

LIA - UM SISTEMA ADAPTATIVO APLICADO AO ENSINO A DISTÂNCIA

Aline Donini Valsirolli¹, Claudia Midori Nakahata², Cristiane Gomes Bastos³, Margaret Yea Wei Ng⁴

Abstract—This project shows an adaptive system that supports learning features by using Intelligent Tutorial System techniques. The knowledge base is modeled using Conceptual Maps, which leads to a more efficient apprentice modeling. The main goal is to optimize the learning process in a way that it would be possible to divide the knowledge in interdependent levels. Thus, intelligent tests can be used to determine in which level of knowledge apprentices' skills must be classified, as well as to evaluate if the student has the needed skills to start learning at some desired level, or if it is advisable to review some previous topics, in order to get the expected background for learning at such level. The work proposes a reusable, adaptive framework for supporting pedagogical contents' modeling. In this way, it is expected to increase students' motivation, since they can start accessing such contents in a more profitable, individualized form, according to their own learning needs.

Index Terms—Adaptive Systems, Apprentice Modeling, Intelligent Tutorial Systems, Learning by Distance.

1. INTRODUÇÃO

Com as crescentes mudanças na sociedade, muitas formas de treinamento e ensino generalizado não se justificam mais. Perde-se muito tempo ensinando, desperdiça-se muito dinheiro com cursos e acaba-se aprendendo pouco, muito menos do que o esperado.

Isto vem acontecendo porque encontram-se pessoas em vários graus de maturidade, motivação e de competência. Assim torna-se cada vez mais difícil colocá-las em uma única sala de aula para aprenderem as mesmas coisas.

Para uniformizar os níveis de conhecimento dessas pessoas, é necessário utilizar abordagens diferenciadas que atendam às necessidades específicas de cada aluno. Desta forma, este artigo vem apresentar um sistema adaptativo de apoio ao aprendizado caracterizado pela utilização de técnicas de ITS – (*Intelligent Tutoring System* - Sistemas Tutores Inteligentes).

Na segunda seção será feita uma explanação do problema a ser abordado. A seção 3 mostrará como é obtida a adaptabilidade do sistema através dos Sistemas Tutores

Inteligentes. A seção 4 apresenta um exemplo de modelagem para a base do conhecimento do sistema. A seção 5 trata da arquitetura do sistema. A seção 6 mostra como o sistema funciona. A seção 7 tem como objetivo apresentar as tecnologias utilizadas no protótipo do sistema. A seção 8 apresenta as considerações finais.

2. CONSIDERAÇÕES INICIAIS

Atualmente encontramos sistemas de ensino generalizados, isto é, que procuram atender a todos sem se interessar com as necessidades de cada aprendiz individualmente.

Estes sistemas de ensino, geralmente apoiados em uma estratégia pedagógica expositivista, exibem o mesmo conteúdo para todos os aprendizes e não se preocupam em analisar o aproveitamento deste, tampouco analisar se o mesmo possui certas competências prévias necessárias ao aprendizado do conteúdo em questão. Não existe um mecanismo nestes sistemas de ensino que guie - o aprendiz - ao ponto do conteúdo mais adequado, a fim de ajudá-lo a tirar maior proveito.

Conforme já mencionado acima, em geral, esses sistemas educacionais sugerem uma abordagem linear e expositiva. Isto pode fazer com que o aprendiz perca a motivação para continuar a estudar, uma vez que todo o processo de ensino-aprendizagem torna-se rotineiro e massificado.

3. ADAPTABILIDADE DO SISTEMA

A adaptabilidade do sistema está baseada na utilização de técnicas dos Sistemas Tutores Inteligentes (ITS), que são sistemas computacionais com modelos de conteúdo instrucionais que especificam o que ensinar e estratégias de ensino que especificam como ensinar. Esses sistemas educacionais são modelados de maneira a levar em consideração o tipo de aluno que está interagindo com o sistema.

Estes sistemas estão divididos em 4 módulos: módulo *tutor*, módulo *aprendiz*, módulo *domínio* e módulo *comunicação* ou *interface*[1].

