

Um Ambiente de Computação Ubíqua para o Ensino baseado em *ABP*

Caroline B. Perlin, Luiz H. Z. Santana, Diogo S. Martins, Helen de Freitas Santos¹,
Antonio F. do Prado, Wanderley L. de Souza, Mauro Biajiz

Grupo de Computação Ubíqua (GCU)
Departamento de Computação (DC)
Universidade Federal de São Carlos (UFSCar)
Caixa Postal 676
São Carlos – SP, 13565-905
BRASIL

{caroline_perlin, luiz_santana, diogo_martins, prado, desouza, mauro}@dc.usfcar.br

¹Programa de Pós-Graduação em Informática em Saúde
Universidade Federal de São Paulo (UNIFESP)
Rua Botucatu, 740
São Paulo – SP, 04023-900
BRASIL
helen-pg@dis.epm.br

Abstract

Problem Based Learning (PBL) is a teaching-learning process that aims to stimulate the student to build his own knowledge, mainly by means of interactions that occur in small groups of students and professors. This paper proposes an Ubiquitous Computing environment, whose kernel is a Web system with collaborative tools, to support the interactions among different actors involved in the PBL teaching-learning process. In order to allow this environment to be ubiquitous, the Web pages of this system are adapted, through Semantic Web Services, so they can be accessed by a huge variety of small mobile devices (e.g., cell phones, smart phones, tablets, Personal Digital Assistants-PDAs).

Keywords: Ubiquitous Computing, Problem Based Learning, Mobile Communication, Web Services, Computer Supported Collaborative Learning.

Resumo

Aprendizagem Baseada em Problemas (ABP) é um processo de ensino-aprendizagem que busca estimular o estudante a construir o seu próprio conhecimento, sobretudo via interações que ocorrem em pequenos grupos de estudantes e professores. Este artigo propõe um ambiente de Computação Ubíqua, cujo núcleo é um sistema *Web* com ferramentas colaborativas, para dar suporte computacional às interações entre os diferentes atores envolvidos no processo de ensino-aprendizagem baseado em ABP. A fim de tornar esse ambiente ubíquo, as páginas desse sistema *Web* são adaptadas, através de Serviços *Web* Semânticos, para que possam ser acessadas por uma grande variedade de pequenos dispositivos móveis (e.g., celulares, *smartphones*, *tablets*, *Personal Digital Assistants-PDAs*).

Palavras chave: Computação Ubíqua, Aprendizagem Baseada em Problemas, Comunicação Móvel, Serviços Web, Aprendizagem Colaborativa Suportada por Computador.

1. Introdução

Com o advento da Internet e dos inúmeros dispositivos computacionais que a acessam, via redes com ou sem fio, a troca de informações em qualquer lugar, a qualquer hora e usando qualquer tipo de dispositivo, entre pessoas de um grupo, que exercem algum tipo de atividade colaborativa (e.g. trabalho, estudo), tornou-se vital para o sucesso dessa atividade. Para que um ambiente possa suportar essa comunicação ubíqua, este deve prover ferramentas para o gerenciamento de usuários, de dispositivos, de redes de acesso e do próprio conteúdo que está sendo compartilhado.

Em contraste à era dos *mainframes*, onde um mesmo computador costumava ser usado por vários usuários, na era da Computação Ubíqua ou *Pervasiva* cada usuário possui diversos tipos de dispositivos computacionais (e.g., *desktops*, *laptops*, dispositivos móveis, *wearable devices*, sensores) com diferentes capacidades de processamento, memória, energia e tela [1]. A Computação Ubíqua concentra-se na interoperabilidade entre esses dispositivos, visando tornar a computação e a comunicação completamente transparentes aos usuários [2].

Uma grande variedade de aplicações em diferentes domínios é alvo da Computação Ubíqua. No domínio pedagógico, um dos principais objetivos é tornar o processo de ensino-aprendizagem ubíquo, ou seja, estendê-lo para além da sala de aula. Para que esse processo seja eficaz, é necessário que o conteúdo possa ser apresentado ao estudante tão logo este o solicite, independentemente do local onde este se encontra e do tipo de dispositivo que está usando para acessá-lo.

Paralelamente à evolução de processos computacionais em direção à ubiquidade, há processos de ensino-aprendizagem que caminham neste mesmo sentido. Em particular, nos processos baseados na metodologia *Problem Based Learning (PBL)* [3], também denominada Aprendizagem Baseada em Problemas (ABP), o estudante é incentivado a buscar o conhecimento tanto através de trabalhos individuais quanto através de trabalhos coletivos. Nas atividades desses processos ocorrem interações estudante-estudante e estudante-professor, que colaboram para que o conhecimento seja construído cognitivamente por cada estudante. Neste sentido, trabalhos em *Computer Supported Collaborative Learning (CSCL)* [4] buscam prover ferramentas computacionais para suportar tais interações.

A fim de buscar soluções que permitam aos processos de ensino-aprendizagem baseados em ABP terem o suporte de um computador, este artigo propõe um ambiente de Computação Ubíqua que também permite a tais processos estarem disponíveis a qualquer hora e em qualquer lugar. Esse ambiente tem uma arquitetura que possibilita o seu uso através da *Web* e o seu acesso via computadores pessoais ou via dispositivos móveis. No caso do acesso via dispositivos móveis, os diferentes conteúdos são adaptados em função das características de hardware e software desses dispositivos.

A seqüência deste artigo está estruturada da seguinte forma: a seção 2 fornece uma visão geral sobre Computação Ubíqua, a seção 3 discorre sobre a área CSCL; a seção 4 trata da abordagem proposta; a seção 5 descreve o estudo de caso desenvolvido; a seção 6 discute trabalhos correlatos comparando-os ao apresentado neste artigo; finalmente a seção 7 tece algumas conclusões relativas a esse trabalho assim como aponta para trabalhos futuros.

2. Computação Ubíqua

A Computação Ubíqua, que tem por lema “em qualquer lugar e a qualquer momento”, representa uma nova era por radicalizar a descentralização do poder computacional, na medida em que o distribui através de uma grande variedade de dispositivos, que possuem funcionalidades e tarefas específicas. Cada um desses dispositivos contribui para a formação de um ambiente computacional globalmente heterogêneo, sendo que estes devem cooperar mutuamente, via padrões de comunicação, linguagens de marcação e plataformas de *software*, visando a interoperabilidade entre os mesmos.

