

SGEMEOS: UM GESTOR DE EVENTOS SOCIAIS BASEADO EM SERVIÇOS WEB

Luiz Marcus Monteiro de Almeida Santos e Admilson de Ribamar Lima Ribeiro

Departamento de Computação – Universidade Federal de Sergipe

Jardim Rosa Elze – São Cristóvão – SE – Brasil

RESUMO

A grande variedade de linguagens e plataformas para o desenvolvimento de sistemas transforma-se em problema quando surge a necessidade de compartilhamento de informações entre uma ou mais aplicações heterogêneas. Como solução, existem componentes como os serviços *Web* que padronizam o formato e a transmissão dos dados. Diante da possibilidade de interoperabilidade oferecida pelos serviços *Web*, este trabalho apresenta o SGEMEOS, um sistema gerenciador de eventos sociais no modelo cliente-servidor que faz uso destes componentes para viabilizar a troca de dados entre o servidor e os clientes de plataformas distintas.

PALAVRAS-CHAVE

Eventos sociais, Web Services, Dispositivos Móveis, Sistemas Distribuídos

1. INTRODUÇÃO

Em eventos sociais como congressos, feiras e apresentações culturais, existe uma quantidade significativa de informações a serem processadas. Nesse tipo de negócio estão inseridos dados como as inscrições dos interessados, os artigos submetidos, os horários e os locais do evento. Como soluções para a organização e manipulação desses dados foram produzidas aplicações de gerenciamento de conferências como as descritas por [Cheng et al. 2008] e [SHIH et al. 2001].

Atualmente, as ferramentas que realizam a gestão de eventos sociais podem ser desenvolvidas em diversas linguagens de programação como PHP [PHP Group 2001] ou Java [Deitel 2006] e executadas em diferentes plataformas como a *Web*. Essa variedade tecnológica apresenta problemas quando é necessária a troca de informações entre uma ou mais aplicações heterogêneas. Sendo assim, se um participante do evento preferir usar uma aplicação móvel para acessar seus dados pessoais que estão persistidos em um sistema *Web*, o recebimento deles pode ser prejudicado ou não efetuado devido às diferenças entre as plataformas.

Para que a integração entre sistemas atinjam os níveis de custo-benefício satisfatórios, é preciso que a execução desta tarefa seja padronizada. Para atender a essas especificações os *Web Services* [Erl 2004], ou em português, serviços *Web* foram desenvolvidos. Eles facilitam a troca de informações entre as aplicações através do envio e recebimento de dados no formato XML (*Extensible Markup Language*) [W3C 2000] via protocolos baseados em Internet. A ação dos serviços *Web* garante a interoperabilidade das aplicações envolvidas de modo que o fluxo de informações entre elas não dependa da tecnologia ou linguagem de programação em que cada aplicação foi desenvolvida.

Sendo uma das características dos serviços *Web* a transmissão de dados entre diferentes ambientes, por meio deles torna-se possível a criação de uma ferramenta de gerenciamento de eventos sociais com transmissão flexível de informações, a qual pode utilizar-se da integração entre a plataforma móvel e a *Web* para melhor atender as necessidades dos usuários. A essa ferramenta deu-se o nome de SGEMEOS que será discutida neste artigo.

Este artigo está organizado como segue: na seção 2, são apresentados os trabalhos relacionados; na seção 3 é descrita a metodologia usada no desenvolvimento do SGEMEOS; na seção 4, a arquitetura do SGEMEOS é exposta; na seção 5, são explicitadas algumas questões de implementação; na seção 6, são expostos os resultados encontrados através da utilização do sistema; e na seção 7 são feitas as considerações finais.

2. TRABALHOS CORRELATOS

Como mencionado na introdução, atualmente, a maior parte de sistemas propostos para o gerenciamento de eventos sociais são voltados para uma única plataforma. Geralmente, a escolhida é a *Web* por motivos como acessibilidade e escalabilidade. Ferramentas como as descritas em [Cheng et al. 2008], [Shih et al. 2001] e [Flisys, 2007] possuem como plataforma alvo a *Web*.

