

# Capítulo 2

## Engenharia de Sistemas

Este capítulo tem por objetivo estudar e compreender o significado de um sistema, como o processo de engenharia afeta o processo de software e como a complexidade de um sistema de larga escala pode ser diminuída quando compartimentalizamos um sistema em seus subsistemas. Para isto abordaremos os seguintes tópicos:

- Apresentação dos conceitos mais amplos de engenharia de sistemas.
- Ilustração dos problemas inerentes ao desenvolvimento de sistemas computacionais os quais envolvem hardware, software e pessoas.
- Explicar porque os sistemas precisam ter um bom relacionamento com o ambiente em que estão instalados.
- Explicar o processo de obtenção de sistemas.

O objetivo deste capítulo é tornar o leitor apto a estruturar sistemas computacionais como parte de sistemas de engenharia em geral, entretanto esta é uma habilidade que só se desenvolve com a experiência. Portanto, pratique!

### 2.1 Os sistemas e seus ambientes

Um sistema é um conjunto de componentes interrelacionados que trabalham juntos para atingir um objetivo. Um sistema pode incluir software, partes mecânicas, elétricas e eletrônicas, e ainda ser operado por pessoas.

Um componente de um sistema é sempre dependente de outros componentes do mesmo sistema. Um componente que trabalha de forma independente não faz parte daquele sistema. Do mesmo modo o sucesso do trabalho de um componente no sistema é dependente do sucesso de outros componentes.

Esta complexidade do sistema revela que um sistema é mais do que a soma de suas partes, ou seja, um sistema possui uma série de **propriedades emergentes** que são aquelas propriedades que não são indetificadas em nenhum dos seus componentes mas

somente no sistema como um todo. Exemplos de propriedades emergentes em um sistemas são o tamanho, a confiabilidade e a usabilidade.

### 2.1.1 A temática da engenharia de sistemas

A engenharia de sistemas é uma área de pesquisa aplicada que busca abordar o problema do desenvolvimento de sistemas de tecnologia dentro de metas, prazos e custos estabelecidos. Esta área lida basicamente com três principais problemas:

- Grandes sistemas são em geral desenvolvidos para resolver problemas muito imprecisos, ou suficientemente grandes de forma que sua especificação formal se torne inviável.
- A engenharia de sistemas requer uma grande interdisciplinaridade das pessoas envolvidas. As alternativas para se resolver um sub-problema em um grande sistema são infinitas e a falta de versatilidade dos desenvolvedores entre as diversas áreas é normalmente um problema de difícil abordagem.
- Sistemas são criados para durarem anos adaptando-se sempre as mudanças no ambiente.

A quantidade de software utilizada em sistemas é cada vez maior. Cada vez mais, sistemas eletrônicos baseados em software substituem com vantagens os sistemas de hardware de propósito específico.

Os problemas da engenharia de software, entretanto, são muito similares aos problemas da engenharia de sistemas. Contudo o software é (infelizmente) visto como um grande problema na engenharia de sistemas pois muitos projetos de larga escala são abandonados devido ao insucesso na produção do software.

### 2.1.2 O sistema e seu ambiente

Sistemas não são entidades isoladas, eles existem dentro de um ambiente o qual este sistema pode agir, modificando-o. Do mesmo modo o ambiente afeta o funcionamento geral do sistema. A consideração do ambiente organizacional e mesmo físico do sistema é importante para o seu desenvolvimento.

Todo o sistema possui subsistemas do mesmo modo que todo sistema é um subsistema de um sistema maior, ou seja, todo o sistema pertence a uma hierarquia de sistemas. O ambiente de sistema é por sua vez composto por outros sistemas que interagem com este afetando-o e/ou governando o seu funcionamento. Em suma: é importante compreender que **o conceito de sistema é um conceito recursivo**, isto é, um sistema é sempre parte de outro(s) sistema(s), do mesmo modo que é composto por um ou mais outros sistemas.