A disseminação desse tipo de computação, impulsionada pela pesquisa de novas abordagens e técnicas de engenharia de *hardware* e *software*, vem ocasionando um impacto significativo em diferentes áreas da sociedade. Entre essas áreas destaca-se a Educação [5], na qual ambientes de Computação Ubíqua permitem que pessoas aprendam em qualquer lugar e a qualquer momento; disponibilizam a informação necessária, no momento exato e da melhor forma possível, uma vez que, a educação ocorre em múltiplos lugares (e.g., escolas, domicílios, trabalho) e em interações com diferentes pessoas (e.g., família, amigos), ao contrário da noção restrita de que esta acontece apenas no ambiente escolar. O aprendizado ubíquo é uma evolução das novas metodologias pedagógicas que envolvem o uso do computador como uma ferramenta que pode auxiliar na construção do conhecimento pelos estudantes, pois permite que o usuário

acesse ferramentas cognitivas, tanto em seu computador, quanto em dispositivos móveis, tornando-se imerso no ambiente e no processo de aprendizagem [6].

Em [7] são descritas cinco vantagens do uso de dispositivos ubíquos na Educação:

- Portabilidade: o dispositivo móvel pode ser utilizado em diferentes lugares;
- Interatividade social: pode-se colaborar e compartilhar informações com outras pessoas (e.g., estudantes, docentes);
- Sensibilidade a contexto: pode-se colher informações sobre a situação em que o estudante se encontra (e.g., localização, meio-ambiente, clima), incluindo-se tanto dados reais quanto simulados;
- Conectividade: pode-se conectar dispositivos móveis para coletar dados de outros dispositivos, ou prover-se uma rede de acesso para compartilhar informações e trabalho; e
- Individualidade: pode-se prover uma plataforma personalizada para a trajetória de cada estudante.

Estas vantagens contribuem, entre outros fatores, para: suprir a necessidade de um conhecimento imediato (e.g., construção de gráfico, fórmula para determinada equação, consulta a um dicionário), sincronizar experiências colaborativas (muitos usuários interagindo entre si), armazenar informações de experiências diárias para uso futuro e acessar informações armazenadas sobre as experiências diárias a qualquer momento e de qualquer lugar.

Um ambiente de computação ubíqua para ABP deve guiar o processo de ensino-aprendizagem, reforçando as idéias da ABP em que: (1) os estudantes são responsáveis pela construção de seu conhecimento, fomentando a busca pela informação e o desenvolvimento do raciocínio crítico; (2) o processo ensino-aprendizagem deixa de ser unilateral e passa a depender da busca pelo conhecimento por parte do estudante, o verdadeiro condutor de seu próprio processo de aprendizagem; (3) os professores passam a auxiliar o estudante a construir seu conhecimento e exercem diversos papéis, tais como facilitador do processo ensino-aprendizagem, consultor, orientador, autor de situações-problemas; (4) uma habilidade fundamental deve ser desenvolvida no estudante: aprender a aprender. Utilizando um ambiente ubíquo o estudante pode compartilhar seu conhecimento com os demais estudantes e professores, os professores podem observar qual é o conhecimento adquirido pelos estudantes no contexto educacional e o ambiente pode direcionar o estudante para o aprender a aprender. Conseqüentemente, descentraliza-se a figura do professor e ressalta-se a figura do estudante. Dessa forma, o ambiente ubíquo está em perfeita sintonia com o método ABP, uma vez que permite ao estudante tornar-se um participante ativo do processo ensino-aprendizagem.

3. *Computer Supported Collaborative Learning*

O *Computer Supported Collaborative Learning* (CSCL) é um ramo do CSCW (*Computer Supported Collaborative Work*) que trata do suporte computacional a processos educacionais em grupo, possibilitando que esses processos não precisem ser realizados presencialmente. Esse suporte consiste em oferecer uma plataforma para que os estudantes possam aprender em grupo, através de comunicação e troca de informações, além de facilitar que os estudantes retornem ao grupo as impressões percebidas durante seu aprendizado. De acordo com [8], CSCW é uma área de pesquisa que trata do projeto de sistemas, baseados em computador, para o suporte à melhoria do trabalho de um grupo de usuários que possuem tarefas ou objetivos em comum. Para isso, um sistema CSCW utiliza uma infra-estrutura de rede e provê uma interface compartilhada para os diferentes papéis de usuários que podem acessar o sistema, cooperando entre si, conforme ilustra a Figura 1.

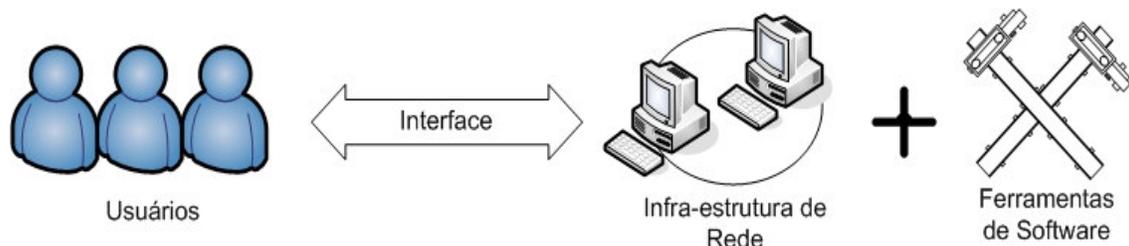


Figura 1. Estrutura de um ambiente *Computer Supported Collaborative Work*.

Um sistema CSCW oferece, por meio desta interface, diversas ferramentas para a interação entre os usuários, como reconhecimento de grupo, interfaces multi-usuário, controle de concorrência, comunicação e coordenação entre o grupo, espaço para compartilhamento de informações e suporte para um ambiente aberto e heterogêneo para integrar aplicações já existentes [9]. Há duas dimensões que agregam as aplicações do domínio de CSCW: tempo e espaço. A dimensão de tempo compreende as aplicações síncronas, que são aquelas em que as informações são trocadas em tempo real; e aplicações assíncronas, que são aquelas que as informações são trocadas em tempos diferentes. No CSCL exemplos de ferramentas síncronas são: videoconferência, comunicadores instantâneos, sistemas de suporte à decisão e chats; e de ferramentas assíncronas: e-mails, *newsgroups*, listas de distribuição de e-mails e fóruns de discussão. Em relação à dimensão espaço, reúnem-se aplicações que se subdividem em aplicações locais, quando os participantes do trabalho cooperativo compartilham o mesmo espaço; e aplicações remotas, quando os integrantes estão distribuídos em localizações diferentes.

