

CENTRO UNIVERSITÁRIO DE BRASÍLIA - CEUB

GUILHERME AUGUSTO DE SOUZA DANTAS

ISABELLA CALAZANS CUNHA

PIETRO VICTOR MACIEL

**RESUMO EXPANDIDO DO ARTIGO FERRAMENTA DE BACKUP TRANSPARENTE
BASEADA EM P2P**

BRASÍLIA - DF

2021

RESUMO

O problema a ser abordado neste artigo está relacionado aos sistemas distribuídos. Desse modo, foi necessário trazer autores para explicar conceitos sobre este tema. Através do estudo, conseguiu-se entender que sistemas distribuídos são dispositivos separados geograficamente que se comunicam através de um canal de comunicação. Desse modo, é possível compartilhar arquivos, objetos de software e hardware através de serviços. Assim, escolheu-se a arquitetura P2P descentralizada, na qual os *nós* do sistema comunicam-se sem a atuação de um servidor central. Para isso, serão abordados conceitos de JXTA, *juxtapose*, que são regras que orientam o funcionamento de uma rede P2P. Com relação ao objetivo deste resumo expandido, pode-se dizer que é trazer conceitos relacionados ao tema. Para isso serão verificados artigos correlatos, além do resumido no presente artigo, que é Ferramenta de backup transparente baseada em P2P. Ao final, serão apresentadas as funcionalidades da solução proposta pelo autor do artigo dito. É importante dizer que não será feito o desenvolvimento da aplicação, apenas será abordado o seu funcionamento. Com relação à metodologia, foi utilizada uma abordagem qualitativa, de modo que fosse possível trazer conceitos e bibliografias sobre o tema, sem a necessidade de números. Ao pensar nos objetivos do artigo, pode-se dizer que se utilizou uma metodologia descritiva, na qual se pensa em descrever o tema. Assim, será possível entregar conhecimentos aos leitores, tirar dúvidas sobre o conteúdo e gerar curiosidade para novos estudos advindos desta leitura. Com relação aos procedimentos, escolheu-se uma metodologia bibliográfica, em que se buscou ler artigos científicos, livros e documentos online sobre o conteúdo e apresentá-lo de forma resumida e direta. Sobre a natureza do trabalho, considera-se ele aplicado, uma vez que, apesar de não desenvolver diretamente a solução, o presente resumo propõe uma solução que aplica os conceitos trazidos. Ao final, perceber-se-á que os objetivos terão sido alcançados, uma vez que conceitos sobre o tema terão sido apresentados e a solução terá sido proposta. Com a aplicação, foi também possível corroborar com os conteúdos do artigo, uma vez que atua em cima de sistemas P2P através de alguns protocolos JXTA. Através de todo o trabalho, percebeu-se que o middleware JXTA possui vários recursos para execução de sistemas P2P. Com relação às dificuldades encontradas, pode-se falar da falta de conteúdo a respeito da JXTA e da manutenção de threads para que todos os fragmentos dos arquivos sejam copiados. É possível salientar que, como módulos futuros, pode ser construído um que recupere os arquivos que foram salvos em nós remotos caso aconteça perda de arquivos. Seria interessante também implementar uma camada de segurança, garantindo que os possuidores dos arquivos pudessem acessar os arquivos em repositórios remotos.

Palavras-chave: P2P; Sistemas distribuídos; JXTA; Backup; Arquitetura.

SUMÁRIO

1 INTRODUÇÃO	4
Objetivos	4
2 FUNDAMENTAÇÃO TEÓRICA	4
3 METODOLOGIA	7
3.1 Ferramenta de backup	9
3.1.2 Módulo de configuração	9
3.1.2 Módulo de backup	9
4 RESULTADOS E DISCUSSÃO	11
5 CONSIDERAÇÕES FINAIS	11
REFERÊNCIAS	13
APÊNDICES	14

1. INTRODUÇÃO

O presente artigo deseja apresentar informações relevantes quanto à arquitetura **P2P** e sobre como utilizá-la para realizar serviços, como o de backup. Alguns artigos com temas correlatos foram estudados, de modo que fosse possível entregar uma explicação coerente sobre o tema abordado. O propósito deste artigo é a necessidade de realizar backups em uma arquitetura **P2P**, *Peer-to-Peer*. Sendo assim, é importante falar sobre Sistemas Distribuídos, tecnologia P2P e os protocolos JXTA. Todo o material apresentado nas seções a seguir são baseados no artigo Ferramenta de backup transparente baseada em P2P, sendo responsabilidade deste artigo, realizar um resumo dos pontos mais importantes apresentados pelo autor. Por último, será apresentada uma solução proposta pelo autor, em seu artigo, que mostra como é possível desenvolver uma aplicação para fazer backups transparentes em sistemas baseados na arquitetura P2P.