Através de [Snodgrass 1999] é possível ter acesso a uma lista ferramentas, resultantes de trabalhos científicos, que têm como objetivo comum o gerenciamento de conferências. A listagem dessas ferramentas é feita seguindo alguns critérios como suporte a envio de artigos, atribuição de revisores, submissão de revisões, inscrição de participantes e a linguagem de programação usada no desenvolvimento. A listagem conta também com a descrição de cada ferramenta.

As ferramentas apresentadas em [Snodgrass 1999], apesar de garantirem o gerenciamento das informações de um evento social, não possuem componentes que viabilizem a apresentação desses dados em plataformas distintas. Além disso, elas estão direcionadas a apenas um tipo de evento social, no caso, conferências, desse modo, não podem ser usadas em outros tipos de eventos. No SGEMEOS essas limitações foram corrigidas. O SGEMEOS permite que os dados possam ser acessados em múltiplas plataformas heterogêneas através do uso de serviços *Web*. Por serem suportados por várias linguagens de programação como Java e PHP, os serviços *Web* permitem que os desenvolvedores fiquem livres para escolherem a linguagem sobre a qual um cliente para o sistema poderá ser desenvolvido.

3. METODOLOGIA UTILIZADA

Sendo o SGEMEOS um sistema baseado em serviços *Web*, faz-se necessária a utilização de uma metodologia que suporte a construção de aplicações que façam uso destes componentes. Sendo assim, para a definição e construção dos serviços foi usada a metodologia de desenvolvimento de *software* orientado a serviços descrita em [Sommerville 2007]. Essa metodologia trata do desenvolvimento de sistemas de *softwares* confiáveis que usam os serviços, por conta própria ou em conjunto com outros tipos de componentes. Essencialmente, a principal preocupação é o desenvolvimento de *software* com reuso.

A metodologia orientada a serviços consiste, basicamente, em três estados lógicos:

- **Identificação dos serviços candidatos:** Nesta etapa são identificados os possíveis serviços que devem ser implementados e ter seus requisitos definidos.
- **Projeto do serviço:** Nesta etapa é feito o projeto lógico e as interfaces WSDL (*Web Service Definition Language*) [W3C 2001] do serviço. Ela envolve, basicamente, pensar sobre as operações do serviço e nas mensagens a serem trocadas quando o serviço for socilitado.
- **Implementação e implantação do serviço:** Nesta fase os serviços são implementados e disponibilizados para uso. A disponibilização envolve a publicação do serviço usando UDDI (*Universal Description, Discovery, and Integration*) [Mckee et al. 2002] e a sua instalação em um servidor *Web*.

Seguindo cada etapa especificada por esta metodologia foi possível o desenvolvimento dos seguintes serviços:

- **Autenticação:** tem por objetivo dar acesso ao sistema para o usuário.
- **Administração:** é constituído por operações ligadas a questões administrativas como o cadastramento dos responsáveis pela organização do evento.
- **Organização:** agrega operações referentes aos dados dos organizadores do evento e suas tarefas.
- **Exibição:** possui operações relacionadas ao que será exposto no evento, como palestras e minicursos, incluindo seu local e horário.
- **Participação:** possui operações referentes aos interessados em participar do evento social.

4. ARQUITETURA DO SGEMEOS

O SGEMEOS é um sistema distribuído no modelo cliente-servidor. O lado servidor gerencia e manipula os dados por meio de classes, sendo ele responsável também pela disponibilização dos serviços *Web* no formato

WSDL através da Internet. Por sua vez, o lado cliente é uma aplicação baseada em J2ME (Java 2 *Micro Edition*) [Knudsen 2005], mais especificamente um MIDlet [Riggs et al. 2003], que consome os serviços, por meio de processos de requisições e respostas, à medida que lhe forem necessários. A figura 1 mostra o esquema dessa arquitetura.