Pedagogicamente, CSCL trata das formas pelas quais a tecnologia pode apoiar os processos de ensino-aprendizagem promovidos através de esforços colaborativos de estudantes e docentes trabalhando numa certa tarefa, fundamentando-se em pesquisas das disciplinas de antropologia, sociologia, lingüística, comunicação, entre outras com ênfase no aspecto social. De maneira geral, pode ser definida como uma estratégia educativa em que duas ou mais pessoas constroem conhecimentos através de discussão, reflexão e tomada de decisões, onde os recursos computacionais são os mediadores do processo de ensino-aprendizagem.

Dessa forma, a aprendizagem pode ser convidativa, apresentando atividades que estão relacionadas com o contexto social, histórico e cultural no qual o estudante se insere, sendo particularmente interessante para processos que utilizem a concepção construtivista. Isto justifica a utilização desses ambientes nesse projeto de pesquisa, já que forma a base do Aprendizado Baseado em Problemas. Nesta metodologia é estimulado a sempre comparar ou realizar atividades que tenham relação com os contextos sociais, culturais e históricos em que vive. Porém, nesse caso, o professor, chamado de facilitador, não passa o conhecimento aos estudantes, mas ajuda-os a buscar e construir o conhecimento por si mesmos a partir de situações ou problemas propostos a cada estudante ou a pequenos grupos. Além disso, o aprendizado baseado em problemas avalia os estudantes de forma diferente, sendo que existem dois tipos de avaliação: a formativa e a somativa. A avaliação formativa faz com que o facilitador acompanhe o aprendizado do estudante, visando entender suas dificuldades e fornecer-lhe um auxílio adequado, já a avaliação formativa visa quantificar a aprendizagem de cada estudante durante um período de tempo para verificar se o mesmo adquiriu conhecimentos suficientes para continuar nas próximas etapas de aprendizagem.

Para essa metodologia de ensino-aprendizado, um ambiente CSCL tem grande valor, principalmente devido ao fato de permitir uma quebra do paradigma convencional, no qual esse processo é centralizado na figura do professor. O ambiente CSCL possibilita que todos os membros, inclusive estudantes, colaborem para construir o aprendizado, estando em perfeita sintonia com a prática do aprendizado baseado em problemas, pois nesse tipo de ambiente, a figura central é o estudante.

4. Ambiente Proposto

A arquitetura do Ambiente Proposto está representada na Figura 2. Este ambiente é composto de dois módulos principais: ABP e o Framework para Adaptação de Conteúdo. No primeiro estão disponíveis ferramentas que dão suporte a cursos que utilizam ABP como metodologia de ensino-aprendizagem. O segundo permite que tais ferramentas sejam acessadas através de dispositivos ubíquos, que utilizem diferentes redes de acesso e a partir de locais que não ofereçam suporte computacional. Os dois módulos compartilham os repositórios de dados: Informações de Contexto, que armazena informações que caracterizam o cenário de utilização do usuário (e.g., dispositivo, rede de acesso, localização); e Conteúdo ABP, que armazena as informações geradas durante o processo de ensino-aprendizagem. No ambiente tem-se ainda o Servidor Intermediário responsável pela recuperação dos conteúdos requisitados pela utilização dos diferentes serviços para adaptar esses conteúdos e pelo envio aos usuários (e.g., docentes, estudantes) desses conteúdos adaptados.

O fluxo, representado na Figura 2 pelas setas numeradas, tem início (1) quando um usuário acessa o Ambiente Proposto através de um certo dispositivo (móvel ou não). Depois da requisição desse usuário, o Servidor Intermediário identifica-o e recupera suas informações de contexto. Essas informações caracterizarão as preferências de adaptação do usuário, o tipo de dispositivo e a rede de acesso, permitindo que a adaptação de conteúdo seja realizada sem a necessidade de interação com o usuário. Após identificar

o usuário, o servidor requisita (3) e recebe (4) o conteúdo a ser adaptado (e.g. Páginas Web, Dados Multimídia, e Outros Arquivos). De posse do conteúdo a ser adaptado, esse servidor verifica quais são as adaptações necessárias para satisfazer as necessidades do usuário de acordo com as informações de contexto recuperadas. As aplicações que realizarão tais adaptações estão disponíveis em servidores distribuídos, cujos serviços podem ser compostos a fim de aumentar a capacidade de adaptação do Ambiente Proposto. Após encontrar os servidores que podem realizar essa adaptação, o Servidor de Intermediário requisita as adaptações (5) e recebe o conteúdo adaptado. Finalmente, retorna este conteúdo adaptado (6) ao usuário que o requisitou.

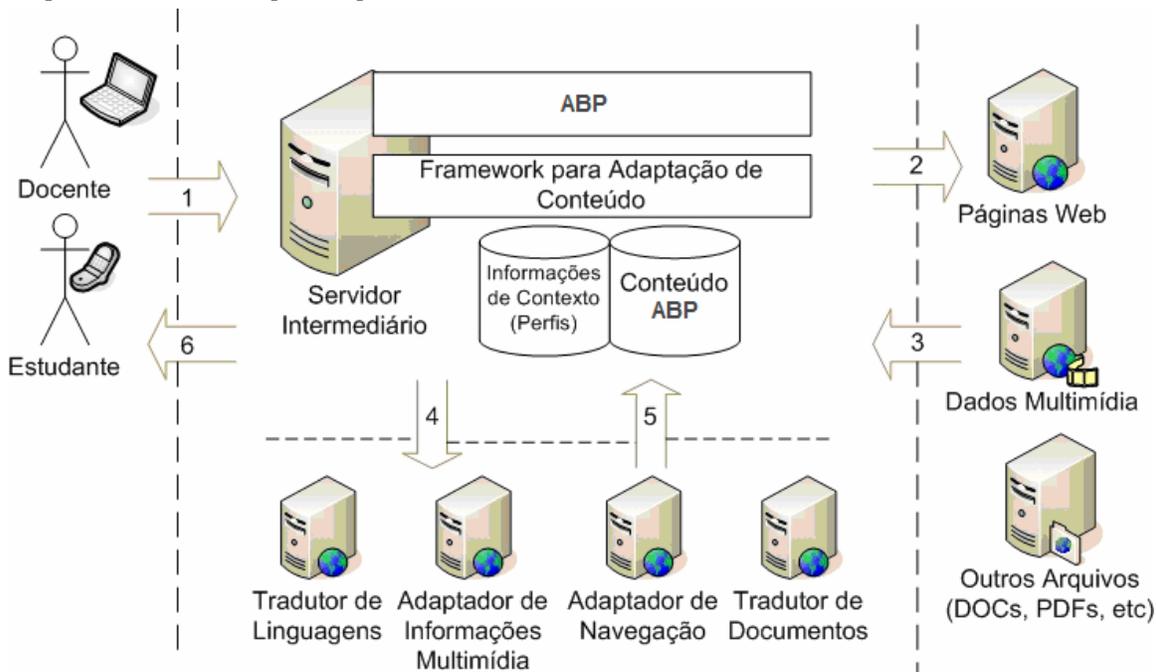


Figura 2. Arquitetura do Ambiente Proposto.