OBJETIVOS

O objetivo do artigo é trazer conteúdos relevantes referentes aos sistemas distribuídos, de modo que seja possível explicar qual a essência da arquitetura P2P. Para isso, serão estudados artigos com temas e assuntos correlatos para que seja possível embasar o conteúdo apresentado no presente artigo. O objetivo é gerar curiosidade no leitor, de modo que surjam novos estudos e novas pesquisas científicas sobre o tema. Além disso, busca-se apresentar o conceito de protocolos JXTA e mostrar regras que permeiam as redes com sistemas distribuídos P2P. Este artigo não tem o objetivo de desenvolver uma solução de backup, uma vez que ele é um resumo expandido. Assim, apenas será abordada qual a solução e qual é o funcionamento dela em sistemas P2P.

2. FUNDAMENTAÇÃO TEÓRICA

Segundo Coulouris, em 2007, sistemas distribuídos são aqueles nos quais componentes conectados em uma rede comunicam-se através da troca de mensagens. Fazem isso de modo que possam compartilhar recursos, como hardware, arquivos e objetos de software. É importante lembrar que esses componentes não precisam estar geograficamente próximos. Podem estar separados por uma distância continental ou dentro de uma mesma empresa.

Com relação ao compartilhamento de recursos, pode-se falar de hardware, software. O primeiro é quando deseja-se utilizar o serviço de impressão, por exemplo. Uma parte do sistema gerencia as impressoras, de modo que as pessoas possam escolhê-la e realizar as funções (TAYLOR, 2005). Porém, além do serviço de impressão, os usuários podem acessar informações relacionadas às suas atividades, sendo que elas não estão presentes localmente, e sim, em um servidor online.

Percebe-se que, para enviar e receber esses recursos, deve-se desenvolver uma conexão confiável e consistente entre os componentes, nós. Para isso, é necessária uma interface de comunicação entre eles, de modo que informações possam trafegar até chegar ao destino final. Isso é importante no que tange à heterogeneidade dos sistemas, uma vez que se trata de nós que operam em sistemas operacionais diferentes, que possuem protocolos escritos em linguagens distintas e que estruturam dados de forma divergente (COULOURIS, 2007). Tudo isso é importante ao estruturar um sistema distribuído e não pode atrapalhar na comunicação entre os nós.

Sobre a tecnologia P2P, diz-se que ela ocorre em um ambiente onde todos os nós podem comunicar-se entre si espontaneamente sem a necessidade de um nó principal (SCHRODER E FISCHBACH, 2003). Desse modo, todos possuem duas funções, ora são clientes, ora servidores, pois, em uma hora, respondem e em outra realizam requisições. Segundo Gradecki, em 2002, a topologia que mais se aproxima dessa definição é a P2P descentralizada (Figura 1 no Apêndice A). Como o próprio nome diz, é uma arquitetura descentralizada, sem uma autoridade central. É formada apenas por dispositivos que se comunicam com qualquer um dos outros na rede. Desse modo, todos os nós gerenciam consultas, pedidos de downloads e, ao mesmo tempo, realizam buscas e transferem mídias requisitadas (GRADECKI, 2002).

Para esconder os detalhes da plataforma onde o sistema P2P será executado dos desenvolvedores de aplicações, é necessário possuir uma pilha de software. Ela divide o sistema em duas camadas, sendo a de aplicações - mais próxima dos usuários - e a middleware, responsável por interagir com as camadas mais baixas da rede. Assim, é possível lidar com a heterogeneidade de redes e computadores. Com esse encapsulamento, é possível entregar facilidades de alto nível aos desenvolvedores, tendo eles a responsabilidade de focar quase que exclusivamente nas funcionalidades desejadas do sistema. Segundo Steinmetz e Wehrle, em 2005, esconder detalhes dos módulos dos desenvolvedores, facilitar a manutenção do sistema e diminuir a complexidade para os usuários são benefícios de usar este middleware.