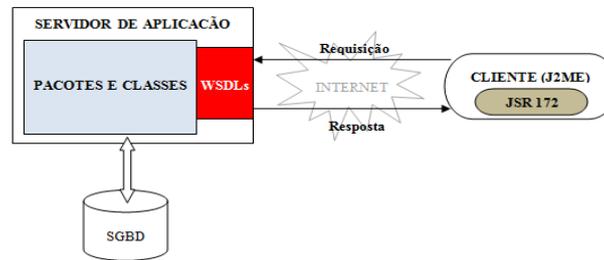


Figura 1. Arquitetura do SGEMEOS.

4.1 Servidor

O servidor é o componente do SGEMEOS responsável por disponibilizar os serviços que servem como meio de transmissão de dados entre o SGBD (Sistema Gerenciador de Banco de Dados) e o cliente. Seguindo o processo definido pela metodologia orientada a serviços, antes de identificarmos os serviços, é importante que, primeiramente, sejam definidos os perfis de usuário a serem atendidos pelo SGEMEOS. Esses perfis foram definidos de modo que fossem diretamente ligados ao domínio do gerenciamento de eventos sociais. Os perfis do sistema são:

- **Administrador:** possui como funções o cadastramento e a edição dos dados dos organizadores. Ele também tem a capacidade comandar a preparação do evento através da atribuição de atividades aos organizadores. O administrador também pode editar informações dos outros perfis.
- **Organizador:** gerencia os participantes e os expositores, além de organizar questões relativas às exposições, como os locais e os horários. Um organizador não possui acesso aos dados do administrador nem de outros organizadores.
- **Expositor:** acompanha os detalhes de suas exposições através da visualização dos locais e horários em quem elas serão realizadas.
- **Participante:** esse perfil é o único que possui autonomia de registrar-se no sistema. Além disso, cada participante pode visualizar os dados das exposições disponíveis e acessar informações sobre o expositor responsável por ela. A sua participação no evento é confirmada por um organizador.

Cada serviço aloca operações que estão associadas a um perfil. Dessa forma, as operações que envolvem os dados de um participante foram atribuídas a um mesmo serviço, e assim por diante. A nomenclatura dos serviços e de suas operações também foi pensada visando facilitar o uso posterior dos serviços.

A figura 2 exibe o fragmento das operações vinculadas ao serviço Organização, o qual faz referência ao perfil organizador. Nela também é possível perceber outras informações atreladas às operações como seus dados de entrada e saída.

Operação	Descrição	Entrada	Saída
obterOrganizador	Utilizada para obter as informações de um Organizador	ID do organizador	Instância de um organizador
obterOrganizadores	Utilizada para obter a lista de organizadores cadastrados	Sem entrada	Lista de organizadores
atualizarOrganizador	Utilizada para atualizar os dados do organizador	Instância de um organizador	<i>Boolean</i>

Figura 2. Operações do serviço Organização.

4.2 Clientes

O cliente é o componente do SGEMEOS responsável por disponibilizar as funcionalidades e dados aos usuários através do consumo de serviços *Web*. Ele pode ser desenvolvido em qualquer plataforma desde que

a linguagem a ser utilizada tenha suporte a serviços *Web*. Em um primeiro momento um cliente móvel para a plataforma J2ME foi desenvolvido. Posteriormente, outros clientes para diferentes plataformas como Android e *Web*, também foram criados.

5. IMPLEMENTAÇÃO

Para a codificação do SGEMEOS a linguagem de programação Java foi escolhida. Essa escolha se deu, primeiramente, pelo fato de ela ser uma linguagem bastante difundida entre os profissionais da área de TI, o que garante um maior número de usuários, e conseqüentemente, um número relevante de citações. Outro motivo foi a possibilidade de desenvolver tanto para a plataforma *Web* quanto para móvel sem a necessidade de mudança de linguagem.

