



Trabalho de Graduação

**Construção de um Ambiente Distribuído e
Colaborativo para o projeto PDSCE**

Tiago Antônio Rizzetti

Curso de Ciência da Computação

Santa Maria, RS, Brasil

2006

Construção De Um Ambiente Distribuído E Colaborativo Para O Projeto PDSCE

por

Tiago Antônio Rizzetti

Trabalho de Graduação apresentado ao Curso de Ciência da Computação da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para a obtenção do grau de

Bacharel em Ciência da Computação.

Orientador: Prof. Ms. João Carlos Damasceno Lima

Co-orientador: Prof^a. Dr^a. Iara Augustin

Trabalho de Graduação nº 234

Santa Maria, RS, Brasil

2006.

Universidade Federal de Santa Maria
Centro de Tecnologia
Curso de Ciência da Computação

A Comissão Examinadora, abaixo assinada, aprova o Trabalho de Graduação

**CONSTRUÇÃO DE UM AMBIENTE DISTRIBUÍDO E COLABORATIVO
PARA O PROJETO PDSCE**

elaborado por
Tiago Antônio Rizzetti

como requisito parcial para obtenção do grau de Bacharel em Ciência da
Computação.

Comissão Examinadora:

Prof. Msc. João Carlos Damasceno Lima
(Orientador)

Prof^a . Dr^a . Iara Augustin
(Co-orientador)

Msc. Celio Trois

Prof^a . Msc. Oni Reasilvia Sichonany

Santa Maria, 01 de Março de 2007.

Agradecimentos

Quero expressar meu profundo reconhecimento e admiração a minha Mãe e ao meu Pai. Pela abnegação que tiveram, assim como pelo incentivo e apoio, sem os quais não teria conseguido trilhar essa fase da minha jornada.

A Fernanda, minha namorada, que sempre me incentivou, principalmente nos momentos difíceis, me ajudando a ver que tudo é possível. A compreensão e paciência, nos momentos conturbados, onde por vezes, dediquei muito tempo as tarefas, deixando o lado pessoal em segundo plano. E aos momentos felizes e as vitórias que conquistamos desde que nossas vidas se cruzaram.

Ao professor Caio, meu orientador, pela oportunidade de ingressar no grupo Gmob, à confiança depositada no meu trabalho e a oportunidade de me aprimorar, tanto profissionalmente quanto pessoalmente, através da convivência e dos trabalhos desenvolvidos no decorrer desses 2 anos que participo do grupo. Não poderia deixar de mencionar a professora Iara, minha co-orientadora, pela receptividade e ótimo ambiente de trabalho proporcionado no grupo, além da oportunidade e confiança depositada através da escolha para ingresso no Mestrado. Certamente, ambos contribuíram de forma decisiva para que essa fase da minha jornada fosse concluída, agradeço a vocês a oportunidade e ajuda. Que durante o mestrado, através do trabalho em conjunto, possamos ir além das expectativas.

Aos amigos de longa data, pelo companheirismo. Aos colegas, alguns dos quais se tornaram amigos. Certamente todos possibilitaram minha evolução, de uma forma ou de outra. Ao pessoal do Gmob, onde amizades se formaram. Aos professores pelos ensinamentos recebidos, e principalmente pelas lições, as vezes aparentemente bastante exigentes, mas sem dúvida necessárias.

A todos os mencionados e as demais pessoas que participaram dessa jornada, os meus mais sinceros agradecimentos.

Resumo

Trabalho de Graduação
Curso de Ciência da Computação
Universidade Federal de Santa Maria

Construção De Um Ambiente Distribuído E Colaborativo Para O Projeto PDSCE

Autor: Tiago Antônio Rizzetti

Orientador: Prof. Msc. João Carlos Damasceno Lima

Co-Orientador: Prof^a. Dr^a. Iara Augustin

Local e data da defesa: Santa Maria, 01 de Março de 2007.

Este trabalho visa construir um ambiente distributivo e colaborativo à aplicação epos-xconfig, no âmbito do projeto PDSCE, desenvolvido junto ao grupo LISHA/UFSC. Para alcançar tais objetivos, o trabalho é desenvolvido em duas etapas distintas: a) escolha, configuração e adaptação de uma ferramenta de gerenciamento de conteúdo (CMS) como meio para prover a distribuição da aplicação epos-xconfig via web; b) implementação de um sistema colaborativo, utilizando a tecnologia peer-to-peer (P2P) através da API JXTA, para disponibilizar um ambiente colaborativo junto à aplicação epos-xconfig.

Palavras-chave: PDSCE, CMS, P2P, OpenCMS, Ambiente Colaborativo, JXTA, Mensagens Instantâneas, SuperPeer, PCIM.

Lista De Abreviaturas

P2P	Peer-to-Peer
JXTA	Juxtapose
PDSCE	Plataforma de Desenvolvimento de Sistemas Computacionais Embarcados.
PCIM	PDSCE Collaborative Instant Messenger
CMS	Content Management System
API	Aplicattion Programming Interface
XML	Extensible Markup Language
IP	Internet Protocol
ID	Identifier
NAT	Network Address Translation
LAN	Local Area Network
HTTP	HyperText Transfer Protocol
TCP	Transmission Control Protocol
JCR	Java Content Repository
HTML	Hipertext Markup Language
WAP	Wireless Application Protocol
SSL	Secure Socket Layer
JSP	Java Server Page
JVM	Java Virtual Machine
JNLP	Java Network Launch Protocol

Glossário De Termos

Peer - Qualquer dispositivo participante de uma rede P2P habilitado a comunicar-se com a rede.

Rendezvous - Consiste num peer especial, atuando também como mecanismo de cache de mensagens para proporcionar funcionalidade e eficiência à rede P2P.

Broadcast – Transmissão emitada com destino a toda uma rede.

SuperPeer – Peer especial, que controla todo o serviço de indexação de recursos em uma rede P2P.

Lista De Figuras

Figura 1: ScreenShot da Ferramenta epos-xconfig.....	12
Figura 2: Diagrama Inicialização epos-xconfig com Portal.....	22
Figura 3: ScreenShot do Portal.....	23
Figura 4: Integração epos-xconfig com o Portal.....	24
Figura 5: Plataforma Colaborativa.....	27
Figura 6: Diagrama de Classes PCIM.....	30
Figura 7: Diagrama de Sequência SuperPeer.....	32
Figura 8: Diagrama de Sêquencia Inicialização PCIM Cliente.....	33
Figura 9: Duas Instâncias da aplicação PCIM, utilizando diferentes usuários, rodando no mesmo computador.....	34
Figura 10: Troca de mensagens entre duas instâncias da aplicação PCIM rodando no mesmo computador.....	36

Lista De Tabelas

Tabela 1: Tabela Comparativa das Ferramentas CMS avaliadas.....	21
---	----

Sumário

1. Introdução.....	1
2. Revisão Bibliográfica.....	4
2.1. Sistemas de Gerenciamento de Conteúdo Web (CMS).....	4
2.2. Tecnologias P2P.....	5
2.3. Plataforma JXTA.....	7
2.4. Tecnologia Java Web Start.....	9
2.5. O projeto PDSCE.....	10
2.6. Ferramenta Epos-xconfig.....	12
3. Ambiente de Distribuição WEB.....	14
3.1. Ferramentas CMS.....	14
3.1.1. Magnolia CMS.....	14
3.1.2. OpenCMS.....	15
3.1.3. JetSpeed.....	15
3.2. Requisitos.....	15
3.2.1. Características Obrigatórias.....	15
3.2.2. Características Desejáveis.....	17
3.3. Avaliação.....	17
3.3.1. Magnolia CMS.....	17
3.3.2. OpenCMS.....	19
3.3.3. JetSpeed.....	20
3.4. Resumo Comparativo.....	21
3.5. Integração da Ferramenta epos-xconfig ao Portal.....	21
3.6. Testes e Resultados.....	24
4. Plataforma Colaborativa.....	26
4.1. Arquitetura do Projeto epos-xconfig – Colaborativo.....	26
4.2. Funcionalidades do Modelo Colaborativo.....	27
4.2.1. Acesso a Rede Utilizando Credenciais do Portal.....	27
4.2.2. Comunicação na rede P2P.....	28
4.2.3. Envio de Arquivos ao Portal.....	28

4.2.4. Descarga (Download) de arquivos do Portal.....	29
4.3. Implementação da Aplicação PCIM.....	29
4.3.1. SuperPeer.....	29
4.3.2. ClientPeer.....	32
4.4. Testes e Resultados.....	33
5. Conclusão	37
Referências Bibliográficas.....	39

1. Introdução

A Internet torna-se a cada dia mais popular e, com isso, a quantidade de conteúdo disponível na rede aumenta consideravelmente. Com essa popularização não se pode esperar que todos os usuários tenham sólidos conhecimentos técnicos na área de informática para que possam produzir e publicar conteúdos. A construção e publicação desse conteúdo na web é uma tarefa que pode, e deve, passar por processos automatizados, incumbindo ao usuário apenas a tarefa de produzir o conteúdo, deixando a cargo destes programas sua estruturação, no formato de páginas web, assim como sua publicação em portais. Esse é o principal objetivo das ferramentas de gerenciamento, conhecidas como CMS [PEREIRA 2002], as quais propiciam que a publicação de conteúdos sejam feitas em portais Web, independentes dos meios, ou seja, usuários produzem conteúdo e através de interfaces amigáveis disponibilizam este na Web. Dessa forma, facilita-se tanto a produção, quanto melhora-se a qualidade visual e técnica dos sites, visto que esses são padronizados por ferramentas especializadas.

A Web também é um meio que nos possibilita fornecer mecanismos de colaboração entre usuários, acelerando o tempo de resposta, seja na implementação de projetos ou na distribuição de conteúdos. A colaboração é disponibilizada, em grande parte, através da utilização de ferramentas peer-to-peer (P2P) [KAMIENSKY, 2005]. As ferramentas P2P [FOX, 2001] possibilitam a desoneração da estrutura centralizada em prol da comunicação direta entre os clientes envolvidos na transação. Muitas ferramentas já foram desenvolvidas nesse sentido, para os mais diversos fins, desde compartilhamento de arquivos na rede até ferramentas que realizam o aproveitamento de recursos ociosos utilizando processamento distribuído.

Uma área emergente e com uma abrangência cada vez maior consiste na área de construção de sistemas embarcados [UFBA, 2004]. Estes por sua vez, são sistemas especializados e dedicados a funções específicas, tal como chips de controles de sensores, celulares, ou até mesmo brinquedos

eletrônicos. Com a redução de custos de componentes eletrônicos e sua miniaturização, sua utilização está se expandindo para as mais diversas áreas, produzindo com isso uma massificação de tais sistemas. Porém, cada sistema por ter um propósito único é um sistema diferente. Dessa forma sem mecanismos para reaproveitamento de código e especificações de hardware a tarefa construção de um novo sistema pode tornar-se muito onerosa.

Visando desenvolver mecanismos para facilitar a construção desses sistemas foi desenvolvido o projeto PDSCE (Plataforma de Desenvolvimento de Sistemas Computacionais Embarcados) [FRÖHLICH 2005]. Esse projeto consiste em propiciar ao usuário a especificação do sistema em um nível um pouco mais abstrato, deixando as características de implementação serem tratadas por ferramentas automatizadas. Através de uma ferramenta, chamada epos-xconfig o desenvolvedor desse sistema realiza sua especificação, a qual é validada pela aplicação; após concluída a fase de especificação é gerado um arquivo de configuração e, através de ferramentas automatizadas é construído o módulo de hardware; o projeto também desenvolveu um sistema operacional, modularizado, onde os módulos do sistema são agregados conforme necessário para formar um sistema operacional mínimo para o sistema embarcado especificado.

Por ser um projeto bastante abrangente, PDSCE envolve várias instituições, cada uma com um foco específico no âmbito do projeto. UFSC coordena e trabalha no desenvolvimento do sistema operacional modularizado, além da ferramenta epos-xconfig; PUC-RS e UNISINOS são responsáveis pela parte de hardware; e a UFSM que nesse contexto, trabalha com as ferramentas de apoio ao projeto. Essas ferramentas consistem em um ambiente de distribuição da ferramenta epos-xconfig na web, através de um portal web onde estão disponíveis informações sobre sistemas embarcados e a ferramenta epos-xconfig. Além de um meio de distribuição das informações na Web também é objetivo desenvolver um meio colaborativo no contexto da ferramenta epos-xconfig.

Este trabalho dedica-se portanto a dois objetivos principais: a) construir um portal Web, utilizando-se para isso uma ferramenta CMS, e utilizar esse

portal como meio de distribuição da ferramenta epos-xconfig; b) construir uma ferramenta colaborativa, para ser utilizada conjuntamente com a ferramenta epos-xconfig, proporcionando assim um ambiente colaborativo entre os usuários da ferramenta distribuída pelo portal.