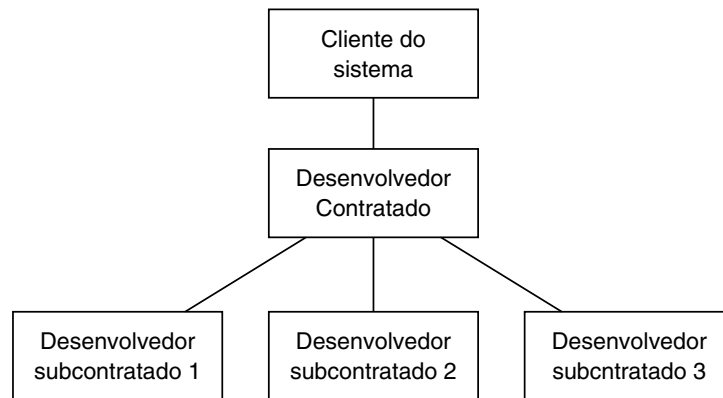


Figura 2.1: Modelo de contrato e sub-contrato

## 2.2 A obtenção de um sistema

A obtenção de um sistema consistem na aquisição de um produto que satisfaça algumas necessidades de uma organização. Algumas especificações do sistema e definição de métricas para o projeto arquitetural são normalmente necessárias antes do sistema ser obtido. Isto é importante para que se possa definir as restrições do contrato de desenvolvimento. A especificação à priori pode possibilitar que os desenvolvedores adquiram componentes pré prontos (off-the-shelf, COTS) que são quase sempre mais baratos se comparados com o desenvolvimento personalizado.

A obtenção de um grande sistema de software e/ou hardware é em geral posta sob responsabilidade de um desenvolvedor (pessoa física ou jurídica) contratado. Este pode trabalhar com sub-contratos, delegando a outros indivíduos ou empresas o fornecimento de partes do sistema. Normalmente um cliente só lida com o contratado principal e não com os seus fornecedores sub-contratados. Veja na Figura 2.1 o modelo de contrato e sub-contrato.

A obtenção de um sistema é um processo que pode passa por algumas fases distintas. Primeiramente é necessário decidir-se se o sistema será desenvolvido ou comprado pronto. Para isto é necessário uma avaliação dos produtos disponíveis no mercado.

Caso encontremos produtos de software genérico que atenda, ao menos parcialmente, as necessidades, o processo de obtenção passa por uma adaptação dos requisitos, pela escolha do sistema a ser utilizado, pela avaliação de propostas de compra e pela escolha do fornecedor. Já se for decidido que o sistema precisa ser desenvolvido deve-se solicitar orçamentos para o desenvolvimento do sistema, escolher a melhor proposta orçamentária, negociar e fechar o contrato de desenvolvimento. Veja na Figura um diagrama deste processo.

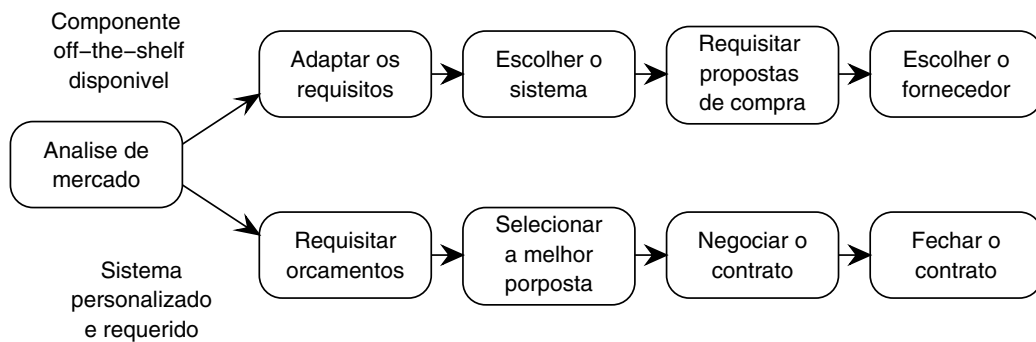


Figura 2.2: Processo de obtenção do sistema.

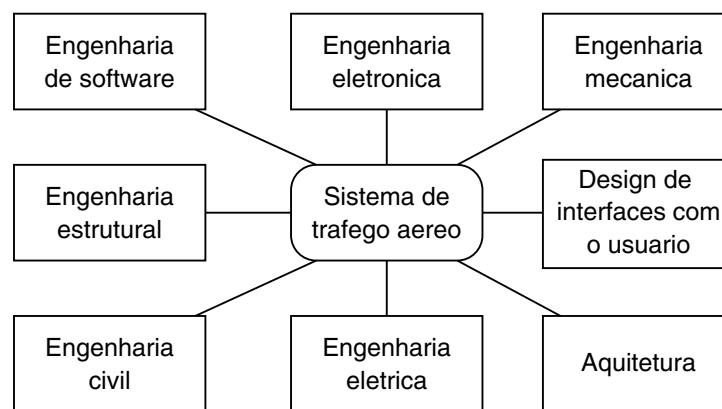


Figura 2.3: Interdisciplinaridade na engenharia de sistemas.

