

Engenharia de Software

Capítulo 3

Engenharia de Sistemas Computacionais

1. INTRODUÇÃO

Embora este curso tenha por objetivo a discussão dos aspectos mais importantes relacionados ao desenvolvimento do software, é importante reconhecer o fato óbvio de que o software não consiste de um elemento autônomo. Para que suas funções possam ser realizadas, ele deverá ser integrado a outros componentes, especialmente elementos de hardware, como processador, circuitos de memória e outros.

Por esta razão, o software deve, antes de tudo, ser visto como um elemento de um sistema mais amplo, ao qual denominaremos um **Sistema Computacional**. No contexto deste curso, um Sistema Computacional deverá ser entendido como o resultado da união de diferentes componentes, entre os quais se enquadram o software, o hardware e elementos de outras naturezas como por exemplo o elemento humano, representado pelo usuário do sistema.

A **Engenharia de Sistemas Computacionais** corresponde ao conjunto de atividades que deverão ser realizadas para definir o sistema computacional como um todo. Dentre as atividades a serem conduzidas, pode-se citar a especificação, o projeto, a implementação, a validação, a instalação e a manutenção.

Cada uma das disciplinas componentes da Engenharia de Sistemas está relacionada a uma tentativa de estabelecimento de uma ordem no desenvolvimento de sistemas computacionais.

Nestes sistemas, o software vem, há muitos anos, substituindo o hardware como o elemento de mais difícil concepção, com menor probabilidade de sucesso em termos de prazos e custo e de mais difícil administração. Além disso, a demanda por software cresce a cada dia, até como consequência do grande desenvolvimento do hardware dos sistemas computacionais.

O objetivo deste capítulo é discutir as principais definições e aspectos relacionados com a Engenharia de Sistemas Computacionais, procurando definir o papel da Engenharia de Software neste contexto.

2. ASPECTOS DA ENGENHARIA DE SISTEMAS COMPUTACIONAIS

2.1. Sistema Computacional

Antes de discutir os principais pontos relativos à Engenharia de Sistemas Computacionais, é imprescindível estabelecer uma definição do que é um sistema computacional. Isto é particularmente importante, uma vez que a palavra "sistema" é uma das mais empregadas em qualquer área da atividade humana (sistema político, sistema educacional, sistema de avaliação, etc...).

Dentro do contexto do que será discutido aqui, pode-se definir um **Sistema Computacional** como sendo *um conjunto de elementos interrelacionados que operam juntos para o alcance de algum objetivo*.

Um sistema de processamento de texto, por exemplo, integra funções de entrada e edição de texto com funções de produção de documentos (software e hardware) manipuladas por um usuário (um elemento humano) para transformar um dado texto (uma entrada) num documento (a saída).

Pode-se listar os elementos de um sistema computacional como sendo:

- **software**, no caso, programas de computador, estruturas de dados, e documentação associada que serve a efetivar os métodos, procedimentos ou processo de controle lógico;
- **hardware**, que são os dispositivos eletrônicos que definem as capacidades de um computador e dispositivos eletromecânicos que oferecem funcionalidades ao ambiente externo;
- **pessoal**, usuários/operadores do hardware e do software;
- **bancos de dados**, uma coleção de informações organizada sistematicamente, acessada através do software;
- **documentação**, manuais, formulários e outros documentos que podem auxiliar no conhecimento do uso e operação do sistema;
- **procedimentos**, as regras que especificam o uso específico de cada elemento do sistema computacional.

As combinações destes elementos podem ser as mais diversas para permitir a transformação das informações. Um robô, por exemplo, é um sistema computacional que transforma um arquivo contendo instruções específicas num conjunto de movimentos e ações.

2.2. Composição dos Sistemas Computacionais e as Propriedades Emergentes

É importante reconhecer o papel das atividades relacionadas à Engenharia de Sistemas Computacionais como determinantes para as tarefas que se desenvolveram com base em seus resultados. Embora tenhamos dito que um Sistema Computacional é composto de diferentes elementos, não se deve negligenciar o fato de que a integração destes diferentes elementos é uma tarefa longe de ser trivial. O fato de se ter elementos concebidos com exemplar nível de qualidade não garante que a reunião destes vá originar um sistema excepcional.

Em tais sistemas, as dependências de um elemento do sistema com relação a outros podem se dar em muitos níveis. Por exemplo, o software só pode automatizar a solução de um problema se o processador estiver funcionando de forma adequada. Por outro lado, uma máquina de comando numérico só irá realizar corretamente o seu trabalho se o software de pilotagem tiver sido adequadamente concebido.

Entender o papel da Engenharia de Sistemas Computacionais é constatar que um sistema é algo mais do que a simples união de seus diversos elementos.

Num sistema de grande porte, existe um conjunto de propriedades que são verificados somente a partir do momento em que os diferentes elementos do mesmo são integrados. São as chamadas Propriedades Emergentes. Alguns exemplos destas propriedades são:

- o peso global do sistema, é um exemplo de propriedade que pode ser previamente determinada a partir de propriedades individuais dos componentes;
- a confiabilidade do sistema, que está fortemente dependente da confiabilidade de cada elemento;

- a facilidade de uso, que vai estar relacionada aos diferentes elementos do sistema como o hardware o software, mas também aos usuários do mesmo.

2.3. Sistemas e Subsistemas

Pode-se observar ainda que elementos de um determinado sistema podem operar de forma autônoma, constituindo-se num sistema independente. No entanto, ao serem incorporados ao sistema, suas funcionalidades vão estar dependendo de outros elementos deste. Uma câmera de vídeo, por exemplo, pode ser vista como um sistema independente. Por outro lado, quando ela está embutida num sistema de segurança automatizado, a maior parte de seus ajustes (enquadramento da imagem, foco, ativação e desativação) serão completamente dependentes de outros componentes do sistema de segurança.

