SocialRAD: Uma Infraestrutura para Telerradiologia Assíncrona, Colaborativa e Segura

João Filho Matos Figueiredo[1], Gustavo Henrique Matos Bezerra Motta[1], Luciano Carvalho de Medeiros Júnior[1], José Hélio de A Fernandes Filho[1]
[1]Centro de Informática (CI), Universidade Federal da Paraíba, João Pessoa, Brasil

Resumo: A popularidade dos serviços de telerradiologia tem possibilitado um importante avanço na prestação dos serviços de saúde em áreas geográficas de difícil acesso. Contudo, este potencial traz desafios elevados: o grande volume de dados, característico dos exames de imagens e os requisitos de segurança que visam garantir a sua confidencialidade e integridade. Desta forma, este trabalho propõe a integração de tecnologias, como modelos *peer-to-peer* utilizados em soluções de VoIP, *Cloud Computing*, DNS Dinâmico, *WebServices RESTful*, além de padrões de segurança e interoperabilidade, a fim de prover um ambiente colaborativo para a realização de telerradiologia de forma assíncrona, segura e eficiente. O modelo desenvolvido está em fase experimental, provendo suporte à telerradiologia em cidades do sertão nordestino, tendo atendido às expectativas planejadas.

Palavras-chave: Telerradiologia, Telemedicina, Sistemas de Informação em Radiologia, Segurança (computação)

SocialRAD: An Infrastructure for asynchronous, collaborative and safe teleradiology

Abstract: The popularity of teleradiology services has enabled major advances in the provision of health services in remote geographical areas . However, this high potential brings challenges: the large volume of data, characteristic of imaging studies and the security requirements designed to keep its confidentiality and integrity. Thus, this paper proposes the integration of technologies, such as the peer-to-peer models used in VoIP solutions, Cloud Computing, Dynamic DNS, RESTful WebServices, and interoperability and safety standards in order to provide a collaborative environment for implementation of teleradiology asynchronously, safely and efficiently. The model is still experimental, providing support for teleradiology in cities of the northeastern of Brazil, having fulfilled expectations planned.

Keywords*: Teleradiology, Telemedicine, PACS (Radiology), Computer Security.*

Introdução

A crescente demanda pelo compartilhamento de informações entre profissionais de saúde situados em locais geograficamente distantes entre si, de forma eficiente, tem o potencial de gerar economia de tempo e recursos, além de proporcionar melhorias na efetividade clínica, na qualidade da assistência e na gestão dos sistemas de saúde. Para isso, a telemedicina, aliada à *Computer-Supported Cooperative Work* (CSCW), tem sido um fator relevante em muitas especialidades médicas[1], em especial na radiologia, que usa novas tecnologias de forma intensiva.

Entre as vantagens da telerradiologia, pode-se destacar a possibilidade da realização de laudos médicos especializados, à distância, para pacientes presentes em áreas de difícil acesso, os quais, outrora, sequer podiam desfrutar de tais exames. Além disso, há a redução considerável de custos, a exemplo do deslocamento de médicos para cidades interioranas, que deixa de ser necessário, além do tempo para produzir o laudo, que é reduzido.

Sistemas modernos desta natureza precisam ser colaborativos, ou seja, devem possibilitar que vários usuários interajam em uma sessão comum, na qual desfrutem de uma visão unificada[2]. Assim sendo, este trabalho propõe uma solução denominada SocialRAD, que provê uma infraestrutura distribuída, assíncrona e colaborativa, bem como uma aplicação de usuário final, a fim de oferecer um serviço robusto para realização de telerradiologia. O SocialRAD aborda o problema tendo em vista suprir as principais necessidades de um ambiente real de telerradiologia, que, muitas vezes, ocorre em condições precárias de infraestrutura, frequentemente em cidades de difícil acesso. Além disso, buscou-se solucionar as principais deficiências presentes em outras abordagens[3], em que é necessário o uso de IPs públicos, o que torna o trabalho pouco praticável

em um ambiente real. A solução comumente adotada[1-3] para a indisponibilidade de IPs públicos é a utilização de um servidor de referência externo que funciona como uma ponte (*relay*) entre os usuários e os hospitais. Tal tipo de solução introduz uma elevada sobrecarga sobre o ponto central, tornando a solução não escalável, além de impactar na velocidade de acesso aos exames (que devem passar pela ponte, antes de estarem disponíveis para os médicos). O SocialRAD realiza uma abordagem para esse problema baseando-se nos mecanismos empregados em soluções de Voz-sobre-IP (VoIP), que permite aos médicos, situados em locais remotos distintos, obterem acesso aos exames diretamente do hospital que os gerou, ainda que este encontre-se inserido em uma rede privada, atrás de um *Network Address Translation* (NAT)[4-5].

