qual o sistema atuará, observando critérios para conseguir boa performance na sua utilização (por meio da normalização de dados). Acompanhando mais efetivamente esse modelo (muito embora já possa existir antes dele) cria-se o dicionário de dados. Também nesta fase elabora-se o DFD particionado por eventos e o DFD hierárquico do sistema, que nada mais é do que o agrupamento de atividades essenciais afins, que enfocam determinado aspecto no sistema.

##### **1.1.2.1 Diagrama de fluxo de dados particionado por evento**

Para cada item da lista de eventos, o analista de sistemas fará um Diagrama de Fluxo de Dados, representando de forma gráfica, individualmente, cada evento existente no sistema. Desta maneira, haverá tantos diagramas de fluxo de dados particionado por eventos quantos forem os itens existentes na lista de eventos.

Depois que a lista de eventos estiver concluída, desenvolve-se o diagrama de fluxo de dados particionados por eventos, também conhecido como diagrama das atividades essenciais. A atividade do desenvolvimento do DFD particionado por evento, onde ocorre o detalhamento dos processos, é também conhecida como Modelagem Funcional, pois o aspecto principal é desenhar um modelo de como as funcionalidades existentes no sistema deverão ocorrer, tudo com base nas ações especificadas na lista de eventos.

## **UNIP – Ciência da Computação**

### **AES – Análise Essencial de Sistemas**

Em última análise, o diagrama particionado por evento é uma forma visual de enxergar-se a lista de eventos, com o acréscimo de uma descrição da funcionalidade da ação e a porção de dados que essa ação irá utilizar. A partir deste ponto, a análise de sistemas passa a incorporar os dados no projeto do sistema, à medida que detalha as ações do sistema, documenta quais são os dados requeridos por determinada ação. Paralelamente pode estar sendo construído o DER.

#### **1.1.2.2 Diagrama hierárquico de macro atividades**

Trata-se de um DFD onde se propiciará uma visão sintética única do sistema. Nesse DFD se aglutinará as funcionalidades existentes na lista de eventos de acordo com os assuntos que tratam. Pegam-se os diagramas de fluxo de dados particionados por eventos e verifica-se quais são aquelas atividades afins (que tratam de determinado assunto). Esses processos são aglutinados em um único, de tal forma que se obterá uma visão mais sintética da representação do sistema, cuja finalidade é, além da documental, a possibilidade de examinar e definir interfaces e locais de processamento. A fim de facilitar a construção do DFD hierárquico (por meio de uma visão mais global do sistema), pode-se antes elaborar o chamado Diagrama preliminar, que consiste em pegar todos os DFDs particionados por evento e torná-lo um só (visão única de um DFD com todos os processos existentes).

#### **1.1.2.3 Diagrama Entidade Relacionamento**

Para modelagem de dados, o analista de sistemas fará inicialmente o DER.

Com esse diagrama terá um poderoso instrumento para mapear como os dados estão organizados e como eles se relacionam. A representação inicial do modelo de armazenamento independe dos dispositivos onde os dados ficarão armazenados.

Depois de concluído, ou paralelamente à atividade de construção do DFD particionado por eventos, o analista de sistemas deve construir a modelagem de dados, empregando para tanto o DER. Existe uma necessidade de estudar mais profundamente como os dados utilizados pelo sistema deverão ser organizados. Esse fato deve-se a fatores de performance na sua utilização cotidiana pelos usuários.

A modelagem de dados é a fase do método essencial em que o analista de sistemas busca especificar, a partir dos fatos essenciais que estejam associados ao domínio de conhecimento analisado, a perspectiva dos dados, permitindo organizá-los em estruturas bem definidas, estabelecer as regras de dependência e restrições entre eles, produzindo um modelo expresso por uma representação, ao mesmo tempo descritiva e diagramática.

No método da análise essencial, a modelagem de dados começa a surgir no momento em que é definido algum depósito de dados no DFD particionado por evento, uma vez que tal depósito dará origem a uma ou mais entidades na modelagem de dados.

#### **1.1.2.4 Normalização**

Definições estão presentes na Parte 03 da disciplina AAES.

Acesse: [http://www.noginfo.com.br/arquivos/SI\\_AAES\\_Parte\\_03\\_MER.pdf](http://www.noginfo.com.br/arquivos/SI_AAES_Parte_03_MER.pdf)

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

### 1.1.2.5 Diagrama de Transição de Estado

Para elaborar um diagrama de transição de estados (DTE), algumas definições se fazem necessárias:

Um estado de um sistema representa uma situação, um cenário ou um modo de comportamento em que encontramos um sistema ao observá-lo em determinado momento. Assim, se o sistema que estivermos observando for uma lâmpada elétrica, podemos encontrá-la em um dos seguintes estados: acessa e apagada.

Uma transição representa a passagem do sistema de um estado para outro. Assim, ao acender ou ao apagar uma lâmpada, estaremos provocando uma transição de estado.

Uma ação representa a atividade do sistema que efetua a transição de estado.

Uma condição representa a causa necessária para que haja a transição de estado. Decorre da ocorrência de um evento ou circunstância que propicia a transição de estado. Assim, ao apertar o interruptor, provocamos a condição para que seja executada a ação de acender ou de apagar a lâmpada, provocando desta forma uma transição de estado.