O middleware usado pelo autor é o JXTA, que remete à juxta-*pose*. Isso significa que esta ferramenta anda lado a lado com modelos mais conhecidos hoje, como: WEB e Cliente-Servidor. JXTA são protocolos (regras) abertos e gerais usados para que, dentro de uma rede, todos os componentes possam comunicar-se, desde um celular até um *PDA*, Personal Digital Assistant (TAYLOR, 2005). As principais metas desse middleware são: independência de sistemas operacionais, de linguagens de programação, fornecimento de serviços e de infraestrutura para aplicações P2P (Brookshier, 2002).

É importante salientar que o JXTA é um framework, e não uma aplicação; que não define os tipos de aplicações que podem ou não ser desenvolvidas. Ainda que os protocolos JXTA tenham sido implementados em JAVA (GRADECKI, 2002), é válido dizer que podem ser implementados em quaisquer linguagens. A seguir, serão citados seis protocolos, encontrados em (FLENNER et al., 2002): **Protocolo de consulta de nó** - usado para enviar consultas a quaisquer nós e receber respostas; **Protocolo de descoberta de nó** para enviar e descobrir conteúdo; **Protocolo de informação** para verificar o status dos nós; **Protocolo de ligação** para conectar os nós; **Protocolo de roteamento** para encontrar a rota até chegar ao nó determinado; e **Protocolo de encontro** para propagar mensagens na rede.

Com esses protocolos, um nó consegue existir em uma arquitetura P2P descentralizada, tendo a capacidade de distribuir conteúdos disponíveis, descobrir os que são interessantes, criar ou entrar em grupos públicos ou privados e ajudar na transferência de mensagens dentro da rede. Para a criação de um nó ou até de grupos, são utilizados os

anúncios (Figura 2 no Apêndice B). Eles são arquivos XML que passam de um nó a outro exibindo informações relevantes para a operação da rede P2P.

Diferentemente da arquitetura P2P, a JXTA é dividida em três camadas (Figura 3 no Apêndice C). Segundo Taylor, em 2005, a mais baixa é o núcleo, responsável pela conexão, pela comunicação e pelo roteamento entre os nós. A segunda camada é a de serviços, que entrega funções de procura e compartilhamento de arquivos. Essa camada faz uso maciço das ligações feitas no núcleo. Na terceira, é onde se encontram os sistemas P2P desenvolvidos. Ela dá suporte para a construção de aplicações de compartilhamento de arquivos, armazenamento distribuído e mensagens instantâneas (FLENNER et al., 2002).

Dentro da arquitetura JXTA, há também o Serviço de Gerenciamento de Conteúdo (CMS) que capacita os nós para o compartilhamento de arquivos de áudio, imagem, vídeo. O CMS utiliza JXTA pipes para transporte e anúncios para apresentar informações sobre os conteúdos compartilhados. Além disso, o CMS mantém um registro com os arquivos compartilhados pelos nós locais, de modo que haja mais eficiência na propagação dos conteúdos (GRADECKI, 2002).

Os JXTA pipes são usados para receber consultas e conteúdos compartilhados por outros nós. Quando uma consulta é realizada, envia-se uma mensagem LIST_REQ para receber a lista de conteúdos compartilhados dos outros nós na rede. Desse modo, o nó que recebe a requisição retorna a mensagem LIST_RES, entregando anúncios com os conteúdos solicitados. Após isso, caso seja necessário fazer download, o nó envia uma mensagem GET_REQ baseada no anúncio recebido. Todos esses processos são construídos em cima dos protocolos JXTA.

3. METODOLOGIA

O artigo retratado utiliza uma abordagem qualitativa, na qual busca aproximar o leitor dos seguintes assuntos: sistemas distribuídos, arquitetura P2P e protocolos JXTA. Desse modo, o resumo traz informações e conceitos de modo que o leitor possa compreender qual o tema, levantar questionamentos e esclarecer dúvidas. O foco deste artigo não é trazer números (quantidade) e sim, qualificar o tema.

Com relação aos objetivos do artigo, foi utilizada a metodologia descritiva. Escolheu-se essa visto que o objetivo era apresentar autores nacionais e internacionais que estudaram sobre o universo de comunicação entre componentes computacionais. Desse modo, o objetivo é descrever os conceitos levantados por eles. Assim, é possível fomentar novos conhecimentos e gerar curiosidade nos leitores. Dessa forma, futuros estudos podem ser feitos a partir e novas soluções práticas podem ser construídas.