O restante do texto está estruturado da seguinte forma: o capítulo 2 faz uma revisão dos principais conceitos e tecnologias utilizadas no desenvolvimento do trabalho; o capítulo 3 aborda o ambiente distribuído na Web utilizando uma ferramenta CMS para construção de um portal, assim como um comparativo entre algumas dessas ferramentas; o capítulo 4 faz a abordagem do modelo colaborativo proposto, bem como sua implementação; e por fim no capítulo 5 a conclusão sobre o trabalho é apresentada.

2. Revisão Bibliográfica

Este capítulo aborda os temas: gerenciamento de conteúdo web, computação peer-to-peer (P2P) e a plataforma JXTA, além do projeto PDSCE.

2.1. Sistemas De Gerenciamento De Conteúdo Web (CMS)

Content Management System (CMS) é a expressão utilizada para descrever ferramentas que promovem meios de gerenciamento, publicação e manutenção de informações [PEREIRA 2002]. Atualmente, estas ferramentas são utilizadas na construção de portais web, onde as características dos sistemas CMS vêm ao encontro das capacidades desejadas para tal sistema. Ferramentas CMS promovem um aumento de produtividade, referente a implementação de portais web, disponibilizando ferramentas que satisfazem as necessidades gerais de um portal. Além disso, em função de ser um sistema de gerência de conteúdo, pode ser aplicado aos mais diversos fins.

As ferramentas CMS possibilitam uma forma fácil e padronizada, através de *templates*, de publicar conteúdo na Web e distribuí-lo. Além disso, promove uma abstração do conteúdo a ser publicado, da forma como ele é publicado. Ou seja, para publicar informações através do portal, o usuário não necessita ter conhecimentos sobre qualquer linguagem de programação, é suficiente que ele utilize o conjunto de ferramentas que acompanham a ferramenta CMS. Dessa forma, por exemplo, membros que atuam em um projeto podem disponibilizar informações a respeito do seu trabalho, independente dos seus conhecimentos sobre linguagens de programação.

Outras características úteis incluem (i) estrutura para *workflow*, (ii) instâncias públicas e privadas para desta forma efetuar o controle de publicação de conteúdo, (iii) sistema de fórum, (iv) listas de discussões, (v) carga de arquivos, (vi) modelos para construção de páginas.

Existem ferramentas CMS baseadas nas mais variadas linguagens de programação, sendo que, freqüentemente encontram-se ferramentas baseadas na linguagem Java e ferramentas baseadas na linguagem PHP.

Foi realizada uma avaliação de algumas ferramentas CMS, baseadas na linguagem Java, sendo este assunto discutido no capítulo 3.

2.2. Tecnologias P2P

Para a elaboração deste projeto foram utilizadas tecnologias para desenvolvimento de sistemas distribuídos baseados em redes ponto a ponto, isto é, sistemas P2P. Esta tecnologia permite a criação de redes colaborativas descentralizadas.

A evidência da arquitetura P2P veio através dos programas de compartilhamento de arquivos na Internet. Há uma grande quantidade de programas desse tipo, tendo seu marco no programa Napster [KOVER 2000].

Ao contrário da arquitetura cliente/servidor, a arquitetura P2P busca a descentralização da computação e pode ser levada a qualquer tipo de recursos e dispositivos. A descentralização tem como principal vantagem o aproveitamento de uma pequena quantidade de recursos ociosos em cada máquina podendo fornecer grandes capacidades computacionais ao conjunto como um todo. Alguns projetos foram desenvolvidos para aproveitamento de recursos ociosos na resolução de problemas complexos; um exemplo bastante conhecido é o projeto SETI@Home [KORPELA 2001].

Porém, o modelo não hierárquico característico de sistemas P2P traz algumas complicações que não existem no modelo hierárquico e que devem ser resolvidas. Entre essas destacam-se: a descoberta de *peers*¹ e serviços na rede, o roteamento de mensagens, o gerenciamento de segurança e grupos de usuários.

Para contornar esses problemas desenvolveram-se técnicas já consolidadas para os sistemas P2P, como Super-Peer [TRAVERSAT 2006]. Nessa, tem-se um *peer* central onde os avisos de descoberta, grupos, e serviços da rede são indexados. Ou seja, o *peer* central funciona simplesmente como um índice para descoberta de serviços e *peers*, mas a comunicação efetivamente ocorre ponto-a-ponto entre os *peers* envolvidos sem

¹ *Peer* é a definição dada a qualquer equipamento final da rede P2P; pode ser um computador, um PDA ou qualquer outro equipamento pertencente a rede P2P.

interferência do *peer* central. Esse foi o modelo empregado na versão inicial do programa Napster, onde havia um banco de dados central indexando os arquivos aos *peers* nos quais eles se encontravam.

Na visão de muitos pesquisadores, a primeira abordagem não corresponde a um sistema P2P puro mas híbrida, visto que ainda há um *peer* central mesmo que somente utilizado para indexação de recursos aos *peers*. A abordagem P2P pura não admite essa centralização, e distribui através da rede a responsabilidade de roteamento e indexação de *peers*, serviços e conteúdo. Dessa forma, a rede distribui, de maneira assíncrona, o serviço de descoberta através de *peers* especiais chamados *rendezvous*. Um *rendezvous* pode ser qualquer *peer* da rede que esteja armazenando rotas e avisos que transitam pela rede e fornecendo esses dados aos *peers* que os requisitarem.

Ainda que na abordagem do serviço de indexação distribuída exista uma disseminação das informações da rede, existe a necessidade de se ter alguns *peers* estáticos que sirvam de entrada para a rede P2P. Do contrário, não há como um *peer* conectar-se a rede sem conhecer previamente ao menos um dos *rendezvous* disponíveis.

Uma das funcionalidades mais importantes para a computação P2P é a descoberta de serviços, *peers*, e outros recursos da rede P2P. Há basicamente duas formas de efetuar a descoberta de avisos² na rede, cada qual adequada a determinadas situações [WILSON 2002]:

- Descoberta Direta: utilizada em redes locais onde há suporte nativo a *broadcast*, não sendo necessário qualquer *peer* roteador ou *rendezvous* para realizar a descoberta;
- Descoberta Indireta: quando um computador dentro de uma LAN deseja acessar um sistema não pertencente à LAN há necessidade de um *rendezvous* que faça a ligação entre a rede externa e o(s) *peer(s)* interno(s) da LAN. Esse *rendezvous* deve ser previamente conhecido. O *peer rendezvous* pode utilizar duas estratégias para a descoberta de serviços: propagar a mensagem para outros *rendezvous* ou guardar os

² Define-se por avisos mensagens de descoberta de serviço, *peers*, grupos e qualquer outro recurso de uma rede P2P.

avisos que trafegam na sua rede em uma cache e acessá-la quando consultado.

Existem outras questões que devem ser cuidadosamente avaliadas na descoberta de serviço. Uma delas é a derivada da introdução de sistemas de *firewall* presentes na imensa maioria das organizações, visto que eles podem bloquear protocolos e/ou portas utilizadas pelas redes P2P. Outra dificuldade é gerada pelo serviço NAT (*Network Address Translation*) onde os computadores da rede interna são dotados de números IP privados e os sistemas externos à rede não tem acesso direto; portanto, sendo impossível iniciar uma conexão externa a menos que o computador da rede interna tenha começado a comunicação. As soluções para essas questões são a utilização de um protocolo e portas padrões normalmente liberadas. Por exemplo, HTTP e porta 80 respectivamente, no caso do *firewall*. No caso da solução para o problema do NAT, foi fazer com que o *peer* interno da rede acesse o seu *rendezvous*, em pequenos intervalos de tempo, verificando se existe alguma mensagem que ele deva receber.

2.3. Plataforma JXTA

O projeto JXTA [JXTA.ORG 2006], idealizado pela Sun Microsystems, tem como principal objetivo fornecer uma plataforma base para sistemas P2P. Os conceitos chaves da arquitetura P2P já são bastante conhecidos, portanto, a existência de uma plataforma que forneça os serviços básicos auxilia o desenvolvedor, pois ele não precisa se preocupar com os aspectos de mais baixo nível da rede.

A plataforma JXTA, através de um conjunto básico de protocolos, fornece flexibilidade e robustez para o desenvolvimento de sistemas P2P. O desenvolvedor, a menos que explicitamente necessário, não precisa se preocupar com os aspectos de rede, como protocolos de rede e de enlace de dados. O conjunto de protocolos básicos que fornecem o núcleo de sistemas P2P na plataforma JXTA são:

- Protocolo de Descoberta de *Peers* (*peer discovery*): possibilita a descoberta de serviços fornecidos pelos *peers* na rede P2P;

- Protocolo de Resolução de *Peers* (*peer resolver*): envio e processamento de requisições gerais;
- Protocolo de Encontro (*rendezvous*): trata a propagação de mensagens na rede P2P;
- Protocolo de Informações do *Peer*: permite uma maneira de obter as informações de estado de um *peer* da rede;
- Protocolo de tunelamento (*Pipe Binding*): promove um canal de comunicação ponto-a-ponto;
- Protocolo de Roteamento Final (*EndPoint Routing*): Promove um conjunto de mensagens utilizadas para roteamento entre *peer* origem e destino.

A Plataforma JXTA é genérica, independente de linguagem de programação. Implementações da plataforma podem ser encontradas nas linguagens Java e C, por exemplo. O controle e comunicação entre os protocolos básicos do JXTA se dão por mensagens XML, independentes de sistema operacional, topologia ou tipo de rede. Sendo, portanto, portátil para muitos sistemas.

A robustez do JXTA está exatamente nessa generalidade do sistema, no uso do XML para troca de mensagens de coordenação da rede, na independência de rede e de linguagem somando-se a construção de um núcleo bem consolidado de funções básicas promovidas pela rede P2P. Como consequência direta, tem-se maior facilidade na concepção de sistemas P2P assim como reusabilidade do código.

Este núcleo básico construído na plataforma JXTA refere-se a manipulação de todos os objetos constituintes da rede: grupos de *peers*; transporte de rede (*pipes, endpoints, mensagens*); avisos (*Advertisements*); IDs (identidades); protocolos (descoberta, comunicação, monitoramento); primitivas de segurança e autenticação.

Um aspecto que deve ser ressaltado é a identificação de recursos e acesso à rede JXTA. Todos os recursos da rede P2P em geral, recebem um

identificador único (ID) utilizado para indexá-lo de maneira única na rede. Dessa forma, o primeiro passo na conexão de um *peer* na rede é a criação de um ID, dito *PeerID*, que será o identificador do *peer* na rede. Ainda, além de criá-lo é essencial publicar esse recurso para que os demais *peers* da rede conheçam seu *PeerID* e possam, quando necessário, realizar a comunicação com o *peer* em questão.

Além disso, na inicialização da rede P2P, é necessário que o *peer* seja acrescentado a um grupo padrão, no caso o *NetPeerGroup*. Recorrendo a instância do serviço de descoberta (*discovery*) fornecido pelo *NetPeerGroup* o usuário pode enviar e receber avisos (*Advertisements*) através da rede P2P.

A partir do *NetPeerGroup* o usuário pode utilizar serviços, criar ou entrar em grupos, enviar mensagens para outros *peers*, listar *peers* do grupo, listar grupos na rede, enfim, usufruir de todos os serviços disponibilizados pela rede P2P.

2.4. Tecnologia Java Web Start

Essa é um tecnologia desenvolvida pela *Sun Microsystems* que permite que aplicações remotas sejam executadas no computador do usuário como se fossem aplicações locais. O Java Web Start [SUN 2006] possibilita também a atualização automática, quando necessário, do ambiente de execução Java, sendo essa informação disponibilizada no arquivo de lançamento do Java Web Start. Para lançamento destas aplicações, é utilizado o protocolo JNLP (*Java Network Launch Protocol*), que contém toda configuração necessária a execução da ferramenta, tais como: bibliotecas necessárias (jars), arquivo principal e classe principal. A seguir um exemplo de um arquivo jnlp que efetua o lançamento de uma aplicação chamada Notepad.

```
<?xml version="1.0" encoding="utf-8"?>
<jnlp spec="1.0"
    codebase="URL of application on your Web server"
    href="Notepad.jnlp">
<information>
```

```
<title>Notepad Demo</title>

<vendor>Sun Microsystems, Inc.</vendor>

<offline-allowed/>

</information>

<resources>

    <jar href="Notepad.jar"/>

    <j2se version="1.3+"

        href="http://java.sun.com/products/autodl/j2se"/>

</resources>

<application-desc main-class="Notepad"/>

</jnlp>
```

Essa tecnologia fornece vantagens para todos os envolvidos, tanto desenvolvedor quanto usuário. Do ponto de vista do usuário, não é necessário efetuar processos de instalação nem tampouco configuração, o usuário executa a ferramenta através de um simples *click* em um *link* disponibilizado em uma página na Web. A aplicação pode disponibilizar consigo todas as bibliotecas necessárias, carregadas através do Java Web Start, sendo uma operação transparente ao usuário. Quanto ao ponto de vista do desenvolvedor, há a flexibilidade proporcionada na migração de ferramentas legadas para o ambiente Web, sendo normalmente um procedimento trivial através da utilização dessa tecnologia. Além disso, através da atualização, da ferramenta, no site que à disponibiliza, o desenvolvedor garante que os usuários sempre utilizarão a ferramenta atualizada.

