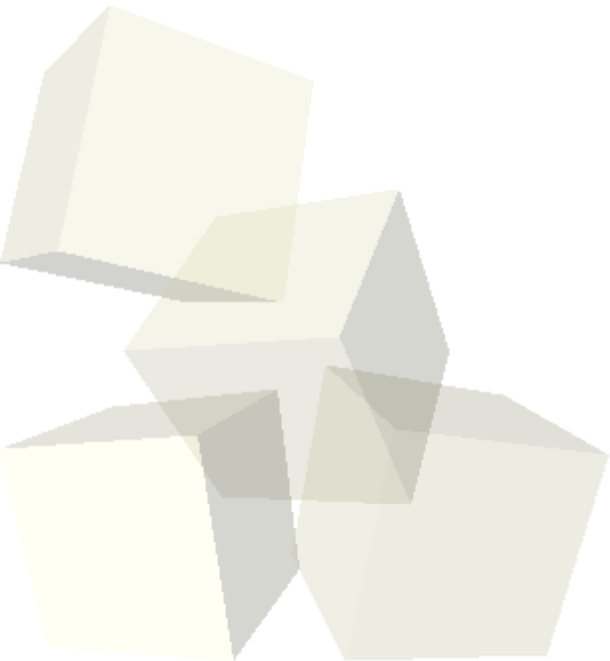


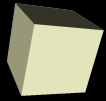


## Programação para Redes de Computadores

Sistemas Distribuídos com RMI, CORBA e SOA

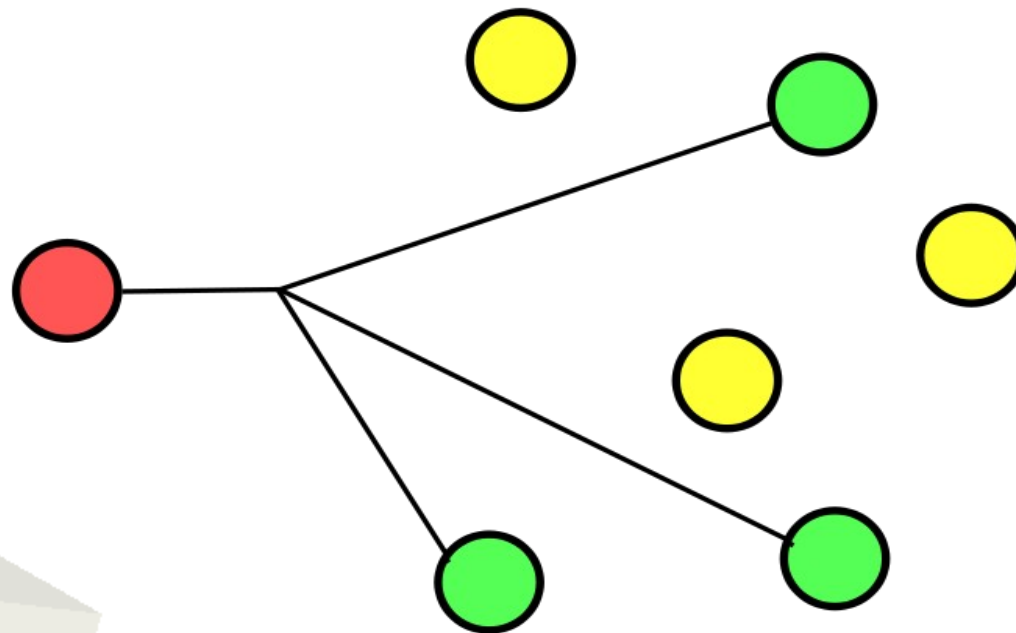


Líus Fontenelle Carneiro  
<http://hypercast.info/>



## ■ Multicast IP

- ♦ Otimização do uso do link eliminando redundância;
- ♦ Múltiplas notificações divididas por grupos.
- ♦ **Facilidade de desenvolvimento;**
- ♦ **Permite que a aplicação cresça.**





