

Desenvolvimento Distribuído de uma Aplicação de Telessaúde com a Metodologia Ágil SCRUM

Alexandre J. H. de O. Luna^{1,2}, Cleyverson P. Costa², Raphael F. A. Patrício², Ana C. M. L. Araújo², Haglay A. N. Silva², João T. F. S. Pessoa², Raony M. Araújo², Marcela B. S. Moraes², Maria S. B. Andrade², Sérgio F. T. O. Mendonça², Fabrício da C. Dias^{1,2}, Marcello R. de Mello^{1,2}, Magdala A. Novaes¹, Jones O. Albuquerque³, Sílvia R. L. Meira²

¹Grupo de Tecnologias da Informação em Saúde, Núcleo de Telessaúde (NUTES) – Universidade Federal de Pernambuco (UFPE), Hospital das Clínicas (HC), Universidade Federal de Pernambuco (UFPE), Brasil

²Centro de Informática, Universidade Federal de Pernambuco (UFPE), Brasil

³Departamento de Estatística e Informática, Universidade Federal Rural de Pernambuco (UFRPE), Brasil

Resumo – A partir da proposta de criação de uma Fábrica de Software, visando o desenvolvimento distribuído de software, realizada durante a disciplina IN953 – Engenharia de Software do Curso de Mestrado em Ciência da Computação do Centro de Informática da UFPE, foi construída uma Biblioteca Multimídia, apoiado na metodologia SCRUM, para compartilhamento de instrumentos de Telemedicina/Telessaúde, promovendo o cuidado à saúde através de programas de educação e prevenção ao nível da atenção primária.

Palavras-chave: Telessaúde, Telemedicina, Engenharia de Software Distribuída, Biblioteca Multimídia.

Abstract – From the proposed creation of a Software Factory, aiming for a distributed software development, held during IN953 - Software Engineering class, of master program in Computer Science offered by Center of Informatics of UFPE, was built a Multimedia Library, based on SCRUM methodology, for sharing content of Telemedicine/Telehealth, improving health care through programs of education and prevention primary care level.

Key-words: Telehealth, Telemedicine, Distributed Software Engineering, Multimedia Library.

Introdução

Engenharia de software é uma área do conhecimento voltada para a especificação, desenvolvimento e manutenção de sistemas de software, aplicando tecnologias e práticas da ciência da computação, gerência de projetos e outras disciplinas, objetivando organização, produtividade e qualidade.

Pressman [1] destaca que a Engenharia de software abrange três componentes básicos:

- **Métodos:** proporcionam os detalhes de como construir o software. Englobam tarefas como planejamento e estimativa de projeto, análise de requisitos de software e de sistemas, projeto da estrutura de dados, arquitetura de programa e algoritmo de processamento, codificação, teste e manutenção;
- **Ferramentas:** existem para sustentar cada um dos métodos. Algumas ferramentas existentes para apoio são as Computer-Aided Software Engineering, conhecidas como ferramentas CASE;

- **Procedimentos:** constituem o elo entre métodos e ferramentas. Definem a seqüência em que os métodos são aplicados.

O Desenvolvimento Distribuído de Software (DDS) é uma evolução da Engenharia de Software e visa o desenvolvimento de produtos de software através de equipes de desenvolvimento geograficamente distribuídas. Esta abordagem, em linhas gerais, tem por objetivo diminuir custos, proporcionar a contratação de mão-de-obra qualificada e aumentar a produtividade de desenvolvimento, proporcionando assim um maior poder de competitividade para as organizações [2].

O SCRUM é uma abordagem para desenvolvimento de software, sob o ponto de vista de metodologia. O termo SCRUM vem de um estudo feito por Takeuchi e Nonaka [3]. Como resultado deste estudo, foi percebido que projetos usando equipes pequenas e multidisciplinares (cross-functional) produzem os melhores resultados, abordagem esta que foi adotada neste trabalho.