Outras tecnologias definidas para a implementação do sistema foram: o IDE NetBeans [Oracle 2011], o software *Umbrello UML Modeler* [Hensgen 2001] usado para a modelagem dos diversos diagramas, em UML (*Unified Modeling Language*) [Booch et al. 2006], necessários para a documentação do sistema, a exemplo dos diagramas de caso de uso e de classes. A persistência dos dados é de responsabilidade do SGBD (Sistema Gerenciador de Bancos de Dados) MySQL 5.5 [Oracle 2011]. Por fim, a hospedagem do sistema foi atribuída ao GlassFish 3.0 [Oracle 2011], um servidor de aplicação com suporte para linguagem Java.

5.1 Servidor

A implementação do servidor envolveu a criação de pacotes de classes Java. Cada pacote foi criado para agrupar classes que tivessem objetivos em comum. Assim, o servidor é constituído de pacotes que possuem as classes que representam os perfis, as classes que fazem comunicação com o SGBD, as classes que representam as entidades presentes no domínio do negócio e as classes que descrevem os serviços.

O pacote *ufs.dcomp.sgemeos.perfis* armazena as classes Administrador, Organizador, Participante e Expositor, as quais representam, respectivamente, os perfis de mesmo nome. Essas classes fazem uso da JAXB (*Java Architecture for XML Binding*) [Grebac 2006] para que o cliente J2ME possa receber as informações dos usuários de maneira estruturada. Além do pacote de perfis, existe o pacote que agrupa o conjunto de classes que representam as entidades presentes no domínio do negócio. Essas classes complementam os perfis para que todo o cenário do gerenciamento de eventos seja abstraído. O pacote que armazena essas classes é o *ufs.dcomp.sgemeos.negocio*. Nele estão presentes as classes Local, Horário, Exposição, Evento, Atividade e Autenticar.

No SGEMEOS também existem as classes que são responsáveis pela comunicação com o SGBD. Essas classes estão armazenadas no pacote *ufs.dcomp.sgemeos.dao* e, como o nome do pacote sugere, foram implementadas no padrão DAO (*Data Access Object*). Os métodos presentes nas classes desse pacote são utilizados pelos serviços *Web* para a manipulação dos dados. Com a base de pacotes estabelecida, foi possível criar as classes que descrevem os serviços. É nessas classes onde estão definidos os métodos que representam as operações dos serviços. Cada classe é referente a apenas um serviço.

Através dessas classes são gerados os serviços, mais especificamente, os WSDLs correspondentes. A geração dos serviços a partir das classes é feita automaticamente pelo IDE (*Integrated Development Environment*). O pacote que contém as classes geradoras de serviços foi nomeado de *ufs.dcomp.sgemeos.servicos*. A figura 3 exhibe um resumo da estrutura de pacotes do SGEMEOS.

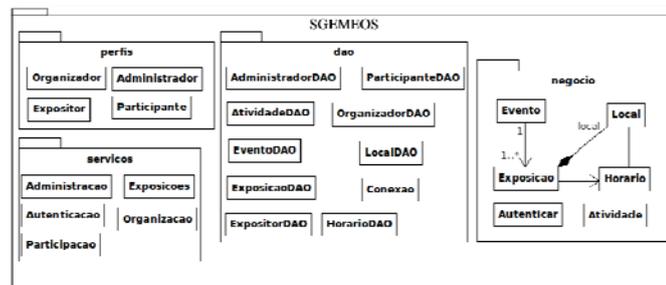


Figura 3. Resumo da estrutura de pacotes do SGEMEOS.

5.2 Clientes

No projeto inicial do SGEMEOS foi definido que o cliente primário seria desenvolvido para a plataforma J2ME. Porém, para uma melhor avaliação da flexibilidade de plataforma do SGEMEOS, foram criados diferentes clientes, sendo um para plataforma Android e outro para *Web*.

5.2.1 Cliente J2ME

Consiste de um Midlet desenvolvido através da composição de formulários, listas e menus. Enquanto alguns comandos são apenas de entrada e saída de telas, outros possuem chamadas a serviços *Web* associadas. Porém, antes de utilizar os serviços *Web* é preciso que os WSDLs sejam importados para o cliente. Esse procedimento foi feito usando o IDE em conjunto com a JSR 172 [Ellis et al. 2004], onde ao ser fornecida a URL do WSDL são geradas classes Java chamadas *stubs*. O procedimento tem o contexto similar ao RMI (*Remote Method Invocation*) [Coulouris 2007]. Os *stubs* são classes Java que implementam o comportamento do serviço. Dessa forma, o acesso às operações dos serviços é feita por meio dos *stubs*. A figura 4 mostra a tela referente ao perfil administrador do cliente J2ME.