¹ Aline Donini Valsirolli, Universidade Presbiteriana Mackenzie, São Paulo, SP, Brasil, lia@mackenzie.com.br

² Claudia Midori Nakahata, Universidade Presbiteriana Mackenzie, São Paulo, SP, Brasil, lia@mackenzie.com.br

³ Cristiane Gomes Bastos, Universidade Presbiteriana Mackenzie, São Paulo, SP, Brasil, lia@mackenzie.com.br

⁴ Margaret Yea Wei Ng, Universidade Presbiteriana Mackenzie, São Paulo, SP, Brasil, lia@mackenzie.com.br

O *módulo tutor* faz o reconhecimento do usuário, no caso o aprendiz, e suas necessidades para que se possa propor conteúdo, percursos ou mesmo uma estratégia de aprendizado adequada a cada tipo de aprendiz.

O *módulo aprendiz* guarda as informações de cada aprendiz. Mantém um histórico de todas as ações do aprendiz: seus erros, seus acertos, o que já estudou e o que falta aprender. Representa o conhecimento e as competências do aprendiz. Já o *módulo domínio* armazena todo o conteúdo que precisa ser ensinado ao aprendiz, ao passo que o *módulo de comunicação (interface)* é a interação entre o módulo tutor e o módulo aprendiz, mostrando o conteúdo adequado ao conhecimento do aprendiz.

A interação entre estes módulos pode ser vista na Figura 1[1].

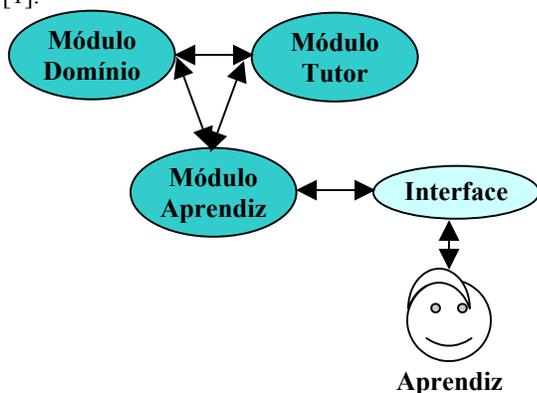


FIGURA 1
ARQUITETURA CLÁSSICA DE STI

A Figura 1 mostra como os módulos são relacionados em um Sistema Tutor Inteligente, mas para obtermos a adaptabilidade desejada, é necessário um relacionamento diferente. Este relacionamento pode ser observado na Figura 2.

A Figura 2 mostra que a adaptabilidade do sistema de ensino é obtida através da ligação direta do módulo interface com o módulo tutor. Assim, o módulo tutor consulta o modelo estático do aluno no módulo aprendiz e acessa o conteúdo disponível no módulo domínio. Com essas duas informações ele provê avaliações e fornece o conteúdo adequado ao aprendiz.

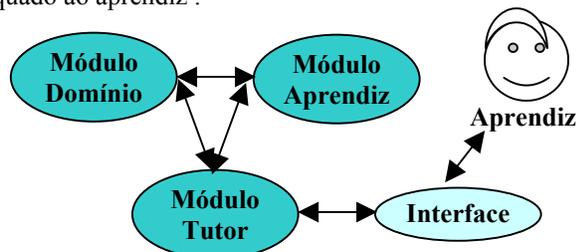


FIGURA 2
ARQUITETURA DOS SISTEMAS ADAPTATIVOS

O módulo tutor obtém algumas dessas necessidades do aprendiz através de avaliações eletrônicas. Vários métodos de avaliação podem ser utilizados como: questionários, entrevistas com alunos, comparação com resultados de cursos tradicionais e observação direta do instrutor online. Com vários tipos de questões, pode-se tentar aproximar o ambiente cibernético com a situação real do aluno. Alguns tipos de questões são: corrigir uma frase invertida, completar frases, verdadeiro ou falso. As questões podem ter vários graus de dificuldades como: fácil, médio, difícil [8].

4. MODELAGEM DO CONHECIMENTO

A base de conhecimento do sistema é modelada usando Mapas Conceituais. Isto permite uma modelagem do aprendiz eficaz o suficiente para otimizar o processo de aprendizado, de forma que torna possível dividir o conhecimento em níveis.

Esses Mapas Conceituais são representações gráficas que auxiliam na ordenação e na sequenciação hierarquizada dos conteúdos do ensino, de forma a oferecer estímulos ao aprendiz [6].

Os Mapas Conceituais servem como instrumentos para facilitar o aprendizado do conteúdo sistematizado em conteúdo significativo para o aprendiz.

Esse tipo de modelagem do conhecimento torna a aprendizagem mais significativa, pois ajuda o aprendiz a fazer ligações entre o novo conhecimento e o conhecimento que já possui.

