• Um engenheiro de software deve aceitar que o seu trabalho envolve responsabilidades mais amplas do que a simples aplicação de habilidades técnicas. Também deve se comportar de maneira ética e moralmente responsável se quiser ser respeitado como um engenheiro profissional.

1- Confiabilidade: A confiabilidade dos seus empregadores ou clientes deve ser respeitada, independente de um acordo ter sido assinado.

2- Competência: Não se deve apresentar um nível de competência que não seja verdadeiro. Trabalhos que estejam fora da sua competência não devem ser aceitos.

3- Direitos de propriedade intelectual: Deve-se estar a par das leis locais que regem o uso da propriedade intelectual, com patentes e direitos autorais. É necessário ter o cuidado de assegurar que a propriedade intelectual dos empregadores e clientes esteja protegida.

4- Mau uso do computador: Não se deve aproveitar habilidades técnicas para usar indevidamente os computadores de outras pessoas. O uso indevido de computador varia desde o relativamente trivial (jogar na maquina de um empregador) até o extremamente grave (disseminação de vírus ou outro tipo de malware).

ACM -> Association for Computing Machinery
IEEE -> Institute of Electrical and Eletronic Engineers

-> Cooperam para produzir um código de ética e prática profissional.

Os computadores têm um papel fundamental e crescente no comércio, na industria, no governo, na medicina, na educação, no entretenimento e na sociedade no geral. Os engenheiros de software são aqueles que contribuem para análise, especificação, projeto, desenvolvimento, certificação, manutenção e teste dos sistemas de software, participando diretamente ou ensinando. Em virtude de seus papéis no desenvolvimento dos sistemas de software, os engenheiros de software tem oportunidades significativas para fazer o bem ou provocar danos, capacitar terceiros a fazer o bem ou causar danos, ou influenciar terceiros a fazer o bem ou causar danos. Para assegurar o máximo possível que seus esforços serão utilizados para o bem, os engenheiros de software devem se comprometer a fazer da engenharia de software uma profissão util e respeitada. De acordo com esse compromisso, os engenheiros de software devem seguir o seguinte código de ética e prática profissional.

O código contém oito princípios relacionados ao comportamento e às decisões tomadas pelos engenheiros de software profissionais, incluindo os praticantes, educadores, gestores, supervisores e criadores de politicas, bem como estagiários e alunos da profissão. Os princípios identificam as relações eticamente responsáveis em que individuos, grupos e organizações participam e as obrigações primárias dentro dessas relações. As clausulas de cada princípio são exemplos de algumas obrigações incluidas nessas relações. Essas obrigações estão fundamentadas na humanidade da engenharia de software, com uma atenção especial dedicada às pessoas afetadas pelo trabalho dos engenheiros de software, e nos elementos exclusivos da prática da engenharia de software. O código prescreve essas obrigações de qualquer um que reivindique ou aspire ser um engenheiro de software.

