

ENGENHARIA WEB E SISTEMAS COLABORATIVOS: UM ESTUDO APLICADO AO CONTROLE DA DOCUMENTAÇÃO DE PRODUÇÃO ACADÊMICA

WEB ENGINEERING AND COLLABORATIVE SYSTEMS: A STUDY APPLIED TO ACADEMIC DOCUMENT MANAGEMENT

CHEMIN, Rayan Marlus¹; LAMB, Juliano Rodrigo²; MALLMANN, Anderson Algacir³
^{1,2,3}Universidade Tecnológica Federal do Paraná – UTFPR, Medianeira, Paraná, Brasil
¹chemin.rayan@gmail.com; ²julianolamb@gmail.com; ³a.a.mallmann@gmail.com

Resumo

Com o constante avanço de webapps e websites a Internet passou a tornar-se um ambiente colaborativo onde constantemente é mantido seu desenvolvimento. Com base nessa abordagem colaborativa, este trabalho apresenta o desenvolvimento de um protótipo baseado em sistemas colaborativos para realizar o gerenciamento/compartilhamento de arquivos (papers). Por meio do sistema gerenciador, o usuário obtém melhor controle sobre os papers publicados e seus certificados, mantendo assim sua base de dados atualizada; permitindo ainda que todos os autores de um paper possam compartilhar facilmente o documento final publicado junto aos anais de um evento, por exemplo, bem como a certificação deste.

Palavras-chave: gerenciamento bibliográfico; webapp; currículo lattes; engenharia de software.

Abstract

In this paper, we intend to develop a software to manage bibliography production. The tool must allow the sharing process of papers and certificates, between all his authors, working like a network social, where an author can keep a network of collaborators and all documents associated.

Key-words: management bibliography; webapp; lattes curriculum; software engineering.

1. Introdução

Em constante evolução, a Internet está sempre expandindo sua abrangência já que inicialmente eram utilizados apenas por computadores, e atualmente diversos eletrônicos (como celulares e *tablets*) possuem a capacidade de acesso. Com todo esse avanço, surgem necessidades para aumentar e melhorar a comunicação, bem como a troca de informações. Com o crescimento, surgiram maneiras de se compartilhar informações, síncronas e assíncronas, tais como e-mails, que inicialmente possibilitavam a troca de imagens e informações, e bate-papos. Estes canais, ainda são utilizados, mas hoje são sombreados pela ascensão de redes sociais, bem como *websites/webapps* de relacionamentos, compartilhamento de arquivos, ferramentas para ensino a distância, entre outros.

Sistemas consistem em um conjunto de informações que estão constantemente interagindo e integrando-se, sempre com o propósito de atingirem objetivos e alcançarem resultados (REZENDE e ABREU, 2003). Laudon e Laudon (2004) explicam que estes sistemas podem contribuir para a solução de vários problemas, independentemente do seu tipo ou do seu uso. Um sistema de informação deve produzir alguma saída, produzir algum resultado útil, caso contrário não pode ser caracterizado como sistema.

Cada sistema possui suas exclusividades e padrões, a definirem a qualidade do sistema. Para *websites* e sistemas *web* (*webapps*), existe a chamada Engenharia *Web*, que define normas, padrões entre outros, voltados para o ambiente *web*. Esta, incorpora abordagens, metodologias, ferramentas, técnicas e normas para atender os requisitos exclusivos dos sistemas para a *web*. O desenvolvimento de sistemas de informação para a *web* apresenta características distintas do desenvolvimento de *software* tradicional, de forma que a aplicação de seus padrões é exclusiva, acrescentando

qualidade ao desenvolvimento do mesmo.