# Tipos de aplicações de redes

- Apenas dois tipos:
  - ◆ Transferência de dados e/ou informações:
    - FTP
    - SMTP
    - HTTP
    - POP
    - IMAP
    - etc...
  - ◆ Execução de programas em outros hosts:
    - RPC
    - Telnet
    - SSH
    - rlogin

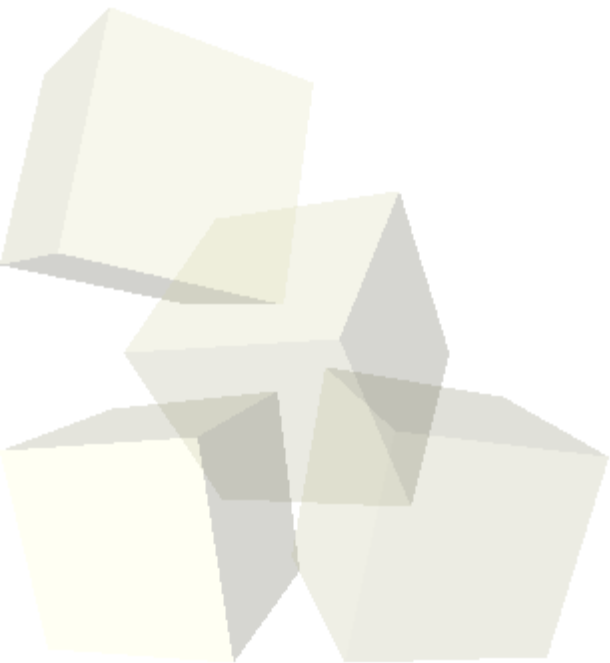
Onde podemos encaixar os sistemas distribuídos?