Metodologia e Solução

O processo de desenvolvimento do SocialRAD se deu de forma progressiva, por meio de uma metodologia evolucionária. Coadjuvante, a prototipação foi utilizada a fim de capturar e validar os requisitos de maior interesse em um ambiente com carências reais da telerradiologia. Estes, por sua vez, foram confrontados com características presentes em outras abordagens, obtidas por meio de uma análise crítica das propostas disponíveis na literatura (no período de 2009 até 2012). Identificou-se, dessa forma, um conjunto de características relevantes, com potenciais de agregar valor ao estado da arte.

Uma dessas características é a capacidade de permitir o acesso remoto aos exames de imagens de forma direta (ponto-a-ponto), eximindo a necessidade do envio destes para uma terceira parte externa (e.g, um centro de dados na Internet). Com isso, o armazenado dos dados sigilosos dos pacientes fica restrito ao Centro de Saúde que gerou/adquiriu tais dados, conforme exigências dos órgãos reguladores, e, ainda assim, os médicos autorizados podem acessá-los.

O processo é conduzido de forma transparente para o profissional de saúde, que, por meio de um aplicativo, desfruta de uma visão unificada do ambiente distribuído (compostos por vários hospitais cooperantes). A Figura 1 ilustra como se dá o fluxo deste processo, desde a aquisição do exame, no Centro de Saúde/Clínica, até a conclusão do laudo médico especializado, realizado, neste exemplo, por dois médicos radiologistas. Os passos de 1 a 8 a seguir descrevem o fluxo do processo e, nos parágrafos subsequentes, as tecnologias utilizadas são explanadas em maiores detalhes.

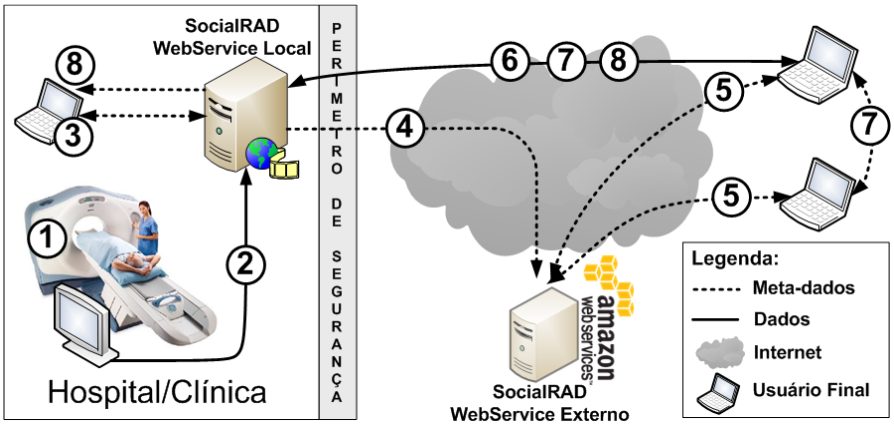

Figura 1: Fluxo em alto nível do processo de telerradiologia via SocialRAD

1. Aquisição do exame de imagem.
2. Exame de imagem é enviado para o *WebService* Local do SocialRAD, via protocolo da especificação *Digital Imaging and Communications in Medicine*[7] (DICOM).
3. O exame torna-se disponível para um operador técnico local, que poderá definir parâmetros, como: urgência do laudo do exame, quais médicos são autorizados, inclusão da história do paciente, dentre outros parâmetros desejáveis.

4. Os metadados do exame são enviados para um servidor de referência externo.
5. O aplicativo SocialRAD Cliente, usado pelos profissionais de saúde, realiza a sincronização em intervalos regulares de tempo, quando são buscadas as novas atualizações disponíveis no servidor externo (nuvem). As atualizações são exibidas ao usuário, conforme ilustrado na Figura 2.
6. Os profissionais de saúde podem, a partir desse ponto, escolher um método para visualizar o exame. Por meio de informações incluídas nos metadados e da técnica de comunicação desenvolvida, o SocialRAD Cliente obtém o exame diretamente do Centro de Saúde/Clínica que o produziu, abstraindo o cenário distribuído do profissional de saúde.
7. Laudos e outros documentos podem ser enviados e anexados ao exame, fazendo-se disponíveis, de forma colaborativa, aos profissionais que também tenham autorização para o referido exame. O armazenamento dos documentos, semelhantemente, é realizado pelo centro de saúde/clínica que produziu o exame.
8. O laudo definitivo é concluído pelos profissionais e sinalizado para entrega ao paciente.