Isto significa que, nos sistemas computacionais é comum um dado sistema compor um outro sistema computacional de nível hierárquico superior. Diz-se então que o sistema considerado é um **macroelemento** ou **subsistema** do sistema de nível superior.

Um exemplo típico desta situação é aquela dos equipamentos de manufatura, como por exemplo os *robôs*, *máquinas de comando numérico* e *controladores lógicos programáveis*, que são sistemas computacionais e ao mesmo tempo subsistemas de um sistema computacional de nível hierárquico superior: a *célula de manufatura*. A célula de manufatura pode ser definida como um sistema computacional, uma vez que esta é caracterizada por todos os elementos definidos acima. Se for feita uma análise considerando outros níveis hierárquicos, é possível visualizar a célula como um macroelemento de outro sistema computacional: o *sistema de manufatura*. A figura 3.1 ilustra esta relação entre os diferentes sistemas computacionais descritos.

2.4. Hierarquia e ambiente de um sistema

Reconhecido o fato de que um sistema pode ser obtido a partir de outros sistemas, como o exemplo ilustrado à figura 3.1, surge então o conceito de hierarquia de sistemas, que vai permitir utilizar a definição de subsistema, introduzida anteriormente.

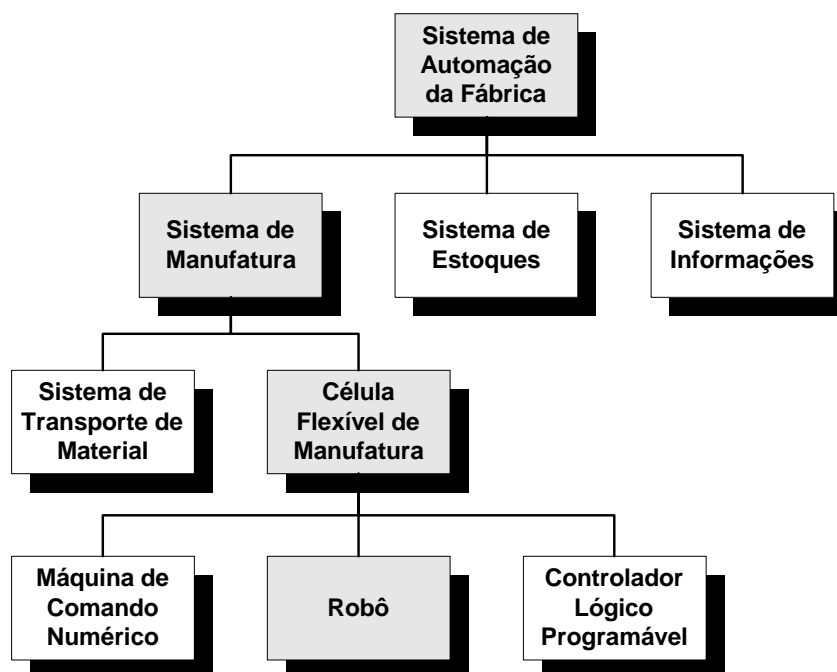


Figura 3.1 - Relação hierárquica entre sistemas computacionais.

Por esta razão, o ambiente de um sistema deve ser cuidadosamente estudado nesta etapa de concepção. A importância do entendimento completo do ambiente de um sistema é justificada pelos fatores seguintes:

- em grande parte das vezes, o sistema é concebido para provocar alterações no ambiente; sendo assim, a compreensão dos principais aspectos de funcionamento do ambiente serão fundamentais para especificar de que forma o sistema deve agir para provocar as alterações de forma precisa; exemplos de tais sistemas são um sistema de iluminação, um sistema de aquecimento (ou refrigeração) de ambiente;
- mesmo quando a função do sistema não está relacionada a alterações de seu ambiente, é importante se conhecer o quanto o ambiente pode influenciar no funcionamento do sistema; por exemplo, numa rede de computadores a ser instalada numa unidade fabricação de uma empresa, o fato de naquela unidade existir um grande número de motores e inversores operando pode ser determinante para decidir que tipos de protocolos e meios de transmissão deverão ser empregados, devido à alta incidência de fontes de interferência que podem causar erros de comunicação.

Um outro aspecto a ser considerado é o conjunto de requisitos originado no próprio ambiente. Os exemplos deste aspecto são os mais diversos, mas pode-se citar normas de qualidade ou segurança de produtos, requisitos de interface numa dada plataforma, requisitos de vedação, para um sistema computacional que irá operar num ambiente úmido ou sujeito a poeira.

Por todas estas razões, é fundamental que um sistema seja visto sempre como um componente de um sistema maior, o estudo aprofundado das características mais importantes relativas ao ambiente e dos requisitos impostos por este sendo uma tarefa essencial.

2.5. A Aquisição de Sistemas

No desenvolvimento de sistemas, principalmente daqueles mais complexos, é comum que determinados componentes sejam adquiridos em lugar de serem completamente desenvolvidos pela equipe ou empresa envolvida no projeto. Em alguns casos, o próprio sistema pode ser completamente adquirido num fabricante específico e, em outros, partes do sistema serão obtidas de um mesmo ou de diferentes fornecedores. Por outro lado, pode haver sistemas onde todos os componentes podem ser completamente desenvolvidos pela equipe.

O processo de decisão sobre a compra ou o desenvolvimento de um sistema ou de parte dele é uma tarefa que requer um esforço e um tempo considerável, principalmente se o grau de complexidade do mesmo for elevado. Uma decisão baseada numa análise superficial pode conduzir ao fracasso do projeto.

Normalmente, para servir de referência à decisão, uma especificação do sistema e um projeto arquitetural devem ser conduzidos. Os resultados obtidos nesta tarefa são importantes pelas seguintes razões:

- para que o processo de aquisição seja encaminhado com sucesso, é necessária a existência de uma especificação e de uma visão arquitetural do sistema;
- o custo de aquisição de um sistema é, na maioria dos casos, menor que do desenvolvimento deste; por esta razão, o projeto arquitetural é importante como forma de decidir quais subsistemas serão adquiridos e quais serão desenvolvidos.