Em virtude dos avanços das tecnologias da informação e das telecomunicações, a

Telemedicina e a Telessaúde, bem como outras áreas de pesquisa e desenvolvimento da Informática Médica, tomaram um grande impulso.

A Telemedicina pode ser definida como o uso das tecnologias da informação e de telecomunicações para apoiar a assistência médica à pacientes em locais distantes, baseando-se na troca de informações. Já a Telessaúde pode ser definida como o uso da tecnologia da informação e telecomunicação para prover serviços de saúde, conhecimento e informação nos contextos onde a distância é um fator crítico. Os termos se confundem em algumas abordagens, mas são melhores descritos como: complementares [4].

Em um país de dimensões continentais, como o Brasil, e com oferta assimétrica de serviços na área de Saúde, a Telemedicina se mostra como uma oportunidade/opção de resolução da dificuldade de comunicação entre instituições de saúde, garantindo à população uma maior qualidade na oferta de serviços e benefícios do sistema nacional de saúde.

Baseando-se nestes conceitos, o Grupo de Tecnologias de Informação em Saúde (TIS), entidade de pesquisa vinculada ao Núcleo de Telessaúde do Hospital das Clínicas (NUTES/HC) da Universidade Federal de Pernambuco (UFPE), ao longo de anos vem desenvolvendo projetos de pesquisa com foco na Telessaúde/Telemedicina.

A Rede NUTES é resultado do projeto de Telessaúde no Programa de Saúde da Família (PSF), um dos projetos do Grupo TIS, que visa disponibilizar instrumentos de Telemedicina/Telessaúde como uma alternativa para melhorar o acesso à saúde para a população carente, em zonas rurais ou em áreas desprovidas de serviços médicos especializados, promovendo o cuidado à saúde através de programas de educação e prevenção ao nível da atenção primária.

Ligado à Rede NUTES, o projeto BIBLIOTECA MULTIMÍDIA dataNUTES (BMD), tem o objetivo de permitir o armazenamento, recuperação, controle e disponibilização inteligente de conteúdo médico nos formatos de áudio, vídeo e texto, o que compreende, entre outros, o gerenciamento de informação obtida através de: sessões de tele-educação, raios-X, tomografias computadorizadas e ultra-sons, visando o aperfeiçoamento de profissionais de saúde, proporcionando apoio direto aos subprojetos Videoconference, DVD e Radiologia [5].

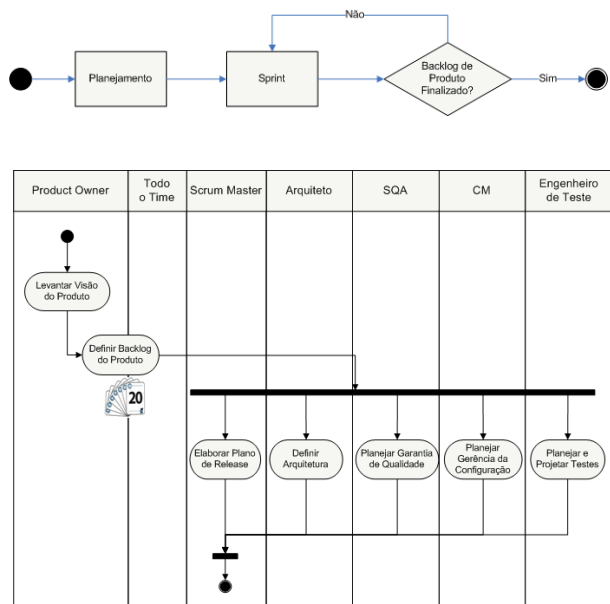
Neste contexto, processo de desenvolvimento do Projeto BIBLIOTECA

MULTIMÍDIA dataNUTES (BMD), experiência recentemente vivenciada pelos autores, será abordado neste trabalho, como caso prático para o estudo da aplicação dos conceitos de DDS, na abordagem da Engenharia de Software em projetos de desenvolvimento de software nas áreas de telessaúde e telemedicina.