A complexidade do fluxo ilustrado acima é abstraída do usuário final, que interage com o ambiente por meio de uma aplicação escrita usando a biblioteca Swing[8], do Java 6. Além disso, a própria aplicação é instalada e iniciada automaticamente no computador usuário, por meio da tecnologia *Java Web Start*[9]. É necessário apenas que ele autorize a instalação e realize a autenticação no sistema.

Por meio da biblioteca *Jersey*[10], que é uma implementação de referência para a construção de *WebServies RESful* em linguagem Java, o SocialRAD Cliente se comunica com o SocialRAD *WebService* externo, na nuvem, e obtêm os metadados dos exames que foram realizados nos hospitais. Estes dados são exibidos ao usuário, conforme ilustra a Figura 2, e armazenados em um *cache* local da aplicação, implementado como um banco de dados relacional e puramente escrito em Java, o *H2 Database*[11]. Esta base de dados é criptografada com o algoritmo *Advanced Encryption Standard* (AES), com chave de 128 bits e modo de operação *Cipher-Block Chaining* (CBC). A chave de criptografia é gerada com a função de *hash* SHA-256 sobre a senha do usuário concatenada a um valor de *salt*, de 64 bits, gerado por uma função *random*. Por sua vez, o resultado dessa operação é aplicado, novamente, à função SHA-256, e por mais 1024 vezes, de forma encadeada, gerando, então, a chave de criptografia do AES-CBC-128. Tal estratégia visa ampliar a difusão e, com isso, inviabilizar ataques de dicionário contra a senha do usuário. O Vetor de Inicialização (IV), para operação no modo CBC, é calculado com base no *hash* da chave, mais uma vez submetido à função SHA-256. Assim, o IV não é público e, neste caso, não o precisa ser, uma vez que apenas o conhecedor da senha simétrica do usuário terá acesso aos dados. Ademais, manter o IV secreto é uma proteção contra ataques *watermarking*[12].

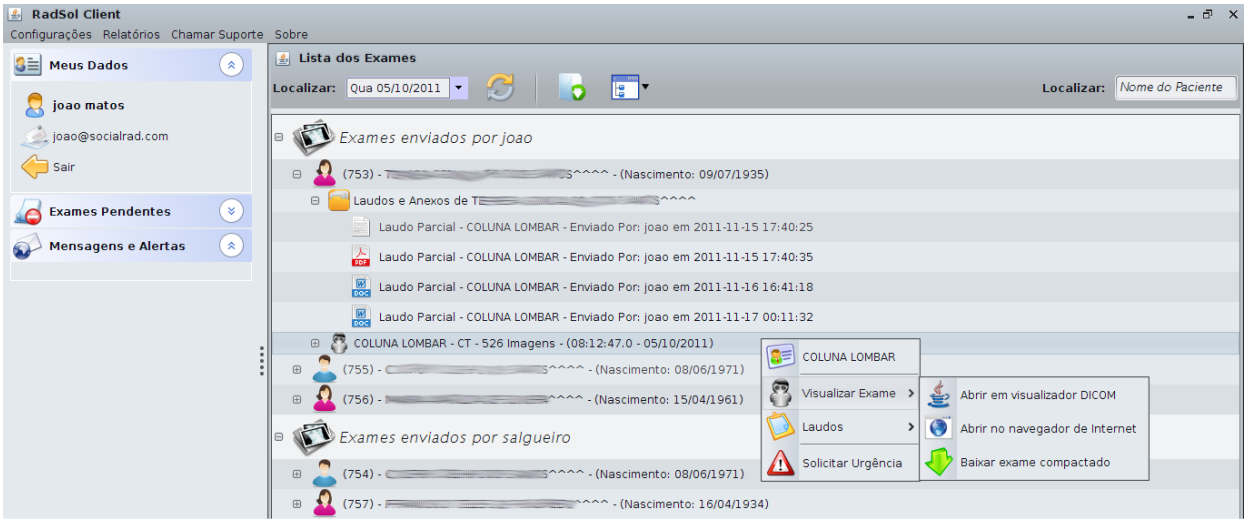

Figura 2: Aplicação SocialRAD Cliente

Modelo em camadas – A solução foi separada em três camadas lógicas: Camada 1, Serviços de Armazenamento; Camada 2, Serviços de Integração e Camada 3, Serviços de Distribuição. Assim, obteve-se uma arquitetura mais flexível, na qual tecnologias específicas podem ser substituídas sem causarem impactos nos serviços das demais camadas, desde que se adequem às interfaces. Ademais, a separação facilitou o acoplamento de *softwares* livres, provendo serviços especializados de armazenamento e exibição/apresentação dos exames de imagens. A Figura 3 ilustra o modelo em camadas, detalhado subsequentemente.