2.5. O Projeto PDSCE

Este trabalho desenvolve-se como parte do projeto PDSCE (Plataforma de Desenvolvimento de Sistemas Computacionais Embarcados) [FRÖHLICH 2005] que visa automatizar o desenvolvimento de sistemas computacionais dedicados. O projeto PDSCE é um esforço multi-institucional que agrega diversas universidades da região sul: UFSM, UFSC, PUC-RS e Unisinos. O

foco é no desenvolvimento de ferramentas que facilitem e permita a elaboração de sistemas embarcados, isto é, utilizam-se técnicas de *co-design* para produzir um hardware e um software de tamanho mínimo para atender uma finalidade específica.

O projeto pode ser dividido em três frentes de trabalho distintas:

- **Sistemas Operacionais:** Através do desenvolvimento de um sistema operacional modularizado, busca fornecer um conjunto de módulos de sistema operacional que possam suprir as necessidades da maioria dos sistemas embarcados, dessa forma, quando da especificação de um sistema embarcado os módulos necessários para fornecer um sistema operacional a esse dispositivo serão agregados formando um sistema operacional mínimo e funcional. A grande vantagem nesse caso é o reaproveitamento de código e o baixo tempo necessário no desenvolvimento.

- **Hardware:** Da mesma forma, busca-se construir alguns componentes de hardware, mais largamente empregados na construção de sistemas embarcados, para que possam se utilizados nos novos dispositivos projetados.

- **Ferramentas de Apoio:** um dos objetivos do projeto é construir um Portal Web, construído a partir de uma ferramenta de gerenciamento de conteúdo (CMS), onde esteja disponível a ferramenta epos-xconfig [FRÖHLICH 2006]. A ferramenta epos-xconfig também deve interagir com o Portal, fazendo upload e download de arquivos de configuração de/e para este. Além disso, nesse Portal devem estar disponíveis informações relevantes para construção de sistemas embarcados. Também buscando melhorar a integração entre os usuários da ferramenta epos-xconfig, esta deve possuir um mecanismo colaborativo para interação entre os projetistas de sistemas embarcados.

No contexto das ferramentas de apoio, a construção deste Portal, assim como apresentação de uma solução colaborativa utilizando o ambiente peer-

to-peer (P2P) para integração dos usuários da ferramenta epos-xconfig é objeto de estudo deste trabalho.

2.6. Ferramenta Epos-xconfig

É através dessa ferramenta (Figura 1) que é possível a especificação do sistema embarcado. Implementada na linguagem java, é portátil, podendo ser utilizada tanto em ambiente Linux quanto Windows. A ferramenta gera o arquivo de descrição do sistema embarcado no formato XML.

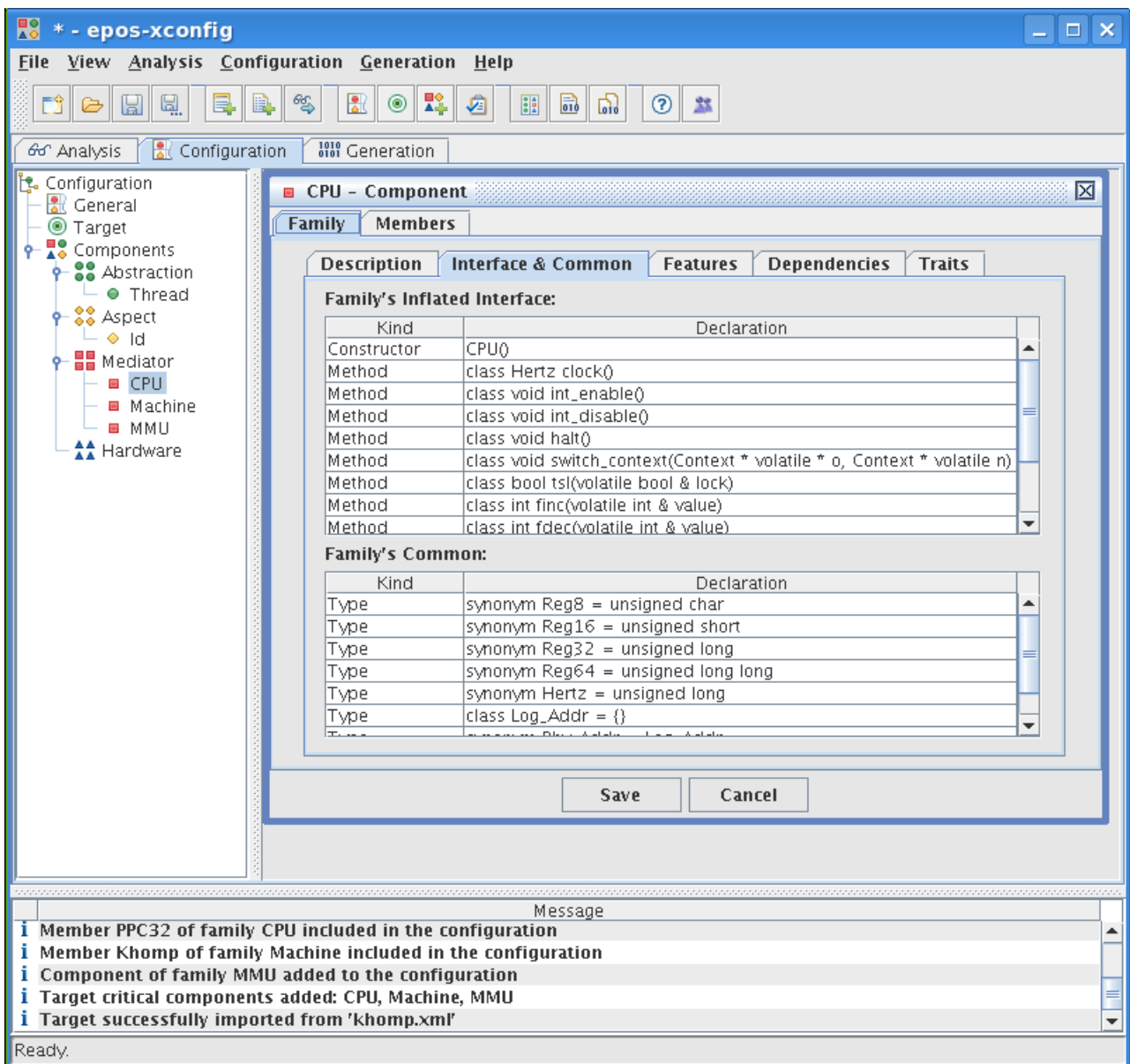


Figura 1: ScreenShot da Ferramenta epos-xconfig

As modificações nessa ferramenta consistem em possibilitar a interação dela com o Portal lendo e escrevendo neste os arquivos XML utilizados pela ferramenta, bem como carregar do Portal o repositório de componentes disponíveis na ferramenta. Também, através desta, deve ser executada uma outra ferramenta que forneça o ambiente colaborativo, no caso, o PCIM que será tema do capítulo 4.

3. Ambiente De Distribuição WEB

Na primeira fase deste trabalho, tem-se a construção de um Portal web que possibilita aos usuários fornecerem detalhes sobre o sistema embarcado requisitado, e então, através de processos manuais ou automatizados, realizar a construção do sistema operacional. O Portal foi desenvolvido com o auxílio de uma ferramenta CMS (*Content Management System*).

Buscando optar pela ferramenta mais adequada às necessidades do Portal do projeto, avaliou-se algumas ferramentas CMS e escolheu-se a mais indicada para construção do Portal, baseando-se nos requisitos necessários ao Portal.

3.1. Ferramentas CMS

Foram escolhidas três ferramentas CMS para serem instaladas e avaliadas com vistas à análise e escolha da ferramenta mais adequada ao contexto do projeto PDSCE. As ferramentas escolhidas para análise são: Magnolia CMS (versão 2.0), OpenCMS (versão 6.0) e JetSpeed (versão 2.0). Essa escolha se deve ao fato de que todas utilizam a linguagem Java, serem implementações de código aberto (*opensource*) e figurarem entre as ferramentas CMS mais conhecidas.

3.1.1. Magnolia CMS

Desenvolvido pela empresa Obinary, Magnolia [Magnolia, 2005] consiste na primeira ferramenta CMS de código aberto a estar de acordo com a API JCR (*Java Content Repository*) [Barik, 2005].

Essa ferramenta não utiliza qualquer banco de dados externo, ou seja, é autocontido e de fácil instalação. Além disso, utiliza a tecnologia Java e XML e uma API orientada à facilidade de uso, além de propiciar acesso transparente e uniforme a diversos repositórios de dados e ser disponibilizada para os sistemas operacionais mais utilizados. Também, utiliza o servidor *servlet container*, Tomcat produzido pela *Apache Foundation* [Apache, 2005].

3.1.2. OpenCMS

Desenvolvido pela Alkacon Software [OpenCMS, 2005], consiste em uma ferramenta CMS baseada na tecnologia Java e XML, de nível profissional para *sites* que necessitam de sistemas gerenciadores de conteúdo. Possui uma estrutura modular permitindo conectar e desconectar módulos ao sistema.

Possui uma interface web completa para administração da ferramenta, permitindo desde a criação de páginas HTML simples, até a importação ou exportação de módulos do sistema.

3.1.3. JetSpeed

Constitui-se na próxima geração de portais produzidos pela *Apache Foundation*, conhecida por seus produtos amplamente utilizados, como o servidor apache.

Utiliza também a tecnologia Java e XML e, além disso, propicia a disseminação de conteúdo para diversos tipos de dispositivos, como por exemplo, WAP para celulares, *Web Browsers*, e *paggers*.

Pode ser considerado como uma central de distribuição de informações para os mais variados dispositivos e meios. Assim, permite a construção de portais de maneira rápida.

3.2. Requisitos

Dentro do contexto do projeto PDSCE, têm-se algumas características que devem ser avaliadas para que se possa escolher a ferramenta CMS mais adequada para a criação do Portal. Essas características subdividem-se em duas categorias: obrigatórias e desejáveis.

3.2.1. Características Obrigatórias

- **Sistema de Edição de Conteúdo:** Sua utilização no contexto do projeto está relacionada ao objetivo de se utilizar o Portal também como um meio de distribuição de conteúdo. Assim, todos os membros atuantes no projeto, de alguma forma, poderão publicar conteúdos de relevância e torná-los de conhecimento público. A identificação se faz necessária

por razões de segurança.

- **Agenda (*jobs*):** O objetivo do Portal é promover uma interface web onde os usuários possam especificar o sistema que atende as suas necessidades. Em um momento inicial alguém deverá fazer a construção desse sistema de maneira manual. Nesse aspecto encaixa-se a característica de agenda onde é possível organizar um cronograma para realização desta tarefa. Além disso, para uma próxima etapa, a construção do sistema deverá ser realizada de forma automatizada.

- **Fórum:** É utilizado como uma ferramenta para usuários e desenvolvedores do projeto interagirem entre si, promovendo discussões sobre o projeto. Dessa forma, tem-se uma forma de documentação *online*, ou seja, dúvidas ou discussões referentes ao projeto ficarão documentadas e disponíveis para consulta, tanto a integrantes do projeto, quanto ao público em geral.

- **News:** Sua utilidade é justamente informar aos usuários do sistema todo o progresso realizado, ou divulgar qualquer outra informação de cunho relevante.

- **SSL:** Um requisito básico à segurança das informações trocadas no *site; pois*, através de requisições feitas pelo site serão construídos sistemas embarcados, não podendo existir dúvidas sobre a autenticidade da requisição.

- **Carga (*upload*) de Arquivos:** Uma característica útil à medida que propicia que pessoas envolvidas no projeto disponibilizem arquivos através do site, tanto para acesso público, quanto para acesso interno aos demais interessados no desenvolvimento do sistema.

- **Modelos (*Templates*):** Tem como objetivo promover agilidade e flexibilidade na construção e manutenção do Portal. A flexibilidade é conseqüência da liberdade de se implementar diversos modelos que atendam diferentes necessidades do Portal, e, através da utilização desses modelos, promover a agilidade na manutenção.