Com relação aos procedimentos, utilizou-se a metodologia bibliográfica. Isso ocorreu a partir do momento que se decidiu buscar bibliografias sobre o tema através da ferramenta Google Scholar. Assim, livros, sites, artigos científicos foram consultados, de modo a trazer o que há de mais atual sobre o conteúdo apresentado ao leitor. Ainda que haja certa dificuldade em encontrar informações sobre o assunto, elas também servirão de base para a construção do sistema de backup para sistemas P2P.

Com relação à natureza, pode-se dizer que o artigo é classificado como pesquisa aplicada. Isso ocorre porque, além de trazer bibliografias sobre o assunto, também tem como objetivo final propor uma solução que aplica os conceitos de backup transparente baseado em P2P apresentados. Dessa forma, como também haverá uma parte em que se apresenta uma solução prática, classifica-se este artigo como pesquisa aplicada, não havendo investimento nem lucro.

Este artigo foi todo desenvolvido em ambiente computacional, sendo todos os artigos encontrados online, na Internet. Não houve necessidade de contato humano, nem de questionários, entrevistas e de deslocamento. Desse modo, conseguiu-se ler os artigos de forma virtual, a partir de desktops e smartphones. Utilizaram-se algumas ferramentas, como o Google Docs para desenvolvimento do artigo e o Google Drive para armazenar alguns dos artigos referência.

Como foi possível observar, o objeto de estudo deste artigo são os sistemas distribuídos, sendo a P2P a arquitetura abordada no resumo. Desse modo, foi necessário apresentar conceitos, como: definição de sistemas distribuídos, topologia P2P, serviços possíveis de utilizar, protocolos JXTA, entre outros. Com relação à delimitação do conteúdo, dentre todas as arquiteturas, focou-se na arquitetura JXTA e no serviço de backup, dentre vários outros possíveis.

Para cumprir o objetivo proposto neste resumo, bastou apenas a leitura do artigo Ferramenta de backup transparente baseada em P2P, além de outros utilizados para sanar dúvidas que apareceram a partir do artigo resumido. Por tratar-se de um resumo expandido, não houve necessidade de desenvolver a solução, apenas de apresentá-la e mostrar seu funcionamento. A seguir, será apresentado o passo-a-passo da solução e sua funcionalidade.

3.1 FERRAMENTA DE BACKUP

Com relação à aplicação, pode-se dizer que ela se divide em dois módulos, um de configuração e outro de backup. É importante dizer que um depende do outro, de modo que o de backup utiliza as configurações feitas no outro módulo.

3.1.1 MÓDULO DE CONFIGURAÇÃO

O objetivo final deste módulo é gerar um arquivo XML que apresenta as informações de execução de um nó, sendo este arquivo utilizado pelo módulo de backup. Abaixo serão descritos os campos do arquivo. Antes, é válido dizer que todos os campos são obrigatórios:

- **<nomepeer>**: Indica o nome do nó local e, aos nós remotos, mostra quem é o dono dos arquivos compartilhados;
- **<pastacompartilhada>**: Mostra a localização dos arquivos de backup, lembrando que ela é sempre procurada pelo módulo de backup;
- **<repositorio>**: Caminho usado para armazenar arquivos recebidos de nós remotos;
- **<pastasistema>**: Caminho usado para apresentar as informações de conexão da ferramenta;
- **<pastapeer>**: Caminho onde serão armazenadas informações relacionadas ao nó local.

3.1.2 MÓDULO DE BACKUP

O módulo de backup (Figura 4 no Apêndice D) começa quando ele lê o arquivo XML gerado pelo módulo de configuração. Primeiramente, ele verifica se existem arquivos dentro

de <pastasistema> e de <pastapeer>, de modo que, caso exista, eles serão apagados e recriados sempre que este módulo for iniciado.

Salienta-se que todo computador que tiver esse módulo em execução mantém o nó local e um servidor. Desse modo, tanto o nó local quanto os remotos podem conectar-se ao servidor. A partir disso, busca-se arquivos na <pastacompartilhada>, sendo eles publicados pelo CMS. Sempre que ele publicar, serão feitas referências a ele e sobre sua origem e integridade.

Para verificar a integridade do conteúdo do arquivo, utilizou-se a assinatura *hash*. Assim, sempre que for realizado o download de arquivos, consegue-se fazer um cálculo e comparar com as referências do arquivo. Se estiver diferente, haverá uma mudança e o arquivo deverá ser atualizado. Com relação à referência do arquivo, utiliza-se o <nomepeer>. Assim, é possível saber quem é o dono do arquivo.