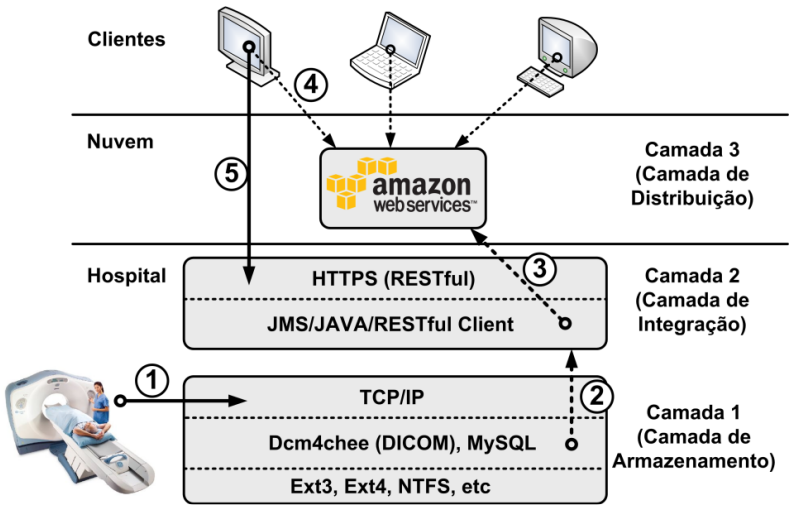

Figura 3: Visão do Modelo em Camadas

A arquitetura será inicialmente descrita por meio dos fluxos indicados na Figura 3, de maneira a facilitar o entendimento. Em (1), o exame de imagem é adquirido e submetido a um servidor local, por meio do protocolo DICOM, implementado por meio do *dcm4chee*[13]. Após o armazenamento no sistema de arquivos, um módulo, escrito em Java, detecta novas entradas no banco de dados, extrai os meta-dados e converte-os para o formato *JSON*[14]. Na sequência, em (2), os dados são entregues à Camada de Integração, onde ocorre o armazenamento em um *cache* de sincronização (para fins de controle de transação) e, por fim, são submetidos à Camada de Distribuição, em (3). Ressalta-se que o servidor da Camada de Distribuição é um ponto externo, na nuvem, onde os clientes, distribuídos, realizam a sincronização e obtém os metadados de novos exames disponíveis, ilustrado em (4). Uma vez sincronizados, os clientes podem, por fim, acessar os exames de imagens, obtendo-os diretamente do Centro de Saúde/Clínica que os produziu, ilustrado em (5). Desta forma, o controle de acesso é realizado no próprio Centro de Saúde que gerou o exame.

Conforme exposto anteriormente, os exames, bem como seus anexos (e.g, laudos parciais, laudos finais, história), são efetivamente armazenados na Camada 1 (Armazenamento), que é implantada em todos os centros de saúde/clínicas que geram exames, e conta, principalmente, com as seguintes tecnologias: servidor DICOM; sistema de gerenciamento de banco de dados (SGBD) MySQL; e *stateful firewall* com política padrão restritiva. Os serviços executam sobre uma plataforma Linux CentOS[15], que é implantada e operacionalizada de forma semiautomática, por meio de um módulo de configuração.

A Camada 2 (Integração) tem o importante papel de disponibilizar, de forma controlada, os recursos da Camada 1 para os clientes distribuídos. As principais tecnologias empregadas nesta camada são: **1) *Proxy* Reverso:** implementado com o nginx[16]. O *proxy* reverso intercepta as requisições dos usuários (realizadas pelo SocialRAD Cliente), realiza novas requisições para o *WebService RESTful* e, por fim, responde ao cliente. Assim, atua como uma camada intermediária, provendo um perímetro de segurança e disponibilizando o acesso HTTP sobre SSL/TLS (HTTPS); **2) *WebService RESTful*:** implementado em Java JSR-311[17], comunica-se apenas com o *proxy* reverso, que lhe solicita recursos em nome do cliente. Com isso, a autenticação e