Websites e *webapps* diferem entre si, uma vez que aplicativos para a *web* possuem características exclusivas: são dirigidos a conteúdo, estão em constante evolução, têm curto prazo de desenvolvimento, dentre outras. Um dos desafios é adaptar as técnicas existentes, para uso na *web* por meio da Engenharia para a *web* (*web engineering*), ou WebE (BREVE, 2002). Ainda quanto essa comparação, Jacyntho (2008) descreve um exemplo dessas diferenças a qual é o conteúdo: a *web* é essencialmente um meio de informação. Além da funcionalidade, uma aplicação *web* é orientada a conteúdo que pode ser compreendido por dados estruturados (banco de dados, por exemplo) e não estruturados (arquivos textos, vídeos, etc.). Além disso, o conteúdo é dinâmico, precisa ser continuamente atualizado e de qualidade em termos de consistência e confiabilidade. Isto implica em um efetivo projeto de informação, bem como gerenciamento de conteúdo apropriado. Um *website* usualmente apresenta conteúdo estático, utilizando de textos e/ou imagens, e navegação entre este conteúdo.

Para *softwares* o termo “sistema colaborativo” parte de um princípio onde o sistema possui usuários que interagem com o intuito de compartilhar dados ou algum tipo de informação, sendo utilizado posteriormente para outros. Borba (2010) citado por Borges (2013) afirma que “sistemas colaborativos são ferramentas especializadas em fornecer subsídios para facilitar a coordenação, comunicação e o controle das atividades entre os envolvidos e com isso destaca-se o suporte e a promoção da colaboração como também a redução de problemas produzidos pelo tempo e espaço”.

Utilizado em diversas áreas da computação, os sistemas colaborativos acabam por se tornar importantes e utilizados amplamente, trazendo vantagens tais como: organização, permitir a utilização de vários usuários, entre outros. Suas

características principais dizem respeito a requisitos de funcionamento, protocolos, conexões, compartilhamento de recursos, concorrência de acesso, transações distribuídas, sincronização e gerenciamento de sessões.

Pensando nesse contexto, pertinente ao controle de documentos, uma demanda específica diz respeito à gerência de documentos relativos ao currículo de um pesquisador. Por vezes o pesquisador não consegue ter todo o seu currículo documentado, visto que suas produções podem ser feitas em equipe e não existe uma comunicação simplificada entre os membros desta, o que ocasiona restrição de acesso aos documentos/certificados de apresentação.

Existem aplicativos como o Mendeley, EndNote e Zotero (YAMAKAWA, 2014) destinados a gerenciamento bibliográfico, no entanto o gerenciamento proposto por estes softwares, restringe-se ao utilizador, sendo o compartilhamento/solicitação de artigos um processo trabalhoso. Aprimoramentos como este, já são observados por Marchiori et al (2010) citados por Yamakawa (2014), em que considerando-se o fato de o Mendeley trabalhar com redes sociais, melhorias devem ser feitas no sentido da extração de metadados e a busca de textos em formato PDF completos.

O objetivo deste trabalho é desenvolver uma ferramenta *web* colaborativa (baseada nos conceitos de sistemas colaborativos) que possibilite o compartilhamento das informações relativas do meio acadêmico, tendo também como intuito permitir que as informações nele inseridas possam ser comparadas com as da plataforma Lattes¹. Desta forma amplia-se o gerenciamento e controle de artigos unificando em um ponto único as informações contidas em ambos os sistemas.

¹ Plataforma que integra currículos de pesquisadores, grupos de pesquisas e instituições. Disponível em <http://lattes.cnpq.br/>

2. Material e métodos

A análise em um todo pode ser considerada a parte anterior ao desenvolvimento, pois concentra-se nela como todo o processo deve ocorrer, conter, e como deve ser feito. A utilização da análise pode ser aplicada para todo tipo de sistema, desde aplicações de softwares até para ambientes web. As etapas para realizar a análise de um sistema são semelhantes para diversos ambientes, e podem ser aplicadas de forma diferentes para os sistemas.

Uma forma de análise inicial para um projeto é o levantamento de requisitos, seu processo de análise é utilizado para fornecer qualidade e planejamento em relação ao desenvolvimento, de forma que futuras necessidades sejam estudadas e documentadas, assim poupando tempo e esforço para a construção de projetos.

Para melhor atendimento aos objetivos, o protótipo foi separado em dois módulos sendo eles o *software* (cliente) e o *webapp* (servidor).