2.6. A SUBCONTRATAÇÃO

As situações em que uma empresa desenvolve completamente todos os componentes de um sistema são extremamente raras. Geralmente, a organização "usuária" vai encomendar o sistema a uma organização "fornecedora". A organização fornecedora, eventualmente, vai "terceirizar" o desenvolvimento do sistema, contratando outras empresas fornecedoras dos diferentes subsistemas. A figura 3.2 ilustra este processo.

Um aspecto interessante deste processo é a otimização, no que diz respeito aos contatos, entre a organização do usuário e as organizações desenvolvedoras. Na realidade, o único contato mantido pela organização usuária é com a principal contratante. As subcontratantes vão projetar e construir os subsistemas cujos requisitos foram especificados pela principal contratante que exerce o papel de aglutinadora de todas as empresas envolvidas no desenvolvimento.

2.7. A IMPORTÂNCIA DO SOFTWARE

No desenvolvimento de sistemas computacionais, é comum a utilização de componentes customizados (construídos sob medida) e componentes de "prateleira". Por esta razão, o componente de software assume um papel bastante importante no desenvolvimento do sistema, isto porque é este que pode assumir a função de elo de ligação entre os diferentes componentes, permitindo a adequação do funcionamento dos diferentes componentes de hardware (principalmente aqueles de "prateleira") aos requisitos do sistema.

Um exemplo deste fato está num dos sistemas computacionais mais utilizados nos dias atuais — os microcomputadores da linha PC. O PC da IBM, quando foi concebido, foi totalmente construído a partir de componentes de hardware disponíveis no mercado. Esta decisão foi tomada devido à urgência em ocupar o espaço do mercado no setor de computadores pessoais. Entretanto, a fórmula mágica da IBM para fazer com que todos estes elementos funcionassem harmonicamente foi buscada no software, no caso o ROM-BIOS, que continha todas as rotinas de programação e acesso aos dispositivos de hardware que compunham a arquitetura do PC.

Só a partir do momento em que as outras empresas conseguiram copiar o conteúdo do ROM-BIOS é que surgiram os conhecidos "clones" do PC que difundiram a tecnologia pelo mundo todo.

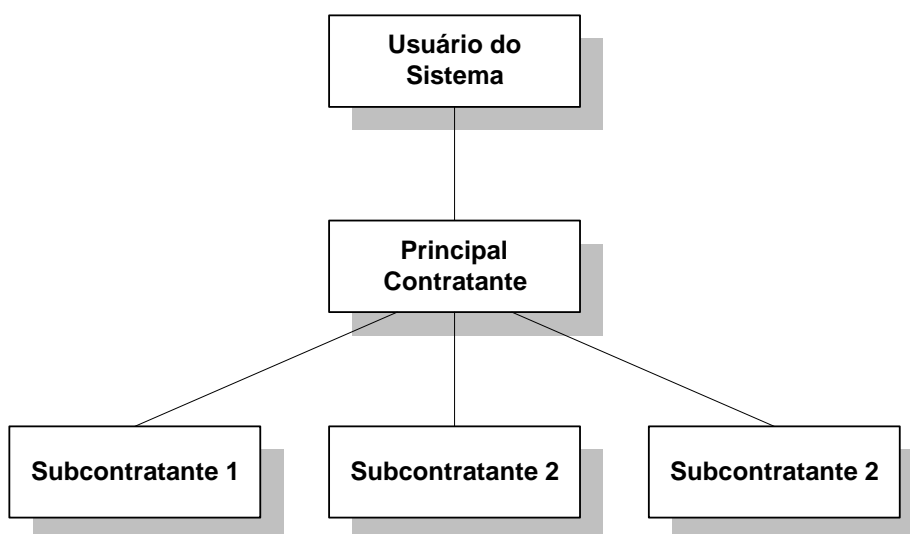


Figura 3.2 - Ilustração do processo de aquisição de um sistema.

3. ENGENHARIA DE SISTEMAS COMPUTACIONAIS

O papel essencial da Engenharia de Sistemas Computacionais é a resolução de problemas, onde a atividade de base é a identificação, análise e atribuição aos elementos constituintes do sistema das funções desejadas para o sistema.

A palavra que melhor define a Engenharia de Sistemas é a **interdisciplinaridade**, uma vez que, na maior parte dos sistemas a serem concebidos, devem entrar em jogo diferentes competências. Além dos Engenheiros de Hardware e de Software, devem intervir nesta etapa especialistas como Engenheiros Eletricistas, Engenheiros Hidráulicos, Engenheiros de Telecomunicações, Químicos, Físicos, Psicólogos, Médicos, etc.

O ponto de partida do trabalho do Engenheiro de Sistemas são os requisitos e restrições formuladas pelo cliente, que vão permitir um primeiro delineamento das funções a serem executadas pelo sistema, de requisitos de desempenho, das interfaces a serem oferecidas, das restrições de projeto e das estruturas de informação que serão processadas por cada um dos elementos constituintes do sistema (software, hardware, pessoal, etc...).

Neste trabalho, é importante que se estabeleça da forma mais precisa possível as funções a serem realizadas. Por exemplo, no caso do sistema de controle de um robô, não é suficiente dizer que *"ele deve reagir com rapidez se a bandeja de ferramentas estiver vazia"*... na realidade, é importante especificar cuidadosamente os diferentes aspectos deste requisito:

- o que vai indicar a bandeja vazia para o robô;
- os limites de tempo relacionados com a reação;
- de que forma deve ser esta reação.

Uma vez identificados e analisadas todas as funções, requisitos e restrições, passa-se à fase de **alocação**, onde o objetivo é atribuir, aos diferentes elementos que constituem o sistema, as funções analisadas anteriormente.