Nessa arquitetura, cada serviço de adaptação fica sob a responsabilidade de um Serviço Web Semântico, a fim de possibilitar uma interoperação dinâmica e flexível entre servidores autônomos e compor de forma automática tais serviços com o intuito de realizar tarefas complexas necessárias para satisfazer uma certa requisição [10,11]. Para o Ambiente proposto foram definidos quatro Serviços Web Semânticos: Tradutor de Linguagens, para a conversão de linguagens de marcação; Adaptador de Informações Multimídia, para o tratamento de imagens; Adaptador de Navegação, para o tratamento da apresentação por mediação de prioridades; e Tradutor de Documentos, para possibilitar o acesso a documentos em formatos não suportados pelos dispositivos móveis. Podem ser adicionados outros serviços de adaptação de conteúdo conforme o domínio de aprendizagem que será apoiado pelo ambiente proposto, por exemplo, um serviço de tradução de idiomas necessite de um grande número de textos estrangeiros.

4.1. Suporte Computacional para a Aprendizagem Baseado em Problemas

A utilização dos computadores em ambientes de aprendizagem colaborativos pode acontecer de diferentes formas, como: colaboração usando um mesmo computador; colaboração baseada numa rede local, onde os estudantes trabalham em diferentes computadores, mas em um mesmo lugar; e colaboração baseada numa rede não local, onde os estudantes trabalham em computadores geograficamente distantes. Neste projeto de pesquisa, inicialmente, foi desenvolvido um ambiente CSCL para atender a última forma de colaboração, oferecendo suporte à aprendizagem através da Web. Posteriormente, esse ambiente foi estendido para suportar o acesso através de dispositivos móveis. Atualmente, o ambiente atende diferentes funcionalidades, implementadas nos componentes da Figura 3a.

As principais funcionalidades do módulo ABP, representadas pela figura 3b, são:

- Gerenciamento de Usuários (*Users Manager*): os usuários do ambiente são classificados de acordo com seus papéis no processo de ensino aprendizagem, sendo que novos tipos de usuário podem ser adicionados de acordo com o curso que será apoiado pelo ambiente. Entretanto, os papéis estudante,

docente e administrador são padrões, e estão disponíveis para todos os cursos que utilizem o ambiente. O papel estudante pode ser especializado, por exemplo, de acordo com o período que um certo estudante está cursando, permitindo o acesso a novas funcionalidades do ambiente. O papel docente pode ser especializado nos diferentes papéis que um docente pode realizar, como: facilitador de pequenos grupos, coordenador de curso e orientador. O papel administrador diz respeito ao gerenciamento do ambiente, por exemplo, um administrador pode cadastrar usuários, moderar fóruns, e enviar avisos.

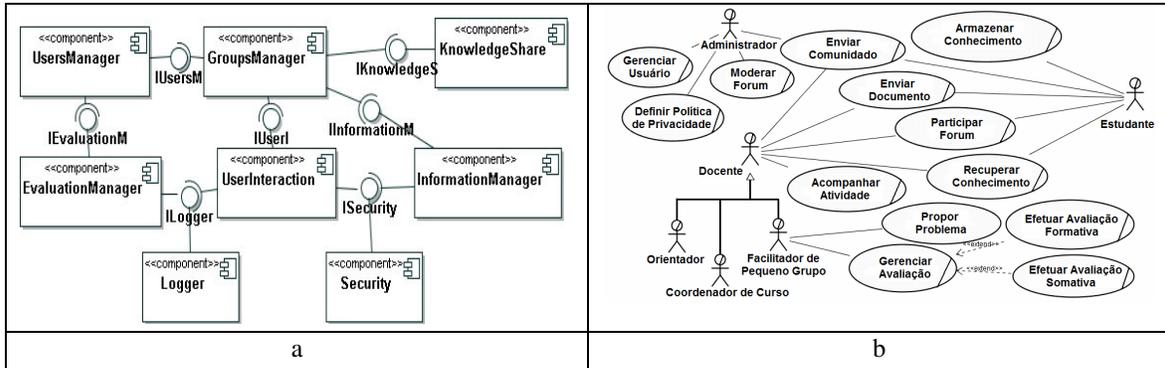


Figura 3. Modelo de Componentes e de Caso de Negócio do ABP.

- Gerenciamento de Pequenos Grupos (*Groups Manager*): os pequenos grupos são a principal forma de interação entre os usuários. Um pequeno grupo é responsável por criar novos conhecimentos a partir de um problema proposto por seu facilitador. No ambiente, os pequenos grupos podem ser reunidos presencialmente, inserindo apenas o conhecimento resultante do processo de ensino e as informações do pequeno grupo (e.g., local de reunião, horário, participantes). Ou podem se reunir através do ambiente, para interação síncrona entre os envolvidos nesse pequeno grupo.
- Gerenciamento de Avaliações (*Evaluation Manager*): o ambiente desenvolvido permite que grupos, pessoas e documentos sejam avaliados de maneira formativa, ao longo do processo de aprendizagem, e de maneira somativa, ao final do processo de aprendizagem.
- Busca e Acesso a informações (*Information Manager*): o conhecimento coletado e formado pelos estudantes é armazenado em documentos de texto livre que compreendem relatórios, artigos técnicos ou científicos, anotações, dentre outras variantes. O ambiente permite que os usuários recuperem esses itens de conhecimento considerando tanto o tema (expresso em palavras-chave ou sentenças em linguagem natural), quanto o contexto (que atividade está realizando, de que grupo está participando, etc.), das suas necessidades informacionais.
- Compartilhamento de Conhecimento (*Knowledge Share*): o conhecimento é compartilhado através da utilização de um repositório de documentos, sendo que um documento pode ser qualquer tipo de informação gerenciada pelo sistema. Por exemplo, um problema que dará início a um processo de ensino-aprendizagem pode ser compartilhado entre vários usuários e pequenos grupos.
- Interação entre Pessoas (*User Interaction*): a interação entre pessoas no ambiente ocorre tanto entre uma pessoa e um grupo, quanto entre pares. No primeiro tipo de interação, um docente ou estudante pode enviar avisos, lembretes, documentos ou dúvidas para seu pequeno grupo através de um fórum de discussão. Além disso, os administradores do sistema podem enviar avisos para conjuntos de usuários. O segundo tipo de interação pode ser entre docentes e estudantes; docentes e docentes; ou estudantes e docentes. Essa interação é feita através de mensagens assíncronas que são armazenadas na caixa de mensagem de cada um dos usuários.
- Segurança de Acesso e das Informações (*Security*): a segurança de acesso é feita pela autenticação de usuários através de senhas e verificação do dispositivo de acesso. A segurança das informações é realizada via protocolo *HyperText Transfer Protocol Secure* (HTTPS), que é uma implementação do protocolo HTTP sobre uma camada para transmissão de dados criptografados, que verifica a autenticidade do servidor e do cliente através de certificados digitais.