O módulo cliente consiste em uma pequena aplicação que deve ficar executando continuamente no computador do usuário. É este módulo quem verifica por atualizações junto ao servidor e responsável por fazer *download* dos arquivos necessários. São requisitos principais do módulo cliente (conforme diagrama de caso de uso, apresentado na Figura 1):

- a. Alterar diretório: sistema permite alterar o diretório padrão onde são armazenados os arquivos;
- b. Recuperar solicitações: o sistema realiza a consulta para verificar e apresentar a quantidade de solicitações que há na conta associada;
- c. Recuperar arquivos do servidor *web*: o sistema realiza o *download* dos arquivos que estão pendentes de serem baixados para o usuário associado.

O módulo servidor por sua vez, compreende o módulo de administração, local em que existe

uma interação maior do usuário com o sistema, por meio de diferentes interfaces. São requisitos principais do módulo servidor (conforme diagrama de caso de uso, apresentado na Figura 2):

- a. Manter usuário: o sistema mantém os dados informados ao momento da autenticação;
- b. Manter evento: o sistema mantém os dados do evento, e possibilita caso o usuário seja o dono (criador) do evento: editar as informações sobre o mesmo;
- c. Manter arquivo: o sistema mantém os dados do arquivo, e possibilita caso o usuário seja o dono (criador) do arquivo: editar o título sobre o mesmo.

A utilização de diagramas torna melhor a apresentação de um sistema, de forma subjetiva, visualizando um diagrama bem definido é possível interpreta-lo, outra forma aplicada onde torna uma melhor visão do sistema em geral é para o desenvolvedor, que consegue basear-se no diagrama para desenvolver uma ação ou estrutura.

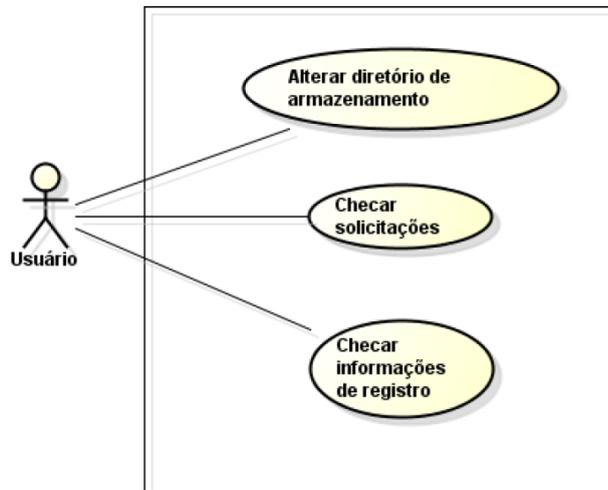
A *Unified Modelling Language* (UML) é uma linguagem de modelagem, que usa de uma notação gráfica, para especificar, visualizar e documentar modelos de software orientados por objetos. A UML é composta por diversos diagramas, sendo que o objetivo deles é fornecer múltiplas visões do sistema que será modelado, analisando-o e modelando-o sob diversos aspectos. Cada diagrama possui suas características diferenciadas, podendo ser classificados entre duas categorias: diagramas estruturais ou comportamentais.

Neste projeto foram utilizados os seguintes diagramas:

- a. *Use case diagram* (Diagrama de caso de uso): diagrama comportamental que descreve as possíveis ações de um envolvido ao sistema;
- b. *Activity diagram* (Diagrama de atividades): diagrama comportamental que descreve as etapas para a ação de uma atividade;

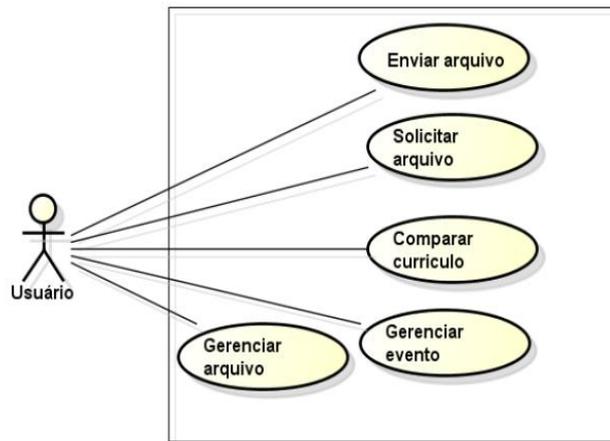
- c. *Class diagram* (Diagrama de classes): diagrama estrutural cujo objetivo é apresentar a estrutura da classe e suas ligações.