Nesta tarefa, é comum estabelecer alternativas de alocação para posterior definição. Consideremos, por exemplo, um sistema de tratamento gráfico, onde uma das funções principais é a transformação tridimensional de imagens. Deste exemplo, é possível imaginar algumas opções de soluções existentes:

- ① todas as transformações serão resolvidas por software;
- ② as transformações mais simples (redimensionamento e translação) serão realizadas em hardware e as mais complexas (rotação, perspectivas, etc...) em software;
- ③ todas as transformações serão realizadas por um processador geométrico implementado em hardware.

A escolha da solução deve ser baseada num conjunto de critérios, onde entram fatores como custo, realizabilidade, desempenho, padronização de interfaces, portabilidade, etc.

Nas seções a seguir, serão descritas atividades encaminhadas no contexto da Engenharia de Sistemas Computacionais.

3.1. Definição dos Requisitos

O objetivo das tarefas a serem desenvolvidas nesta etapa é identificar os requisitos do Sistema Computacional a ser concebido. Considerando que grande parte dos sistemas computacionais são construídos sob demanda, a forma usual de encaminhar esta etapa é caracterizada por um conjunto de reuniões estabelecidas entre a equipe de desenvolvimento do sistema e os clientes ou usuários.

Nesta etapa, pode-se considerar basicamente três categorias de requisitos:

- os **requisitos básicos**, que estão associados às funções a serem desempenhadas pelo sistema; neste momento do desenvolvimento, estes requisitos são definidos num nível de abstração relativamente elevado, uma vez que o detalhamento será necessariamente realizado numa etapa posterior;
- as **propriedades do sistema**, que estão relacionadas às propriedades emergentes do sistema; desempenho, segurança, disponibilidade, são alguns exemplos desta categoria de requisitos;
- as **características indesejáveis**, que são propriedades que o sistema não deve possuir; é tão importante, muitas vezes, definir que características não devem ser encontradas num dado sistema, quanto definir aquelas que são consideradas essenciais; num sistema de comunicação de dados, por exemplo, algumas características indesejáveis são os erros, as perdas e as duplicações de mensagens.

De forma geral, é importante que sejam estabelecidos, da forma mais completa possível, os objetivos e as funções a serem providas pelo sistema. Neste nível, deve-se enfatizar o papel do sistema computacional no ambiente em que ele será instalado, podendo-se deixar para etapas mais à frente, a especificação dos aspectos funcionais do mesmo.

Observemos a distinção através de um exemplo, onde são apresentadas duas versões de especificação dos requisitos de um sistema de segurança para um edifício comercial:

- (a) *O objetivo do sistema é prover um sistema de proteção contra incêndio e intrusão que vai ativar um alarme interno e externo em caso de princípio de incêndio ou invasão por pessoas não autorizadas;*
- (b) *O objetivo do sistema é assegurar a continuidade do funcionamento normal conduzido no edifício, mesmo na ocorrência de eventos como o princípio de incêndio ou intrusão.*

Neste nível do desenvolvimento do sistema, a segunda versão apresentada é a mais adequada, uma vez que ela é, por um lado, mais abrangente e, por outro, limitante em alguns aspectos. A forma como está estabelecido o objetivo do sistema está mais compatível com as necessidades do ambiente, e dá uma maior abertura à adoção de técnicas mais evoluídas de prevenção de incêndio e intrusão. Por outro lado, o fato de estabelecer como requisitos a garantia de continuidade das atividades realizadas no escritório, elimina algumas soluções como o uso de alarme interno ou irrigadores contra incêndio.

3.2. O PROJETO DO SISTEMA

Os principais passos envolvidos no projeto de um sistema computacional estão esquematizados na figura 3.3 e serão descritos nos parágrafos que seguem.

Particionamento dos Requisitos. Uma vez estabelecidos os requisitos do sistema na etapa anterior, o objetivo deste primeiro passo de projeto é particioná-los segundo algum critério lógico. A escolha do critério de particionamento é importante pois deste vai depender as diferentes categorias de requisitos que serão criadas.

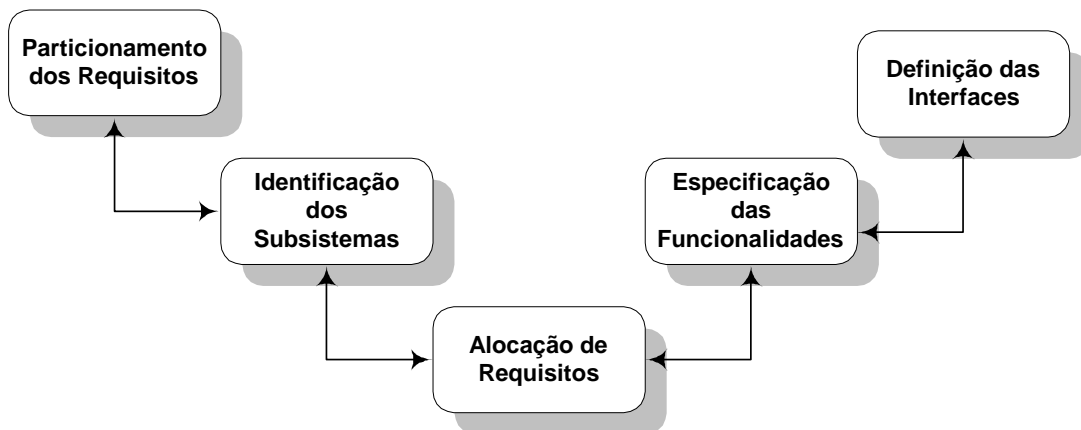


Figura 3.3 - Passos a serem conduzidos no projeto de um sistema computacional.

Identificação dos Subsistemas. Este passo busca definir os diferentes subsistemas que irão compor o sistema computacional. O particionamento realizado no passo anterior será determinante para a obtenção dos subsistemas, uma vez que, dependendo do critério utilizado, cada grupo de requisitos poderá definir um subsistema. Por outro lado, outros aspectos podem ser considerados para a definição dos subsistemas, particularmente aqueles relacionados a questões do ambiente onde o sistema será instalado.