A necessidade apresentada pelo NUTES, denominado Cliente do Projeto, visou o desenvolvimento de uma aplicação que possibilitasse o armazenamento, recuperação, controle e disponibilização inteligente de conteúdo multimídia, denominado Biblioteca Multimídia dataNUTES. Para esta aplicação, o cliente solicitou, em linhas gerais, servidor de Streaming, Player Web, sistema de busca inteligente, independência de sistema operacional, além da compatibilidade com os sistemas legados (ambiente de ensino e aprendizagem) já implantados.

Metodologia

Para o desenvolvimento deste projeto foi viabilizado uma fábrica de software, denominada Factory2u, que trabalhasse em um formato geograficamente distribuído, tendo seu processo de desenvolvimento de software baseado na metodologia ágil SCRUM, como apresentado na Figura 1.



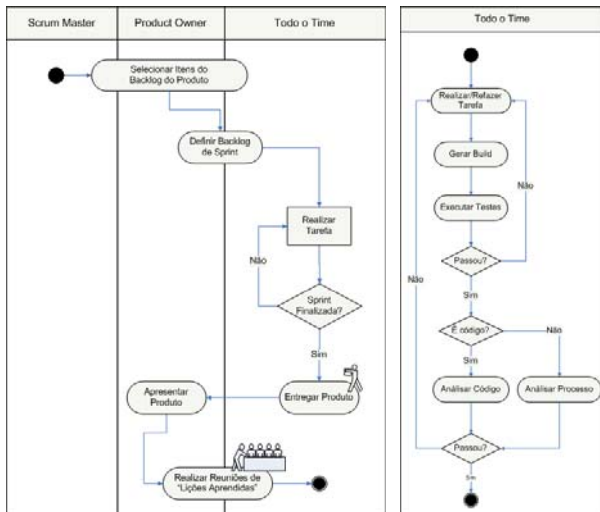


Figura 1: Processo de Desenvolvimento de Software da Fábrica Factory2u

Inicialmente, os integrantes da fábrica decidiram por estabelecer os dias e horários de reuniões, bem como os meios de comunicação, contendo inclusive, informes de tarefas e relatórios de estado da evolução do projeto, acompanhadas pelas análises comparativas de Velocidade e Execução, através de planilhas e gráficos (Sprint Burndown), conforme demonstrados na Figura 4.

Ao longo execução deste projeto fez-se o uso de diversas ferramentas, que viabilizassem o desenvolvimento distribuído, onde, realizamos através de *benchmark*, a escolha pela ferramenta mais indicada à realidade vivida pela Factory2u.

Dentre as ferramentas utilizadas, destacam-se, o Eclipse - Integrated Development Environment (IDE), ambiente de desenvolvimento integrado, disponível em <http://www.eclipse.org>; Subclipse - plugin para o Eclipse que permite trabalhar com servidores SVN para controle de versão de código e artefatos de software, disponível em <http://subclipse.tigris.org>; Cruise Control - framework para o processo de build contínuo disponível em <http://cruisecontrol.sourceforge.net>; e Ant - ferramenta utilizada para automatizar a construção de testes através de um arquivo no formato XML para descrever o processo de construção e suas dependências, disponível em <http://ant.apache.org>.

Como linguagem de programação, foi adotado JAVA para o desenvolvimento do componente Cliente/Servidor, responsável pelo armazenamento e conversão dos arquivos multimídia. Para o ambiente responsável pela publicação do conteúdo, foi adotado Python como linguagem de programação, e utilizado o

framework Django, devido à sua difundida facilidade de aprendizado e alta produtividade.

Dentre as estratégias de comunicação adotadas, ocorreram poucos encontros presenciais, mas, devido à necessidade de comunicação e incompatibilidade de horários, realizaram-se, com maior frequência, reuniões virtuais (Comunicação Síncrona) com dias e horários pré-definidos (dois encontros semanais), fazendo-se uso de softwares de comunicação VOIP como Skype, disponível em <http://www.skype.com> e Windows Live Messenger, disponível em <http://get.live.com/messenger>. Fez-se também o uso de Lista de Discussão e Emails individuais (Comunicação Assíncrona), de forma constante e sempre que necessário. A quantidade de emails gerados na Lista de Discussão durante o desenvolvimento do projeto é apresentado na Figura 3.

