

Arquitetura de Software

Texto complementar para este assunto:

Len Bass, Paul Clements, Rick Kazman  
Software Architecture in Practice, 2<sup>nd</sup> Edition  
Capítulos 1, 2 e 3  
Disponível em:  
<http://www.tar.hu/softarchpract/index.html>

PEARSON  
Prentice Hall

© 2007 by Pearson Education  
Olan Sommerville 2006Engenharia de Software, 8ª edição. Capítulo 11

Slide 1


Programação Modular

PEARSON  
Prentice Hall

© 2007 by Pearson Education  
Olan Sommerville 2006Engenharia de Software, 8ª edição. Capítulo 11

Slide 2

Programação Modular

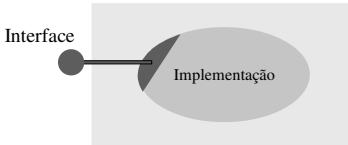


PEARSON  
Prentice Hall

© 2007 by Pearson Education  
Olan Sommerville 2006Engenharia de Software, 8ª edição. Capítulo 11

Slide 3

Programação Modular

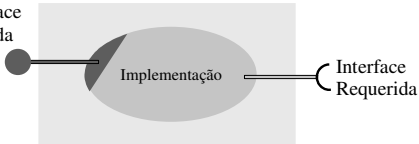


PEARSON  
Prentice Hall

© 2007 by Pearson Education  
Olan Sommerville 2006Engenharia de Software, 8ª edição. Capítulo 11

Slide 4

Programação Modular

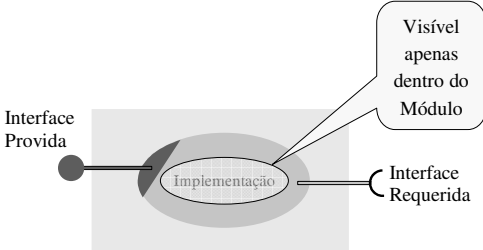


PEARSON  
Prentice Hall

© 2007 by Pearson Education  
Olan Sommerville 2006Engenharia de Software, 8ª edição. Capítulo 11

Slide 5

Programação Modular



PEARSON  
Prentice Hall

© 2007 by Pearson Education  
Olan Sommerville 2006Engenharia de Software, 8ª edição. Capítulo 11

Slide 6

## Benefícios Esperados da Programação Modular [Parnas, 1972]

- (1) Tempo de desenvolvimento encurtado, já que grupos de desenvolvimento separados poderiam trabalhar em um módulo, com pouca necessidade de comunicação
- (2) Possibilidade de realizar mudanças drásticas a um módulo sem a necessidade de mudar outros
- (3) Possibilidade de estudar o sistema olhando para um módulo de cada vez
  - ✓ Interações entre módulos

## Arquitetura de Software

A **estrutura** de um sistema de software, que engloba

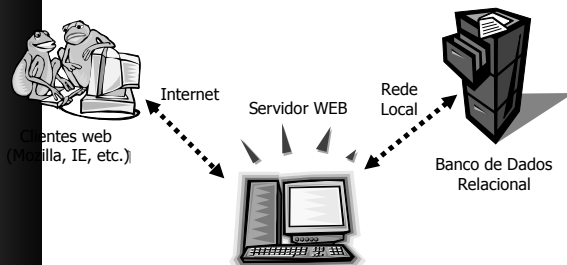
- componentes de software;
- suas propriedades visíveis externamente;
- e os relacionamentos e interações entre eles