Com relação ao tempo de republicação dos arquivos da <pastacompartilhada>, escolheu-se 1 hora de intervalo. Desse modo, a cada hora, os arquivos que se encontram neste caminho são republicados. Assim, os nós poderão verificar os novos arquivos e se houve novas modificações desde a última atualização.

Após a publicação, envia-se uma requisição de arquivos para os nós parceiros da solução. Caso, nos anúncios recebidos dos nós remotos não contenha informações de arquivos, é necessário esperar o tempo pré-determinado para a publicação de novos arquivos e para a análise dos anúncios.

Após a análise do anúncio, deve-se verificar no <repositorio> a existência de arquivos remotos na rede. Caso tenha, para cada arquivo, verifica-se, dentro do repositório local, se existe um com o mesmo nome e extensão do arquivo remoto analisado. Se não existir o arquivo, envia-se uma requisição de download dele. Entretanto, caso ele já exista, verifica-se se ele é idêntico ao que está no repositório local. Para isso calcula-se o algoritmo MD5 e compara-se com a informação anterior.

Sendo as assinaturas hash idênticas, não é necessário realizar um novo download. Caso contrário, o arquivo pode estar corrompido ou pode ter sofrido alterações. Desse modo, envia-se uma nova requisição de download do arquivo presente no nó remoto. Após o

pedido de download, a transmissão do arquivo é realizada através de threads, de modo que o arquivo possa ser enviado em fragmentos. Assim, ganha-se eficiência no processo de download. Entretanto, é importante dizer que não é possível realizar todas as threads de forma simultânea, uma vez que os arquivos chegam incompletos. Assim, criou-se um gerenciador de threads, que limita a 5 threads simultâneas.

4. RESULTADOS E DISCUSSÃO

Ao relembrar os elementos levantados para este resumo, conseguiu-se trazer os conceitos de sistemas distribuídos, de arquitetura P2P e dos protocolos JXTA. Com relação ao objetivo principal de um resumo expandido, foi possível realizar a leitura do artigo “Ferramenta de backup transparente baseada em P2P” e trazer os pontos mais importantes listados pelo autor. Desse modo, a motivação de gerar interesse em ler o artigo resumido foi concluída com sucesso. Com relação à solução proposta pelo autor, foi possível apresentar o seu funcionamento e sua conclusão bem sucedida, ainda que o objetivo deste resumo expandido não tenha sido de construir a solução, e sim de mostrar o passo-a-passo dela.

Por último, através da leitura desse e de outras bibliografias correlatas, foi possível corroborar com o que o autor do artigo “Ferramenta de backup transparente baseada em P2P”. Além disso, através da solução proposta por ele, foi possível perceber o quão alinhado está o módulo de backup desenvolvido com os conceitos abordados. Os conceitos levantados foram trazidos de tal maneira que se encaixam muito bem com sua solução, levando o leitor a uma contextualização para que então possa ter suas próprias conclusões e opiniões, facilitando a possibilidade de desenvolvimentos posteriores envolvendo as mesmas ferramentas e arquiteturas, como, a arquitetura P2P e os protocolos JXTA, contribuindo para o desenvolvimento e a evolução dos mesmos.

5. CONSIDERAÇÕES FINAIS

Com o desenvolvimento da ferramenta foi possível deduzir que a JXTA é um middleware com enormes quantidades de recursos para a execução de sistemas P2P. Essa ferramenta foi aplicada massivamente nos serviços de descoberta e o CMS, para administrar

a publicação e download dos arquivos na rede. Os maiores impedimentos no desenvolvimento foram a insuficiência de conteúdos atualizados a respeito da JXTA, a manutenção das threads para que todas as partes dos arquivos fossem copiados e como as ideias referentes ao arquivo seriam publicadas de maneira que ficassem à disposição aos outros nós da rede. Como tarefas futuras, poderia ser construído, além do módulo de backup, um módulo que, em caso de perda de arquivos do nó local, recuperasse os arquivos que foram salvos em outros nós da rede. Como trabalhos futuros poderiam ser implementados, além do módulo de backup, um módulo que, em caso de dano em arquivos do nó local, resgataria os arquivos que foram salvos em outros nós da rede. Da mesma forma, seria interessante implementar uma camada de segurança, para que apenas os possuidores dos arquivos fossem capazes de acessá-los nos repositórios remotos.

REFERÊNCIAS

ANIDO, R. Atualizações em informática. Rio de Janeiro: PUC-Rio, SBC, 2006.