Outra ferramenta bastante utilizada foi o Google Docs, disponível em <http://docs.google.com/>, para o acompanhamento das atividades e colaboração online, onde foi criado a planilha de registro de requisitos e atividades, denominada pelo SCRUM de Ticket Tracking.

Fazendo uso das ferramentas apresentadas, as estórias para desenvolvimento eram geradas pelo cliente, analisadas e aprovadas ou rejeitadas pela Factory2u. Após aprovação, a estória era então encaminhada aos desenvolvedores para implementação, sendo o resultado gerado encaminhado aos testadores para verificação da correte e completude dos requisitos implementados. Após a execução dos testes por parte de integrantes especializados da fábrica, o componente de software era então liberado para teste final por parte do cliente, sendo ele o responsável final pela aceitação. Maiores detalhes sobre o desenvolvimento deste projeto é abordado na seção de resultados, apresentada a seguir.

Resultados

Existe a necessidade de uma especial atenção na concepção, elaboração e desenvolvimento de uma aplicação web que atenda requisitos de confiabilidade, segurança, e que possibilite controles adequados de acesso, preservando os direitos autorais.

Assim, ente trabalho se enquadra nestas características, uma vez que trabalha com o gerenciamento de conteúdo educacional, protegido por direitos autorais, e conteúdos de saúde, protegido pela confidencialidade médico-paciente no caso de conteúdos para teleconsulta

ou telediagnóstico. Portanto, o processo de desenvolvimento utilizado, teve papel fundamental para assegurar a qualidade do produto. Neste contexto, será descrito a seguir a descrição das seis Sprints de execução realizadas no desenvolvimento deste projeto. Cada sprint é uma iteração que segue o ciclo PDCA (Planejamento, Desenvolvimento, Controle e Avaliação) e entrega um incremento de software pronto.

Sprint 1

A Sprint 1 se preocupou em desenvolver uma interface gráfica do lado do cliente, para que um arquivo de vídeo pudesse ser enviado para um servidor de Streaming. Além do arquivo propriamente dito, também seriam enviadas informações adicionais sobre o arquivo para serem gravadas no banco de dados do NUTES, sendo priorizado apenas o envio do arquivo e suas informações, sem necessariamente, haver a conversão do vídeo.

Toda a interface foi codificada utilizando-se a biblioteca Swing da J2SE. Não foi encontrado problemas em sua utilização. Um servidor FTP Open Source proporcionou uma significativa ajuda no desenvolvimento das Sprints seguintes, pois foi possível inserir mais operações necessárias aos nossos requisitos além do recebimento do arquivo.

Sprint 2:

O principal objetivo da Sprint 2 foi a conversão dos arquivos de vídeo que seriam enviados ao servidor de Streaming. Como formatos de entrada, a aplicação receberia arquivos em MPEG e AVI. Foi definido, junto ao cliente, que uma resolução de 720 x 480 seria adequada e que num outro momento essa resolução seria parametrizável, uma vez que, para alguns vídeos destinados ao uso em telediagnósticos a qualidade da imagem é um fator crítico. O processo de conversão era iniciado no sistema operacional do cliente, após o término da recepção do arquivo pelo servidor de FTP. No final do projeto, a parametrização da resolução foi abortada por questões de tempo hábil ao cumprimento do projeto.

O cliente se mostrou satisfeito com os produtos das conversões, questionando a estratégia de implementação por entender, num primeiro momento, que o upload e a conversão da mídia tratavam-se de uma única operação. Após alguns esclarecimentos e discussões, a estratégia adotada pela factory2u foi aceita, separando em duas operações distintas.