As primeiras decisões tomadas no projeto de um sistema

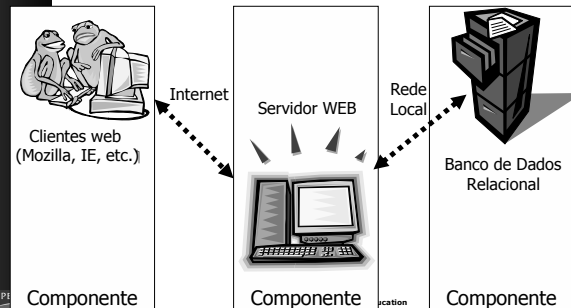
- **As mais importantes!**

Uma arquitetura de *software* é composta por **componentes e conectores**

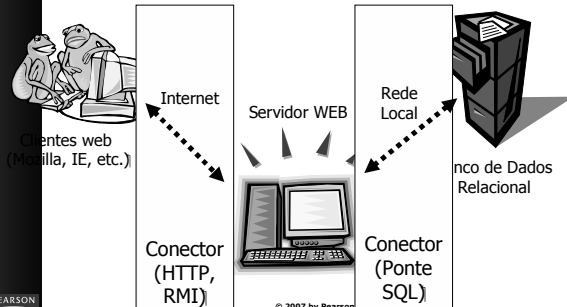
## Uma Arquitetura em Camadas



## Uma Arquitetura em Camadas



## Uma Arquitetura em Camadas



## Projeto Arquitetural

- O processo de projeto que estabelece
  - Os subsistemas que constituem um sistema
  - A maneira como essas componentes interagem
- Incluindo algumas decisões tecnológicas
  - Ex. Plataforma de componentes, SGBD
- A saída desse processo de projeto é uma descrição da arquitetura de software.
- A arquitetura de software lida com os requisitos não-funcionais do sistema

## Projeto Arquitetural

- É o primeiro estágio do projeto do sistema
- Representa a ligação entre os processos de especificação e de projeto
- É freqüentemente conduzido em paralelo com algumas atividades de especificação
  - Às vezes junto com a elicitação de requisitos
- Envolve a identificação dos componentes principais do sistema e sua interação
  - Componentes => unidades de **modularidade**

## Vantagens da arquitetura explícita

- Comunicação de *stakeholder*
  - A arquitetura pode ser usada como um foco de discussão pelos *stakeholders* do sistema.
- Análise de sistema
  - Se há possibilidade de o sistema atender a seus requisitos de qualidade (não-funcionais)
- Reuso em larga escala
  - A arquitetura pode ser reusável em uma variedade de sistemas
  - Suas partes também!

## Características de arquitetura e de sistema

### Desempenho

- Localizar operações críticas e minimizar comunicações. Usar componentes de alta ao invés de baixa granularidade.

### Proteção (*security*)

- Usar uma arquitetura em camadas com itens críticos nas camadas mais internas.

### Segurança (*safety*)

- Localizar características críticas de segurança em um pequeno número de subsistemas.

### Disponibilidade

- Incluir componentes redundantes e mecanismos para tolerância a falhas.

### Facilidade de manutenção

- Usar componentes facilmente trocáveis

## Conflitos de arquitetura

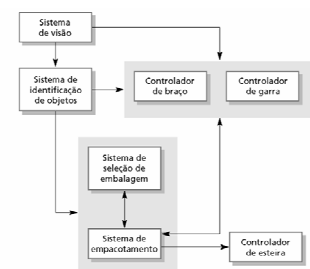
- O uso de componentes de alta granularidade aprimora o desempenho mas diminui a facilidade de manutenção
- A introdução de dados redundantes aprimora a disponibilidade, mas torna a proteção mais difícil
  - E cria dificuldades para tornar o sistema confiável em outras partes
- Localizar as funcionalidades críticas de segurança em poucos locais pode criar gargalos de desempenho

## Estruturação de sistema

- Está relacionado à decomposição do sistema em subsistemas que interagem.
- O projeto de arquitetura é normalmente expresso como um diagrama de blocos
  - Estrutura geral do sistema
- Modelos mais específicos também podem ser desenvolvidos.

## Sistema de controle robotizado de empacotamento

Figura 11.1  
Diagrama de blocos de um sistema de controle robotizado de empacotamento.



## Diagramas caixa e linha

- Muito abstrato – não mostram a natureza dos relacionamento de componentes, nem suas propriedades externamente visíveis
- Contudo, são úteis para comunicação com os stakeholders e para planejamento de projeto.
- Alternativas:
  - Notações formais
  - Notações informais mais organizadas

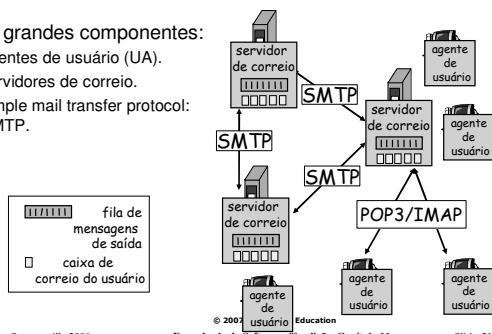
## Visões Arquiteturais

- A arquitetura de um sistema software normalmente é representada através de várias **visões**
- Visões são maneiras diversas de se enxergar uma mesma arquitetura
  - Enfocando diferentes aspectos de interesse
  - Ex.: as várias plantas de uma casa
- Arquiteturas de software são especificadas através de uma ou mais de suas visões

## Correio Eletrônico – Visão 1

Três grandes componentes:

- agentes de usuário (UA).
- servidores de correio.
- simple mail transfer protocol: SMTP.