Os Mapas Conceituais podem ser descritos sob 3 perspectivas diferentes [3]:

- sob uma perspectiva abstrata, onde o mapa é constituído por nós ligados por arcos, podendo ser visto como hipergrafos ordenados, sendo que a ligação entre eles pode ser direcionada ou não;
- sob perspectiva de visualização, onde o mapa pode ser visto como diagramas; e
- sob a perspectiva dialética, onde os mapas conceituais são considerados uma forma de representação e comunicação do conhecimento através de linguagens visuais.

O nível pelo qual o aprendiz deverá começar é definido através de testes inteligentes, na qual é possível identificar por quais níveis o aluno deve passar ou não, ou ainda avaliar se o aluno tem condições de dar início ao aprendizado na etapa em que ele deseja, ou se deve rever alguns tópicos necessários, para só então aprender o que deseja de forma mais proveitosa.

O primeiro protótipo do LIA (*Learning by Intelligent Adaptiveness*) baseia-se em um conteúdo relativo ao ensino de matemática, tendo como sub-domínio específico o estudo de matrizes para ilustrar o uso dos Mapas Conceituais.

A Figura 3 ilustra a modelagem do conhecimento em níveis.

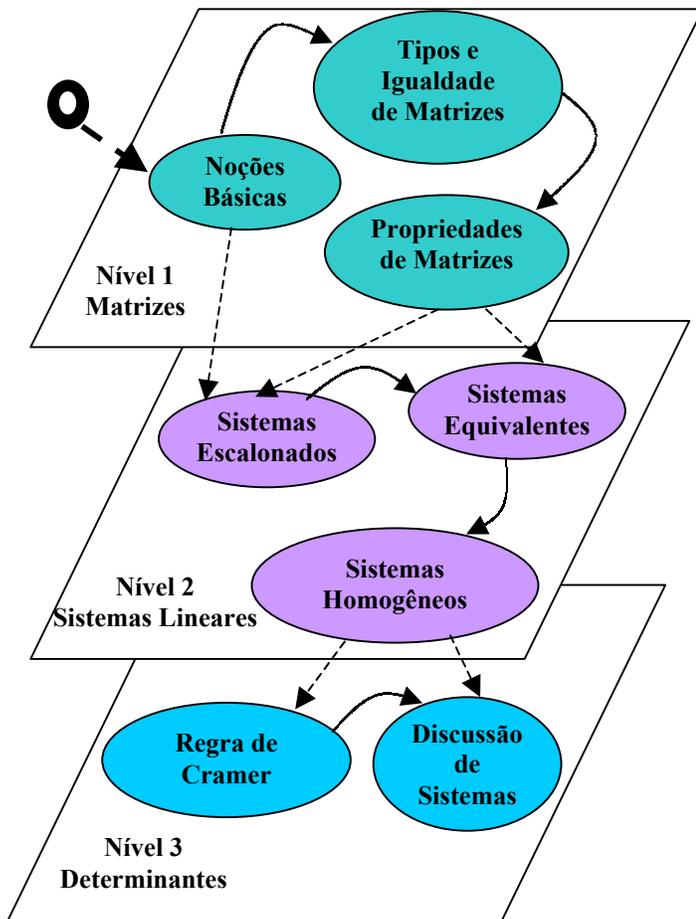


FIGURA 3

MODELAGEM DO CONHECIMENTO EM NÍVEIS

Pode-se notar na Figura 3 que há certas prioridades nos tópicos mostrados. Por exemplo, para o aprendiz aprender sobre *sistemas equivalentes* que está no nível 2, ele precisa ter conhecimento sobre *sistemas escalonados*, que por sua vez, tem como pré-requisito as *propriedades de matrizes* que está no nível 1, que por sua vez, necessita de *tipos e igualdade de matrizes*, que por sua vez requer o conhecimento de *noções básicas*. Assim, para o aprendiz estudar o tópico de sistemas equivalentes, ele precisa ter conhecimento de *sistemas escalonados*, *propriedades de matrizes*, *tipos e igualdades de matrizes* e *noções básicas*.

5. ARQUITETURA DO SISTEMA

Do ponto de vista da arquitetura, esse sistema está dividido em três camadas[7]:

- a primeira camada é a de apresentação. Esta contém toda a implementação relacionada com o *lay-out* do sistema, como ele é apresentado ao usuário aprendiz;
- a segunda camada é a de negócio. Esta camada é o local onde são feitos todos os processamentos e funções do

- programa, guardando toda a implementação relacionada a estas funções. É a interação entre a primeira camada e a terceira camada; e
- a terceira camada é a de dados. Nesta camada são guardadas as implementações relacionadas com o armazenamento de dados e acesso a esses dados, além de demais informações do sistema.