3.2.2. Características Desejáveis

As características desejáveis incluem ferramentas de *Workflow*, as quais permitem uma maior organização na delegação de tarefas; por exemplo, o pedido de um sistema específico envolve, em sua realização, mais de uma pessoa e, além disso, as sub tarefas desenvolvidas por cada uma das pessoas tendem a ser interdependentes. Nesse contexto, é útil que as pessoas envolvidas no processo de realização da tarefa recebam informações sobre em que fase está o desenvolvimento e quem é o responsável pela etapa que está sendo realizada.

3.3. Avaliação

A metodologia utilizada para a avaliação foi à análise de uso. Todas as ferramentas foram instaladas e avaliadas através da interação direta com o sistema. Verificou-se as características disponíveis em cada uma delas e o nível de dificuldade apresentado para sua utilização. As instalações foram realizadas no sistema operacional Linux, nas distribuições Slackware, Debian e Knoppix.

Todas as ferramentas analisadas executam sobre o servidor Tomcat. Para o OpenCMS e o JetSpeed foi utilizado o Tomcat 5.5.9; já o Magnolia CMS traz consigo o servidor Tomcat 5.0.24. Esse *Servlet Container* possui boa documentação *online* que facilita sua instalação e configuração.

A seguir é discutida a apresentação dos resultados obtidos na experiência de avaliação de cada uma destas ferramentas.

3.3.1. Magnolia CMS

Como característica mais marcante tem-se a facilidade de uso.

- **Instalação:** O Magnolia CMS 2.0 apresenta um instalador de uso extremamente intuitivo com uma interface bastante amigável. Basta baixá-lo e executar o programa de instalação que está disponível para plataforma Linux, Unix, Windows e Mac. O instalador já traz consigo as ferramentas necessárias para o Magnolia rodar, dentre elas o servidor Tomcat 5.0.24.

- **Configuração e Execução:** Não há necessidade de configurações adicionais, exceto configurar a variável de ambiente \$JAVA_HOME para que o *script* de execução do Magnolia possa executar encontrando o caminho para o *Java runtime*. O Magnolia possui duas instâncias: instância pública e instância autor. Ambas são idênticas exceto pelo conteúdo não publicado contido na instância do autor. Dessa forma, faz-se necessário à ativação de ambas as instâncias através de seus *scripts* de inicialização. Por padrão, a instância autor roda na porta 8181 e a instância pública roda na porta 8182.

- **Características:** Possui uma interface Web muito amigável, mecanismos eficientes e práticos para inclusão de conteúdo, permitindo assim, que qualquer usuário possa incluir ou modificar conteúdo. Na instância em modo autor, onde são permitidas as alterações, as páginas são exibidas tal como no modo usuário; porém, existem os controles para alterar, excluir e editar permitindo que o conteúdo seja manipulado. Essas características são conseguidas através dos *servlets* que rodam no lado servidor.

Existe ainda uma interface de administração do CMS, onde se pode criar novas páginas, modificar a estruturação de páginas, incluir usuários, e realizar todas as demais configurações necessárias ao portal. O controle de segurança é granular, ou seja, existe o conceito de regras das quais os usuários fazem parte e essas regras são associadas às páginas, permitindo ou restringindo o acesso aos usuários vinculados à regra em questão. Também é através da administração que se pode ativar ou desativar conteúdo, ou seja, copiar o conteúdo disponível da instância autor para a instância pública através da ativação ou desativação de páginas. Pode-se, inclusive, executar esse controle em cada nó da estrutura e, por conseguinte, aplicá-lo a todos os nós filhos deste.

Apesar de suas vantagens quanto à facilidade de uso, o Magnolia possui algumas deficiências: não possui estrutura para *workflow*, fórum, ou listas de discussão.

- **Documentação:** Apesar de ser uma ferramenta de uso relativamente fácil existe pouca documentação gratuita disponível no *site* da ferramenta. Existe uma documentação mais completa para essa ferramenta CMS, porém é paga.

3.3.2. OpenCMS

Destacam-se os aspectos: ferramenta ampla e interface amigável.

- **Instalação:** A instalação do OpenCMS, apesar de não ser tão simplificada quanto na ferramenta anterior, não apresenta dificuldades. A ferramenta está disponível como um *Web Archive* (.war) e deve ser extraída em um *servlet container* tal como Tomcat. Após isso, é necessário executar o *setup* que é intuitivo. Além disso, por padrão o OpenCMS utiliza como banco de dados o MySQL, sendo necessário sua instalação separadamente. Há suporte a outros bancos de dados também, mas o MySQL, em virtude de ser livre, é o mais utilizado.

- **Configuração e Execução:** Como o OpenCMS está dentre as ferramentas carregadas automaticamente pelo Tomcat, após a instalação, é necessário apenas a inicialização do próprio Tomcat. Como único passo adicional, deve-se configurar a variável de ambiente \$JAVA_HOME para que o *script* de execução do Tomcat execute corretamente.

- **Características:** Possui mecanismo de *workflow*, tarefas e *news*. Suporta SSL (*Secure Socket Layer*), pois o servidor Tomcat no qual roda o CMS tem suporte a esse protocolo. Ainda, tem-se uma interface web completa no que diz respeito a configurações da ferramenta permitindo acesso a mecanismos fáceis e rápidos para publicação de conteúdo. Seu esquema de *templates* possibilita a rápida criação de páginas visualmente atrativas e enxutas. Ao se editar uma página, esta estará disponível apenas aos usuários autenticados, a menos que se tenha explicitamente publicado. Dessa forma, pode-se criar versões *offline* do

portal, tendo apenas uma versão publicada (*online*). É importante ressaltar que o OpenCMS possui estrutura modularizada, permitindo a inserção ou remoção de módulos de acordo com as necessidades e propósitos estabelecidos.

- **Documentação:** Possui boa documentação disponível através do *site* da ferramenta.

3.3.3. JetSpeed

Destaca-se o aspecto: flexibilidade, porém com dificuldade de uso.

- **Instalação:** A versão 2.0 é distribuída com um pacote onde tem-se o Tomcat 5.5.9 já configurado para rodar o JetSpeed. A instalação é facilitada, não sendo necessário recurso adicional exceto o *java toolkit*. Como banco de dados por padrão é utilizado o HyperSonic Database, que também vem como parte do pacote da ferramenta. Outros bancos de dados também podem ser utilizados, mediante esforços adicionais de configuração.

- **Configuração e Execução:** Necessário apenas a configuração do Tomcat para se utilizar o protocolo SSL. Para executar a ferramenta inicializa-se o servidor Tomcat, e a ferramenta é automaticamente carregada em sua inicialização. Como nas ferramentas anteriores, deve-se configurar a variável de sistema \$JAVA_HOME.

- **Características:** Ao contrário das ferramentas apresentadas anteriormente, esta não possui uma interface web tão amigável e interativa. Os desenvolvedores da ferramenta, optaram pela simplicidade visual em prol da capacidade e flexibilidade dada ao usuário. Visando flexibilidade, a ferramenta apresenta um *framework* que permite a execução de páginas utilizando as linguagens Perl, JSP ou PHP. A ferramenta em si não traz sistemas de *news*, fórum, *workflow* ou editores web para edição de páginas.

- **Documentação:** O *site* do desenvolvedor apresenta uma documentação razoável, ainda que o uso da ferramenta não seja muito

amigável.

3.4. Resumo Comparativo

Uma característica comum a todas as ferramentas analisadas é o mecanismo de controle de acesso. Todas elas utilizam mecanismos semelhantes no processo de validação de permissões de usuários, utilizando-se de regras onde os usuários são vinculados. Dessa forma, as páginas são vinculadas às regras e não diretamente ao usuário. Sendo assim, o processo de estabelecer permissões sobre determinada área do portal a um determinado número de usuários é bastante facilitado, sendo necessário apenas vincular esses usuários a uma regra, e esta, às páginas desejadas.

A Tabela 1 apresenta um sumário das questões discutidas acima para efeito comparativo.

	Edição de Conteúdo	Fórum	Jobs	News	SSL	Upload	Templates	WorkFlow
Magnolia CMS	Sim	Não	Não	Parcial	Sim	Sim	Sim	Não
OpenCMS	Sim	Não	Sim	Sim	Sim	Sim	Sim	Sim
JetSpeed	Não	Não	Não	Não	Sim	Não	Sim	Não

Tabela 1: Tabela Comparativa das Ferramentas CMS avaliadas

Sendo assim, o Portal do projeto PDSCE foi desenvolvido com a ferramenta OpenCMS, porque agrega de forma transparente as características de segurança e a interação entre os usuários. Além disso, dentre as ferramentas analisadas, essa foi a que demonstrou maior adequação aos requisitos estabelecidos no projeto.

3.5. Integração Da Ferramenta Epos-xconfig Ao Portal

O Portal é o ponto de partida para toda a interação e utilização do sistema por parte do usuário. É através dele que o usuário pode acessar o conteúdo disponível sobre sistemas embarcados, inclusive arquivos de configuração utilizados pela ferramenta epos-xconfig. Também é através do Portal, que é possível lançar a execução da ferramenta epos-xconfig, sendo inclusive esse, um dos principais objetivos do Portal. Verificou-se que a

melhor forma de alcançar este objetivo é através da tecnologia Java Web Start. Essa tecnologia permite que aplicativos disponibilizados na web sejam executados como se fossem aplicativos locais em qualquer computador remoto que possua a Java Virtual Machine (JVM). A aplicação é lançada através do Portal onde ela se encontra, e então, executada localmente no computador solicitante. A implementação de segurança do Java Web Start não permite que sejam executados aplicativos remotos que não estejam digitalmente assinados. Assim, o usuário não precisa preocupar-se com a segurança, visto que essa é uma forma de assegurar que a aplicação é autêntica. Existem ainda funcionalidades visando o desempenho, pois o Java WebStart trabalha com o conceito de cache, mantendo no host cliente a cópia da aplicação obtida remotamente. Essa cache é automaticamente substituída assim que houver uma nova versão da aplicação no Portal.

Na Figura 2 há um diagrama que ilustra o processo de inicialização da ferramenta epos-xconfig com o Portal.

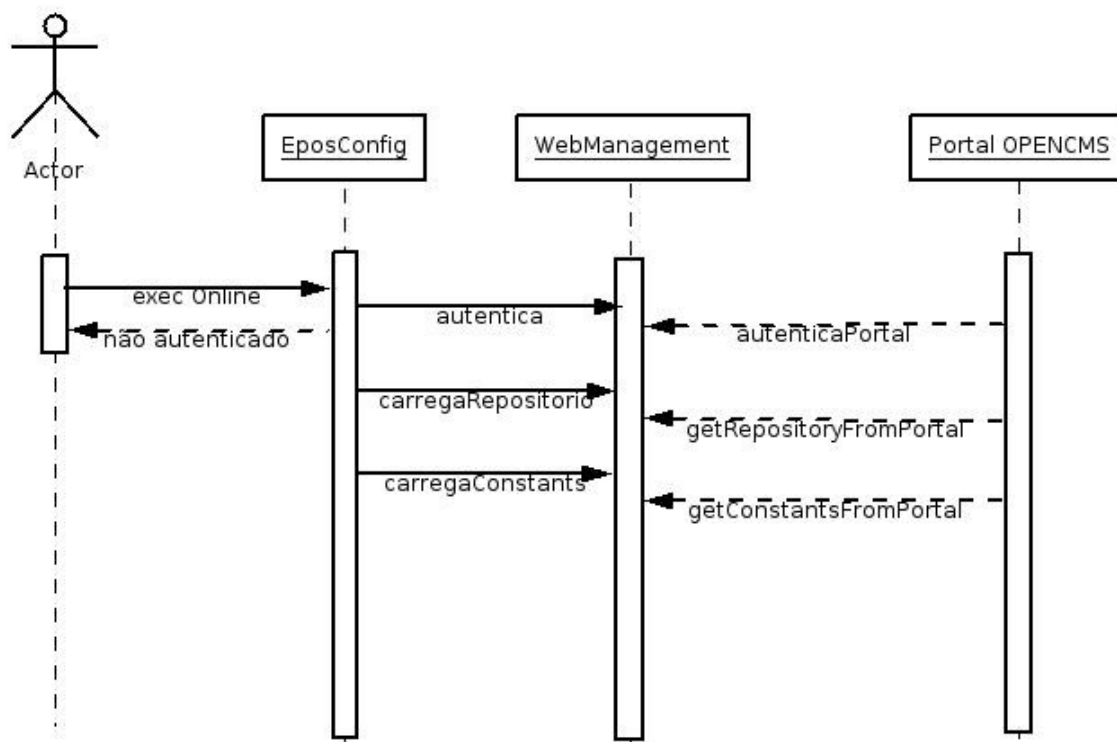


Figura 2: Diagrama Inicialização epos-xconfig com Portal

A utilização da tecnologia Java Web Start ainda minimiza o impacto de alterações de versão na ferramenta epos-xconfig, visto que quando esta é alterada, basta assinar digitalmente os arquivos *.jar* correspondentes à aplicação e, novamente, disponibilizá-los no Portal. Na figura Figura 3 pode ser vista a imagem do Portal.