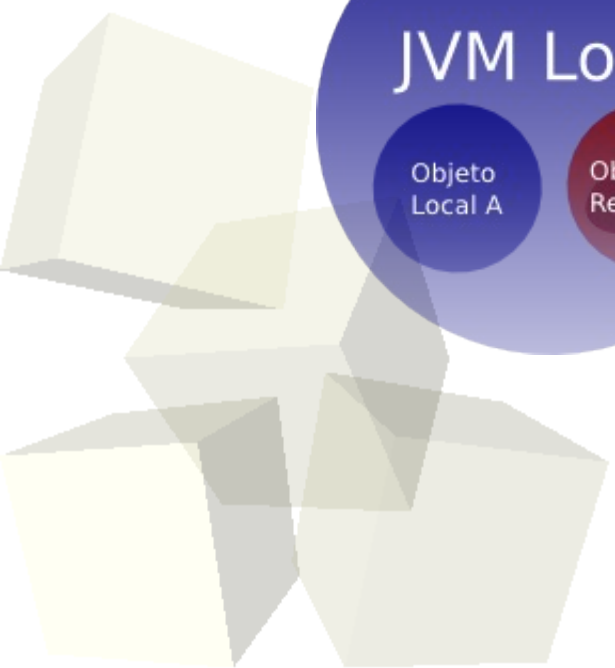
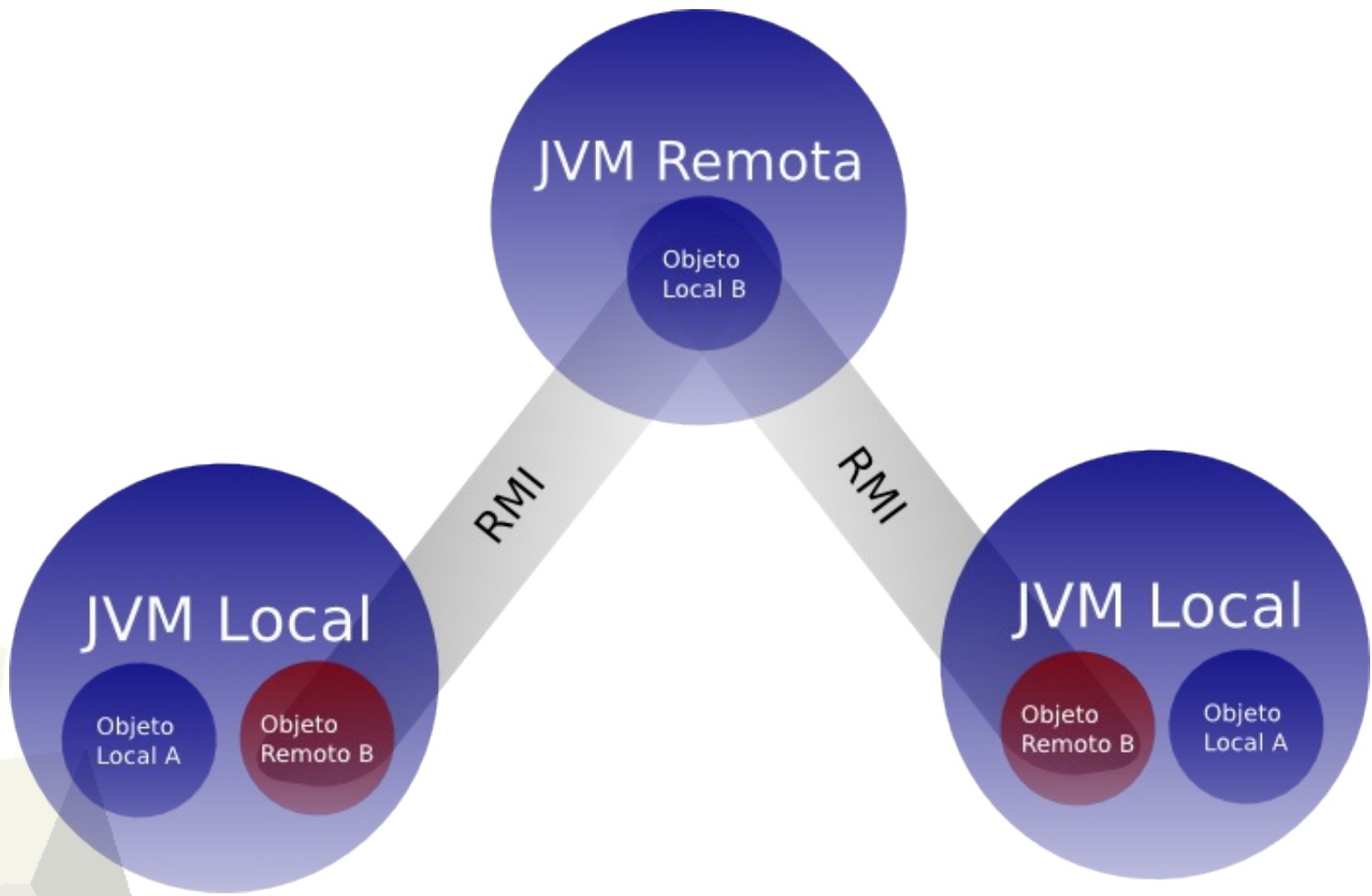


- Principais demandas:
  - ◆ Escalabilidade;
  - ◆ Extensibilidade;
  - ◆ Interoperabilidade.
  
- Principais padronizações:
  - ◆ RMI;
  - ◆ CORBA;
  - ◆ SOAP.
  
- Conhecimento:
  - ◆ Literatura;
  - ◆ Na ementa;
  - ◆ Casos de uso.



- RMI (“*Remote Method Invocation*”);
- Implementação em Java como uma Core API disponível na JVM da Sun;
- Abstração na localização dos objetos. Para o desenvolvedor e para a JVM, todos eles estão no host local;
- Ilusão para a JVM de um único espaço de memória com todo o código e dados necessários.