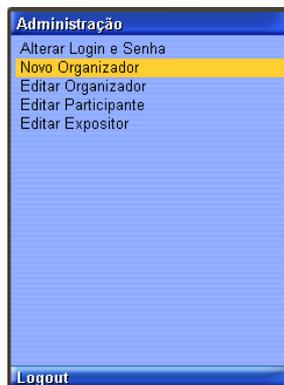


Figura 4. Tela de Administração do cliente J2ME.

5.2.2 Cliente Android

Este cliente foi construído através do uso do Android SDK [Google 2011]. Além disso, a IDE utilizada para o desenvolvimento foi o Eclipse. O Eclipse foi escolhido pelo fato de suportar o SDK que faz uso da linguagem Java em suas APIs (*Application Programming Interface*). A IDE também permite a emulação e a depuração da aplicação.

Como a base do SGEMEOS são seus serviços *Web*, fez-se necessário a integração de uma biblioteca que desse suporte a esta tecnologia no projeto. A KSoap2 [Haustein 2006] foi a API escolhida para ser a responsável por essa tarefa.

A implementação do cliente Android decorreu da mesma forma que no J2ME, ou seja, associando comandos de telas com as operações dos serviços *Web*. A tela de login do cliente Android é apresentada na figura 5.



Figura 5. Tela de login do cliente Android.

5.2.3 Cliente Web

Além dos clientes voltados para plataformas móveis, foi desenvolvido também um cliente alternativo. Esse cliente foi produzido direcionado ao ambiente *Web*, como mostra a figura 6.



Figura 6. Tela de login do cliente *Web*.

No desenvolvimento deste cliente foi utilizada a linguagem de programação PHP. A escolha da PHP foi motivada pelo fato de que todos os clientes e o servidor do SGEMEOS terem sido construídos por meio da Java. Desse modo, não só a versatilidade de plataforma como também a flexibilidade com relação à linguagem de programação pôde ser testada.

6. RESULTADOS

Como o principal objetivo do SGEMEOS é gerenciar eventos sociais, o melhor cenário para sua avaliação seria a sua utilização em um ambiente real. Visando isso, o sistema foi usado para gerir a II Semana de Computação da Universidade Federal de Sergipe, a qual ocorreu entre os dias 5 e 8 de Outubro de 2011.

Ao SGEMEOS coube a responsabilidade de gerenciar as inscrições, exibir relatórios e apresentar aos inscritos as informações vinculadas ao evento como datas, locais de palestras e responsáveis pelas palestras.

Para atender as necessidades não só dos organizadores como também dos participantes ficou definido que além da implantação do servidor seriam utilizados também os clientes para a plataforma Android e para a plataforma *Web*. O cliente J2ME foi excluído devido à falta de serviços de Internet que proporcionassem aos usuários acesso gratuito para a visualização das informações.

O cliente *Web*, após ser hospedado, tornou-se acessível a partir de uma URL. Nele, os participantes podiam inscrever-se no evento, escolher quais minicursos gostariam de participar e realizar o pagamento da inscrição como mostra a figura 7.