Uma demonstração dessa estrutura de camadas é dada na Figura 4.

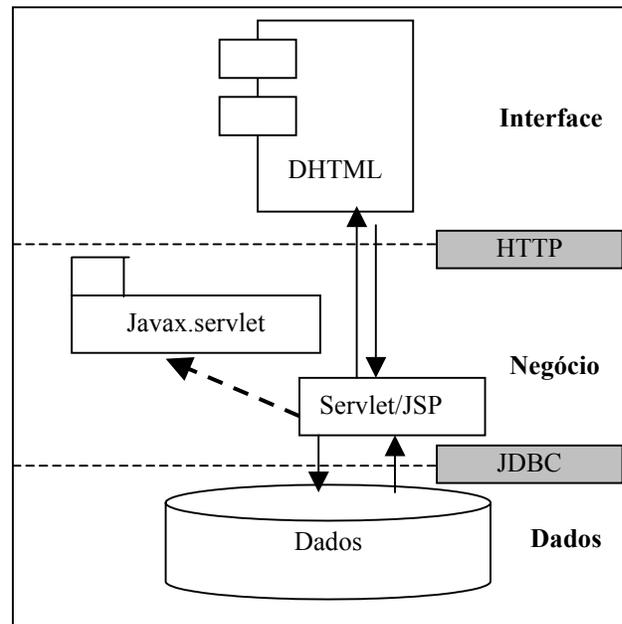


FIGURA 4

ARQUITETURA DE CAMADAS

Pode-se analisar uma ligação entre esta arquitetura de camadas e a divisão dos módulos de um Sistema Tutor Inteligente, explicados na seção 3.

A camada de dados representa o módulo aprendiz e o módulo domínio, haja visto que o módulo aprendiz armazena os dados sobre o aprendiz e o seu nível de conhecimento, já o módulo de domínio armazena todo o conteúdo a ser ensinado.

A camada de negócio representa o módulo tutor, pois busca os dados do aluno e cruza essas informações com o conteúdo do conhecimento e o mostra através da interface ao aprendiz. Já a camada de interface representa o módulo de comunicação, sendo implementada através da página DHTML em si.

6. FUNCIONAMENTO DO SISTEMA

Ao entrar no sistema pela primeira vez, o aprendiz precisa realizar seu cadastro. Em seguida, lhe é exibido um menu com todos os tópicos disponíveis no sistema, de forma

a que o mesmo possa escolher o tópico que mais lhe interessa.

É a partir dessa escolha, que depende única e exclusivamente do arbítrio do aprendiz, que o LIA gera exercícios de acordo com os pré-requisitos do tópico escolhido, analisando os mapas conceituais relativo ao domínio do conhecimento mapeado. Tais exercícios têm caráter avaliativo, pois pretendem verificar se o aprendiz possui as competências necessárias para estudar o tópico desejado.

Caso o aprendiz demonstre tais competências, o LIA permite a entrada deste no tópico desejado. O aprendiz pode, inclusive, visitar o tópico desejado mesmo que não demonstre as competências requeridas para tal. Porém, o LIA gera mensagens de avisos, indicando quais tópicos seriam mais proveitosos ou até quais tópicos o aprendiz deveria revisar, deixando assim o aprendiz com o direito de escolha, mas consciente de suas necessidades. Tal estratégia é adotada, entre outras razões, de forma a não frustrar as expectativas do aprendiz em relação ao aprendizado.

As avaliações eletrônicas são o cerne do processo de modelagem heurística do aprendiz no LIA, sendo que durante todo o processo o aprendiz passa por diversos testes que qualificam o nível de seu conhecimento dentro do mapa conceitual sob o qual está organizado o currículo da área de conhecimento sendo estudada.

Todas as informações do aprendiz são armazenadas em um banco de dados, para um acompanhamento continuado do mesmo, de modo a poder informar quais competências já foram adquiridas, quais ainda necessitam ser, além de fornecer uma avaliação de até como foi seu desempenho nas avaliações realizadas. A adaptabilidade do LIA depende grandemente da manutenção e recuperação de tais informações, assim como o acompanhamento de todas as ações por ele efetuadas.