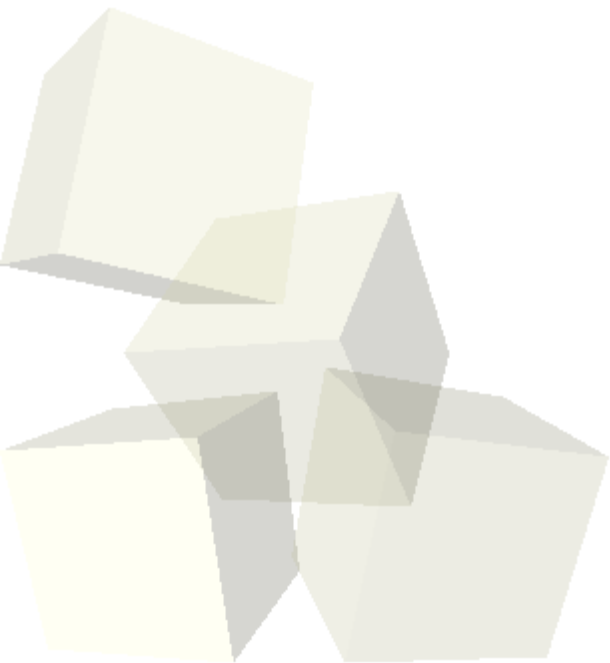






# RMI – Problemas que precisam ser resolvidos

- O servidor precisa de uma interface para expor seus métodos para JVMs remotas;
- O cliente precisa pensar que está acessando objetos locais;
- A passagem de objetos e variáveis por valor e por referência precisa ser resolvida.
  - ♦ Resposta: Serialização.





## ■ Serialização em Java:

### Classes

Tipos primitivos

Tipos básicos que implementam Serializable

Containers caso os objetos sejam serializáveis

Classes que herdam tipos serializáveis (Integer herda Number)

Exceções e erros

Containers, componentes e eventos AWT e Swing

Classes matemáticas (java.math)

Classes de reflexão (java.lang.reflect)

Adapter, filters e classes filhas

Streams, readers, writers e outras classes de I/O

### Serialização permitida

Sim

Sim

Sim

Sim

Sim

Sim

Sim

Não

Não

Não





Internet  
↕





## ■ Implementação:

- ♦ Definir interfaces de exposição remota;
- ♦ Escrever implementações da classe remota;
- ♦ Gerar arquivos Stubs da classe de implementação remota;
- ♦ Escrever inicialização do servidor;
- ♦ Escrever chamada à classe remota no cliente;

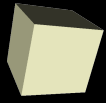
## ■ Execução:

- ♦ Iniciar serviço de registro no servidor;
- ♦ Iniciar criação do objeto a ser compartilhado e exposição dos métodos;
- ♦ Acessar o método remoto com o cliente (precisa da classe Stub gerada no servidor para a interface de exposição).