- Acompanhamento das Atividades (*Logger*): para acompanhamento das atividades foi utilizada a API log4j¹, que permite que todas as atividades do usuário sejam registradas em arquivos de log. Esses arquivos são consolidados em informações disponibilizadas apenas para os coordenadores do curso ou para docentes responsáveis por um pequeno grupo, para que seja possível obter informações como a utilidade do ambiente para o curso e as formas que o ambiente está sendo utilizado. Essa funcionalidade pode ser desabilitada de acordo com a política de privacidade acordada entre os usuários e os administradores do ambiente.

Para que estas funcionalidades sejam plenamente utilizadas pelos diferentes dispositivos ubíquos se faz necessário adaptar conteúdos considerando as restrições e características desses dispositivos, das redes de acesso, e dos usuários. Essa necessidade requer uma plataforma computacional conforme descrito a seguir.

4.2. Suporte Computacional para a Adaptação de Conteúdo

Para facilitar o desenvolvimento desse ambiente, foi utilizado o *Framework para Adaptação de Conteúdos em Celulares* estendido (FACCe), uma versão do *Framework para Adaptação de Conteúdo da Internet* (FACI) [12,13] especializada na adaptação de conteúdo para dispositivos que utilizem o *Micro Information Device Profile (MIDP)* do *Java MicroEdition (Java ME)*. O FACCe é composto pelos pacotes *AdaptationProxy* e *AdaptationServer* e foi estendido para a utilização de Serviços Web Semânticos, conforme ilustra a Figura 4.

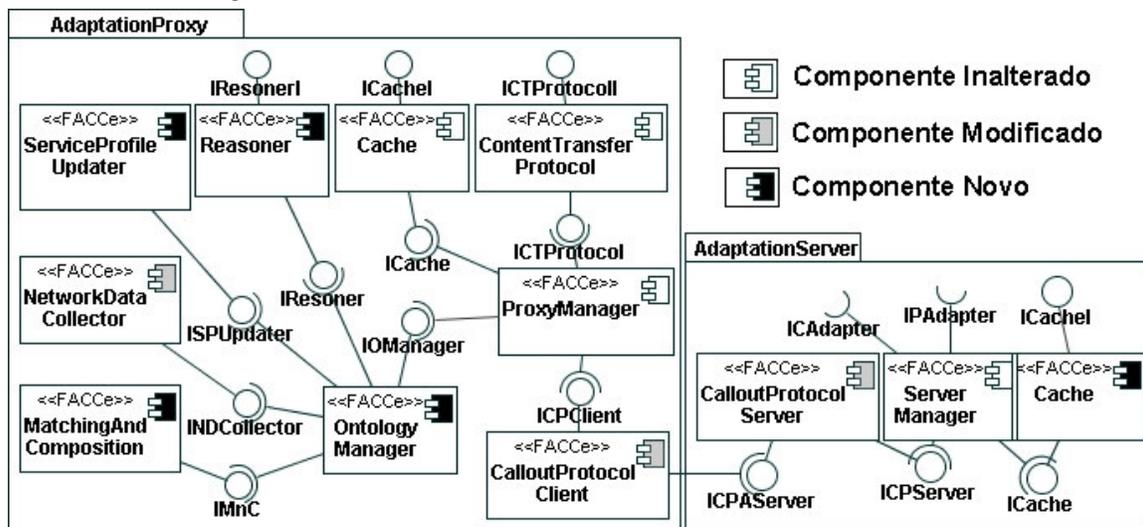


Figura 4. Plataforma de Software do FACCe.

O componente *Content Transfer Protocol* é responsável pela comunicação com o servidor de origem e o dispositivo do usuário, podendo lidar com diferentes protocolos (e.g., HTTP, WAP). O componente *ProxyManager* é responsável pela gerência de cada requisição e pela consulta ao componente *Cache* em busca de adaptações previamente realizadas. A comunicação com o pacote *Adaptation Server* é de responsabilidade do componente *Callout Protocol Client* que, nesse caso, utiliza o protocolo *Simple Object Access Protocol (SOAP)*.

O componente *Ontologies Manager* trata da política de adaptação de conteúdo, cujo papel é definir quais serviços de adaptação serão oferecidos, quais adaptadores executarão esses serviços e quando estes deverão ser solicitados. Para que a política de adaptação possa ser eficaz, é necessário que o contexto de entrega contenha diversos tipo de informações: as características e as capacidades do dispositivo de acesso; os dados pessoais e as preferências do usuário; as condições da rede de comunicação; as características do conteúdo requisitado; e as resoluções contratuais entre o provedor de serviços e o usuário. Neste trabalho essas informações são disponibilizadas via os perfis de dispositivo, usuário, rede, conteúdo e contrato de serviços, respectivamente, e esses perfis são especificados em *Web Ontology Language (OWL)*². Nesse

¹ <http://logging.apache.org/log4j/docs/>

² <http://www.w3.org/2004/OWL/>

sentido, o componente *Network Data Collector* atua como agente de atualização do perfil de rede e o componente *ServiceProfileUpdater* trata do perfil de *Service Level Agreement (SLA)*.

As informações da política de adaptação são recuperadas pelo componente *Matching and Composition* através de consultas em *SPARQL*³. Tais informações são usadas pelo componente *Reasoner* para decidir a necessidade ou não de uma determinada adaptação. Em caso positivo, as pré-condições e os efeitos dos serviços de adaptação, descritos nos perfis de serviço, são verificados para indicar se é preciso ou não mais de um serviço para realizar essa adaptação. Em caso afirmativo, esses serviços são determinados e uma ordem é estabelecida para as invocações dos mesmos.