autorização é realizada no *WebService* Local, que é mantido isolado da rede pública por meio do *proxy* reverso. **3) Serviço de Resolução de Nomes Dinâmico (*Dynamic* DNS):** para que fosse possível aos clientes localizarem o servidor real, que disponibiliza os exames de imagens, foi necessário disponibilizar um serviço de DNS Dinâmico. Sem este módulo, os centros de saúde/clínicas precisariam dispor de um endereço de IP estático, além de público. Tal necessidade está distante da realidade, principalmente ao se considerar o cenário da telerradiologia, no qual a maior contribuição se dá em cidades interioranas, muitas vezes com infraestrutura de internet precária e, frequentemente, sem opção de IP estático. Assim, foi utilizado o serviço *Amazon Route 53*[18] que, por meio da sua API, possibilitou a construção do serviço de DNS Dinâmico. Desta forma, os metadados dos exames carregam um URI que identifica a fonte geradora, no seguinte formato: *nome_do_hospital.socialrad.net*. Um módulo, instalado na Camada 2, detecta alterações no endereço IP do servidor (por exemplo, quando a conexão for interrompida e reestabelecida com outro endereço) e realiza a atualização junto ao serviço *Route 53*, que atualizará o mapeamento entre a URI e o novo endereço IP.

Diversos outros aspectos são tratados nesta camada, a exemplo do mecanismo para estabelecer conexões *Peer-to-Peer*, permitindo o acesso dos clientes ao servidor do hospital, ainda que ele esteja inserido dentro de uma rede privada (sobre um *Network Address Translation* – NAT[4]). Ao contrário de outras abordagens, as quais utilizam uma terceira parte realizando a função de ponte (ou *relay*), a solução do SocialRAD permite que a conexão seja estabelecida diretamente entre os dois pontos, sem a necessidade do *relay*, baseada no mecanismo adotado pela ferramenta *Skype*[5]. Desta forma, a terceira parte é necessária apenas para o estabelecimento da conexão, quando, por fim, o cliente e o hospital se comunicam diretamente, sem sobrecarregar o servidor externo. Esse mecanismo assemelha-se ao *Transversal Network Address Translation* (NAT-T)[19], por vezes utilizado em conjunto com o protocolo de Segurança IP (IPSec).

A Camada 3, disposta em um servidor externo, realiza, principalmente, a função de manter um espelho unificado dos metadados de exames realizados em todos os hospitais participantes. Além disso, as configurações de segurança, tais como permissões de acesso aos exames, são definidas localmente aos hospitais e, periodicamente, replicadas para o servidor externo (esta camada), a fim de refletirem no sistema globalmente. O SocialRAD Cliente comunica-se com essa camada por meio da interface RESTful, através da qual obtém os dados de sincronização, no formato *JSON*. Assim como na camada 2, o WebService da camada 3 é acessado por meio de um *proxy* reverso.

Discussão

O modelo vem sendo testado e avaliado por médicos radiologistas, distribuídos entre os estados do Ceará, Paraíba e Pernambuco, os quais obtêm exames realizados nas cidades de Brejo Santo – CE e Salgueiro – PE. Em ambos os hospitais, o SocialRAD Local foi implantado em computadores desktop, de baixo custo, que funcionam 24h por dia, 7 dias por semana. Em caso de falha, o ambiente pode ser rapidamente reestabelecido por meio de um *script* utilizado para instalação e configuração.

O Servidor Externo (camada de distribuição) está funcionando em uma *Micro Instância* do *Amazon EC2*. Além de outras funcionalidades, a API do *Amazon EC2* permitiu a introdução de elasticidade e escalabilidade com baixa complexidade. Assim, essa camada, que é um ponto centralizado, passa a ter uma maior confiabilidade.

O mecanismo de instalação automática do SocialRAD Cliente, nos computadores dos médicos radiologistas, por meio da tecnologia *Java Web Start*, permitiu uma maior efetividade no uso do sistema, uma vez que os profissionais de saúde, frequentemente, precisam utilizar computadores distintos, em diferentes momentos e localidades.

Foram utilizados apenas *softwares* e bibliotecas livres em todos os módulos desenvolvidos. Ainda, o SocialRAD Cliente integra-se a dois visualizadores DICOM *opensource*, além de permitir que os profissionais utilizem o visualizador que melhor lhes convenha.