Figura 1 - Diagrama de caso de uso CLIENTE



Fonte: Autoria Própria

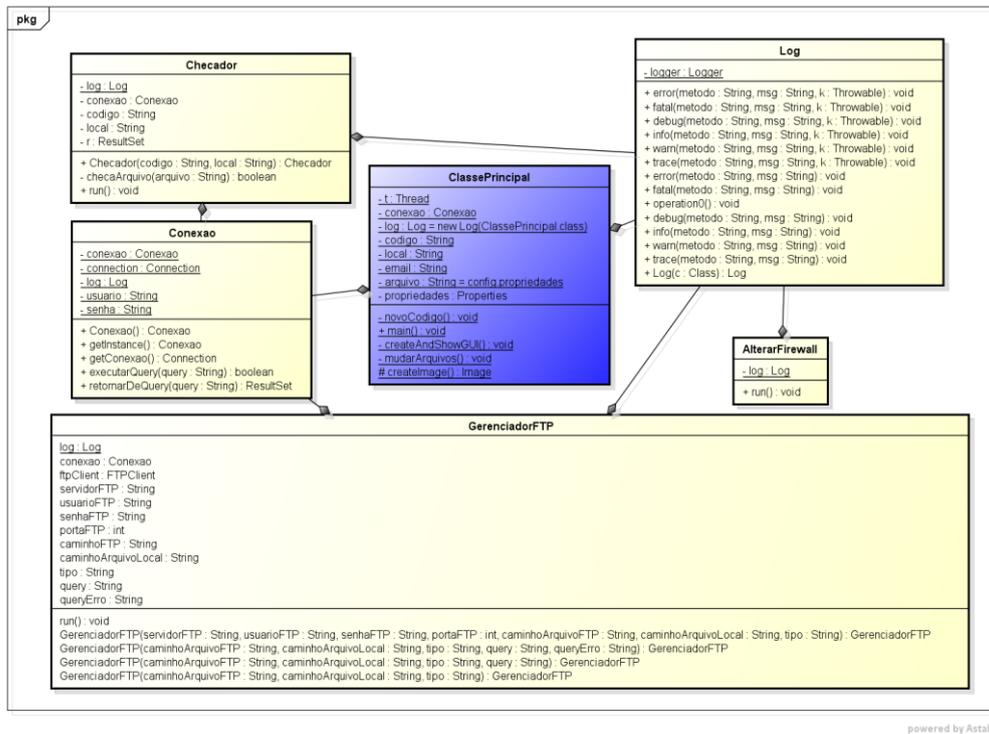
Figura 2 - Diagrama de caso de uso SERVIDOR



Fonte: Autoria Própria

Na Figura 3 é apresentado o diagrama de classes do *software* cliente, nele são descritos todas as classes e métodos contidos no *software*.