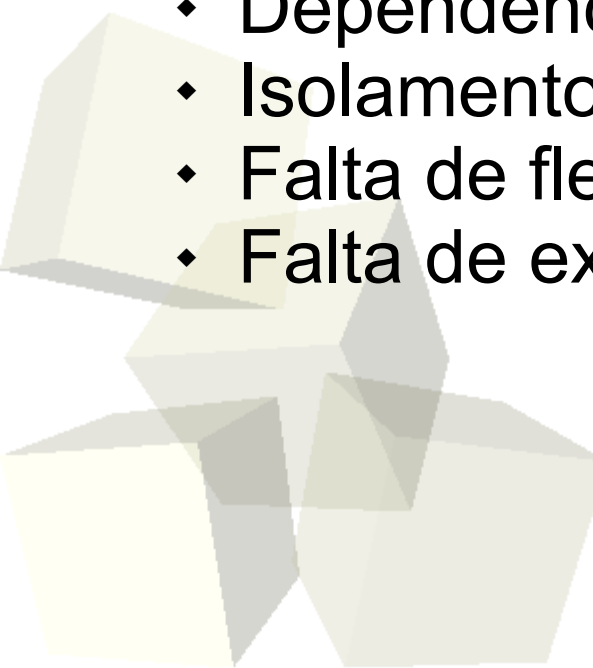


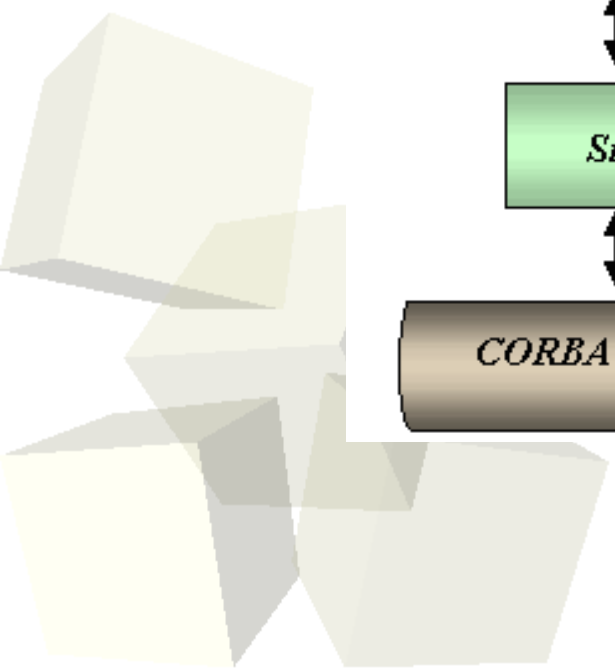
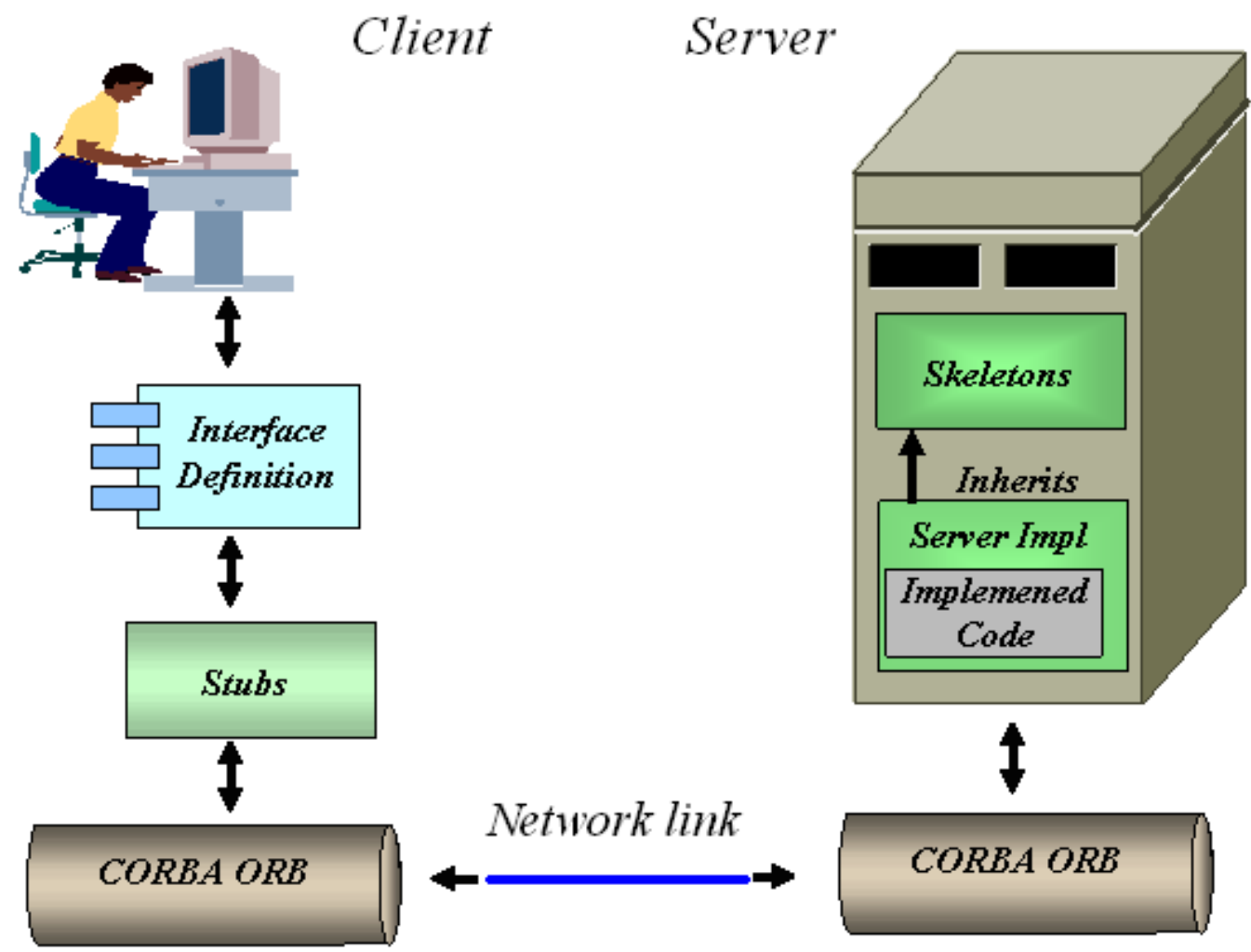
- CORBA (*“Common Object Request Broker Architecture”*)
- Especificação da OMG (*“Object Management Group”*) para sistemas distribuídos multi-plataforma, independente de fornecedor;
  - ◆ Outra especificação bastante conhecida: UML
- Possui muitas implementações, para várias linguagem e arquiteturas;
  - ◆ Inclusive para sistemas embarcados e sistemas em tempo real (CORBART);