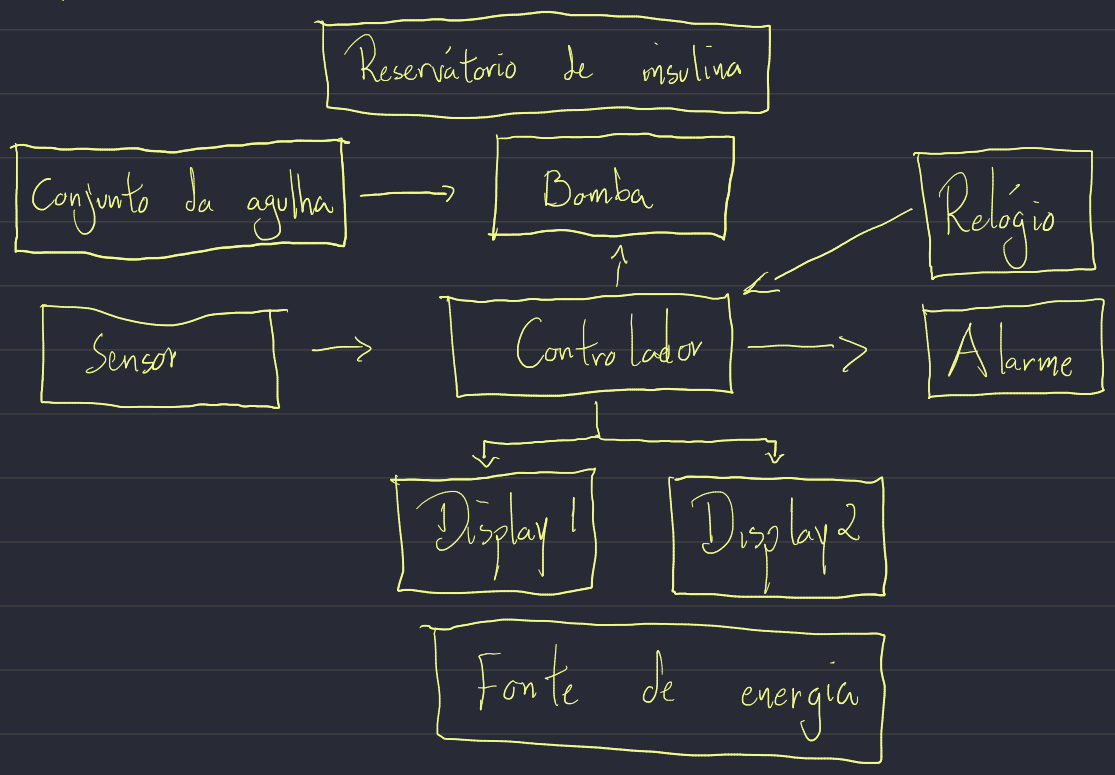
GOTTERBARN; MILLER; ROGERSON, 1999

# Estudos de caso

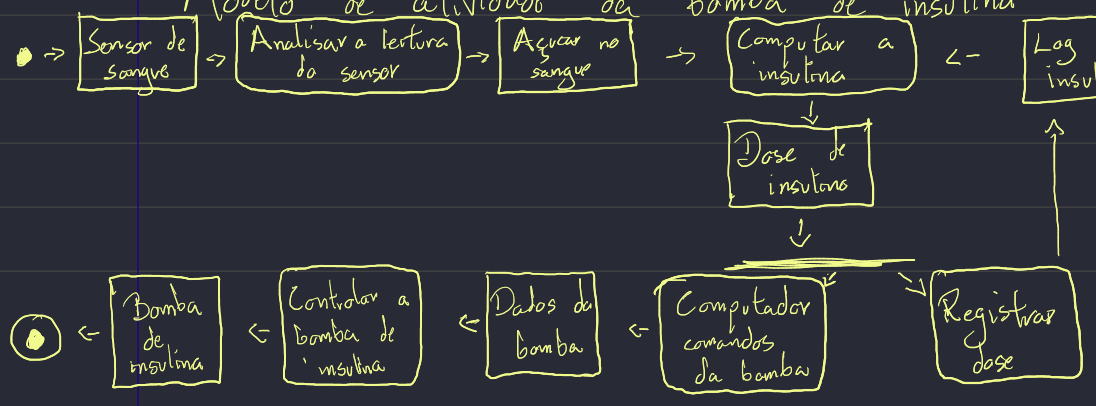
## 1- Um sistema embarcado

Esse é um sistema em que o software controla algum dispositivo de hardware e está embarcado neste dispositivo. As questões nos sistemas embarcados incluem, normalmente, o tamanho físico, a capacidade de resposta, o gerenciamento de energia etc. O exemplo de sistema embarcado que utilizo é o software para controlar uma bomba de insulina para portadores de diabetes.

## Arquitetura do hardware de uma bomba de insulina




## Modelo de atividade da bomba de insulina (UML)



Claramente, trata-se de um sistema crítico em segurança. Se a bomba não funcionar ou trabalhar de modo incorreto, a saúde do usuário pode ser prejudicada ou ela pode entrar em coma em virtude de seus níveis de açúcar no sangue estarem altos ou baixos demais. Portanto, esse sistema deve cumprir dois requisitos de alto nível essenciais:

1. O sistema deve estar disponível para fornecer insulina sempre que necessário.

2. O sistema deve funcionar de maneira confiável e fornecer a quantidade correta de insulina para compensar o nível atual de açúcar no sangue.