## 2.3 Processo de engenharia de sistemas

O processo de engenharia de um sistema inevitavelmente envolve engenheiros com várias especialidades e que devem trabalhar juntos. Se pegarmos como exemplo o desenvolvimento de um sistema de controle de tráfego aéreo (Figura 2.3) temos um caso onde profissionais de áreas tão diversas como as engenharias civil, mecânica, elétrica, eletrônica, de software, além de arquitetos e projetistas de interfaces devem trabalhar juntos.

Nestas atividades interdisciplinares, conflitos de escopo são comuns. As diversas disciplinas utilizam nomenclaturas próprias e muita negociação é necessária para que o processo flua.

O processo de engenharia de sistemas normalmente utiliza o modelo em cascata devido a necessidade de desenvolvimento em paralelo das várias partes do sistema. Um grande sistema de engenharia pode ter partes de hardware que não conversam entre si naturalmente e o software do sistema deve ser desenvolvido para normalizar este processo, servindo como uma *cola* entre os componentes disjuntos.

O processo de engenharia de sistemas passa por diferentes fases como nos mostra a Figura 2.4 e como estudaremos a seguir.

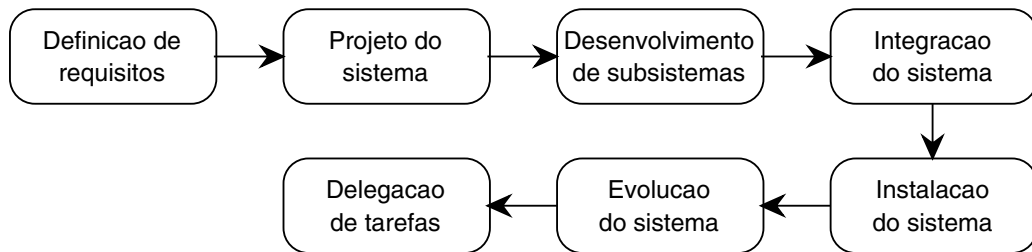


Figura 2.4: Processo de engenharia de sistemas.

### 2.3.1 Definição de requisitos

Três tipos de requisitos são definidos nesta fase:

- **Requisitos funcionais gerais.** As funções do sistema são definidas de uma forma bem abstrata.
- **Propriedades do sistema.** Requisitos não funcionais em geral são definidos.
- **Características indesejáveis.** São especificados os comportamentos não aceitáveis do sistema.

Nesta fase deve-se também definir a organização geral do sistema e os seus objetivos. Os objetivos de um sistema dividem-se em objetivos funcionais e organizacionais.

Para ilustrar este conceito vamos considerar como exemplo um sistema de alarme contra incêndio e invasão de um prédio. Este sistema tem o seguinte objetivo funcional:

*Prover um sistema de alarme contra fogo e invasão para o prédio que irá acusar interna e externamente a ocorrência de incêndio ou invasão.*

Já um dos objetivos organizacionais seria:

*Garantir que o funcionamento normal dos trabalhos no interior do prédio não seja muito afetado pela ocorrência de alarmes de incêndio ou invasão.*

O principal problema em se definir os requisitos de um sistema de grande porte é que este tipo de sistema é em geral desenvolvido para atacar **problemas ardilosos** (*wicked problems*). Um problema ardiloso é um problema que possui um conjunto tão grande de requisitos e restrições que é praticamente impossível prever todas as possibilidades que o sistema irá enfrentar.

Um exemplo extremo de um problema ardiloso seria um sistema de minimização dos efeitos de terremotos: é impossível saber o que fazer quando ocorre um terremoto, antes que o terremoto aconteça. Todo o sistema de grande escala esconde requisitos imprevisíveis como este.

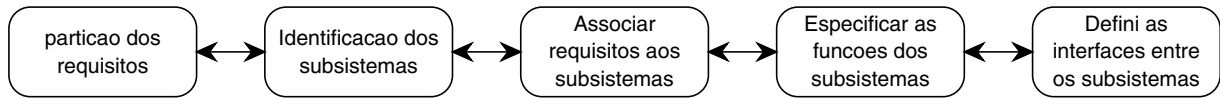


Figura 2.5: Atividades do projeto do sistema.