Pretende-se evoluir o mecanismo de comunicação *peer-to-peer*, que apresenta algumas incompatibilidades em decorrências das variações nas implementações do NAT em diferentes dispositivos. Além disso, experiências estão sendo realizadas, tendo em vista avaliar a melhor estratégia para criptografar os dados dos exames que são encapsulados nos arquivos no formato DICOM (ao invés de criptografar todo o arquivo DICOM), a fim de prover uma garantia a mais, sem, contudo, impactar negativamente na eficácia do uso.

Conclusão

O desenvolvimento do SocialRAD contemplou uma série de vertentes, abordando desde questões de usabilidade, até mecanismos mais específicos para comunicação assíncrona e ponto-a-ponto entre os diversos elementos do ambiente distribuído. Também foram consideradas características concernentes a problemas legais e de privacidade desde os fundamentos da pesquisa, tendo sido desenvolvidos métodos específicos para o sigilo dos dados com o uso das boas práticas das tecnologias de criptografia e *WebServices*.

O seu desenvolvimento se faz no contexto de uma pesquisa de mestrado e tem como objetivo final disponibilizar, para a comunidade, um mecanismo robusto, adaptável e gratuito, em forma de serviço, para a realização de telerradiologia com qualidade.

O conjunto de soluções possibilitou um resultado de fácil utilização, prático, robusto e eficaz, fatores relevantes para o sucesso de aplicações desta natureza[6].

Referências:

[1] F. del Pozo, J.A. Quiles, M.T. Arredondo, H. Rahms, M. Sanz, P. Cano. A Telemedicine system for remote cooperative medical imaging diagnosis. Computer Methods and Programs Biomedicine. 1996; 37-48.

[2] Łukasz Czekierda, Tomasz Masternak, Krzysztof Zieliński. Evolutionary Approach to Development of Collaborative Teleconsultation System for Imaging Medicine Computers and Communications. 2009; 417-423.

[3] Yonggang Huang,ChunYang Hu.Yongwang Zhao and Dianfu Ma, Web-Based Remote Collaboration over Medical Image Using Web Services. Information Infrastructure Symposium. 2009; 1-8.

[4] The IP Network Address Translator (NAT) – Disponível em: http://www.ietf.org/rfc/rfc1631.txt , Acesso em 24 de Junho de 2012.

[5] Salman A. Baset and Henning Schulzrinne. An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol. New York : Columbia University; 2004.

[6] Ivan Drnasin, Domagoj Vucica, Stanko Tonkovic, Success of Teleradiology as a Confirmation of Radiological Excellence. Information Technology Interfaces. 2009; 73-78

[7] Digital Imaging and Communications in Medicine (DICOM), National Electrical Manufacturers Association, DICOM Committee, 2004.

[8] JDK 6 Swing (Java Foundation Classes) – Related APIs and Developer Guide - Disponível em: http://docs.oracle.com/javase/6/docs/technotes/guides/swing/, Acesso em 01 de Junho de 2012

[9] JSE Desktop Technologies – Java Web Start Technology - Disponível em: http://www.oracle.com/technetwork/java/javase/javawebstart/index.html Acesso em 01 de Junho de 2012

[10] Jersey - Disponível em: http://jersey.java.net/ , Acesso em 24 de Junho de 2012.

[11] H2 Database Engine – Disponível em: http://www.h2database.com/html/main.html , Acesso 24 de Junho de 2012.

[12] François Cayre, Caroline Fontaine, Teddy Furon. Watermarking Attack: Security of WSS Techniques. Lecture Notes in Computer Science, 2005;171-183

[13] Max J. Warnock, Christopher Toland, Damien Evans et al. Benefits of Using the DCM4CHE DICOM Archive. Journal of Digital Imaging, 2007; 125-129;

[16] JSON: The fat-free alternative to XML – Disponível em: http://www.json.org/fatfree.html , Acesso 24 de Junho de 2012.

[14] CentOS – The Community Enterprise Operating System – Disponível em: http://www.centos.org/ , Acesso em 24 de Junho de 2012.

[15] Will Reese. Nginx: the high-performance web server and reverse proxy. Linux Journal 2008

[16] JSR 311 – Disponível em: http://jsr311.java.net/ , Acesso em 24 de Junho de 2012.

[17] Amazon Web Services LLC. Amazon Route 53. Disponível em: http://aws.amazon.com/route53/, Acesso de em 20 Maio de 2012

[18] Address Allocation for Private Internets – Disponível em: http://tools.ietf.org/html/rfc1918 , Acesso em 24 de Junho de 2012.

Contato:
João Filho Matos Figueiredo
joaomatosf@gmail.com