A composição dos serviços pode ser estática ou dinâmica. No primeiro caso deve-se especificar previamente, por meio de um fluxo de trabalho, como os serviços interagem entre si para oferecer o conjunto de funcionalidades de uma determinada adaptação. Já na composição dinâmica os serviços são selecionados automaticamente de acordo com a política de adaptação.

Para determinar os serviços necessários são utilizadas regras de adaptação, descritas em *Semantic Web Rule Language (SWRL)*, nas ontologias do perfil de serviço. Tal abordagem permite que novas regras sejam inseridas, pelo mecanismo de inferência, na base de conhecimento, minimizando assim o impacto de modificações sobre os componentes responsáveis pela política de adaptação.

Para realizar o tratamento dos aspectos de apresentação da página Web e suportar a utilização de um segundo nível de cache têm-se o componente *Cache* e as interfaces *ICAdapter* e *IPAdapter* do pacote *Adaptation Server*. Após realizar essas tarefas o *Cache* armazena os conteúdos em uma representação intermediária. Posteriormente, essa representação é adaptada através de serviços especificados na interface *ICAdapter* e *IPAdapter*, e cujas implementações são providas por componentes específicos das aplicações. Ainda, nesse pacote têm-se os componentes: *Callout Protocol Server* responsável pela comunicação com o *Adaptation Proxy*, e *ServerManager* responsável pela gerência de adaptações.

5. Estudo de Caso

Como estudo de caso foi desenvolvido um protótipo de um sistema que apóia o processo de ensino-aprendizagem do curso de Medicina da Universidade Federal de São Carlos (UFSCar). O diferencial desse curso é a utilização de uma abordagem ensino-aprendizagem que busca substituir os processos de memorização e de transferência unidirecional e fragmentada de informações – do professor para o aluno – pela construção e significação de saberes a partir do confronto com situações reais ou simuladas, que estimulem o desenvolvimento de capacidades crítico-reflexivas e de aprender a aprender, integrando teoria e prática. A avaliação do processo de ensino-aprendizagem é permanente e a avaliação formativa assume um papel determinante na melhoria desse processo. O processo de avaliação somativa visa abranger em amplitude e profundidade a análise das capacidades construídas, preferencialmente em ação. Essas avaliações e o progresso de cada estudante são sintetizados num portfólio que representa e qualifica sua trajetória [14].

O portfólio é uma coleção de trabalhos de um estudante, que mostra sua evolução em um determinado período de tempo. Entretanto, a utilização dos portfólios tradicionais possui desvantagens em relação a manutenção (e.g., perda ou esquecimento de itens do portfólio necessários numa certa atividade), acesso (e.g., falta de suporte para recuperação de informações específicas) e transporte (e.g., um portfólio pode atingir um grande volume, dificuldade de utilização dos portfólios em aulas práticas) [15].

Além disso, o curso de Medicina da UFSCar possui características que justificam a utilização de um ambiente de ensino ubíquo, como: alta mobilidade, o que obriga os estudantes a utilizar seu portfólio em diferentes locais (e.g., nas unidades de saúde e nos hospitais do município); grande quantidade de pessoas envolvidas, desempenhando diferentes papéis (e.g., estudantes, docentes, profissionais de saúde da rede municipal, pacientes); e a necessidade de uso de diferentes dispositivos de acesso (e.g., computadores pessoais para atividades de aprendizagem e dispositivos móveis durante a atuação em hospitais).

Nesse estudo de caso o ambiente proposto foi utilizado, conforme ilustram os pacotes de componentes da Figura 5: Tradutor de Documentos, Informações Multimídia, Tradutor de Linguagens e Adaptador de Navegação.

³ <http://www.w3.org/TR/rdf-sparql-query/>

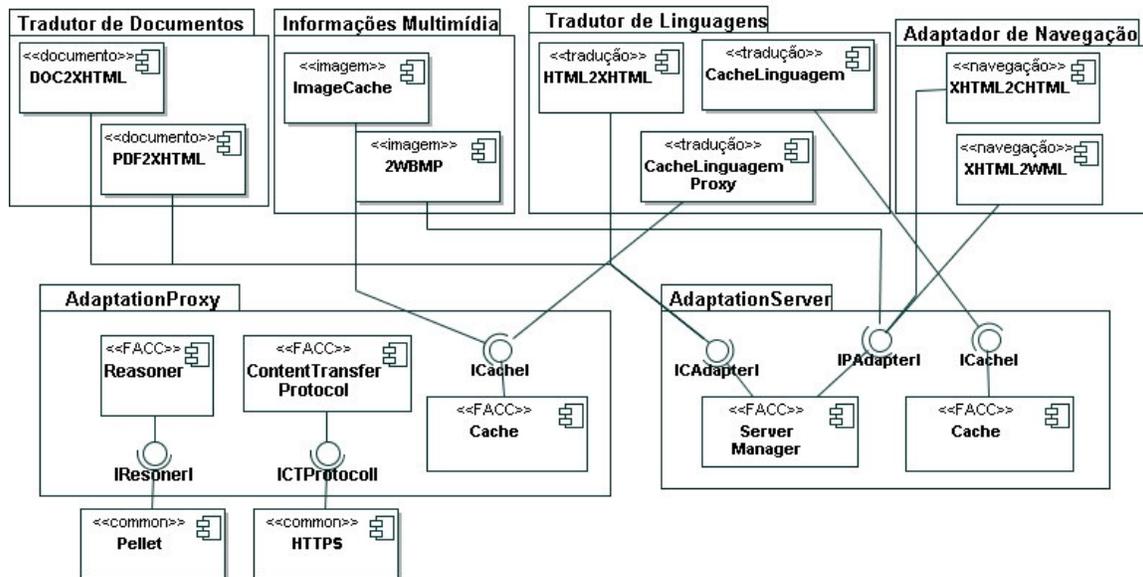


Figura 5. Reuso do FACCe.

O serviço *Tradutor de Documentos* permite que o usuário visualize documentos em formato Word (DOC) ou *Portable Document Format* (PDF) através de sua tradução para a linguagem de marcação do seu dispositivo, passando pelas mesmas adaptações que uma página Web comum.

O *Adaptador de Informações Multimídia* usa a interface *ICAdapter* para realizar transformações de escala, de cores ou o eventual descarte de imagens. Já a interface *IPAdapter* é usada para a transcodificação entre formatos (e.g., JPEG, PNG, GIF, WBMP) por meio da API *Java Advanced Imaging (JAI)*. Após ser adaptada, cada imagem é enviada ao *Adaptation Proxy* para que seja encaminhada ao dispositivo do usuário.