Sprint 3:

Durante essa Sprint, a equipe definiu qual Player Web de conteúdo multimídia seria utilizado para a reprodução de áudio e vídeo. Foram levadas em consideração portabilidade e manutenibilidade como características desejadas para o Player Web. Finalmente, foi selecionado o Flowplayer, cuja licença é GPL 3, o que se adaptou ao objetivo da equipe e do cliente. Além disso, bugs e pendências das Sprint 1 e 2 também foram tratadas.

Mediante necessidades apontadas na redefinição das estórias, além da dependência e implementação dessas redefinições, algumas estórias da Sprint 4 foram detalhadas já nessa Sprint, facilitando sua posterior execução.

Sprint 4:

Num primeiro momento poderia parecer que seriam muitas estórias para uma Sprint de 15 dias, uma vez que seria implementado as estórias de Visualizar Conteúdo, Cadastro/Classificação, Configuração de Acesso e Publicação de Conteúdo.

Como na Sprint anterior, priorizou-se um trabalho de preparação para essa Sprint, referente à estória de Visualização de Conteúdo. Em seguida, foi verificado que as estórias Cadastro/Classificação, Configuração de Acesso e Publicação de Conteúdo se complementavam e poderiam compor uma única estória, a qual foi denominada unicamente de Cadastro/Classificação.

A partir dessa Sprint, as estórias começaram a ser desenvolvidas em Python, o que dificultou a programação, pois, somente dois integrantes da equipe dominavam a linguagem, e a estratégia para a utilização desta tecnologia, previa, que através da utilização do framework Django, e com uma breve capacitação pelos desenvolvedores mais experientes, os demais entrassem no ritmo planejado de produtividade, que não ocorreu, ficando a codificação do projeto à encargo de menos integrantes que o planejado originalmente. Conseqüentemente, só foi possível concluir a estória Visualizar Conteúdo, restando a estória de Cadastro/Classificação, que embora iniciado o processo de codificação pelos programadores, não foi concluída, além dos testes da estória Visualização de Conteúdo, postergados para a Sprint seguinte.

Sprint 5:

Essa Sprint objetivava concluir todo o projeto, assim, ficou programado para ser finalizada a estória de Cadastro/Classificação (unificada com as estórias de Configuração de

Acesso e Publicação de Conteúdo) e para ser executada a estória de Pesquisa Inteligente.

A falta de conhecimento da linguagem Phyton diminuiu a velocidade de programação das estórias. Mesmo assim, a estória de Cadastro/Classificação teve seu desenvolvimento concluído. Entretanto, a estória Pesquisa Inteligente foi detalhada junto ao cliente, e embora tenha sido iniciado seu desenvolvimento, a finalização além e os testes ficaram pendentes para Sprint seguinte.

Sprint 6:

A Sprint 6 (final) foi marcada pelo desenvolvimento da Pesquisa Inteligente, chamada assim por possibilitar a realização de operações com AND, OR, + e -, permitindo pesquisas mais eficientes. Além do desenvolvimento da suíte de testes, base para a validação dos requisitos. Também houve a correção de bugs de Sprints anteriores.

Ainda nesta última Sprint foi programado uma atividade extra – teste de fluxo completo, onde todo o sistema foi novamente testado, a fim de garantir uma maior qualidade e confiabilidade do produto a ser entregue ao cliente.

Após as seis Sprints de desenvolvimento, o cliente recebeu o produto final, dando o aceite como positivo, estando este ainda em fase de implantação. A Figura 2 apresenta algumas interfaces construídas durante o processo de desenvolvimento.

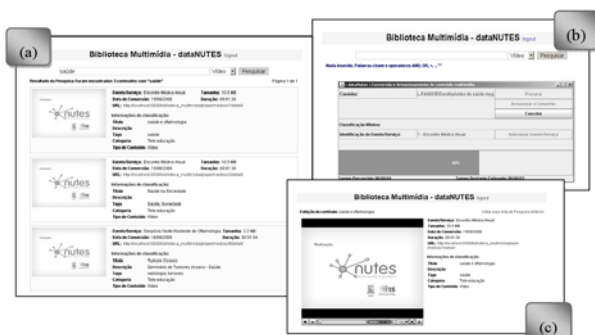


Figura 2: (a) Resultados do sistema de busca por palavras-chave; (b) O módulo de upload e conversão de arquivos multimídia; (c) Exibição do vídeo em modo Streaming, através do Player Web.