Módulo de Participante									
Menu Alterar Dados Minicursos Disponíveis Meus Minicursos	Atenção, Fulano da Silva. Se você deseja conhecer o acesse ESTE LINK . Minicursos Disponíveis <table border="1"> <thead> <tr> <th>Título</th> <th>Descrição</th> </tr> </thead> <tbody> <tr> <td>Aplicações Web em PHP utilizando o framework Symfony</td> <td></td> </tr> <tr> <td>Conheça o Linux</td> <td>Conhecimentos teóricos e práticos sobre a Plataforma Linux e suas distros. Demonstrações práticas dos programas em Ambiente Gráfico incluso no Ubuntu.</td> </tr> <tr> <td></td> <td>Introdução a sistemas de controle de versão. Instalando e configurando o</td> </tr> </tbody> </table>	Título	Descrição	Aplicações Web em PHP utilizando o framework Symfony		Conheça o Linux	Conhecimentos teóricos e práticos sobre a Plataforma Linux e suas distros. Demonstrações práticas dos programas em Ambiente Gráfico incluso no Ubuntu.		Introdução a sistemas de controle de versão. Instalando e configurando o
Título	Descrição								
Aplicações Web em PHP utilizando o framework Symfony									
Conheça o Linux	Conhecimentos teóricos e práticos sobre a Plataforma Linux e suas distros. Demonstrações práticas dos programas em Ambiente Gráfico incluso no Ubuntu.								
	Introdução a sistemas de controle de versão. Instalando e configurando o								
Status A pagar R\$: 27 Pagar com									

Figura 7. Fragmento da tela de escolha de minicursos do cliente *Web*.

Os organizadores ao acessarem o cliente *Web* tinham acesso a informações como total de inscritos no evento, total de inscritos em um determinado minicurso, locais, horários, entre outras. Além disso, era possível também alterar os inscritos de um minicurso, alterar os dados cadastrais de um inscrito e gerar listas de presença no formato PDF.

Em quanto à parte burocrática do evento podia ser resolvida por meio do cliente *Web*, as informações mais relevantes para os participantes eram disponibilizadas através do cliente Android. Ao entrar no cliente, o participante tinha acesso às informações associadas ao evento como horários, locais e responsáveis pelas palestras, além de saber quais palestras estavam acontecendo no momento (figura 8).



Figura 8. Tela de inicial do cliente Android com menu de opções ativo.

Como o acesso ao sistema podia ser feito em múltiplas plataformas, os participantes tinham a possibilidade de escolher do ambiente mais conveniente para a visualização das informações. Além disso, a organização, o gerenciamento e a persistência dos dados possibilitou aos responsáveis pela organização do evento uma economia de tempo nas realizações das tarefas.

A importância do gerenciamento das informações do evento feita pelo SGEMEOS ficou ainda mais evidente quando os dados quantitativos foram analisados ao final do evento. A tabela 1 mostra as estatísticas finais do evento.

Tabela 1. Tela de Administração do cliente J2ME.

Total de inscritos	410
Total de palestras	16
Total de minicursos	12
Nº de locais utilizados	5

Através das estatísticas do evento foi possível constatar que independentemente do tamanho físico, do volume de pessoas e do tipo do evento é importante que exista um sistema de informação capaz de manipular, persistir e apresentar as informações de maneira legível e eficiente. Quando esse objetivo é alcançado e junto a ele está agregada a flexibilidade de acesso a essas informações, os organizadores ganham em eficiência nas tarefas e os participantes em qualidade de serviço.

7. CONCLUSÃO

Utilizando-se da flexibilidade oferecida pelos serviços *Web*, este trabalho apresenta um sistema chamado SGEMEOS direcionado ao gerenciamento de eventos sociais. O sistema foi desenvolvido no modelo de cliente-servidor, de modo que o servidor e o cliente pudessem pertencer a plataformas distintas.

A partir do sistema foram disponibilizadas funcionalidades como o cadastramento de organizadores, visualização de exposições, registro de participantes, entre outras. Essas funcionalidades foram avaliadas

através da utilização do sistema em um evento social real. Como resultado constatou-se que o uso dos serviços *Web* garantiu o fluxo de dados entre plataformas distintas, de modo que os clientes do sistema pudessem acessar as informações persistidas em um servidor vinculado a um ambiente heterogêneo. Além disso, a avaliação também mostrou que plataformas podem ser integradas, desde que as informações estejam estruturadas e os protocolos de transmissão sejam abertos.