Alocação dos Requisitos. O próximo passo a ser conduzido é a alocação dos requisitos aos subsistemas estabelecidos no passo anterior. Pode parecer estranho considerar tal etapa, quando se utiliza o resultado obtido no Particionamento dos Requisitos para realizar a Identificação dos Subsistemas, mas, na verdade, isto não ocorre de forma tão imediata. Por exemplo, no caso da adoção de sistemas de prateleira para implementar algum subsistema, as limitações do sistema adotado podem impor a realocação de requisitos.

Especificação das Funcionalidades. Neste passo, serão definidas as funções que serão implementadas por cada subsistema. No caso de um subsistema de software, esta atividade pode ser encaminhada como parte da análise de requisitos do mesmo.

Definição das Interfaces dos Subsistemas. Este é um passo de extrema importância na condução do projeto do sistema. A definição, de forma precisa das interfaces entre os subsistemas é fundamental para as etapas posteriores de integração dos diferentes componentes. Não se deve negligenciar o fato de que muitos subsistemas podem ser adquiridos em lugar de serem desenvolvidos e mesmo aqueles que são especialmente construídos no contexto de um projeto, usualmente são desenvolvidos por diferentes equipes ou por diferentes empresas. Qualquer má definição neste item pode comprometer o sucesso do projeto ou implicar num aumento de custo ou de prazo de desenvolvimento.

As linhas contendo setas nas duas direções entre cada par de passos sugere a necessidade de revisões e realimentações em cada passo. Na maioria dos projetos, o número de soluções possíveis é bastante grande, podendo-se chegar a diferentes visões de como as funções são alocadas a subsistemas de hardware, de software ou aos elementos humanos que vão interagir com o sistema.

A escolha da solução a ser encaminhada vai ser feita em função de critérios técnicos, mas também terá influência de critérios políticos e organizacionais da instituição. No caso de projetos desenvolvidos para instituições militares, por exemplo, fornecedores nacionais serão escolhidos em lugar de fornecedores estrangeiros, mesmo que em termos técnicos, a solução oferecida não seja a ideal.

3.3. Desenvolvimento dos Subsistemas

Esta etapa agrupa as atividades relacionadas ao desenvolvimento de cada subsistema definido no projeto. Cada subsistema terá suas funções bem definidas e suas especificidades quanto às competências necessárias ao seu desenvolvimento. No caso de um subsistema de software, um processo de desenvolvimento caracterizado pelas etapas clássicas de desenvolvimento será iniciado.

Em raras situações, o desenvolvimento de um sistema impõe a construção, a partir do zero, de todos os subsistemas que o compõem. O mais comum é que alguns subsistemas sejam adquiridos e incorporados ao sistema. Embora seja o caso mais comum, nem sempre é fácil integrar subsistemas "de prateleira" a um sistema; em muitos casos, são necessárias modificações no sentido de adaptar o subsistema adquirido às necessidades do sistema a ser desenvolvido. Além disso, nem sempre é possível prever a disponibilidade de um subsistema de prateleira no momento em que ele é necessário.

A **Engenharia de Hardware**, para efeito de definição de um dado sistema, consiste no trabalho de seleção dos componentes que vão compor o hardware do sistema e que serão capazes de assumir as funções atribuídas a este elemento. Atualmente, o trabalho de seleção dos componentes de hardware é mais simples do que o de software, sendo facilitado pelos seguintes fatores:

- os componentes são montados como blocos de construção individuais;
- as interfaces entre os componentes são padronizadas;
- existem numerosas alternativas "de prateleira" à disposição;
- aspectos como o desempenho, custo e disponibilidade são de fácil obtenção.

A Engenharia de Hardware é baseada na execução de três principais fases:

- o **Planejamento e a Especificação**, cujo objetivo é estabelecer a dimensão do esforço necessário ao desenvolvimento, assim como o estabelecimento de um roteiro para o projeto e implementação do hardware; ainda nesta fase, é conduzida a análise de requisitos, cujo resultado deve ser uma especificação a mais completa possível das funções e outros requisitos para o hardware e das suas restrições;
- o **Projeto e Prototipação**, a qual consiste na fase que, com base nos requisitos e restrições especificadas na fase anterior, define uma configuração preliminar do hardware; atualmente, as tarefas relativas a esta fase são semi ou completamente automatizadas com o auxílio de ferramentas de software CAE/CAD; o resultado final desta fase é um protótipo montado que será testado para garantir a satisfação de todos os requisitos;
- a **Produção, Distribuição e Instalação**, onde o protótipo obtido na fase anterior vai sofrer as evoluções para tornar-se verdadeiramente um produto de hardware; alterações na embalagem, interfaces, componentes, etc..., são então realizadas; definição de métodos de controle de qualidade é um ponto fundamental nesta fase; a criação de um estoque de peças de reposição é uma outra preocupação conduzida neste momento, assim como a definição de esquemas de instalação e manutenção.

Um dos resultados da Engenharia de Sistemas é a definição de aspectos como funcionalidade e desempenho do software. O trabalho essencial do engenheiro de software é acomodar os requisitos de funcionalidade e desempenho da forma mais eficiente possível, tendo que, para isto, adquirir e/ou desenvolver os componentes de software. A dificuldade que surge na Engenharia de Software é a falta de padronização nos componentes de

software (o que não ocorre no caso do hardware), uma vez que, na maior parte dos projetos, estes componentes são "customizados" para atender às exigências de software do sistema a ser desenvolvido.