### 2.3.2 Projeto do sistema

No projeto do sistema é determinado como as funcionalidades do sistema serão providas pelos diferentes componentes. Podemos dividir esta fase, como sugere a Figura 2.5, nas seguintes atividades:

- **Particionar os requisitos.** Organizar os requisitos por similaridade em grupos de forma que possam ser atacados por subsistemas.
- **Identificar os subsistemas.** Baseado nos grupos de requisitos escolhidos identificar um grupo de subsistemas que seja capaz de alcançar os objetivos gerais do sistema.
- **Associar requisitos aos subsistemas.** Identificar quais grupos de requisitos serão atacados por quais subsistemas. Esta atividade causa problemas particulares quando COTS são integrados, pois pode significar que requisitos precisam ser adaptados.
- **Especificar a funcionalidade de cada subsistema.**
- **Definir as interfaces de cada subsistema.** Esta atividade é crítica quando vai se desenvolver subsistemas em paralelo pois se as interfaces não forem bem definidas, os subsistemas diferentes podem não acoplar na fase de integração.

O particionamento de requisitos entre subsistemas de software, hardware e *peopleware* – isto é, definir quais partes do sistema serão implementadas em software ou hardware, ou ainda, quais partes não serão automatizadas – pode envolver um trabalho maior de pesquisa e negociação. Nesta fase é comum que problemas de maior dificuldade de projeto sejam em geral mais facilmente resolvidos em software, entretanto isto nem sempre é verdade.

As plataformas de hardware disponíveis podem ser desapropriadas para os requisitos do software, entretanto, é trabalho da engenharia de software, dentro de um projeto, compensar estas limitações ao máximo.

### 2.3.3 Desenvolvimento de subsistemas

Em um grande sistema é comum que os diferentes subsistemas sejam desenvolvidos em paralelo. Tipicamente, atividades de desenvolvimento de hardware, software e instalação de infra-estrutura de comunicação são realizadas em paralelo.

Esta fase do processo é aquela onde o risco de os diferentes subsistemas não possam integrar-se corretamente é mais latente. Este risco se agrava com a falta de comunicação

entre as diferentes equipes trabalhando nos diferentes subsistemas. Com isto também cria-se um mecanismo burocrático e lento para a proposição de modificações no sistema, pois, já que as pessoas não se comunicam com facilidade, é mais difícil para as equipes identificarem e demonstrarem umas às outras, quais modificações precisam ser feitas.

### 2.3.4 Integração do sistema

A integração do sistema é o processo de juntar o hardware, o software e as pessoas para que trabalhem como um sistema unificado.

O processo de integração deve ser desenvolvido de forma paulatina para que os subsistemas e as pessoas se integrem um de cada vez. Isto reduz os custos de localização de erros, entretanto é necessário que o desenvolvimento dos componentes seja pensado para ser integrado desta forma. Entretanto é muito difícil agendar com precisão todos os processos de desenvolvimento de todos os subsistemas.

Nesta etapa são encontrados os problemas de má especificação e projeto de interfaces o que provoca a necessidade de modificações nos diferentes componentes. Neste caso, e quando se trabalha com diferentes fornecedores, é comum que cada fornecedor procure atribuir o erro ao componente vizinho de forma que este fornecedor tenha o menor esforço na correção do sistema. Isto é um problema que precisa ser resolvido pelas pessoas que gerenciam projeto.

### 2.3.5 Instalação do sistema

A instalação do sistema é a atividade de instalar o sistema dentro do ambiente no qual este vai operar. Esta fase não é tão simples quanto parece e alguns problemas mais comuns podem acontecer:

- O ambiente no qual o sistema será instalado não é exatamente aquele ambiente especificado. Isto é comum quando se instala sistemas de software: um bom exemplo ocorrem quando as máquinas às quais o software se destina não são exatamente aquelas esperadas.
- Pode haver muita resistência humana na instalação de um sistema. A implementação de sistemas de engenharia visa a otimização de determinados processo e as pessoas normalmente se sentem ameaçadas (normalmente com razão) com a introdução deste tipo de sistema.
- Novos sistemas precisam coexistir com sistemas antigos por algum tempo. Não se pode esperar que um sistema novo mude instantaneamente a forma de operação de uma organização. Um período de adaptação tanto das pessoas como dos dados precisa ser considerado.
- Podem haver problemas de ordem física que impeçam a instalação do sistema. Os mais comuns são problemas relacionados a falta de infra-estrutura de rede já que a maioria dos prédios foi construída antes dos computadores se tornarem tão populares.

