

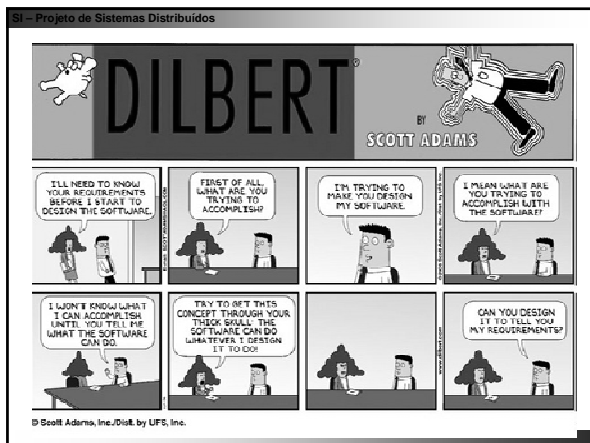
## Projeto de Sistemas Distribuídos

### Requisitos de Software

Prof. Msc. Marcelo Iury de Sousa Oliveira  
marceloiury@gmail.com  
<http://sites.google.com/site/marceloiury/>

### Requisito é...

- Uma condição ou capacidade que deve ser atendida ou possuída por um sistema, produto, serviço, resultado ou componente para satisfazer um contrato, uma norma, uma especificação ou outros documentos impostos formalmente.
  - Os requisitos incluem necessidades, desejos e expectativas quantificados e documentados do patrocinador, do cliente e de outras partes interessadas.




### Levantamento de Requisitos

- Obter os requisitos dos especialistas no domínio – clientes e usuários
- Principal problema: comunicação
  - Usuários e especialistas não compartilham o mesmo vocabulário
- Técnicas de Comunicação
  - Entrevistas
  - Observação direta
  - Encontros
- Extração dos conceitos

SI - Projeto de Sistemas Distribuídos

## Requisitos básicos em SD

- Compartilhamento de recursos
- Abertura
- Concorrência
- Escalabilidade
- Tolerância a falhas
- Transparência



SI - Projeto de Sistemas Distribuídos

## Compartilhamento

- Recursos fisicamente encapsulados em um computador
  - Encapsulamento → acesso através de interface
  - Alguém para gerenciar
- Gerente de recursos
  - Software que gerencia um conjunto de recursos
  - Políticas e métodos diferentes

SI - Projeto de Sistemas Distribuídos

## Compartilhamento...

- ... de recursos
  - Hardware: disco, impressora
  - Software: janelas, dados, arquivos
- Usuário acostumado
- Concebido nos '60s
  - Time sharing multiusuário
  - UNIX multiusuário




SI - Projeto de Sistemas Distribuídos

## Compartilhamento

- Não há acesso direto!
  - Cliente fala com o gerente
  - Gerente atende o pedido
  - Gerente devolve resultado
- Dois paradigmas de interação
  - Cliente/servidor
  - Modelo baseado em objeto
- Maior compartilhamento → Maior problemas de segurança

SI – Projeto de Sistemas Distribuídos

## Compartilhamento de dados

- Redes de computadores
  - Centralização das atualizações
- Bases de dados
  - Acesso compartilhado a registros
  - Controle de concorrência/atualização (calma!)
- Arquivos
  - Sistemas P2P



SI – Projeto de Sistemas Distribuídos

## Abertura não é isso!



SI – Projeto de Sistemas Distribuídos

## Requisitos básicos em SD

- **Compartilhamento de recursos**
- Abertura
- Concorrência
- Escalabilidade
- Tolerância a falhas
- Transparência



SI – Projeto de Sistemas Distribuídos

## Sistemas precisam de ...

- Extensão de hardware
  - Adição de memória, periféricos...
- Extensão de software
  - Recursos de software, protocolos, serviços...
- A inserção de novos módulos no sistema deve ser uma tarefa simples
- Grau de não interferência e não duplicidade quando da inclusão de algo novo

## Abertura

- A abertura de um sistema determina o modo como pode ser estendido e re-implementado e por quem:
  - Sistemas abertos
    - Interfaces e modelo (incluindo protocolos de comunicação) públicos
    - Evolução controlada por organismos de normalização independentes ou consórcios industriais
    - Permite a interoperabilidade de componentes com diferentes implementações
  - Sistemas proprietários
    - Fechados e apenas podem ser modificados pelo seu “dono”

## Interoperabilidade vs Portabilidade

- Interoperabilidade
  - Caracteriza até que ponto duas implementações devem coexistir e trabalhar em conjunto
- Portabilidade
  - Caracteriza até que ponto uma aplicação desenvolvida para um sistema A pode ser executada, sem modificação, para um sistema B que implementa as mesmas interfaces do sistema A
- Exemplo: Interfaces Wireless

## Abertura

- Resulta da especificação de interfaces, de tornar as especificações públicas e de padronizá-las
- Especificações podem ser
  - padrões estabelecidos por organização de padronização
  - padrões estabelecidos pelo uso (de fato)
- Especificações adequadas devem ser neutras e completas
  - Estas propriedades são importantes para Interoperabilidade e Portabilidade

## Requisitos básicos em SD

- **Compartilhamento de recursos**
- **Abertura**
- Concorrência
- Escalabilidade
- Tolerância a falhas
- Transparência



## Concorrência

- O acesso simultâneo a um recurso por vários clientes pode causar problemas;
- Torna-se necessário serializar as operações para que os dados permaneçam consistentes.
  - Num único computador, soluções possíveis são o uso de semáforos, monitores, transações
  - Num sistema distribuído, é necessário recorrer a técnicas de sincronização distribuídas.