Como já foi discutido anteriormente, a Engenharia de Software envolve as seguintes fases:

- **Definição**, que é iniciada com o Planejamento do Desenvolvimento do Software, englobando todas as tarefas discutidas no capítulo III do curso, obtendo um documento de Plano do Software, o qual será produzido e revisado pelo Gerente do Projeto; ainda nesta fase é realizada a Análise de Requisitos do Software, a qual vai permitir que funções, dentro do sistema como um todo serão atribuídas ao software; a última tarefa relacionada a esta fase é a revisão da Especificação de Requisitos do Software, o qual é o documento resultante da Análise de Requisitos;
- **Desenvolvimento do Software**, que é a fase que tem, como ponto de partida, os documentos produzidos na fase anterior, particularmente, o Plano do Software e a Especificação de Requisitos do Software; com base nestes documentos, inicia-se a etapa de Projeto do Software, onde serão descritos aspectos relacionados ao funcionamento do software como a sua arquitetura e as estruturas de dados; após avaliada esta definição, inicia-se a etapa de Projeto Detalhado, onde os aspectos algorítmicos e comportamentais do software são definidos; finalmente, a etapa de codificação é encaminhada, seja com base no uso de uma linguagem de programação clássica ou com o auxílio de uma ferramenta CASE, o resultado desta etapa sendo a listagem dos programas-fonte do software em desenvolvimento;
- **Verificação, Entrega e Manutenção**, é a última fase do processo, a qual envolve as atividades de teste do software, preparando-o para a entrega; uma vez entregue, inicia-se, ao longo de toda a vida útil do software a etapa de manutenção, a qual permitirá manter o software em funcionamento a partir da correção de novos erros que tenham sido detectados com o software em funcionamento, da introdução ou melhorias de funções do software, da adaptação do software para novas plataformas de hardware existentes.

Por razões óbvias, os diferentes subsistemas definidos na etapa de projeto são desenvolvidos em paralelo. No caso de ocorrência de algum problema relacionado à interface dos diferentes subsistemas, uma solicitação de modificação no sistema pode ser necessária. Em sistemas envolvendo a utilização de muitos subsistemas de hardware, a realização de modificações após o término da implementação podem representar um aumento considerável no custo de desenvolvimento. Uma forma bastante utilizada nos projetos, neste caso, é utilizar os subsistemas de software para facilitar estas modificações, devido, particularmente, à flexibilidade inerente dos sistemas de software. É lógico que, para que as modificações nos subsistemas de software não representem custos tão elevados ou maiores quanto o dos subsistemas de hardware, é necessário que estes subsistemas tenham sido concebidos tendo como princípio sua manutenibilidade.

3.4. Integração do Sistema

O conjunto de atividades a ser desenvolvido nesta etapa é o de conexão dos diferentes subsistemas construídos ou adquiridos para compor o sistema. É uma atividade bastante complexa, devido principalmente, à grande diversidade de tecnologias envolvidas na concepção dos diferentes sistemas.

Um problema comumente encontrado nesta etapa é o mal funcionamento de um subsistema como consequência de uma definição imprecisa de funcionalidade de outro subsistema.

Uma forma de realizar a integração de um sistema é o processo "big-bang", onde todos os seus subsistemas são conectados num passo único. Entretanto, por razões técnicas e gerenciais, a forma mais adequada é a integração incremental dos diferentes subsistemas. As principais razões para adoção desta estratégia são:

- 1) *Na maior parte dos casos, é impossível sincronizar o fim do desenvolvimento de todos os subsistemas;*
- 2) *A integração incremental reduz os custos de identificação e correção de erros de integração. Quando um número muito grande de subsistemas são integrados simultaneamente, uma falha no sistema pode ser consequência de um ou mais erros presentes nos diferentes subsistemas. No caso da integração incremental, a ocorrência de uma falha pode ser mais facilmente controlada, pois ela será, muito provavelmente, consequência de um erro no subsistema mais recentemente integrado.*

No caso de sistemas em que os subsistemas foram construídos por diferentes fornecedores, é muito comum ocorrer desentendimentos quando um problema de integração acontece. Os fornecedores tendem a acusar um ao outro como o responsável do problema detectado. Este tipo de ocorrência é bastante prejudicial ao desenvolvimento do sistema, podendo tomar muito tempo para que o problema seja resolvido.

3.5. Instalação do Sistema

Esta etapa envolve todas as atividades relacionadas à colocação do sistema em funcionamento no ambiente para o qual ele foi concebido. Embora possa parecer um problema menor, a instalação de um sistema pode ser caracterizada pela ocorrência de muitos problemas que coloquem em risco o cumprimento de estimativas de prazo ou de custo.

Alguns problemas típicos desta etapa são:

- O ambiente para o qual o sistema foi concebido não é precisamente o mesmo que foi especificado no início do projeto; este é um problema bastante comum em sistemas de software, particularmente no que diz respeito à utilização de certas facilidades do sistema operacional; quando a versão do sistema operacional do ambiente não é a mesma que foi utilizada para o desenvolvimento do sistema, o resultado pode ser catastrófico;
- A resistência dos usuários à implantação do novo sistema é um outro problema importante; nem sempre a adoção de um sistema computacional para a realização de uma tarefa que vinha sendo realizada de outra forma é bem vista pelos funcionários, pois este vai, na quase totalidade dos casos, na necessidade de treinamento ou de alteração na sistemática de trabalho; um exemplo bastante óbvio é o da utilização de sistemas de software para a declaração de imposto de renda; quando os primeiros programas foram lançados, muitas pessoas que tinham computadores ainda insistiam em utilizar os formulários impressos para realizar suas declarações;
- A necessidade de coexistência entre um sistema antigo e o novo sistema é outra fonte de problemas de instalação. No caso dos sistemas compartilharem algum recurso, pode ser impossível instalar completamente o novo sistema sem desativar o antigo;
- Obstáculos físicos à instalação do sistema são problemas muito comuns nesta etapa; ausência de pontos de energia ou tubulação apropriada, falta de sistema

de ar-condicionado, são exemplos de obstáculos comumente encontrados na instalação de sistemas computacionais.

3.6. Ativação do Sistema

Uma vez instalado, o sistema deverá ser ativado. Uma etapa de apresentação e treinamento relativo à utilização pode estar associada à ativação do sistema. Os problemas que podem ocorrer nesta etapa são os mais diversos. No caso de sistemas que deverão operar conjuntamente a outros sistemas existentes, problemas de incompatibilidade podem ocorrer e a dificuldade de resolução destes problemas pode provocar grandes atrasos à sua colocação em operação.