7. TECNOLOGIAS UTILIZADAS

Nos sistemas adaptativos é necessário usar técnicas de inteligência artificial para se obter um resultado como visto anteriormente. Dentro deste ramo, tem-se por base as técnicas de modelagem do aprendiz, que usualmente são utilizadas em Sistemas Tutores Inteligentes. No LIA, o modelo do aprendiz é construído a partir de Mapas Conceituais implementados como grafos, onde cada nó do mesmo representa uma competência dentro de um dado currículo.

Em termos de implementação, conforme já demonstrado anteriormente na Figura 4, a interface da ferramenta é feita utilizando-se DHTML e Javascript. A camada de negócio é implementada em Servlets/JSP. A ligação da camada de negócio e a camada de dados é feito através do padrão JDBC.

Os tópicos a seguir irão explanar brevemente sobre cada uma dessas tecnologias.

DHTML

DHTML, ou HTML Dinâmico, é uma tecnologia da *Web* que torna os elementos de uma página muito mais dinâmicos que em HTML padrão[2]. Algumas das características principais de DHTML são:

Performance: O processamento é realizado localmente, ou seja, no *browser* do usuário, o que garante boa performance já que não exige o tráfego de informações pela rede durante a interação. Dado que o DHTML funciona no *browser* do lado cliente, uma vantagem pode ser imediatamente apontada: a redução drástica dos tempos de resposta. Utilizando o DHTML para deslocar a maior parte do trabalho para o lado cliente, torna-se possível proporcionar ao utilizador novos graus de interatividade. Por exemplo, a utilização de "*pre-caching*", que faz o carregamento em *background* de conteúdo, permite ao utilizador navegar entre páginas com um tempo de espera mínimo. Finalmente, não sendo menos importante, esta tecnologia abre novas perspectivas no que se refere ao desenvolvimento de conteúdo para operar do lado do cliente. Utilizando *scripts* (Em Javascript, por exemplo) combinados com DHTML é perfeitamente viável desenvolver aplicações *Internet* completas.

Compatibilidade: O DHTML não apresenta boa compatibilidade entre os *browsers*. Na verdade não existe um padrão para o DOM (*Document Object Model*), que é o centro dessa tecnologia. Tanto a *Microsoft* como a *Netscape* já suportam esse padrão a partir das versões 4.0 de seus *browsers*, mas cada uma com seu modelo de objetos. Logo, o código *client scripting* deve ser escrito de acordo com o *browser* destino, a menos que se faça uso das propriedades protegidas, que são um subconjunto das funcionalidades comuns a ambos os *browsers*.

Baseado em objetos: Cada elemento de uma página HTML é visto como um objeto, que pode ser acessado e ter suas propriedades, como cor e posicionamento, alteradas dinamicamente.

SERVLETS/JSP

Servlets são módulos escritos em Java que estendem as funcionalidades de servidores orientados a requisição/resposta [4]. São usadas basicamente para tratar dados de formulários HTML, sendo uma alternativa viável aos *scripts* CGI.

Virtualmente qualquer aplicação sob o paradigma cliente-servidor pode ser escrita em Servlets. Por exemplo, Servlets podem suportar videoconferência, pois uma servlet pode manipular múltiplas conexões e tratá-las

concorrentemente através de *threads*, além de reenviar requisições para outros servidores.

As servlets são fáceis de programar, são mais rápidas de serem executadas e adicionam grande nível de portabilidade do lado servidor.

Toda servlet tem o mesmo ciclo de vida: o servidor carrega e inicializa a servlet; a servlet manipula zero ou mais requisições de clientes; e o servidor remove a servlet. Cada um destes estados é governado por um método específico da servlet.

Já JSP é uma tecnologia que oferece uma maneira simples para se trabalhar com uma arquitetura similar à de Servlets, consistindo em código Java *server-side* incorporado em páginas HTML [5].

A arquitetura do LIA conta com implementações do lado do servidor utilizando Servlets e JSPs, notadamente para fazer a montagem dinâmica das páginas.

JDBC

O padrão JDBC (*Java DataBase Connectivity*) permite aos programadores se conectarem com um banco de dados, consultá-lo ou atualizá-lo, usando SQL *standard*[4]. A grande vantagem de adotar um padrão como o JDBC é a total independência de plataforma e modelo de BDs.