A execução deste projeto proporcionou também a obtenção de informações interessantes a respeito de como foi o processo de comunicação, as mudanças de escopo, velocidade de desenvolvimento e retrabalho.

A Figura 3 apresenta a quantidade de emails (comunicação assíncrona) trocados via lista de discussão durante o projeto.

Comunicação Assíncrona

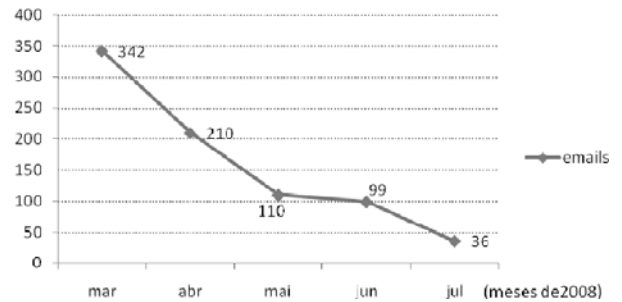


Figura 3: Quantidade de emails trocados

Observando o gráfico pode ser identificado que ao longo do período do projeto, de março a julho, a quantidade de troca de emails decresceu. Esta característica foi devido, principalmente ao amadurecimento da fábrica com relação ao processo de desenvolvimento e entendimento das atividades a ser executada. O mês de Julho obteve uma quantidade mínima, pois o projeto já estava em fase de conclusão.

A Figura 4 apresenta a velocidade de desenvolvimento da equipe, com base no planejado vs realizado.

Burndown

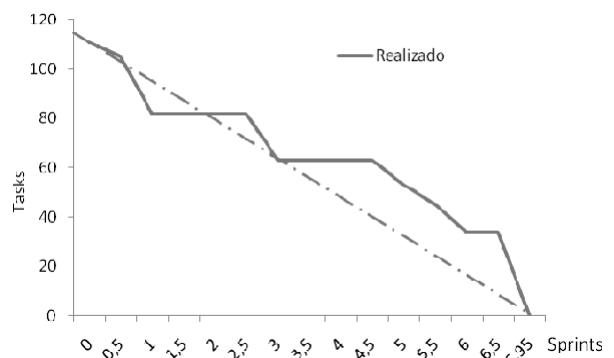


Figura 4: Sprint Burndown em relação às tarefas esperadas, executadas, acumuladas e finalizadas.

Analisando o Sprint Burndown, observa-se que nas primeiras Sprints o executado foi abaixo da média esperada. Esta característica ocorreu, pois o processo de desenvolvimento não estava bem assimilado pela fábrica. Outros fatores identificados como causadores da baixa produtividade na Sprint 1 foram problemas na comunicação, comprometimento com as reuniões virtuais e mudanças no escopo solicitadas pelo cliente.

Com relação aos testes realizados no sistema, foram gerados 52 casos de teste, compreendidos em 4 suites de teste, como pode ser visualizado na Figura 5.

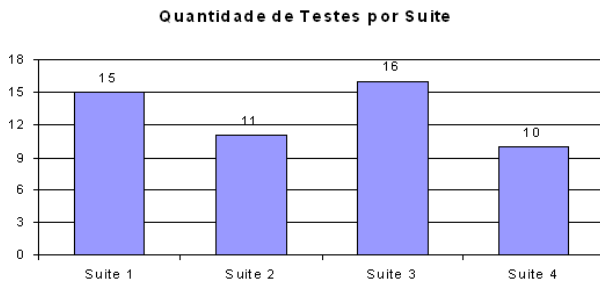


Figura 5: Quantidade de Teste por Suíte

Os testes foram de vital importância para a qualidade final do produto gerado. Outra característica foi a execução dos testes por membros da fábrica que não faziam parte do desenvolvimento. Isto levou a uma independência no processo de testes uma vez que o testador não estava “viciado” com o produto a ser testado.