Figura 3 - Módulo cliente - Diagrama de classes do *software*

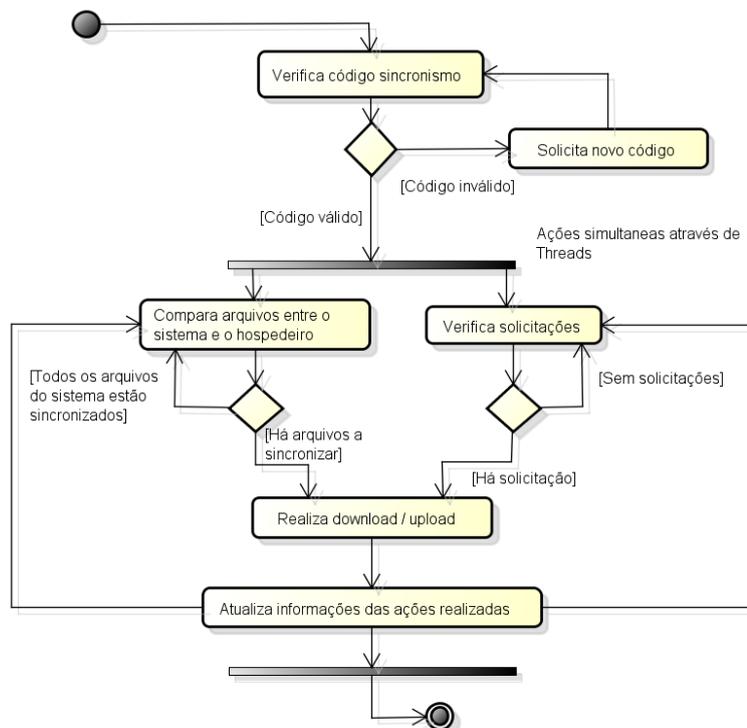


Fonte: Autoria Própria

Ao iniciar o *software* cliente, são realizadas verificações para reconhecimento de conta, com base no código informado em sua primeira inicialização, ou caso o arquivo de

configurações seja removido. A Figura 4 apresenta todo o processo de atividades do *software*.

Figura 4 - Módulo cliente - Diagrama de atividades do *software*



Fonte: Autoria Própria

O desenvolvimento foi feito por meio das tecnologias:

- **PHP (Hypertext Preprocessor):** Mediante as facilidades apresentadas pela linguagem, adotou-se, neste trabalho, a linguagem de desenvolvimento PHP na codificação da *webapp*.
- **HTML (HyperText Markup Language):** No desenvolvimento da *webapp*, o HTML assumiu o papel de apresentar o conteúdo, pois as informações são apresentadas pela linguagem de marcação, estruturando o conteúdo em páginas web.
- **CSS (Cascading Style Sheets):** O CSS3 foi empregado no desenvolvimento do *webapp* simplificando códigos, aumentando a compatibilidade com os navegadores, e aplicando novas formas de estilo aos elementos da estrutura HTML.
- **JavaScript:** No desenvolvimento web do projeto, o Javascript foi responsável pela validação de campos, navegação do menu, e outras verificações *client-side* como capturar a resolução do usuário e verificação do navegador utilizado.
- **OpenID:** Para realizar a autenticação no *webapp* deste trabalho, foi utilizado o OpenID. Ao usar esta biblioteca, é simplificada a forma de como o usuário entrará no sistema, além disso, a segurança também é elevada, pois a autenticação é realizada pelo provedor (Google).
- **Ajax:** Neste projeto, o Ajax foi utilizado com a biblioteca jQuery, que realizou grande

parte de navegação e também validação de formulários.

- **jQuery:** O uso da biblioteca tornou o *webapp* mais dinâmico e compatível com diversos navegadores. Seu uso também foi utilizado no menu em base com a linguagem de programação Javascript.
- **Java:** No projeto, o desenvolvimento do software necessário para o funcionamento do projeto ao todo, foi baseado na linguagem de programação Java, pois atende as demandas necessárias de comunicação com SGDB e realizar ações *server-side*, contando também com a compatibilidade entre os sistemas operacionais.
- **MySQL:** Para a realização do projeto foi escolhido o MySQL, pois atende as necessidades tanto da *webapp* quanto do software cliente. O uso de banco de dados possibilitou a comunicação entre ambas às partes do projeto, em partes centralizando as informações de conteúdo em um único ponto de acesso.

3. Resultados e discussões

A interface inicial da *webapp* desenvolvida pode ser visualizada na Figura 5. A *webapp* permite fazer o gerenciamento de eventos, arquivos, visualizar gráficos referentes aos documentos da conta e obter o *software* cliente.