JDBC consiste em duas camadas: a camada superior é a API (interface de programação) do JDBC. Essa API se comunica com a API do driver gerenciador JDBC enviando para ela as diversas instruções SQL. O gerenciador deve se comunicar com os vários *drivers* de terceiros que efetivamente se conectam com o banco de dados, e retornam as informações da consulta ou executam a ação especificada pela interface.

Pode-se usar o JDBC em aplicativos e *applets*. Em uma *applet*, todas as restrições de segurança se aplicam. Em particular, uma *applet* que usa JDBC só pode abrir uma conexão de banco de dados ao servidor a partir da qual foi descarregada. Isto significa que o servidor *web* e o servidor de banco de dados devem estar na mesma máquina, o que não é uma configuração recomendável. Os aplicativos, por outro lado, têm completa liberdade para acessar servidores de banco de dados remotos. Para programas cliente-servidor fará mais sentido usar um aplicativo para acesso ao banco de dados.

No LIA, tais restrições não se aplicam, uma vez que não se utilizam *applets* em sua interface. Contudo, por se tratar de uma interface baseada em DHTML, futuras versões do sistema que porventura incorporem o uso de *applets* deverão tratar da questão de segurança.

O uso de tais tecnologias traz diversas vantagens para a implementação do modelo de arquitetura em três camadas:

- Separa-se a camada de interfaces (no cliente) da lógica de negócios e dos dados. Portanto torna-se possível

acessar os mesmos dados e as mesmas regras de negócio de múltiplos clientes.

- Caso se queira substituir a comunicação via HTTP entre o cliente e a camada de negócio, pode-se implementá-la através de RMI (*Remote Method Invocation*) ou algum outro mecanismo, uma vez que se trata de camadas desacopladas e independentes. Da mesma maneira, como se utilizou a ponte JDBC para fazer a conexão com o repositório de dados, o mesmo pode ser trocado, se necessário, bastando uma alteração de *driver*.

8. CONCLUSÃO

Com o crescimento do interesse pelo ensino a distância, o aumento da necessidade de criação de sistemas que se adaptem às habilidades e competências de cada aprendiz tem sido muito grande, pois cada vez mais é necessário a criação de sistemas que não só cativem a atenção do aprendiz, como também sejam eficientes no que diz respeito à eficácia do aprendizado obtido através da indicação de conteúdo individualizado, adequado às necessidades de cada aprendiz. Espera-se que o uso da ferramenta apresentada neste trabalho produza bons resultados no sentido de obter um melhor aprendizado, graças à adaptabilidade individualizada.

AGRADECIMENTO

As autoras agradecem o apoio emocional de seus familiares, o apoio didático de seu orientador, Prof. Ismar Frango Silveira, e o apoio financeiro da Universidade Presbiteriana Mackenzie.

REFERENCIAS BIBLIOGRÁFICAS

- [1] BOLZAN, W.; GIRAFFA, L. M. M. . “Estudo Comparativo sobre Sistemas Tutores Inteligentes Multiagentes Web” . Disponível em: [<http://www.inf.pucrs.br/tr/tr024.pdf>] . Visitado em: 13/09/2002.
- [2] CAMARGO, M. . “DHTML – Visão Geral e Introdução” . Disponível em: [<http://www.ibestmasters.com.br>] . Visitado em: 29/08/2002.
- [3] CARNEIRO, M. L. F. . “Mapas Conceituais” . Disponível em: [http://penta.ufrgs.br/~luis/Ativ2/mapas_mara.html] . Visitado em: 26/08/2002 .
- [4] CORNELL, G.; HORSTMANN, C. S. . “Core Java- v.2-Recursos Avançados”. 1ªed. São Paulo: Makron Books,2000.
- [5] HALL, M. . “Tutorial on Servlets and JSP”. 1999. Disponível em: [<http://www.apl.jhu.edu/~hall/java/Servlet-Tutorial/>] . Visitado em : 29/11/2002.
- [6] KONRATH, M. L. P. . “Mapas Conceituais” . Disponível em: [<http://penta2.ufrgs.br/edutools/mapasconceituais/>] . Visitado em: 26/08/02.
- [7] PAULA FILHO, W. de P. . “Engenharia de Software” . 1ªed . LTC,2001.
- [8] SILVA, K. M.; ALVES, F. B. . “Avaliação Eletrônica na Internet(e-Evaluation)”. *Dissertação de Graduação. Universidade Presbiteriana Mackenzie*. . São Paulo: Dezembro /2001.