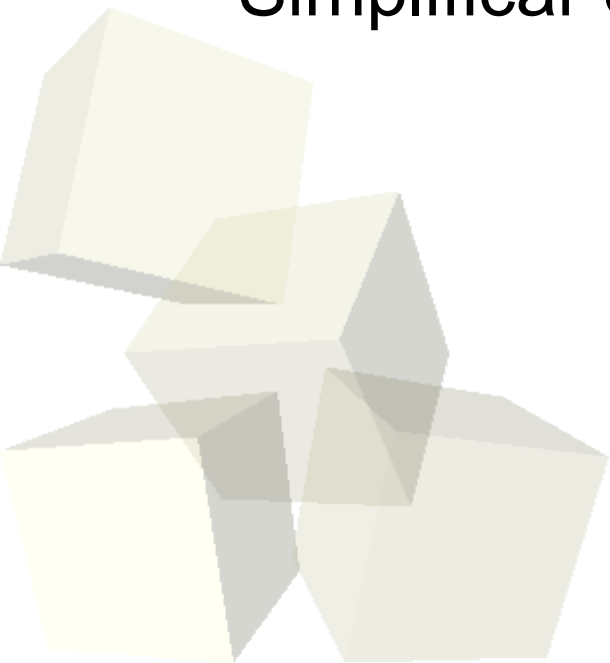
- Interfaces de exposição descritas em linguagem independente (IDL);
- Altamente extensível;
- Interface de comunicação (ORB) é modular, e faz a gerência de transporte e entrega;
- Problemas resolvidos com o CORBA:
  - ◆ Dependência de linguagem;
  - ◆ Dependência de sistema operacional;
  - ◆ Isolamento de tecnologias e sistemas legados;
  - ◆ Falta de flexibilidade na tipagem de dados;
  - ◆ Falta de extensibilidade.





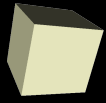


- SOA (*“Service-Oriented Architecture”*);
- Paradigma de exposição de recursos de execução remota de aplicativos de forma de serviços;
- Objetivos principais:
  - ◆ Desacoplamento total entre os componentes do sistema;
  - ◆ Simplificar overhead de infra-estrutura;





- Possui várias implementações, mas uma das mais usadas na Internet são os Web Services:
  - ♦ Baseado no protocolo HTTP para troca de mensagens;
  - ♦ Mensagens escritas em XML;
  
- Em Java, as soluções mais utilizadas são:
  - ♦ Projeto Metro, que é composto pelas APIs:
    - JAX-WS (“Java API for XML-based Web Services”);
    - JAX-RPC (“Java API for XML-based RCP”);
    - JAXB (“Java Architecture for XML Binding”)
  - ♦ Axis Webservices:
    - Tomcat (servidor de aplicações);
    - Apache Axis (exposição de classes como serviços);
    - Xerces (XML parser);



Fim

Contato

Página pessoal: <http://hypercast.info/>

E-mail: [liusfc@gmail.com](mailto:liusfc@gmail.com)