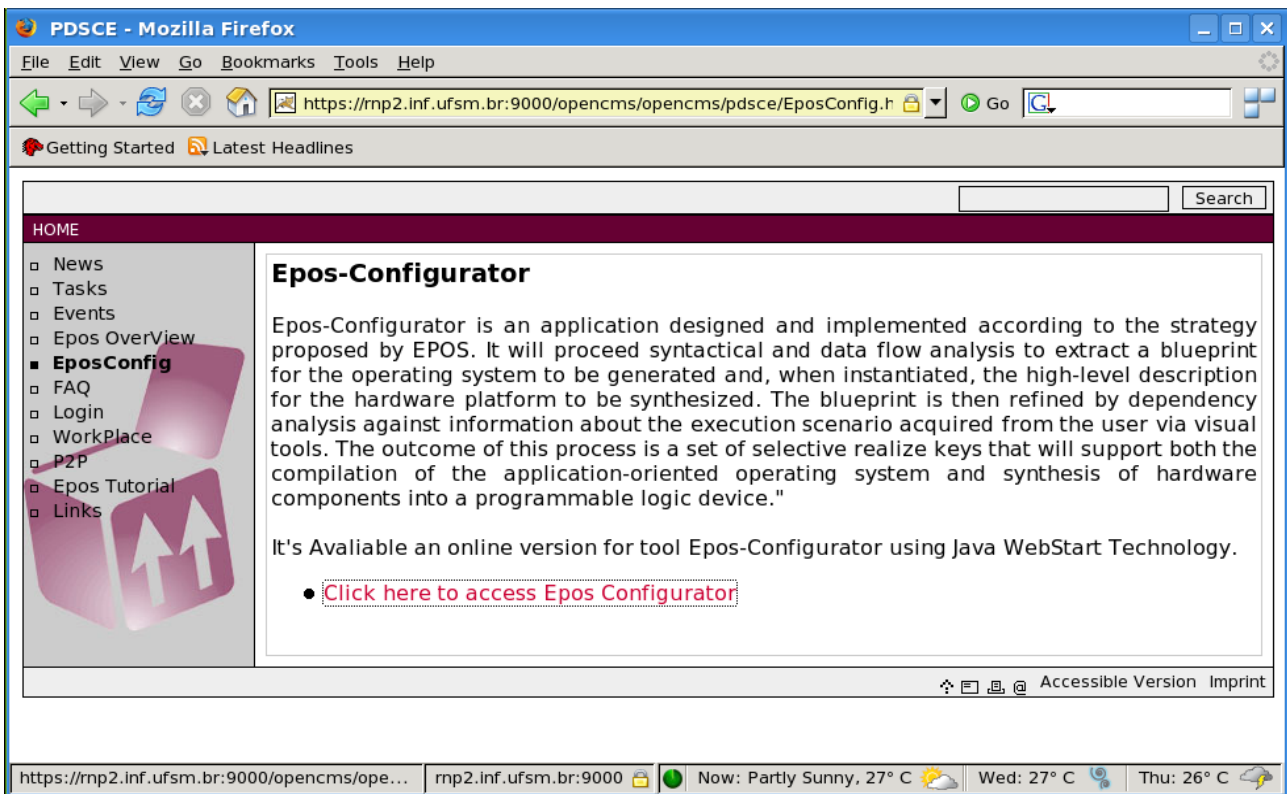


Figura 3: ScreenShot do Portal

A interação propriamente dita da aplicação epos-xconfig com o Portal ocorre de duas formas: *Upload* e *Download* do arquivo de configuração produzido e editado pela ferramenta epos-xconfig. Para realizar operações remotamente, o usuário deve fornecer as credenciais de acesso ao Portal. Se a autenticação for bem sucedida, é possível ao usuário enviar e receber arquivos do Portal dentro dos grupos ao qual ele pertence. Quando selecionada a opção de *upload/download* de arquivos remotos é apresentada ao usuário uma tela, mostrando os grupos aos quais ele tem acesso, conforme Figura 4.

As funcionalidades acima descritas foram implementadas e adicionadas

à ferramenta epos-xconfig, de forma que possam ser desativadas através de um parâmetro na inicialização da ferramenta, permitindo que, quando necessário, o usuário execute a ferramenta *offline* localmente.

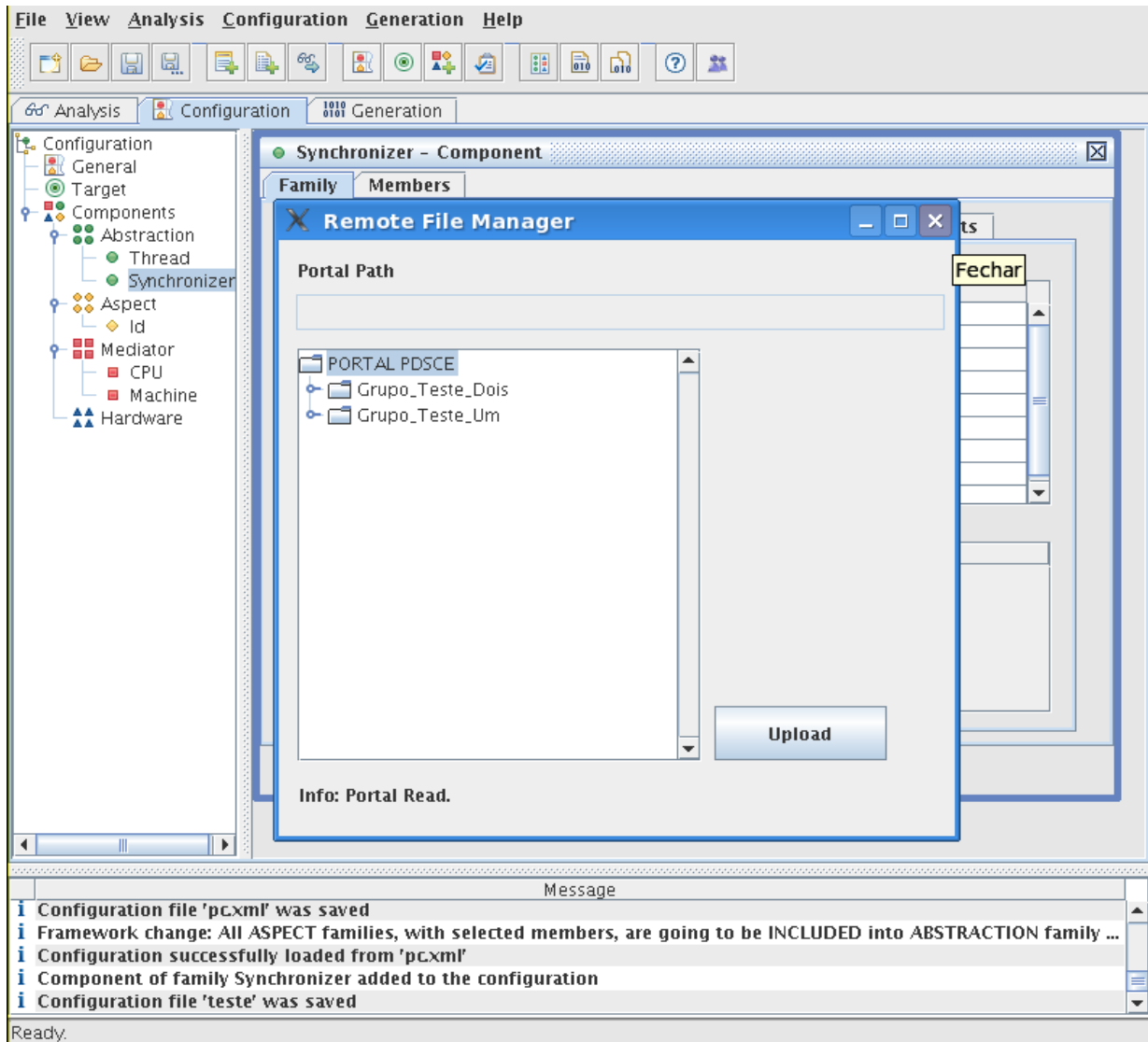


Figura 4: Integração epos-xconfig com o Portal

3.6. Testes E Resultados

O módulo de interação com o portal mostrou-se funcionar adequadamente, observando a concordância de permissões de acesso ao portal. A carga da ferramenta através do Java Web Start apesar de ser um pouco onerosa, possui uma boa relação custo/benefício, visto que a

flexibilidade obtida com a execução da ferramenta através do browser é significativa, além da facilidade para disponibilizar novas versões do epos-xconfig.

Alguns breves comentários relativos a análise de desempenho utilizando o Java Web Start: a execução da ferramenta propriamente dita ocorre normalmente, não apresentando qualquer ponto negativo em relação a execução *offline* da ferramenta e; *upload* e *download* de arquivos do portal ocorre de forma rápida e segura, visto que somente pode ser efetuada utilizando a sessão devidamente criada baseada nas credenciais informadas pelo usuário. Enquanto não houver sido informado usuário e senha válidos a aplicação não irá prosseguir.

Com as funcionalidades acrescentadas, abrir e salvar arquivos no portal, à ferramenta fornece a possibilidade do usuário utilizá-la em qualquer computador bem como acessar os arquivos de descrição pertencentes ao seu grupo diretamente do Portal. Dessa forma, além da alta disponibilidade, a ferramenta oferece um meio fácil de intercambiar os arquivos de configuração, característica importante quando se participa de um grupo de trabalho. Também é importante ressaltar que a ferramenta epos-xconfig carrega, na sua inicialização, um repositório de componentes de software e hardware que possibilita ao usuário sua utilização para especificação do sistema embarcado. Esse repositório de componentes foi incluído no Portal, e a ferramenta epos-xconfig foi modificada de forma a carregar esse repositório do Portal. Com isso, ganhou a flexibilidade na disponibilização de novos componentes, pois basta acrescentá-los ao Portal e a ferramenta epos-xconfig os carregará automaticamente na sua inicialização.

4. Plataforma Colaborativa

Este capítulo apresenta a construção da plataforma colaborativa proposta ao projeto, abordando os principais aspectos quanto ao sistema colaborativo, bem como características da implementação realizada.

4.1. Arquitetura Do Projeto Epos-xconfig - Colaborativo

O processo de elaboração de um sistema embarcado é definido em três fases distintas: (i) especificação de hardware, (ii) configuração de requisitos adicionais e (iii) geração de um sistema operacional para este hardware. A ferramenta epos-xconfig auxilia a execução das duas primeiras fases através de uma interface interativa, na qual o engenheiro pode compor o hardware necessário e as funcionalidades que deseja que o mesmo execute. A ferramenta realiza uma análise das necessidades e adiciona os recursos de hardware para implementar as funcionalidades desejadas. Ao final, epos-xconfig gera os arquivos de configuração que serão compilados, através do EPOS [FRÖHLICH 2006], para gerar a aplicação (software) que será utilizada pelo sistema embarcado especificado.

Na elaboração do projeto do sistema embarcado os engenheiros atuam de forma descentralizada, distribuída e colaborativa. O sistema Portal OpenCMS [RIZZETTI 2005] desenvolvido permite que o processo de distribuição seja transparente aos usuários, mas os mecanismos de colaboração para o desenvolvimento do projeto são limitados e ficam sob a responsabilidade de cada usuário.

Para prover a ferramenta epos-xconfig de mecanismos de colaboração foi implementada uma arquitetura P2P específica para o projeto PDSCE [RIZZETTI 2006], conforme a Figura 5. O fluxo de interação entre o Portal OpenCMS e o epos-xconfig é: 1 - enviar/receber arquivos do Portal; 2 - trocar arquivos e mensagens entre si e acessar o servidor de indexação centralizado P2P.

Com o incremento modular dessas novas funcionalidades, a ferramenta

permite que os engenheiros e/ou usuários realizem projetos, interações e decisões através de comunicações assíncronas, as quais serão armazenadas, nos computadores dos usuários, e que podem auxiliar na documentação do projeto, desde sua concepção até as principais decisões do sistema embarcado.