- A preparação dos operadores pode não ser suficiente. Neste caso é necessário planejar um esquema de treinamento com o novo sistema.

### 2.3.6 Operação do sistema

Quando o sistema é posto em operação, diversos requisitos que não foram observados são trazidos à tona. Isto pode ocorrer devido a falta de clareza dos objetivos do sistema ou mesmo a falta de perspicácia dos projetistas. Alguns dos problemas que são comumente relevados através da interação do usuários com o sistema são os problemas físicos de incompatibilidade, problemas de conversão de datas e erros de operação por falhas no projeto de interfaces.

Além disto é comum que os usuários utilizem o sistema de uma forma que os projetistas não haviam imaginado. Neste caso o usuário pode não conseguir fazer o que deseja ou mesmo este usuário pode levar o sistema a uma falha imprevista.

### 2.3.7 Evolução do sistema

Sistemas de grande escala possuem um grande tempo de vida útil até mesmo para compensar seus custos de desenvolvimento. Durante este tempo os sistemas precisam evoluir adequando-se as mudanças no seu ambiente externo.

A evolução de um sistema é algo inerentemente caro e todas as mudanças em um sistema precisam ser analisadas em uma perspectiva técnica e econômica. O custo de transformação de um sistema aumenta gradativamente pois a cada nova modificação é mais difícil definir com clareza o projeto arquitetural do sistema. Assim, o sistema vai se corrompendo a medida que vai sendo mexido ficando cada vez mais caro manter o sistema funcionando.

Chamamos os sistemas que precisam ser mantidos de **sistemas legados**.

### 2.3.8 Inativação do sistema

A inativação do sistema consiste em tirar o sistema de serviço. Este processo pode requerer a remoção de materiais perigosos que podem poluir o ambiente, como produtos químicos ou mesmo a sucata digital de computadores fora de linha.

Para se retirar um sistema de operação deve-ser em geral ter um cuidado especial para não se perder os dados gerados por este sistema. Pode ser necessário reconstruir e reformatar estes dados para que sejam utilizados em outro sistema mais recente.

## 2.4 Modelagem da arquitetura do sistema

O modelo arquitetural do sistema apresenta uma visão abstrata em alto nível dos subsistemas que juntos formam o sistema completo. Este modelo pode incluir indicações dos principais fluxos de informação entre os componentes.



A forma como o modelo arquitetural é *desenhado* depende de cada sistema e, principalmente, da facilidade, maior ou menor, com que as pessoas envolvidas com o sistema conseguem compreendê-lo, entretanto, é muito comum que um sistema seja modelado através de um diagrama de blocos.

O modelo abstrato do sistema identifica diversos tipos de componentes funcionais. Estes componentes podem ser agrupados de acordo com esta classificação:

- **Componentes sensores** que adquirem informações para o sistema.
- **Componentes atuadores** que ativam processos em um sistema.
- **Componentes computacionais** que processam informações recebidas pelos sensores sob ordem dos componentes atuadores.
- **Componentes de comunicação** que repassam informações entre os componentes do sistema.
- **Componentes de coordenação** que verificam o funcionamento do sistema, aferindo se este se encontra dentro de um conjunto de restrições.
- **Componentes de interface** que permitem a comunicação do sistema com outros sistemas externos ou mesmo operadores humanos.

Veja nas Figuras 2.6 e 2.7 os modelos arquiteturais para os exemplos utilizados anteriormente: o sistema de tráfego aéreo e o sistema de alarme contra incêndio e invasão.

## 2.5 Fatores humanos

Todo o sistema é, de certa forma, operado por indivíduos humanos e portanto afeta o ambiente das pessoas. Os fatores humanos são os que mais influenciam no sucesso do sistema. Assim, alguns pontos envolvendo a interação entre o sistema desenvolvido e os indivíduos devem ser considerados:

- O sistema altera o processo de trabalho no ambiente em que está instalado? Se afirmativo, podem haver resistências ao sistema se algum serviço for perdido.
- O sistema *emburrece* os usuários? Se afirmativo, profissionais bem capacitados no modelo antigo podem oferecer resistência ao sistema.
- O sistema altera as relações de poder dentro da organização? É importante lembrar que gerentes não gostam de ter seu poder diminuído.
- O sistema altera o modo como as pessoas trabalham? A alteração de algumas práticas pode não ser bem aceita.