Para o *Tradutor de Linguagens de Marcação* adotou-se XHTML, como representação intermediária das páginas a serem traduzidas. Se as páginas do servidor de origem já estiverem nesse formato, estas serão processadas diretamente pelo componente *ContentAdapter*, a fim de que este possa extrair as informações a serem armazenadas no perfil de conteúdo. Se as páginas estiverem em HTML, um passo adicional é necessário para a conversão das mesmas para XHTML. Em ambos os casos as páginas XHTML são armazenadas no componente *Cache*. Finalmente, o componente *PresentationAdapter* realizará a tradução das páginas para as linguagens determinadas pela política de adaptação.

Os aspectos de apresentação são tratados pelo *Adaptador de Navegação*, cujo processo é orientado pelas prioridades atribuídas aos blocos lógicos da página (e.g., tabelas, parágrafos, camadas). Para tal, as páginas que são mantidas no componente *Cache* recebem marcações especiais, a fim de que os blocos possam ser identificados. No perfil de conteúdo é associada uma prioridade ao identificador de cada bloco lógico. A métrica adotada, para a definição de prioridades, é a importância do bloco, a qual é calculada como $(qt_acesso + fator_importancia)/qt_links$, onde: *qt_acesso* é o número total de acessos que um bloco lógico obteve; *fator_importancia* é determinado em função da quantidade de informação (em caracteres) do bloco, a sua posição na página e se este contém imagens; e *qt_links* representa a quantidade de links encerrados num bloco. A prioridade de um bloco aumenta à medida que seus *links* são mais acessados ao longo do tempo. Além desses, existem dois componentes que não estão em nenhum pacote: Pellet e HTTPS. O primeiro define o mecanismo de inferência para o sistema criado e o segundo o protocolo de comunicação com computadores de fora do sistema.

Para que se tenha uma idéia da utilização do sistema construído nesse estudo de caso, a Figura 6 contém as informações de uma pesquisa sobre pediatria realizada por um estudante. Essas mesmas informações são acessadas através de *emuladores* de celulares que implementam MIDP, conforme mostra a Figura 7.

Para a tela da Figura 6 foram necessárias adaptações de imagens, da linguagem de marcação (HTML para WML) e de navegação para que apenas as informações mais relevantes sejam apresentadas aos usuários. Assim, na Figura 7, têm-se: a) a página com adaptação de linguagem, mostrando as dificuldades de se apresentar imagens num dispositivo limitado; b) a página com adaptação de linguagens e imagens, mostrando a grande quantidade de informação que é enviada ao usuário; e c) a página completamente

adaptada, apresentando apenas as informações mais relevantes para o usuário. Essas telas mostram a importância da adaptação da apresentação, uma vez que os usuários de dispositivos móveis muitas vezes necessitam que apenas as informações mais relevantes sejam apresentadas, entre outros motivos, pelas limitações de dispositivo, pelo preço da utilização da rede de telefonia celular, por utilizarem esse serviço em movimento.



Figura 6. Ambiente para apoio ao Ensino de Medicina acessado através de um navegador.



Figura 7. Ambiente para Apoio ao Ensino acessado através de um Celular.

Um dos desafios em relação a adaptação de conteúdo é o da melhoria do desempenho na utilização de Serviços Web Semânticos. Assim, a Figura 8 apresenta o gráfico dos tempos de respostas para as requisições de adaptação de conteúdo utilizando Serviços Web Tradicionais ou usando Serviços Web Semânticos. Conforme ilustra o gráfico os tempos de respostas são maiores quando se utiliza Serviços Web Semânticos do que quando não os utilizam. Contudo, sua utilização se justifica pelo aumento da capacidade de adaptação e flexibilidade introduzida por esses serviços. Mais detalhes sobre os testes e a avaliação quantitativa desse servidor pode ser obtido em [16].

O uso do ambiente ubíquo deve fornecer benefícios aos usuários, pois estes não têm como atividade fim a utilização da aplicação, e sim, através desta atingir uma determinada finalidade. Pode-se citar como benefícios oferecidos pelo ambiente ubíquo: informação sempre à disposição e de forma mais atualizada possível; dispositivo para anotação de idéias, registro de conhecimento e registro de conteúdos didáticos; acompanhamento, pelos tutores, da construção do saber dos estudantes, observando seus anseios, angústias,

desejos e experiências de sucesso; compartilhamento do conhecimento; e meio tecnológico para desenvolvimento de métodos ABP.

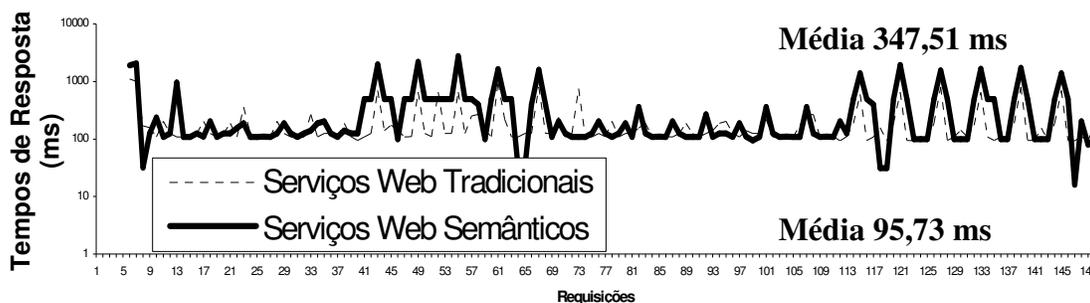


Figura 8. Tempos de resposta às requisições [16].

Um dos aspectos mais importantes no projeto de software para dispositivos móveis é a usabilidade. O W3C (*World Wide Web Consortium*) estabelece os principais fatores que devem ser considerados no projeto de interfaces em dispositivos móveis e alguns fatores considerados no *framework* utilizado por este estudo de caso são: situar o conteúdo mais importante no alto da página; localizar a barra de navegação no alto, com menos opções; criar URL's curtas; explicar os textos nos títulos, com o mínimo de palavras; evitar imagens grandes, não ultrapassando a altura do monitor; as páginas devem conter somente as informações mais importantes; esconder recursos avançados de usuários principiantes; oferecer recursos de uso intuitivo, fáceis de achar, com a menor quantidade de etapas; nos textos, prover uma introdução que permita ao usuário selecionar um link se desejar ler maiores detalhes; dividir os textos em diversos títulos e subtítulos em destaque; usar fonte de letras que facilitem a leitura. Assim, tais aspectos de usabilidade resumem-se a: autonomia orientada, antecipação da experiência, consistência da funcionalidade e contextos culturais.