Para representar graficamente o modelo de transição de estados, diversos tipos de diagramas têm sido usados. Nestes diagramas usam-se as seguintes representações:

- ❖ Retângulos significam estados;
- ❖ Setas significam transições válidas (mudanças de estados);
- ❖ Uma linha horizontal ao lado da seta que representa a transição, significa o par condição/ação, onde a condição aparece acima da linha mencionada e a ação desencadeada pela ocorrência da condição ou evento, aparece abaixo da linha.

### 1.1.2.6 Dicionário de dados

Todos os dados referenciados na construção do sistema deve ter sua definição no dicionário de dados. Para a construção do dicionário existem alguns padrões, nos quais é comum encontrar a convenção simbólica conforme a tabela abaixo.

<b>SÍMBOLO</b>	<b>SIGNIFICADO</b>
=	É composto de
+	E
()	Opcional (pode estar presente ou ausente)
{ }	Iteração (Repetição)
[ ]	Escolha uma das opções
**	Comentário
@	Atributo chave
/ ou	Separa alternativas na construção [ ]

Simbologia para dicionário de dados

Código\_da\_Obra = \* Identificador de uma obra \*  
Tipo: Numérico  
@Código\_da\_Obra

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

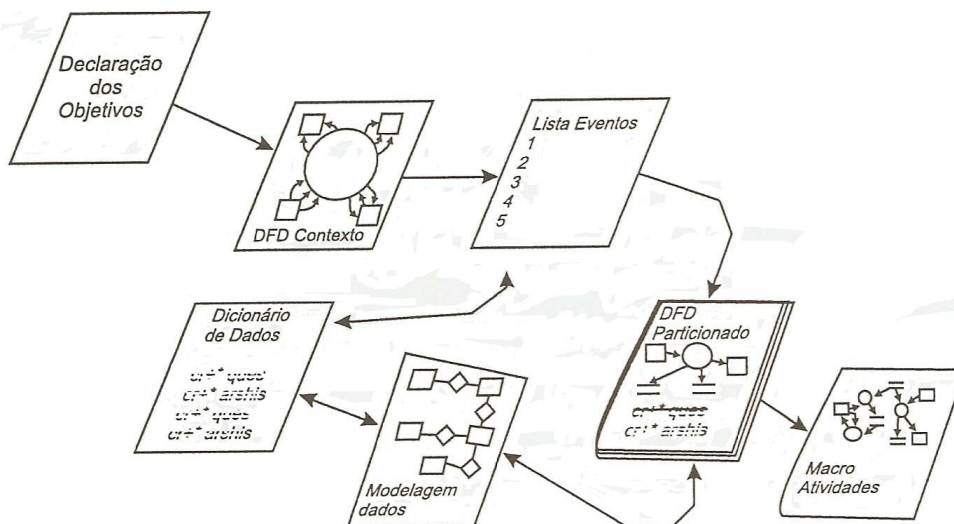
Cod\_Dept\_Func = \* Código do departamento do Funcionário \*  
Tipo: Numérico  
@Código\_Funcionário + @Código\_Departamento

Data\_Registro = \* Data em que ocorreu o evento \*  
Tipo: Data  
Formato: 99/99/9999

Telefone\_Biblioteca = \* Telefone da Biblioteca \*  
Tipo: Alfanumérico  
Tamanho: 14  
Formato: (99)9999-9999

Tipo\_Cliente = \* Descreve o tipo do cliente \*  
Tipo: Alfanumérico  
Tamanho: 08  
Conteúdo: [Física, Jurídica]

Na figura a seguir, tem-se a seqüência de trabalho e a relação entre os documentos que são gerados nas fases descritas anteriormente.



Documentos gerados na análise essencial (TONSIG, 2003).

## 1.2 Modelo de Implementação

Apresenta o sistema num grau de abstração completamente dependente de restrições tecnológicas. É derivado do modelo essencial. Diz respeito à implementação do sistema. Neste modelo, são colocadas todas as características tecnológicas, importando saber se parte de sua implementação vai ser manual ou automatizada, que tipo de hardware ou software vai ser usado. É o que os autores da análise estruturada chamam de modelo físico desejado para o sistema que ainda vai ser implementado.

### 1.2.1 Modelo Físico de Dados

Definições estão presentes na Parte 04 da disciplina AAES.

Acesse: [http://www.noginfo.com.br/arquivos/SI\\_AAES\\_Parte\\_04\\_PF.pdf](http://www.noginfo.com.br/arquivos/SI_AAES_Parte_04_PF.pdf)

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

**1.3 Referências Bibliográficas**

FILHO, WILSON DE PÁDUA PAULA, *Engenharia de Software*, Rio de Janeiro, Ed. LTC, 2003.

IEEE STD. 610 12-1990, *IEEE Standard Glossary of Software Engineering Terminology*, IEEE, Piscataway, NJ, 1997.

MINISTÉRIO DA CIÊNCIA E TECNOLOGIA, Secretaria de Política de Informática, *Qualidade e Produtividade no Setor de Software Brasileiro*, Brasília, N.4, 2002.

Pompilho, S, *Análise Essencial, Guia Prático de Análise de Sistemas*, Editora Ciência Moderna, Rio de Janeiro, 1995.

PRESSMAN, ROGER S., *Engenharia de Software*, Rio de Janeiro, Ed. McGraw-Hill, 2002.

SOMMERVILLE, IAN, *Engenharia de Software*, São Paulo, Ed. Pearson Education, 2003.

Tonsig, Sérgio Luiz, *Engenharia de Software, Análise e Projeto de Sistemas*, Editora Futura, São Paulo, 2003.