Outra fonte de problemas pode ser a inadequação ou a má utilização das interfaces de operador do novo sistema. Muitos erros de operação poderão ser cometidos até que o operador crie intimidade com o novo sistema.

3.7. Evolução do Sistema

Os grandes sistemas computacionais são desenvolvidos para funcionar durante longos períodos tempo. É inevitável que, durante este período, ocorram problemas que imponham modificações no sistema. As razões que podem conduzir a esta necessidade podem ser erros do próprio sistema, novas necessidades em termos de função ou alterações ambientais.

As evoluções a serem promovidas num dado sistema podem representar um custo bastante alto, o que é justificado pelas razões abaixo:

- Necessidade de estudos relativos às modificações a serem feitas para não comprometer o bom funcionamento do sistema;
- O alto grau de dependência dos diversos subsistemas pode conduzir à necessidade de alterações em muitos subsistemas, como consequência da alteração de um componente;
- A falta de documentação relativa ao desenvolvimento do sistema original vai ser, sem dúvida, um grande obstáculo para o desenvolvimento do sistema;
- À medida que o tempo passa e que alterações vão sendo realizadas, a estrutura original dos sistemas vai sendo modificada, de modo que futuras mudanças podem ser fortemente comprometidas.

3.8. Desativação do Sistema

Esta etapa consiste em retirar o sistema de operação, quando seu tempo previsto de funcionamento esgota ou quando ele será substituído por um novo sistema. A desativação deve ser feita de forma bastante rigorosa, principalmente no caso de sistemas cujos componentes podem ser nocivos ao ambiente. Sistemas que utilizam componentes radioativos são os exemplos mais comuns de sistemas cuja desativação deve ser feita com o maior cuidado possível.

Por outro lado, no caso dos sistemas de software, a desativação pode ser extremamente simples. A maior parte dos sistemas de software construídos atualmente já vem dotada de um utilitário de desativação (ou "desinstalação") que retira automaticamente todos os componentes do sistema e as referências a estes componentes do sistema computacional.

Outro aspecto importante da desativação dos sistemas são as informações que este manipulava ou gerava. Esta, normalmente, deverão ser armazenadas em mídia específica para posterior adaptação à utilização num novo sistema.

4. PROJETO ARQUITETURAL

4.1. O Modelo do Sistema

Um importante resultado da especificação de requisitos e do projeto de um sistema computacional é a sua organização em termos de subsistemas e dos relacionamentos entre estes. A forma mais usual de representar esta organização é, sem dúvida, os modelos baseados em linguagem gráfica.

Invariavelmente, a utilização de diagramas em blocos destacando os principais subsistemas e como estes estão relacionados é bastante aceita nos projetos de sistemas em geral, incluindo os sistemas computacionais. A figura 3.4 ilustra tal mecanismo, através do exemplo de projeto arquitetural de um sistema de controle de tráfego aéreo. O objetivo aqui não é descrever o funcionamento de um tal sistema, mas mostrar como se pode representar os diferentes componentes deste e como estes relacionam-se. As setas ligando os diferentes subsistemas indicam os fluxos de informação que existem no sistema.

No caso de sistemas relativamente complexos, é comum que os subsistemas definidos neste primeiro nível sejam, por si só, sistemas complexos. Isto significa que, para se ter uma idéia mais completa de como o sistema deve operar, os subsistemas podem, eventualmente, ser representados utilizando a mesma linguagem.