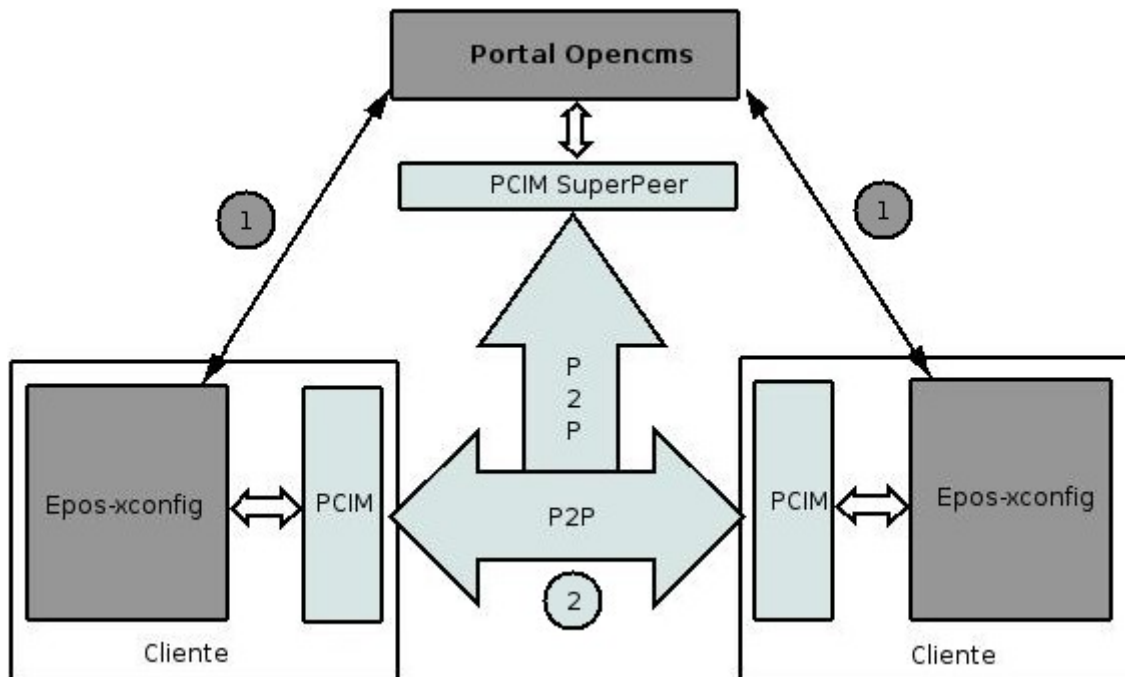


Figura 5: Plataforma Colaborativa

4.2. Funcionalidades Do Modelo Colaborativo

Tendo em mente o objetivo de fornecer um ambiente colaborativo ao projeto epos-xconfig utilizou-se a plataforma JXTA para sua concepção, e definiu-se algumas funcionalidades a serem seguidas na sua implementação, descritas a seguir.

4.2.1. Acesso A Rede Utilizando Credenciais Do Portal

A primeira tarefa a ser realizada para entrada na rede JXTA (através do NetPeerGroup) é o acesso ao servidor *rendezvous*, enviando mensagem para este com as credenciais³ de acesso do usuário ao Portal. O servidor, por sua vez, realiza a conexão com o Portal fornecendo as credenciais do usuário. Se

³ No caso, as credenciais são nome de usuário e senha.

essas estiverem corretas, uma sessão com o Portal é criada. Após a criação da sessão com o Portal, a primeira tarefa a ser realizada é buscar a informação sobre qual grupo (ou grupos) o usuário pertence. Nesse momento, retornando ao ambiente P2P, verifica-se se o grupo já existe; caso não exista, é criado. Por fim, é necessário que o usuário entre nesse grupo para que possa interagir com os demais usuários conectados à rede P2P e pertencentes ao seu grupo.

4.2.2. Comunicação Na Rede P2P

Para descoberta de recursos na rede P2P, optou-se pela indexação centralizada através do *peer rendezvous* acoplado ao Portal. A comunicação cliente-cliente deve ocorrer para efetivar o sistema colaborativo, permitindo envio de arquivos e mensagens tal como programas de mensagens assíncronos bem conhecidos.

A comunicação efetivamente entre clientes JXTA é feita através dos *pipes*, que consistem em mecanismos de comunicação unidirecionais. Existem outras implementações de *pipe* que permitem canais de comunicação bidirecionais que poderiam ser utilizados. Um *pipe*, como qualquer outro recurso de uma rede P2P, recebe um identificador único, o *PipeID*. Esse identificador é utilizado tanto no envio da mensagem como no recebimento desta pelo *peer* que possui o *pipe* de entrada passado pelo identificador. Todos os IDs dos recursos envolvidos na comunicação e que devem ser conhecidos pelos *peers* estão disponíveis no *peer rendezvous*, acoplado ao Portal. Dessa forma, através da alta disponibilidade do Portal e dos recursos de descoberta e resolução de serviços e componentes da rede JXTA, é possível construir uma rede funcional que atenda às características do projeto PDSCE.

4.2.3. Envio De Arquivos Ao Portal

Para o envio de arquivos para o portal, existem duas estratégias que podem ser abordadas: (i) utilizar a rede P2P através do *peer rendezvous* acoplado ao servidor ou (ii) utilizar o mecanismo já implementado, na aplicação epos-xconfig, para carga e descarga de arquivos do Portal. Optou-se pelo mecanismo tradicional já em funcionamento na aplicação epos-xconfig

para comunicação direta com o Portal. Desta forma, evita-se a sobrecarga que seria ocasionada pela comunicação P2P inserida nesse processo.

4.2.4. Descarga (*Download*) De Arquivos Do Portal

Da mesma forma que no sistema de envio de arquivos ao Portal, a descarga de arquivos do Portal se dá de forma direta. Ou seja, o *peer* acessa diretamente o Portal e realiza a descarga dos arquivos de projeto necessários.

4.3. Implementação Da Aplicação PCIM

Tendo em mente as funcionalidades propostas acima, foi implementada a aplicação PCIM (PDSCE *Colaborative Instant Messenger*). Na Figura 6 pode ser observado o diagrama de classe de algumas das classes mais importantes da aplicação.

A aplicação PCIM pode ser executada sob forma de duas instâncias especializadas, que pertencem a mesma aplicação: SuperPeer e ClientPeer. A seleção de qual instância deve ser executada é realizada através de um arquivo de configuração do aplicativo carregado em sua inicialização.

4.3.1. SuperPeer

Consiste no *peer Rendezvous*, centralizando o controle da rede. É responsável por receber e redistribuir os *advertisements* de todos os *peers* conectados a rede. Assim, todos os *peers* pertencentes ao mesmo grupo tem conhecimento dos *advertisements* uns dos outros. Os *advertisements* têm um tempo de duração e, quando esgotado esse tempo são removidos da *cache* cessando sua publicação aos demais *peers*.

Não se pode esperar que sempre os *peers* comuniquem ao Superpeer que irão desconectar-se; sendo assim, faz-se necessário um mecanismo que permita ao SuperPeer saber quando um *peer* não está mais conectado à rede. A solução para tal questão dar-se-á através da republicação periódica dos *advertisements* dos *peers* conectados ao *rendezvous*; assim, se um *peer* cair da rede, os *advertisements* deste não mais chegarão ao *rendezvous* e, após esgotar o tempo limite do último *advertisement* recebido do *peer* em questão, o SuperPeer e a rede JXTA saberão que este *peer* não está mais conectado.

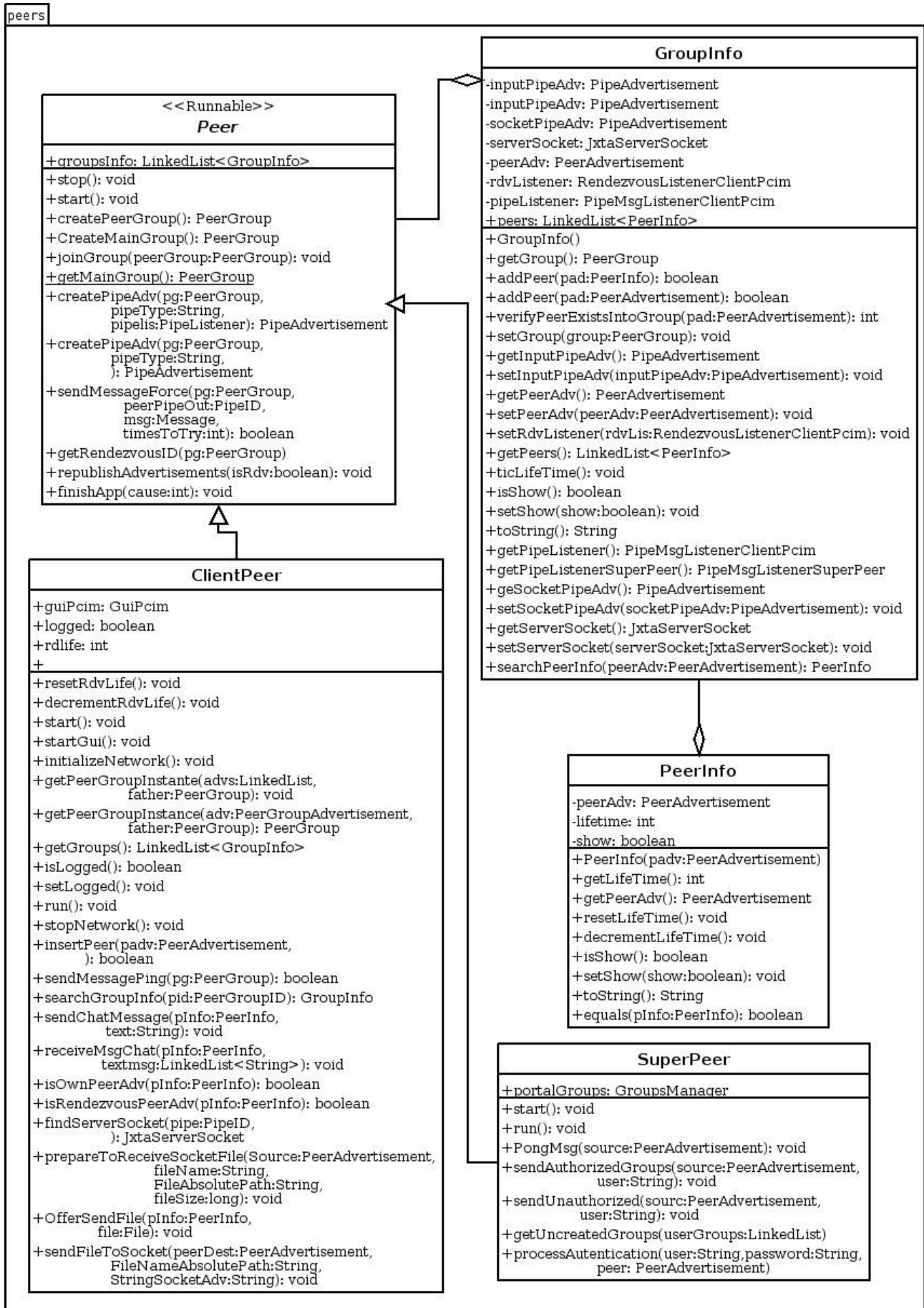


Figura 6: Diagrama de Classes PCIM

Da mesma forma que os *peers* devem comunicar-se periodicamente com o *SuperPeer* para informar que ele ainda continua ativo, o *SuperPeer* deve informar aos *peers* clientes que ele está ativo. Através do *ping* periódico realizado pelos *peers* clientes ao *SuperPeer* estes poderão detectar se algum problema ocorreu com este. Sempre que o *SuperPeer* recebe uma mensagem de *ping*, este responde ao *peer* que originou a mensagem informando que ele está ativo. Se durante um determinado número de tentativas não houver resposta do *SuperPeer*, o cliente detecta o problema, exibe o erro e encerra a aplicação.

A interação com o Portal também é realizada exclusivamente pelo *SuperPeer*, visto que este centraliza os mecanismos de autenticação. Sempre que um cliente conecta-se ao *SuperPeer* manda uma mensagem informando suas credenciais. De posse dessas, o *SuperPeer* acessa o Portal e verifica se essas credenciais estão corretas; sendo a identidade confirmada, o *SuperPeer* envia uma mensagem ao cliente informando à quais grupos de usuários ele pertence. O cliente recebendo essas informações entra no(s) respectivo(s) grupo(s) para que possa comunicar-se com os demais usuários a ele(s) pertencente(s); caso a identidade seja inválida, o *SuperPeer* retorna uma mensagem informando o erro e a aplicação cliente é finalizada.

Na inicialização do *SuperPeer* apenas o *NetPeerGroup*, que é o grupo padrão da rede *JXTA*, é criado. Os demais grupos são criados sob demanda, ou seja, quando um usuário acessa a rede é verificado a quais grupos ele pertence e se esses grupos existem na rede *JXTA*, se não existirem são criados. A criação sob demanda facilita o controle caso novos grupos sejam criados no portal, desta forma, não é necessário reiniciar o *SuperPeer* para que esses novos grupos sejam conhecidos na rede colaborativa, pois o *SuperPeer* deve executar indefinidamente aguardando comunicação dos *peers* clientes. Cabe ressaltar que esta aplicação (*SuperPeer*) não tem interface gráfica, pois não necessita interação humana e deve rodar na forma de *daemon*.