## Escala

- A escala de um sistema distribuído é o âmbito que o mesmo abrange assim como o número de componentes;
  - A escala de um sistema tem várias facetas:
    - recursos e utilizadores
      - âmbito geográfico (rede local, país, mundo, ...)
      - âmbito administrativo (uma organização, inter-organizações)
  - Pequena escala: confinado a uma organização pequena / média, envolvendo poucos utilizadores e poucos componentes;
  - Grande escala: envolve muitos recursos e/ou utilizadores, ou cobrindo uma área geográfica significativa, ou abrangendo muitas organizações.

## Requisitos básicos em SD

- **Compartilhamento de recursos**
- **Abertura**
- **Concorrência**
- Escalabilidade
- Tolerância a falhas
- Transparência



## Escalabilidade

- Um sistema é descrito como escalável se permanece eficiente quando há um aumento significativo no número de recursos e no número de usuários.
- SD tem que trabalhar em qualquer escala
  - 2 computadores em rede
  - Milhares em várias redes



## Escalabilidade: desafios

- O projeto de sistemas distribuídos escaláveis apresenta os seguintes desafios:
  - Controlar os custos dos recursos físicos.
  - Controlar a perda de desempenho.
  - Impedir que os recursos de software se esgotem: IP de 32 bits para IP de 128 bits...
  - Evitar gargalos de desempenho.
- Não deve haver mudanças de SW
- Contra-exemplos
  - Mudança de número de telefone (7 → 8)
  - Justiça seja feita: impossível prever!

## Escalabilidade

- Replicação
  - Garantia de coerência entre cópias
  - Algoritmos de replicação

## Escalabilidade

- Escalabilidade não é só HW ou rede
- Em SD, tudo é “ilimitado”
  - Adicionar estação/servidores
  - Enlaces
- Aumenta usuários → aumenta acesso a arquivos
  - Acessos frequente ao mesmo arquivo → gargalo no sistema
  - Replicação dos arquivos

## Requisitos básicos em SD

- Compartilhamento de recursos
- Abertura
- Concorrência
- Escalabilidade
- Tolerância a falhas
- Transparência



## Tolerância a Falhas

- Sistemas distribuídos devem ser projetados para mascarar falhas
- As falhas em um sistema distribuído são parciais, isto é, alguns componentes falham, enquanto outros continuam funcionando.
- O tratamento de falhas é particularmente difícil, e envolve as seguintes técnicas:
  - Detecção de erros;
  - Mascaramento de falhas.
  - Tolerância a falhas.
  - Recuperação de falhas.
  - Redundância

## Tolerância a falhas

- Disponibilidade dos SD
  - Centralizado
    - Falha no computador central: indisponibilidade
  - Distribuído
    - Falha em serviço: reiniciar em outra máquina
    - Falha de estação: troca de estação
- E a rede???
  - Geralmente não redundante... (FDDI é!)
  - Sobrecarga → indisponibilidade

## Tolerância a falhas

- Software tolerante?
  - Formas de recuperar o estado seguro
  - Roll back
    - Shadow pages... (calma!)
- Disponibilidade
  - Fração do tempo funcionando corretamente
  - Tempo médio entre falhas, para falhar, para recuperar, para detectar...

## Requisitos básicos em SD

- Compartilhamento de recursos
- Abertura
- Concorrência
- Escalabilidade
- Tolerância a falhas
- Transparência

SI – Projeto de Sistemas Distribuídos

## Transparência

- Um sistema distribuído deve ser capaz de se apresentar como se fosse um sistema único.
- Esconder os detalhes.



**Principal Requisito!**

SI – Projeto de Sistemas Distribuídos

## Transparência(s)

- Acesso
  - Local e remoto iguais
- Localização
  - Não interessa o lugar
- Concorrência
  - Compartilhamento
- Replicação
  - Várias cópias sem perceber
- Falha
  - "Non-stop"
- Migração
  - Mudança de lugar sem afetar sistema
- Desempenho
  - 1 ou 99 dá no mesmo
- Escalabilidade
  - Crescer sem mudar algoritmos

SI – Projeto de Sistemas Distribuídos

## Transparência

- Separação dos componentes não influencia na percepção do sistema como um todo
  - Separação é da natureza do SD
- Usuário deve ignorar a distribuição
  - Quanto mais "ignorante", melhor!
- ISO definiu 8 formas de transparência

SI – Projeto de Sistemas Distribuídos

## Transparência


- As mais importantes
  - Acesso e localização
  - Transparência de rede
    - Anonimidade para os recursos
- Exemplo
  - Email: dc@lobato.org
  - Qual é o servidor? Onde está? Onde ELE está?



SI – Projeto de Sistemas Distribuídos

## Requisitos básicos em SD

- **Compartilhamento de recursos**
- **Abertura**
- **Concorrência**
- **Escalabilidade**
- **Tolerância a falhas**
- **Transparência**



SI – Projeto de Sistemas Distribuídos

## Considerações

- Não há somente estes requisitos
  - Concorrência, distribuição de carga, flexibilidade, segurança...
  - Estes são os mais popularmente usados
- Isso não quer dizer que devemos implementar todos os requisitos possíveis.
  - Qual é o problema disso???

SI – Projeto de Sistemas Distribuídos

## Outros Requisitos

- **Segurança**
  - Advém da necessidade em proteger os recursos e a informação gerida num sistema distribuído
  - Implica um recurso que têm valor para os seus utilizadores
  - Segurança tem três componentes principais:
    - **Confidencialidade:** indivíduos não autorizados não podem obter informação
    - **Integridade:** dados não podem ser alterados ou corrompidos
    - **Disponibilidade:** acesso aos dados deve continuar disponível

SI – Projeto de Sistemas Distribuídos

## Considerações

- Estes requisitos são os mais popularmente usados
- Isso não quer dizer que devemos implementar todos os requisitos possíveis. Qual é o problema disso???