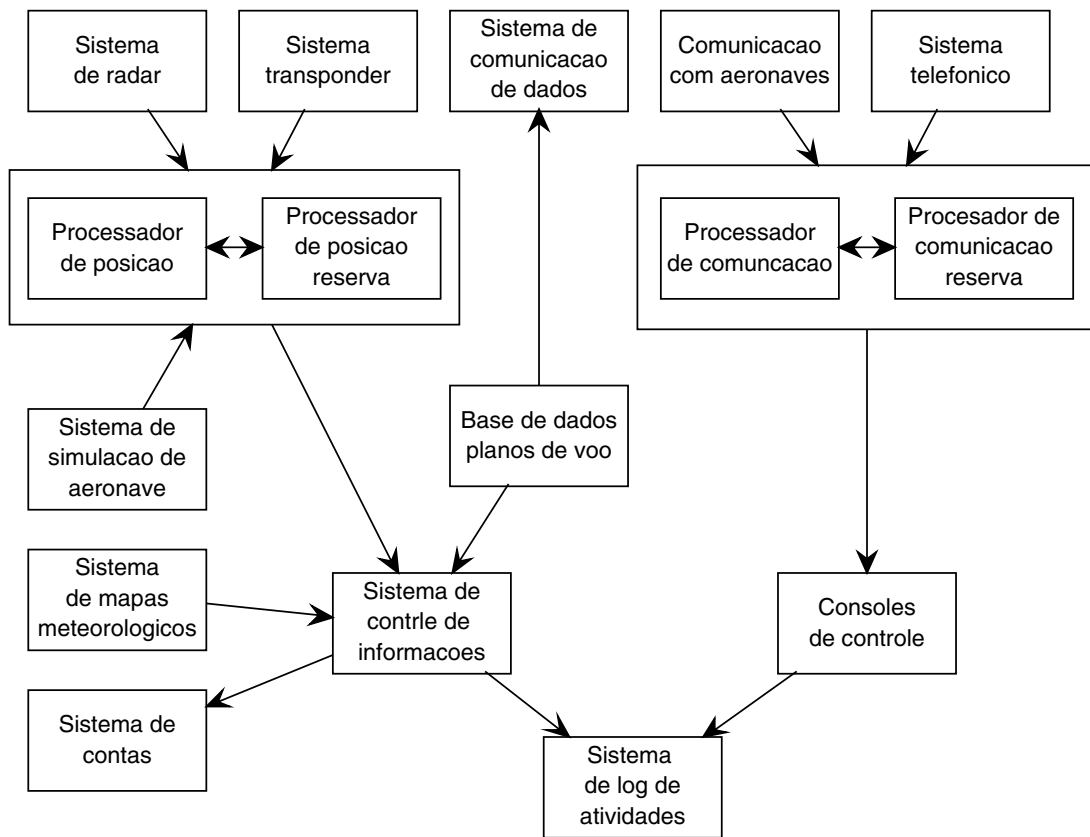


Figura 2.6: Sistema de tráfego aéreo (modelo arquitetural).

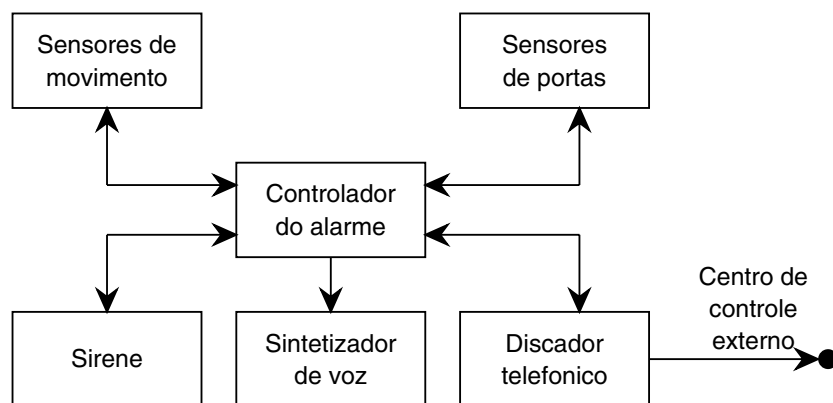


Figura 2.7: Sistema de alarmes simples (modelo arquitetural).