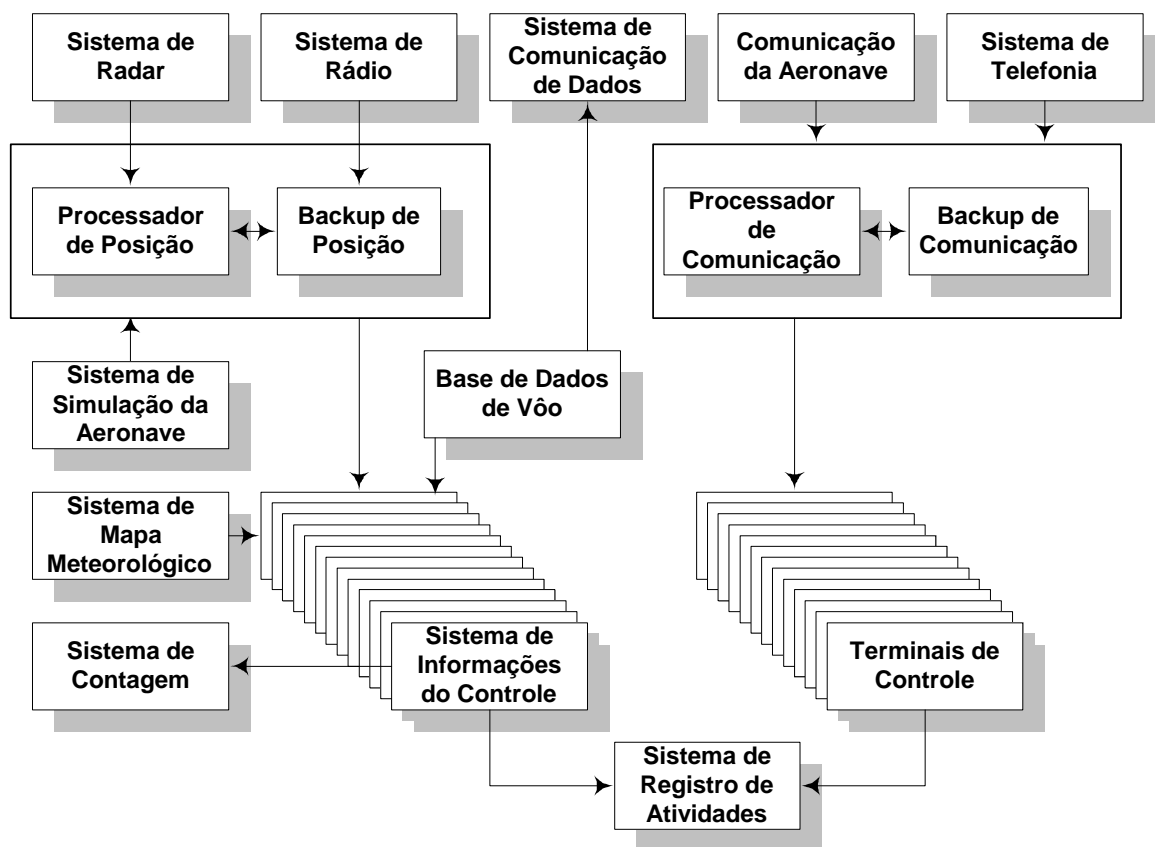


Figura 3.4 - Exemplo de projeto arquitetural: um sistema de controle de tráfego aéreo.

Num sistema computacional, os diagramas de arquitetura destacam, na forma de subsistemas, os elementos que deverão ser desenvolvidos em hardware e em software, permitindo o encaminhamento das atividades relativas às diferentes engenharias, normalmente conduzidas em paralelo.

Na maior parte dos sistemas atuais, é comum que os componentes de hardware incluam um ou mais computadores e o software associado. Isto se justifica pela queda constante nos preços de equipamentos computacionais, permitindo a substituição de outras alternativas pelos sistemas computacionais, além das vantagens de tais equipamentos, particularmente para tarefas de controle e supervisão.

No nível de projeto arquitetural, nem sempre é importante definir que funções serão executadas por software ou por hardware. O mais importante é definir subsistemas com base nas funções a serem executadas pelo sistema. A definição sobre a forma como cada função será implementada (em software ou em hardware) pode ser feita numa etapa posterior, levando em conta não apenas fatores técnicos. Por exemplo, a existência ou não de um componente de hardware que implemente adequadamente uma dada função pode ser determinante para definir se esta função deverá ser implementada em hardware ou software.

4.2. Componentes funcionais

Durante a realização das diferentes etapas que caracterizam o desenvolvimento de um sistema computacional, particularmente na etapa de definição dos subsistemas, diferentes categorias de componentes podem ser identificadas. Abaixo serão descritas aquelas mais freqüentemente encontradas nos sistemas:

Sensores. São os elementos capazes de coletar informações a respeito do ambiente onde o sistema está instalado. Exemplos deste tipo de componente são os radares do sistema de controle de tráfego aéreo ou os sensores de posicionamento de papel numa impressora a laser ou jato de tinta.

Atuadores. Estes componentes tem por função realizar ações no sistema ou em seu ambiente. Válvulas para abertura ou fechamento de fluxo hidráulico, os flaps das asas do avião, o mecanismo de alimentação de papel numa impressora, são exemplos desta classe de componentes.

Componentes computacionais. São os elementos capazes de exercer algum processamento de uma dada informação de entrada para gerar um conjunto de informações de saída. Alguns exemplos são processadores gráficos, processadores aritméticos, um elemento que implemente um algoritmo de controle, etc.

Componentes de comunicação. São os componentes que irão viabilizar a comunicação entre partes de um sistema ou entre o sistema e seu ambiente. Uma placa Ethernet, uma porta serial, um modem, são instâncias de tal classe.

Componentes de Coordenação. São os componentes que executam as funções de coordenação do sistema. Um algoritmo de escalonamento num sistema tempo-real é um bom exemplo deste tipo de componente.

Componentes de interface. São os componentes capazes de adaptar uma forma de representação de informação na forma de representação utilizada pelo componente que vai processá-la. Exemplos deste tipo de componente são uma placa de vídeo, um conversor analógico-digital, etc.

A figura 3.5 mostra, através de um sistema de segurança doméstica, como podem ser classificados os diferentes componentes, o que é explicitado na tabela a seguir.

Classe	Componente	Função
Sensor	Sensor de movimento e da porta	Detectar intrusão

Atuador	Sirene	Gerar um sinal de alarme
Comunicação	Ativador de chamada telefônica	Realizar chamada a um sistema externo
Coordenação	Controlador do alarme	Coordenar os demais componentes
Interface	Sintetizador de voz	Gerar mensagem de localização

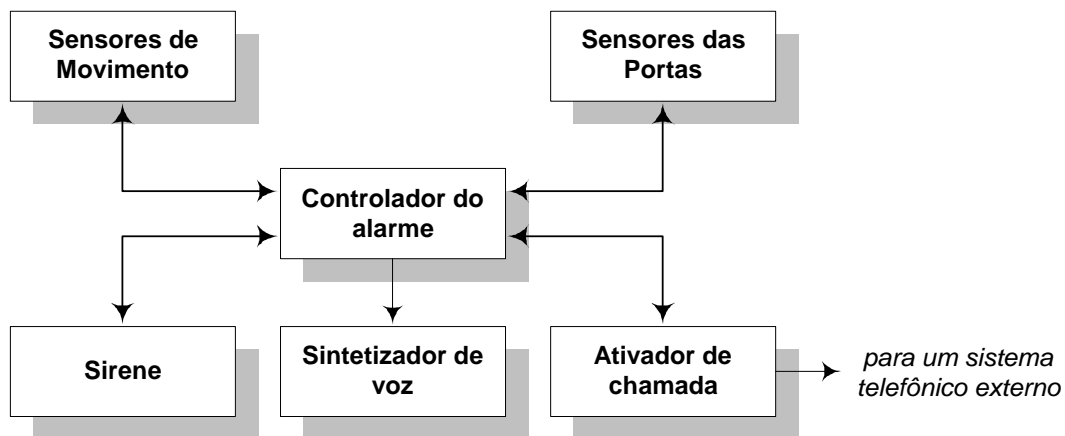


Figura 3.5 - Sistema de segurança doméstica.