—— // ——

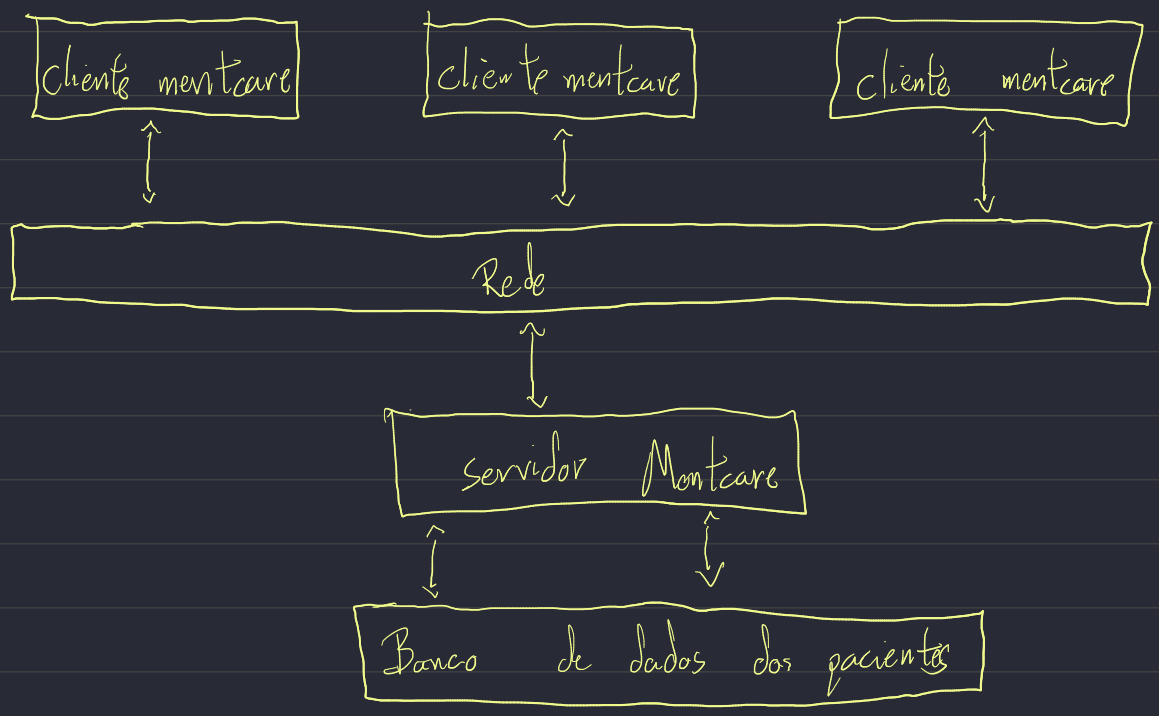
2- Um sistema de informação

→ A finalidade principal desse tipo de sistema é gerenciar e proporcionar acesso a um banco de dados de informações. As questões nos sistemas de informação incluem segurança da informação (security), usabilidade, privacidade e manutenção da integridade dos dados. O exemplo de sistema de informação utilizado é um sistema de registros médicos.

- O sistema Mentcare é um sistema de informação do paciente para ser utilizado em clínicas. Ele utiliza um banco de dados centralizado, que contém informações sobre os pacientes, mas também foi concebido para ser executado em um notebook, de modo que possa ser acessado e utilizado em locais sem acesso a uma conexão de rede segura. Quando os sistemas locais contam com acesso seguro à rede, eles usam as informações de pacientes que constam do banco de dados, mas podem baixar e usar

cópias locais dos registros dos pacientes quando estiverem desconectados. O sistema não é um registro de informações médicas completo e, portanto, não mantém informações sobre outras condições clínicas

- A organização do Sistema Mentcare




Esse sistema tem duas finalidades

1. Gerar informações de gestão que permitam que gestores de serviços de saúde avaliem o desempenho em relação às metas locais e governamentais.
2. Fornecer informações atualizadas à equipe médica para o tratamento dos pacientes.

As características principais desse sistema são:

1 - Gestão de cuidado individual
2 - Monitoramento do paciente
3 - Relatórios administrativos

## 3 - Sistema de coleta de dados baseado em sensores

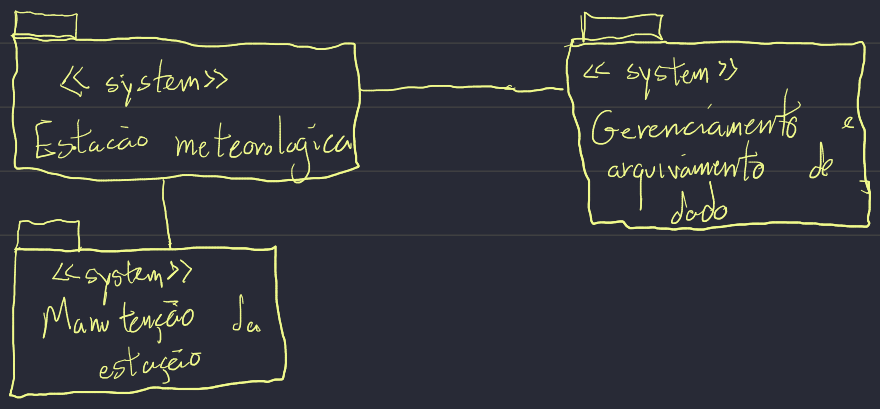
→ Esse é um sistema cujas finalidades principais são coletar dados de um conjunto de sensores e processá-los de alguma maneira. Os requisitos fundamentais desses sistemas são confiabilidade, mesmo em condições ambientais hostis, e manutenibilidade. O exemplo se sistema de coleta de dados que utilizo é uma estação meteorológica na natureza.