No entanto, apesar de o sistema apresentado neste trabalho possuir diversas funcionalidades, existem muitas modificações e extensões que podem ser agregadas a ele. Primeiramente, visando um melhor gerenciamento dos recursos associados a um evento social e uma melhor integração entre seus participantes, seria interessante a construção e disponibilização de serviços que facilitassem a localização das pessoas no ambiente do evento (geolocalização), proporcionassem a comunicação entre usuários do sistema (chat) e minimizassem as dificuldades relacionadas à administração dos recursos financeiros (monitoramento de finanças). Além disso, o desenvolvimento de um cliente voltado para dispositivos que fazem uso do sistema operacional iOS, como o iPhone, também seria uma boa contribuição futura.

REFERÊNCIAS

- BOOCH, G. et al, 2006. *Uml: Guia do Usuário*. Ed. 2. Rio de Janeiro, Elsevier, Campus.
- CHENG Z. et al, 2008. *Design and implementation of a collaborative conference management system*. Computer Supported Cooperative Work in Design, 2008. CSCWD 2008. 12th International Conference on , vol., no., pp.5-10, 16-18 April 2008.
- COULOURIS, G., 2007. *Sistemas Distribuídos – Conceitos e Projeto*. Bookman. 4a. Edição.
- DEITEL, H. M., DEITEL P. J., 2006. *Java: Como Programar*. Ed. 6. Upper Saddle River, New Jersey, Prentice Hall.
- ELLIS, Jon. YOUNG, Mark., 2004. *JSR 172: J2ME Web Services Specification*. Disponível em < <http://jcp.org/en/jsr/detail?id=172> >
- ERL, T., 2004. *Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services*. Prentice Hall PTR.
- _____. *Flisys*, 2007. Disponível em < <http://sourceforge.net/projects/flisys/>>
- GOOGLE., 2011. *Android SDK*. Disponível em <<http://developer.android.com/sdk/index.html>>
- GREBAC, M., 2006. *JSR 222: Java Architecture for XML Binding (JAXB) 2.0*. Disponível em < <http://jcp.org/en/jsr/detail?id=222> >
- HAUSTEIN, S., 2006. *KSoap2*. Disponível em <ksoap2.sourceforge.net>.
- HENSGEN, P., 2001. *Manual do Umbrello UML Modeller*. Disponível em <<http://docs.kde.org/stable/pt/kdesdk/umbrello/>>
- KNUDSEN, J.; LI S., 2005. *Beginning J2ME: From Novice to Professional*. Ed. 3. Apress.
- MCKEE, B. et al, 2002. *UDDI Version 2.04 API Specification*. Disponível em < http://uddi.org/pubs/ProgrammersAPI_v2.pdf >
- ORACLE., 2011. *Glass Fish*. Disponível em: < <http://glassfish.java.net/>>.
- ORACLE., 2011. *MySQL*. Disponível em: < <http://dev.mysql.com/>>.
- ORACLE., 2011. *NetBeans*. Disponível em: < <http://netbeans.org/>>.
- PHP GROUP., 2001. *PHP: Hypertext Preprocessor*. Disponível em:<<http://www.php.net/>>
- RIGGS, R. et al, 2003. *Programming Wireless Devices with the Java 2 Platform, Micro Edition, Second Edition*. Addison Wesley.
- SHIH, T.K. et al, 2001. Virtual conference management system. *Information Networking. Proceedings. 15th International Conference on*, vol., no., pp.776-781.
- SNODGRASS R., 1999. *Summary of Conference Management Software*. Disponível em <http://www.acm.org/sigs/sgb/summary.html>
- SOMMERVILLE, I., 2007. *Software Engineering*. Ed. 8. Addison-Wesley.
- W3C., 2004. *Web Services Architecture*. Disponível em < <http://www.w3.org/TR/ws-arch/>>
- W3C., 2000. *Extensible Markup Language (XML) 1.0 (Second Edition)*. W3C Recommendation. Disponível em <<http://www.w3.org/TR/2000/REC-xml-20001006>>
- W3C., 2001. *Web Services Description Language (WSDL) 1.1*. Disponível em <<http://www.w3.org/TR/wsdl>>