Para ilustrar esse processo, na Figura 7 pode ser visto o diagrama de sequência de acesso do cliente ao *SuperPeer*.

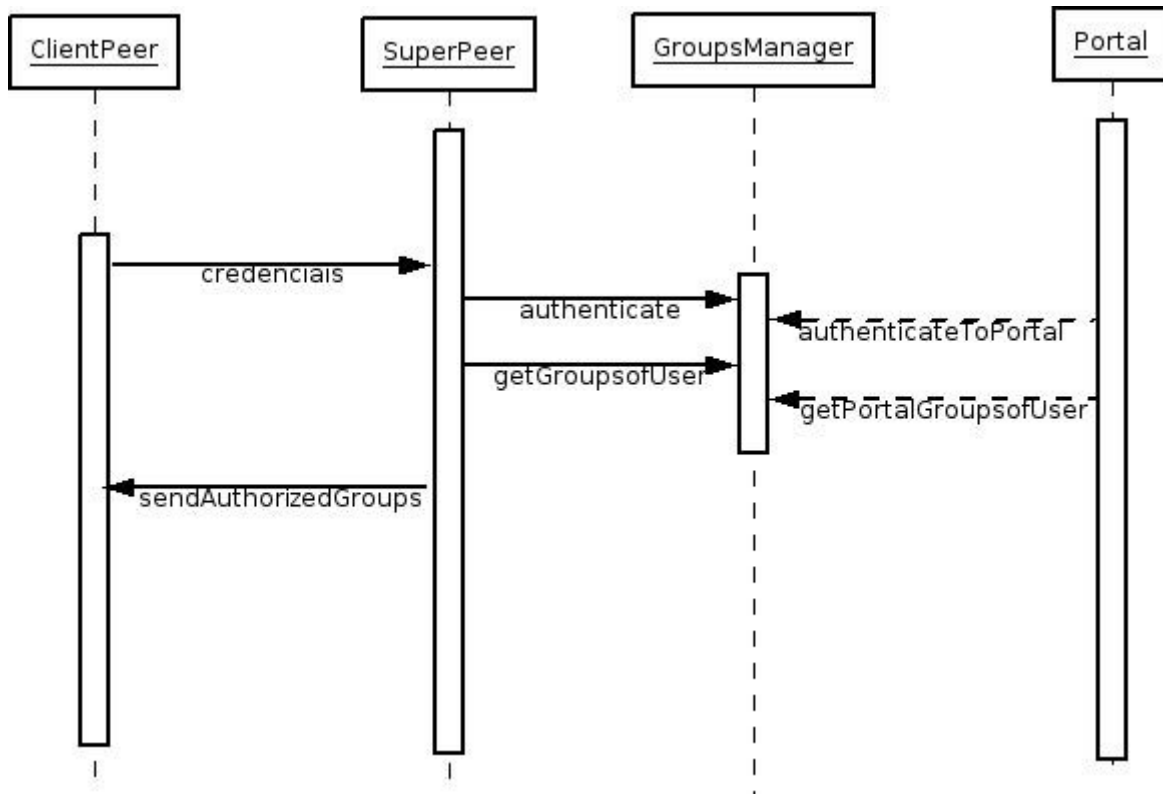


Figura 7: Diagrama de Sequência SuperPeer

4.3.2. ClientPeer

ClientPeer é a aplicação utilizada pelos clientes, disponibilizada junto a aplicação epos-xconfig. Quando o usuário estiver utilizando a ferramenta epos-xconfig através do Portal, poderá utilizar a ferramenta PCIM em conjunto, permitindo a ele saber quais usuários do seu grupo de trabalho estão trabalhando *online* com a ferramenta epos-xconfig. Dessa forma, é possível realizar interação entre esses usuários, proporcionando a estes um mecanismo de discussão *online*, através de mensagens instantâneas. Também está disponível a opção de trocas de arquivos entre os usuários, facilitando a troca de diferentes versões do arquivo de projeto, quando ambos trabalharem simultaneamente neste, por exemplo.

A aplicação PCIM foi construída como uma aplicação independente, e que deve ser chamada a partir da aplicação epos-xconfig. Portanto, as credenciais do usuário devem ser repassadas da ferramenta epos-xconfig para a ferramenta PCIM. Na Figura 8 é apresentado o diagrama de sequência explicativo da inicialização da aplicação PCIM cliente.

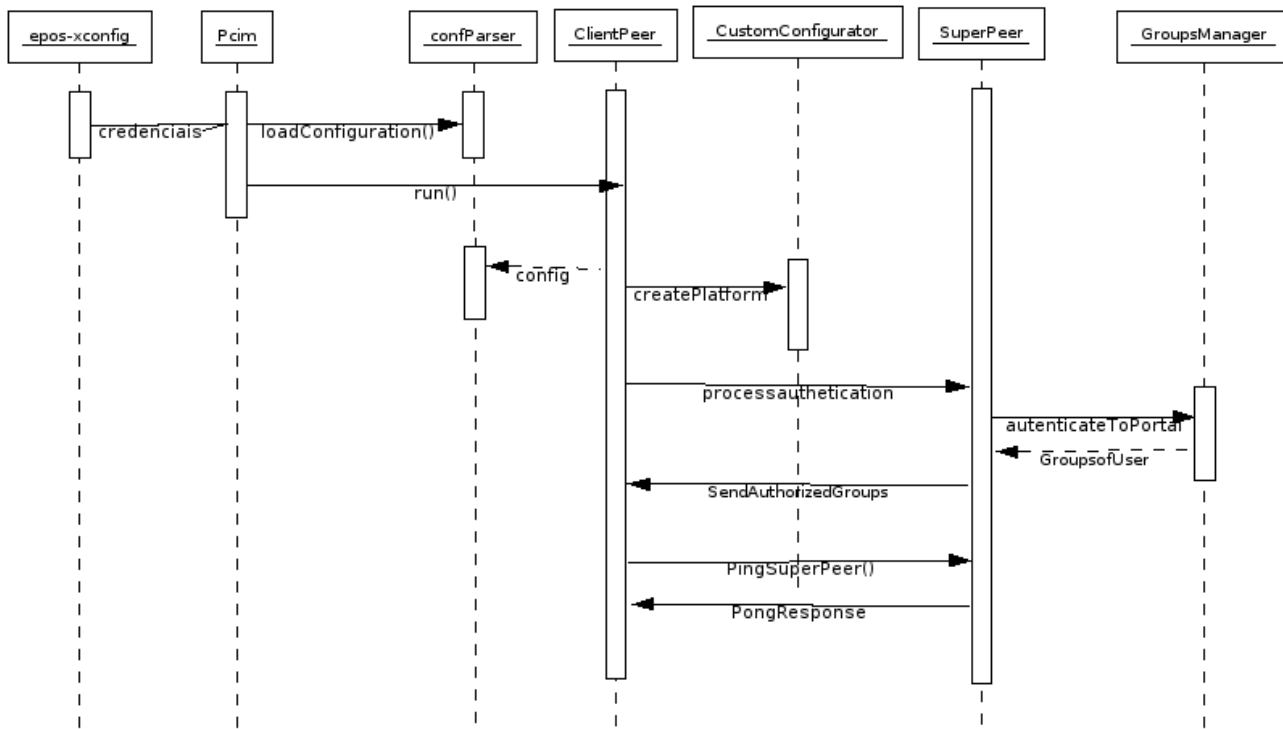


Figura 8: Diagrama de Sêquencia Inicialização PCIM Cliente

OBS: a classe SuperPeer e GroupsManager pertencem ao SuperPeer e portanto representam um acesso remoto.

A aplicação PCIM também é dotada de mecanismo de log, tanto de aplicação quanto de conversas. O *log* de aplicação basicamente é utilizado para detectar possíveis erros na aplicação. Já o *log* de conversa, é armazenado em uma pasta no diretório *home* do usuário e serve como uma forma de documentação das discussões realizadas para o projeto através do ambiente colaborativo.

É importante frisar que a troca de mensagens instantâneas e arquivos entre os usuários se dá de forma direta, ou seja, sem nenhuma interferência do *SuperPeer*.

4.4. Testes E Resultados

A aplicação PCIM foi desenvolvida como um aplicação independente que pode ser utilizada sozinha ou pode ser executada através do epos-xconfig. Em qualquer dos casos as funcionalidades são as mesmas, assim como a

necessidade de autenticação com o Portal.

Para incorporação do PCIM no epos-xconfig adotou-se a seguinte abordagem: as bibliotecas e o arquivo jar da aplicação PCIM são incorporados ao arquivo jar da aplicação epos-xconfig. O arquivo jar do epos-xconfig é descarregado do Portal e, quando o epos-xconfig é inicializado há um procedimento que realiza a cópia dos arquivos do PCIM desse jar para um diretório do sistema de arquivos local. A partir desse ponto, a aplicação PCIM fica disponível para ser utilizada através da aplicação epos-xconfig, que realiza uma chamada a esse arquivo copiado para o sistema de arquivos local.

Buscou-se oferecer uma interface amigável à ferramenta PCIM, sendo semelhante a diversos programas de troca de mensagens comumente utilizados, conforme Figura 9.

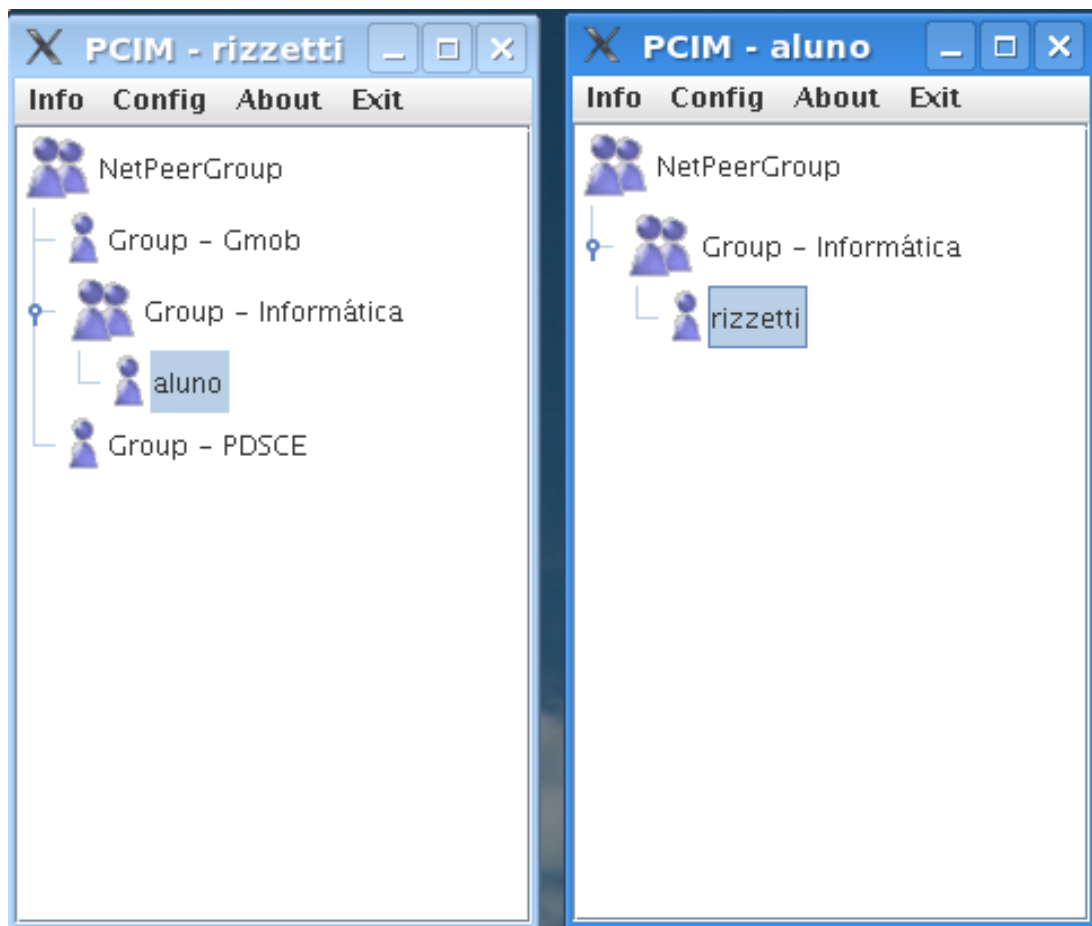


Figura 9: Duas Instâncias da aplicação PCIM, utilizando diferentes usuários, rodando no mesmo computador.