I - Sistema da estação meteorológica → É responsável por coletar dados climáticos, realizar algum processamento inicial e transmitir os dados para o sistema de gerenciamento.

II - Sistema de gerenciamento e arquivamento de dados → Coleta dados de todas as estações meteorológicas, processa e analisa os dados e os arquiva de uma forma que possam ser recuperados por outros sistemas, como os sistemas de previção meteorológica.

III - Sistema de manutenção da estação → Capaz de se comunicar via satélite com todas as estações meteorológicas na natureza para monitorar a condição desses sistemas e fornecer relatórios dos problemas. Ele também pode atualizar o software embarcado nesses sistemas.

O ambiente da estação meteorológica



Uma vez que as estações meteorológicas precisam ser autocontidas e independentes, o software instalado é complexo, embora a funcionalidade da coleta de dados seja bem simples.

— || —

## 4. - Um ambiente de suporte

Esse é um conjunto integrado de ferramentas de software como o eclipse, serão o tipo de ambiente mais familiar aos leitores deste livro. Exemplo: Ambiente digital de aprendizado para escolas.

A arquitetura de um ambiente digital de aprendizagem: iLearn

| | |
|---|---|
| Interface com o usuário baseada em navegador | Aplicativo iLearn |

Serviços de configuração

| | | |
|---|---|---|
| Gerenciamento de grupo | Gerenciamento de aplicação | Gerenciamento de identidade |

Serviços de aplicação

| | | | |
|---|---|---|---|
| E-mail | Mensagens | Videoconferência | Arquivo de imprensa |
| Processamento de texto | | Simulação | Armazenamento de vídeo |
| Localização de recursos | | Planilha | Ambiente virtual |
| | Histórico | | |

Serviços utilitários

| | | |
|---|---|---|
| Autenticação | Logging e monitoramento | Interfaceamento |
| Armazenagem do usuário | | Armazenamento de aplicação |
| | Busca | |

O iLearn foi concebido para utilização em escolas, por alunos dos 3 aos 18 anos de idade. A abordagem adotada é a de um sistema distribuído em que os componentes do ambiente são serviços que podem ser acessados de qualquer lugar através da internet. Não é necessário que todas as ferramentas de aprendizagem estejam reunidas em um único lugar.

O ambiente foi projetado para que os serviços possam ser substituídos à medida que novos serviços forem disponibilizados e para proporcionar diferentes versões do sistema que sejam adequadas à idade dos usuários.

1. Serviços integrados: São aqueles que oferecem uma API (interface de programação de aplicação, do inglês application programming interface) e que podem ser acessados por outros serviços através dessa API. Portanto, a comunicação direta entre serviços é possível. Um serviço de autentificação é um exemplo de serviço integrado, pois em vez de usar mecanismos de autentificação próprios, ele pode ser chamado por outros serviços para autenticar usuários. Se os usuários já estiverem autenticados, então o serviço de autentificação pode passar a informação de autentificação diretamente para outro serviço, por meio de uma API, sem que os usuários precisem ser autenticados novamente.

2. Serviços independentes: São aqueles acessados por meio de uma interface do navegador e que operam de maneira independente dos demais serviços. As informações só podem ser compartilhadas com outros serviços por ações explícitas do usuário, como copiar e colar; a reautentificação pode ser necessária para cada serviço independente.

— // —

Exercícios

1.1 - Explique por que o software profissional desenvolvido para um cliente não consiste apenas nos programas que foram desenvolvidos e fornecidos.

-> Os programas desenvolvidos são o reflexo de uma boa entrevista com o cliente, boas maneiras de se lidar com a equipe e boas escolhas de regras de negócio que levam as boas práticas de criação e documentação na engenharia de software.

1.2 - Qual é a diferença mais importante entre desenvolvimento de software genérico e desenvolvimento de software personalizado? O que isso poderia significar na prática para os usuários dos produtos de engenharia de software genéricos?

-> Software genérico são feitos para atender a necessidade do mercado no geral, que controla a especificação é o proprio desenvolvedor.

-> Software personalizado o comprador controla a especificação, eles são feitos para atender a necessidade deste comprador.