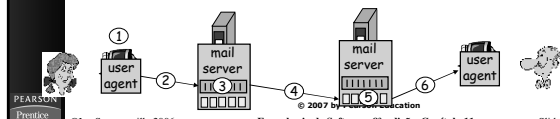
## Correio Eletrônico – Visão 2

Alice usa o UA para compor uma mensagem "para" bob@school.edu

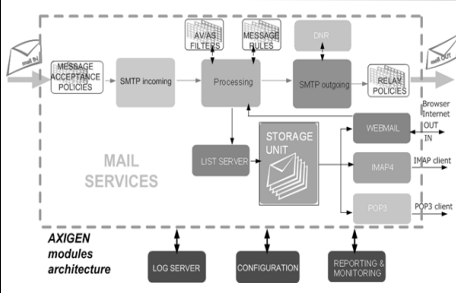
O UA de Alice envia a mensagem para o seu servidor de correio; a mensagem é colocada na fila de mensagens.

O lado cliente do SMTP abre uma conexão TCP com o servidor de correio de Bob.

- O cliente SMTP envia a mensagem de Alice através da conexão TCP.
- O servidor de correio de Bob coloca a mensagem na caixa de entrada de Bob.
- Bob chama o seu UA para ler a mensagem.

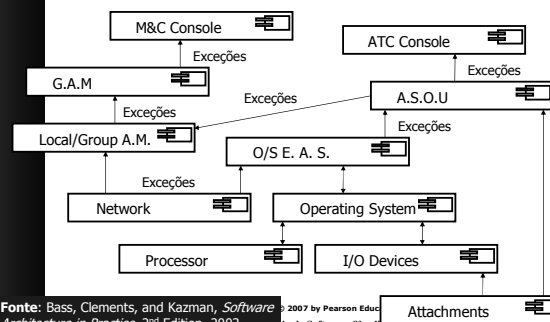


## Correio Eletrônico – Visão 3



Fonte: Axigen Mail Server Documentation - Mail Server Architecture. Consultado em 24 de março de 2008  
[http://www.axigen.com/docs/en/Mail-Server-Architecture\\_85.html](http://www.axigen.com/docs/en/Mail-Server-Architecture_85.html)

## Um Exemplo de Sistema de Controle de Tráfego Aéreo



Fonte: Bass, Clements, and Kazman, Software Architecture in Practice, 2nd Edition, 2003.

## Sobre Visões

- Algumas são genéricas
  - Lógica
  - De interação
  - Física ou de implantação
- Outras servem a fins específicos
  - Fluxo de exceções

## Decisões de projeto de arquitetura

- Projeto de arquitetura é um processo criativo
  - Cada sistema envolve diferentes decisões/requisitos/conflitos/restrições
- Contudo, há algumas questões que são praticamente universais
  - Precisam ser respondidas durante o projeto arquitetural

## Decisões de projeto de arquitetura

- Existe uma arquitetura genérica de aplicação que possa ser usada?
- Como o sistema será distribuído?
- Quais estilos de arquitetura são apropriados?
- Qual será a abordagem fundamental usada para estruturar o sistema?
- Como o sistema será decomposto em módulos?
- Como o projeto de arquitetura será avaliado?
- Como a arquitetura do sistema deve ser documentada?

## Reuso de arquitetura

- Sistemas do mesmo domínio freqüentemente têm arquiteturas similares que refletem os conceitos de domínio.
- Linhas do produto de software são construídas em torno de um núcleo de arquitetura
  - Variantes satisfazem requisitos específicos de clientes.
- Reuso de arquiteturas é capturado através da noção de padrões ou estilos arquiteturais