BARCELOS, M.P.; GASPARY, L.P. Fundamentos, tecnologias e tendências rumo a redes P2P seguras. In: BREITMAN, K..

BROOKSHIER, D. et al. JXTA: Java P2P Programming. Indianapolis: Sams Publ., 2002

COULOURIS, G. Sistemas distribuídos: conceitos e projeto. São Paulo: Pearson Education, 2007 . v.4

DEVMEDIA. JXTA - Parte II. 2015. Disponível em: <<http://www.devmedia.com.br/jxta-parte-ii/6054>>. Acesso em: 17 ago. 2021.

FLENNER, R, et al. Java P2P Unleashed. Indianapolis: Sams Publ., 2002.

GRADECKI, J. D. Mastering JXTA: building java peer-to-peer applications. John Wiley & Sons, 2002.

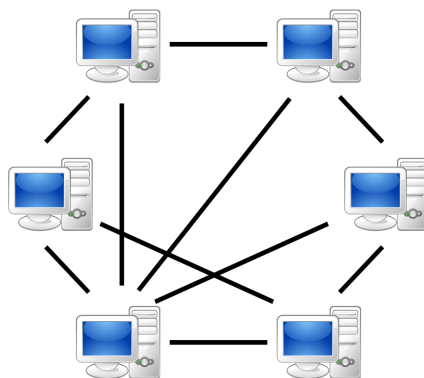
SCHODER, D.; FISCHBACK, K. Peer-to-peer prospects. Communications of the ACM, v.46. n.2, p-27-29, 2003. Disponível em: <<https://doi.org/10.1145/606272.606294>> Acesso em 17 ago. 2021.

STEINMETZ, R.; WEHRLE, K. Peer-to-peer systems and applications. Berlin: Springer, 2005. Disponível em: <<https://doi.org/10.1007/11530657>> Acesso em: 17 ago. 2021

TAYLOR, I.J. From P2P to web services and grids - peers in a client/server world. Unites States: Springer, 2005.

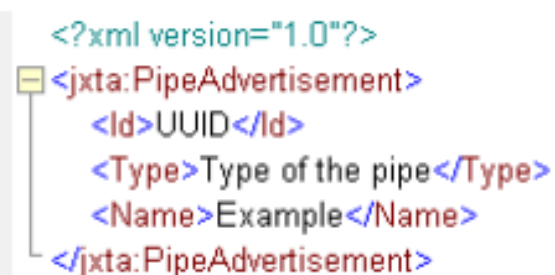
APÊNDICE A - ARQUITETURA P2P DESCENTRALIZADA

Figura 1.



APÊNDICE B - ANÚNCIO XML

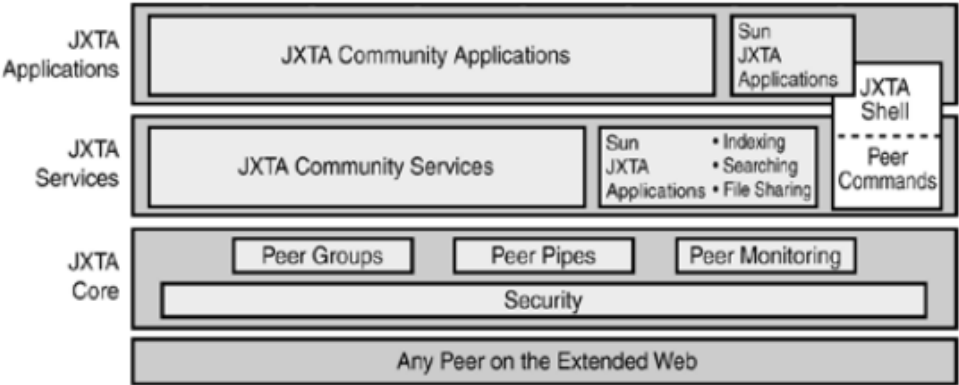
Figura 2.

The image shows a code editor with a light gray background. On the left, there is a vertical toolbar with a yellow icon of a document with a minus sign. To the right of this icon, a vertical line descends and then turns right to point at the opening tag of the XML element. The XML code is as follows:

```
<?xml version="1.0"?>
<jxta:PipeAdvertisement>
  <Id>UUID</Id>
  <Type>Type of the pipe</Type>
  <Name>Example</Name>
</jxta:PipeAdvertisement>
```

APÊNDICE C - CAMADAS JXTA

Figura 3.



APÊNDICE 4 - MÓDULO BACKUP

Figura 4.