6. Trabalhos Correlatos

Algumas experiências já foram feitas em relação à utilização de computação ubíqua no auxílio a atividades de ensino, dentre essas se destacam as apresentadas a seguir.

O trabalho de [17] apresenta um suporte personalizado para aprendizagem em redes de informação e aprendizagem dinâmicas. Neste trabalho é apresentado como personalizar o aprendizado através de tecnologias baseadas na Web Semântica. Para tal é proposta uma arquitetura para o *e-learning* onde diferentes funcionalidades de personalização são oferecidas através de Serviços Web. Em relação a esse trabalho, o proposto estende as funcionalidades de personalização para possibilitar que o acesso ao ambiente seja feito através de dispositivos móveis. Além disso, utiliza Serviços Web Semânticos para permitir que os serviços de adaptação de conteúdo sejam encontrados e compostos automaticamente, aumentando o poder de personalização do ambiente.

O trabalho de [9] apresenta evidências de que a Computação Ubíqua pode integrar experiências de aprendizagem em ambientes internos e externos permitindo que os conhecimentos sejam conectados tanto no mundo real quanto na sala de aula. Entretanto, esse trabalho tem foco na educação infantil, enquanto o ambiente proposto foi criado para o ensino universitário.

O trabalho de [18] apresenta um modelo para suporte ubíquo à aprendizagem, utilizando informações de localização e do gerenciamento de contextos como instrumentos de auxílio do processo de ensino-aprendizagem. Este trabalho se assemelha às aplicações que serão criadas para o ensino de Medicina, porém esse sistema se concentra na utilização de informações de contexto, sem lidar com outros campos da Computação Ubíqua, como mobilidade e adaptação de conteúdo, tratadas pelo trabalho proposto.

7. Conclusões e Trabalhos Futuros

Este artigo apresentou um ambiente ubíquo para auxiliar o processo de ensino-aprendizagem em cursos que utilizem ABP como metodologia de ensino. Este ambiente consiste de uma aplicação Web que pode ser acessada via navegadores comuns, que é adaptado para que seja acessado através de dispositivos móveis limitados (e.g., celulares, *Smart Phones*). Para tal foi empregado o FACCe, que utiliza Serviços Web Semânticos por oferecerem uma interoperação dinâmica entre os diferentes computadores envolvidos na adaptação de conteúdo, tornando-a flexível e distribuída.

Como estudo de caso, foi apresentado o protótipo de um sistema desenvolvido para auxiliar as atividades de ensino-aprendizagem do curso de Medicina da UFSCar. Este curso possui características, como alta mobilidade, que justificam a utilização do ambiente proposto. Conforme apresentado, o ambiente ubíquo proposto facilitou o desenvolvimento do sistema, possibilitando que os diferentes conteúdos fossem adaptados e acessados por vários dispositivos ubíquos, considerando as particularidades dos usuários e das redes de acesso.

Como trabalhos futuros pode-se citar a adição de novos serviços de adaptação de conteúdo como tradução de idiomas, transcodificação de vídeo e áudio, entre outros. Além disso, serão realizados testes qualitativos de aceitação junto aos estudantes, docentes e outros profissionais envolvidos com o curso de medicina da UFSCar.

Referências

- [1] Weiser, M. "Some Computer Science Problems in Ubiquitous Computing". Communications of the ACM, pp. 137-143, 1993.
- [2] Weiser, M., Gold, R. e Brown, J.S. "The origins of ubiquitous computing research at PARC in the late 1980s" IBM Systems Journal, vol. 38, no. 4, pp. 693-696, 1999.
- [3] Boud, D. e Feletti, G. The challenge of problem-based learning. Londres: Kogan Page, 1999.
- [4] Dirckinck-Holmfeld, L. e B. Fibiger, Eds. "Learning in virtual environments". Fredriksberg, Samfundslitteratur, 2002.
- [5] Rogers, Y. et al. "Ubi-learning Integrates indoor and outdoor experiences". Communications of the ACM, vol. 48, no. 1, pp. 55-59. 2005.
- [6] Jones, V. e Jo, J.H. "Ubiquitous learning environment: An adaptive teaching system using ubiquitous technology". Anais do 21st Australasian Society for Computers in Learning in Tertiary Education Conference, pp. 468-474, 2004.
- [7] Klopfer, E., Squire, K. e Jenkins, H. "Environmental Detectives PDAs as a Window into a Virtual Simulated World". Anais do International Workshop on Wireless and Mobile Technologies in Education, 2002.
- [8] Fluckiger, F. "Understanding networked multimedia : applications and technology". Prentice Hall, 1995.
- [9] Törlind, P., Stenius, M., Johanson, M. e Jeppsson, P. "Collaboration Environments for Distributed Engineering". Anais do Computer Supported Cooperative Work in Design, 1999.
- [10] Web Services Activity. Disponível em <http://www.w3.org/2002/ws/>, acessado em abril de 2007.
- [11] Martin, D. et al. "OWL-S: Semantic Markup for Web Services 1.0". OWL-S Coalition, 2003. Disponível em: <http://www.daml.org/services/owl-s/1.0/>
- [12] Claudino, R. A. T., Souza, W.L. and Prado, A.F. Um framework baseado em componentes para o domínio de adaptação de conteúdo na Internet. Anais do XIX Simpósio Brasileiro de Engenharia de Software, pp. 88-103. 2005.
- [13] Forte, M., Souza, W.L. e Prado, A.F. (2006b) "Utilizando ontologias e serviços Web na computação ubíqua". Anais do XX Simpósio Brasileiro de Engenharia de Software, pp. 287-302.
- [14] Site institucional da Universidade Federal de São Carlos, Curso de Medicina. <<http://www2.ufscar.br/graduacao/medicina.php>>
- [15] Niguidula, D., Ring, G. e Davis, H. "Digital Portfolios: A Dozen Lessons in a Dozen Years", 2005.<www.richerpicture.com/dozenLessons.pdf>
- [16] Santana, L.H.Z. et al. "Serviço de tradução de linguagens de marcação para a Internet". Anais do XXV Simpósio Brasileiro de Redes de Computadores, 2007.
- [17] Dolog, P., Henze, N., Nejdl, W. e Sintek, M. "Personalization in distributed e-learning environments", Anais do 13th international World Wide Web, pp. 19-21, 2004.
- [18] Barbosa, J., Hahn, R., Barbosa, D.N.F., e Geyer, C.F.R., "Mobile and ubiquitous computing in an innovative undergraduate course", Anais do Technical Symposium on Computer Science Education, pp. 379 – 383, 2007.