1.3 - Quais são os quatro atributos importantes que todo software profissional deve possuir? Sugira outros quatro atributos que as vezes, podem ser significantes.

atributos importantes

- Manutenabilidade
- Eficiência
- Confiança
- Proteção
- Aceitabilidade

outros atributos

- Portabilidade
- Rastreabilidade
- Facilidade de integração
- Distribuição

1.4 - Além dos desafios de heterogeneidade, mudanças sociais e corporativas, confiança e proteção, identifique outros problemas e desafios que a engenharia de software provavelmente enfrentará no século XXI (Dica: pense no meio ambiente).

-> • Sistemas eficientes energeticamente
• Sistemas humanizados e de fácil compreenção a todos
• Sistemas com grande adaptabilidade as mudanças e necessidades do negócio.

1.5 - Baseado em seu conhecimento de alguns tipos de aplicações discutidos na seção 1.1.2, explique, com exemplos, por que tipos de aplicações diferentes requerem técnicas especializadas de engenharia de software para apoiar seu projeto e desenvolvimento.

-> Cada sistema geralmente requer diferentes requisitos. Desde uma aeronave com sistema fechado de controle em tempo real - tendo a segurança como principal requisito, ou uma máquina em uma fabrica que depende de um sistema que auxilia a pressão exercida no metal, um jogo fluido e com poucos bugs. E cada exemplo exige tecnicas diferentes de construção e manutenção.

1.6 - Explique por que os princípios fundamentais da engenharia de software, como processo, dependabilidade, gerenciamento de requisitos e reúso, são relevantes para todos os tipos de sistema de software.

-> Os fundamentos de engenharia de software pouco envolvem códigos em diferentes linguagens - os fundamentos são amplos, desde ética a prática da engenharia e isso a torna um item tanto atemporal quanto multiplataforma dentro do mercado de TI.

1.7 - Explique como o uso universal da web mudou os sistemas de software e sua engenharia

-> Com a difusão da internet a população, os softwares começaram a funcionar de acordo com a evolução das ferramentas web. Hoje a nuvem, por exemplo, auxilia diferentes mercados na sociedade: negócios, games, armazenamentos de dados e etc. Isso criou novas mudanças ao cenário de software e como as pessoas se relacionam com elas.

1.8 - Discuta se os engenheiros profissionais devem ser licenciados, da mesma maneira que os médicos e advogados.

-> Da mesma forma que um médico é responsável pelo seu paciente, um engenheiro civil pela sua obra e segurança e um advogado trabalhar dentro das leis - um engenheiro de software necessita seguir regras para se tornar um profissional melhor e que não conduza um software ao abismo, caso isso aconteça, a empresa pode quebrar ou algo pior. Ou seja, um engenheiro de software deve seguir as regras assim com qualquer profissão que requer cuidados.

1.9 - Para cada cláusulas no código de ética da ACM/IEEE exibido na figura 1.3, proponha um exemplo adequado que ilustre essa cláusula.

-> • Público - Engenheiros de software devem agir de acordo com o interesse público.
• Cliente e Empregador - Engenheiros de software devem agir de maneira que seja do melhor interesse de seu cliente e empregador e de acordo com o interesse publico
• Produto - Engenheiros de software devem garantir que seus produtos

e modificações relacionadas atendam aos mais altos padrões profissionais possíveis.

- Julgamento – Engenheiros de software devem manter a integridade e a independência em seu julgamento profissional
- Gerenciamento – Gerentes e líderes de engenharia de software devem aceitar e promover uma abordagem ética para o gerenciamento de desenvolvimento e manutenção de software.
- Profissão – Engenheiros de software devem aprimorar a integridade e a reputação da profissão de acordo com o interesse público.
- Colegas – Engenheiros de software devem auxiliar a ser justos com seus colegas.
- Si Próprio – Engenheiros de software devem participar da aprendizagem contínua durante toda a vida, e devem promover uma abordagem ética para a prática da profissão.

1.10 – Para ajudar a combater o terrorismo, muitos países estão planejando ou desenvolveram sistemas de computador que rastreiam um grande número de seus cidadãos e suas ações, o que, claramente, tem implicações na privacidade. Discuta a ética de trabalhar no desenvolvimento desse tipo de sistema.

→ Trabalhar com órgãos governamentais de rastreio é uma questão delicada e que somente o empregador e a pessoa a assumir o cargo devem conversar e chegar a um acordo. Geralmente serviços com estes requerem maior atenção no trabalho e vida social. Snowden, Cambridge Analítica e outros casos demonstram que muitos colaboradores podem sair dos eixos com a pressão do trabalho.