A quantidade de retrabalho gerado pelo resultado dos testes foi relativamente grande. Da primeira suíte 6 casos de teste falharam, da segunda suíte 4 casos de teste falharam, da terceira suíte 6 casos de teste falharam e da quarta suíte 3 casos de teste falharam. Foi ainda executado uma varredura exploratória onde outras 5 novas falhas foram encontradas. Com base nas suítes de teste, 19 falhas foram encontradas, contabilizando um total de 36,53%, como apresentado na Figura 6. Se adicionado as falhas encontradas pela execução dos testes exploratórios, seria obtido o total de 24 falhas o que corresponde a 42,10% do total de testes realizados.

Quantidade de Testes que Passaram/Falharam (Sem Exploratório)



Figura 6: Quantidade de testes que passaram/falharam, sem contar testes exploratórios.

O tamanho final do projeto foi dimensionado em **Linhas de Código** (LOC – Lines of Code), apresentando aproximadamente 8.779 Linhas de Código, representando o desenvolvimento de 128 **story points** determinados pela fábrica através dos Planning Poker e acordadas com o cliente. *Story point* é uma medida relativa do uso de recursos (dificuldade/ complexidade) que é útil para o planejamento e dimensionamento do tamanho de software.

Discussão e Conclusões

O processo baseado no SCRUM, numa primeira análise extremamente simples, se demonstrou bastante eficaz, gerando os resultados desejados. O site da fábrica (www.factory2u.net), entre outras ferramentas, se apresentou como ferramenta essencial de comunicação, embora tenha havido, em algumas circunstâncias, certo atraso de sincronização das informações em função do trabalho distribuído da Equipe.

O produto final gerado, Biblioteca Multimídia do dataNUTES, está em fase de implantação no Projeto REDENUTES [6], e será ferramenta para compartilhamento de uma biblioteca de sessões de videoconferências na Web, tendo como estudo de caso as sessões produzidas pela Rede de Núcleos de Telessaúde (NUTES) de Pernambuco, constituindo-se como um sistema de armazenamento eficiente e de boa qualidade para essas sessões, e ainda permitindo, posteriormente, a elaboração de uma biblioteca de imagens médicas, utilizando a internet como meio para disponibilização.

Referências

- [1] PRESSMAN, Roger. Software Engineering: A Practitioner's Approach, 6ª edição, Mc Graw Hill, 2005.
 - [2] Herbsleb, J.D. e Grinter, R. “Splitting the organization and integrating the code: Conway's Law revisited”, In: ICSE 1999, Carolina do Norte, EUA, 1999.
 - [3] Takeuchi, H. and I. Nonaka, The New New Product Development Game. Harvard Business Review, 1986 (January-February).
 - [3] World Health Organisation – WHO/OMS, <http://www.who.int>, acessado em 03/07/2008;
 - [5] J. O. A. Segundo, C.H.L.Souza, e M.A.Novaes, “Biblioteca Digital de Videoconferências: Uma Estratégia para Ampliar o Acesso a Programas de Capacitação para o PSF”, CBIS 2006, Florianópolis, 14-18 de outubro de 2006, p.1255-1260.
 - [6] Projeto REDENUTES, <http://www.redenutes.ufpe.br>, acessado em 08/08/2008;
- Contato**
 Alexandre J. H. de O. Luna, NUTES-Hospital das Clínicas (HC) da UFPE, Av. Prof. Moraes Rego, s/n, 2º Andar, Cidade Universitária, Recife-PE, CEP.: 50.670-420, Fone: (81) 2126-3903, Fax: (81) 2126-3904,
 E-mail: alexandre.luna@nutes.ufpe.br.