A ferramenta mostrou-se funcionar adequadamente, autenticando

corretamente no portal, informando ao usuário quando erros de autenticação ou erros na rede JXTA ocorrem, como por exemplo, se o SuperPeer tornar-se indisponível. O reconhecimento de conexão e desconexão de usuários também mostrou-se eficaz, ocorrendo apenas um pequeno atraso no reconhecimento dessas situações na rede, porém, esta é uma situação prevista, visto que eventualmente mensagens podem ser perdidas na rede, portanto uma desconexão somente é reconhecida após um número mínimo de tentativas de contato sem sucesso, no caso foi definido um número de três tentativas.

A ferramenta foi testada sob duas formas: várias instâncias do PCIM executando no mesmo computador e; três instâncias da ferramenta PCIM sendo executadas cada uma em um computador diferente, inclusive dois deles utilizando o sistema operacional Linux e outro Windows. A troca de mensagens e arquivos funcionou com sucesso em todos os testes, apresentando também uma interface amigável de utilização, conforme pode ser observado na Figura 10. Foram realizados diversos testes de transferências de arquivos, onde também mostrou-se eficaz a implementação utilizada, independente do tamanho dos arquivos testados. Os tamanhos de arquivos utilizados para testes variaram de alguns Kbytes à até 700 Mbytes, ambos mantendo a integridade do arquivo apresentando portanto resultados satisfatórios.

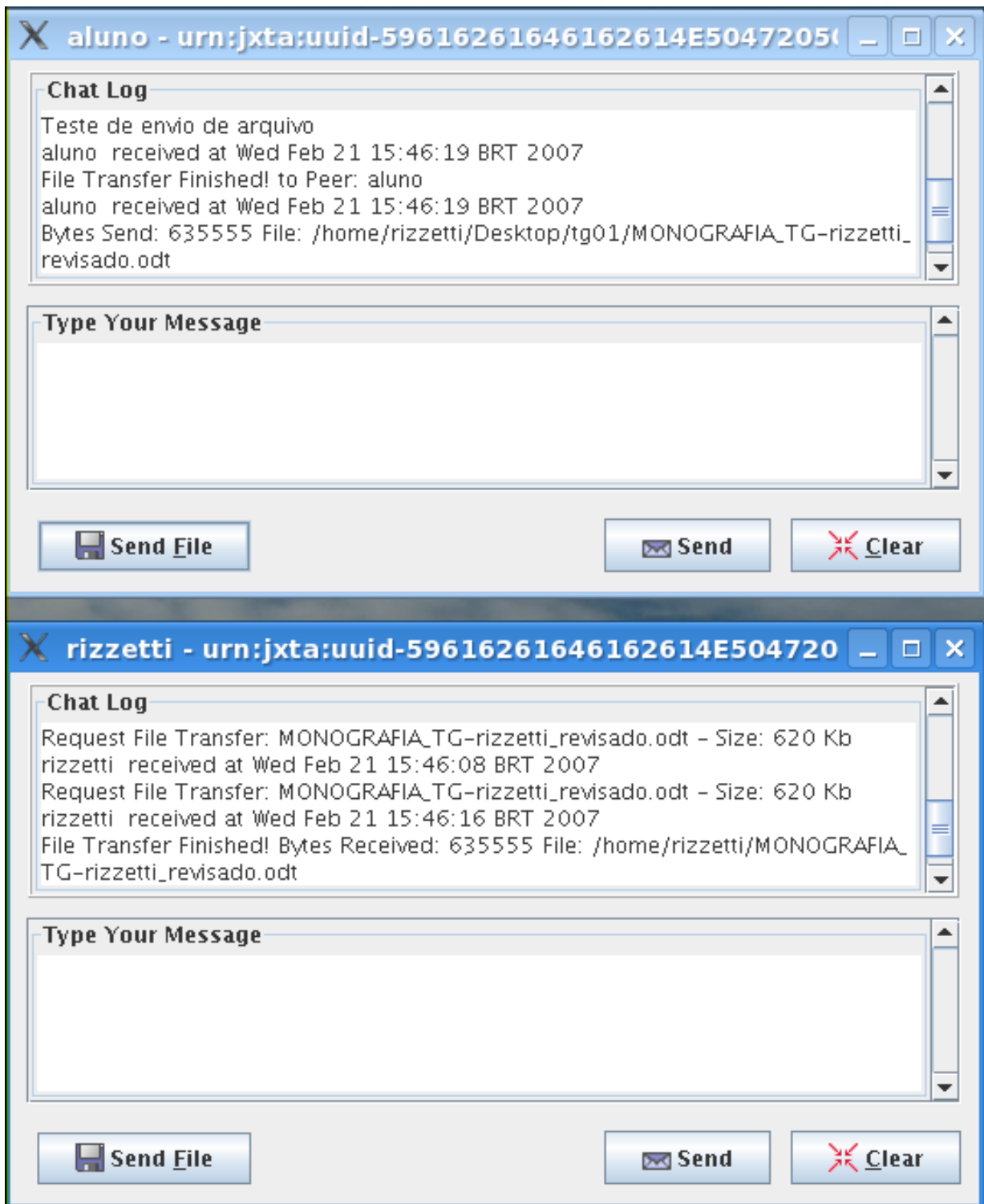


Figura 10: Troca de mensagens entre duas instâncias da aplicação PCIM rodando no mesmo computador

5. Conclusão

A tecnologia Java Web Start aliada as possibilidades fornecidas pelas ferramentas CMS formam um poderoso meio de distribuição de aplicativos através da web. Com o crescente desempenho obtido com as redes de comunicação, a sobrecarga imposta pelo Java Web Start torna-se menos significativa à cada dia. Os mecanismos de *cache* utilizado por esta tecnologia são importantes, visto que minimiza os impactos das transferências, possibilitando que mesmo usuários que possuam conexões com baixa largura de banda venham a utilizar aplicações distribuídas desta forma.

Em conjunto com a internet a utilização das redes P2P vem crescendo muito nos últimos anos, decorrência clara dos benefícios de sua utilização, reduzindo tráfegos redundantes pela internet. Pois, somente os interessados na comunicação participam desta, ao contrário do modelo cliente/servidor, onde sempre ocorre a interferência do servidor. Com isso, em um época que a internet vem a cada dia popularizando-se mais, é importante fornecer mecanismos que promovam eficiência às comunicações. Prover mecanismos que facilitem o desenvolvimento de tais ferramentas é também imprescindível, tal como faz a API JXTA, abstraindo do desenvolvedor várias aspectos de rede, facilitando assim o desenvolvimento, além de prover padronização e integração de diferentes aplicações.

O poder das redes P2P é notório, a integração possibilitada pelos programas de mensagens instantâneas emergentes nos últimos anos são um exemplo. A facilidade de comunicação e eficiência em transferências de dados, são as principais vantagens obtidas. Da mesma forma, a utilização da tecnologia P2P neste trabalho possibilitou um mecanismo fácil e eficiente de comunicação entre os usuários da ferramenta. Podendo com isso até mesmo melhorar a produtividade no desenvolvimento, visto que há um mecanismo de colaboração incorporado a própria ferramenta e, portanto, não sendo necessária qualquer adaptação, que normalmente é frustrante, entre ferramentas não preparadas para tal acoplagem.

Sem dúvida os objetivos iniciais do trabalho foram concluídos com êxito, sendo inclusive possível dizer que os resultados foram melhores que os

esperados. O Portal tornou-se operacional, e através dele pode-se lançar a ferramenta epos-xconfig, através do Java Web Start, que por sua vez contém embutida em seu arquivo de execução a ferramenta PCIM, que culmina na construção do ambiente colaborativo.

Aprimoramentos podem ser feitos, entre eles testes de campo com essas aplicações e com o próprio Portal, visto que os testes realizados foram apenas internos. Como trabalhos futuros, inclui-se a possibilidade de implementar o armazenamento de logs no Portal, dessa forma tornando-o disponível para consulta pelos demais usuários interessados e também como parte da documentação do projeto embarcado, cuja especificação está sendo realizada.

Além disso, pode-se acoplar um sistema de CVS (*Current Version System*) [BAR 2003] ao Portal, proporcionando o versionamento dos arquivos de especificação. Com isso, além do histórico de modificações, tem-se um mecanismo de backup eficiente, evitando que arquivos sejam apagados ou sobrescritos acidentalmente.

Referências Bibliográficas

- APACHE.ORG (2005) “Welcome! – The Apache Software Foundation”, Disponível em <http://www.apache.org/>, acessado em maio de 2005.
- BAR, M.; FOGEL, K. (2003). “Open Source Development with CVS”, Editora Paraglyph, 3ª edição.
- BARIK, Titus (2005) “Introducing the Java Content Repository API”, disponível em <http://www-128.ibm.com/developerworks/java/library/j-jcr/>, acessado em agosto de 2005.
- CMSMATRIX.ORG (2005) “The CMS Matrix - cmsmatrix.org - The Content Management Comparison Tool”, disponível em <http://www.cmsmatrix.org>, acesso em Maio de 2005.
- FOX, G. (2001). “Peer-To-Peer Networks”, IEEE, Web Computing.
- FRÖHLICH, A. A., et al. (2006) “EPOS Project”. <http://www.lisha.ufsc.br/epos/> acessado em abril de 2006.
- FRÖHLICH, A. A., (2005). “PDSCE - Plataforma de Desenvolvimento de Sistemas Computacionais Embarcados”, março. Disponível em <http://www.lisha.ufsc.br/events/pdsce-opening/opening.pdf>, acessado em abril.
- UFBA, Instituto de Matemática (2004). “Sistemas Operacionais para Sistemas Embarcados”. Março. Disponível em <http://twiki.im.ufba.br/pub/MAT154/WebHome/SOE.pdf>, acessado em fevereiro de 2007.
- KAMIENSKY, Carlos, et al. (2005). “Colaboração na Internet e a Tecnologia Peer-to-Peer”, XXV Congresso da Sociedade Brasileira de Computação. Disponível em <http://www.dca.ufrn.br/~marlos/arquivos/artigos/p2p/colaboracao-na-internet-e-a-tecnologia-p2p.pdf>, Julho de 2005.
- KORPELA, E. et al. (2001). “SETI@Home—Massively Distributed Computing for SETI”, Scientific Programming, feb. 2001.
- KOVER, A. (2000). “Napster: The Hot Idea Of The Year”. ISSN 0015-8259. Vol.

142 No. 1 página 128. Fortune 06/26/2000.

JAKARTA, Apache.org (2005) “The Jakarta Site – Apache Tomcat”, disponível em <http://jakarta.apache.org/tomcat/>, acessado em junho de 2005.

JETSPEED2 (2005) “JetSpeed 2 Enterprise Portal”, disponível em <http://portals.apache.org/jetspeed-2>, acessado em Julho de 2005.

JXTA.ORG. (2005) “JXTA v2.3.x: Java™ Programmer’s Guide”, disponível em <http://www.jxta.org/>, acessado em maio de 2006.

OBINARY, Inc. (2005) “Magnolia Content Management Suite”, disponível em <http://www.magnolia.info>, acessado em julho de 2005.

OPENCMS.ORG (2005) “OpenCms - The Open Source Content Management System”, disponível em <http://www.opencms.org>, acesso em junho de 2005.

PEREIRA, Júlio C.L, et. al (2002) “Introdução à Gestão de Conteúdos”. Congresso Anual da Sociedade Brasileira de Gestão do Conhecimento, 2002. Disponível em <http://www.bax.com.br/research/publications/introducaoGestaoConteudos.pdf>, acessado em Janeiro de 2006.

TRAVERSAT, B. ARORA, A. (2006). “Project JXTA 2.0 Super-Peer Virtual Network”. Disponível em <http://www.jxta.org/project/www/docs/JXTA2.0protocols1.pdf>, acessado em março de 2006.

RIZZETTI, T. A. et al. (2005). “Análise de Sistemas de Gerenciamento de Conteúdo para o Projeto PDSCE”, SIRC 2005, Santa Maria – RS.

RIZZETTI, T. A. et al. (2006). “Ambiente Colaborativo P2P para o projeto PDSCE”, SIRC 2006, Santa Maria – RS.

SUN, Sun Microsystems (2005) “Java Servlet Technology”, disponível em <http://java.sun.com/products/servlet/index.jsp>, acessado em agosto de 2005.

SUN, Sun Microsystems (2006) “Java Web Start Overview”, disponível em http://java.sun.com/developer/technicalArticles/WebServices/JWS_2/JWS_White_Paper.pdf, acessado em fevereiro de 2007.

WILSON, B. J. (2002) “JXTA”, Editora New Riders 1ª Edição. Disponível em www.brendonwilson.com/projects/jxta.