Figura 5 - Interface de boas-vindas do sistema



Fonte: Autoria Própria

A aplicação *web* usa o sistema de *login* OpenID baseado em contas do Google para que não seja necessária a criação de mais uma conta e com isso o usuário tenha que armazenar mais uma senha. Este tipo de procedimento é seguro e não armazena nenhuma informação pessoal do usuário junto à aplicação que consome o serviço, com exceção do nome e o e-mail informados, que são necessários para manter a

associação dos dados da conta.

Para ser possível associar um arquivo a um evento, deve-se primeiro existir o evento (Figura 6). Após sucesso ao cadastrar o evento, este estará disponível para ser associado aos arquivos. Após isso, é possível editá-los e também visualizar as informações sobre cada um, listando os *papers* que estão vinculados ao evento, como apresentado nas Figuras 7 e 8.

Figura 6 - Tela de cadastro de evento

Fonte: Autoria Própria

Figura 7 - Tela de gerenciamento de eventos



Fonte: Autoria Própria

Figura 8 - Trabalhos relacionados a um evento



Fonte: Autoria Própria

O cadastro dos eventos possibilita o envio de trabalhos, sendo este processo de envio realizado em três etapas: acessando a opção de Envio no menu, ao primeiro momento são solicitados: o arquivo do documento, o título e o tipo da publicação (conforme apresenta a Figura 9). Ao avançar de tela é exibido outro formulário, em que é solicitado o evento para vincular com o documento (Figura 10).

Figura 9 - Interface para *upload* de trabalhos



Fonte: Autoria Própria

Figura 10 - Vincular evento ao documento



Fonte: Autoria Própria

Após selecionar o evento, a última etapa é inserir os autores que participaram na realização do documento (Figura 11). Para

relacionar autores com o arquivo, cada autor deve estar cadastrado no sistema. Caso ocorra de algum autor não estar cadastrado, posteriormente através do gerenciamento deste *paper*, é possível incluí-lo de forma que convide o mesmo a utilizar o sistema, por meio de um convite no *email*.

Figura 11 - Associação de autores do documento



Fonte: Autoria Própria

O módulo servidor possibilita ainda a solicitação de um *paper*; situação em que um dos autores apresentou, está de posse do certificado/documento e não repassou ainda aos demais autores.

A solicitação do artigo pode ser realizada de duas formas: por meio do título, ou por meio do link em cor azul abaixo do campo do título, como mostra a Figura 12.

Realizando por meio da pesquisa por título, espera-se que o arquivo já esteja no sistema, ou seja, que algum usuário possua o mesmo. Então é realizado um pedido de solicitação de arquivo, e enviado uma mensagem no sistema e também no e-mail do usuário que possua o arquivo. Caso o arquivo não se encontre no sistema, a outra forma de realizar a solicitação, é com base no e-mail da pessoa que eventualmente possua o arquivo, mas que ainda não se encontre no sistema. A Figura 13 apresenta a primeira etapa da solicitação de arquivo, cujo usuário solicitado não faz parte do sistema.

Figura 12 - Solicitação de arquivo



Fonte: Autoria Própria

Figura 13 - Solicitação por e-mail



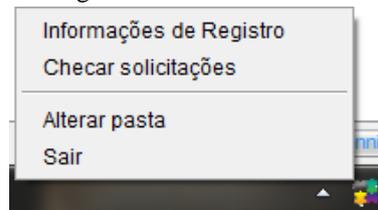
Fonte: Autoria Própria

Outra funcionalidade implementada no módulo servidor é a comparação de currículo entre a plataforma Lattes. Com esta, o usuário possui a capacidade de verificar quais *papers* estão publicados em seu currículo na plataforma e destes quais estão cadastrados no sistema. A comparação é feita com base no arquivo XML obtido na plataforma Lattes.

Para melhor funcionamento do sistema ao todo, foi realizado o desenvolvimento de uma aplicação cliente para que fossem verificadas constantemente possíveis atualizações junto ao sistema web, assim trabalhando em conjunto para um melhor controle e usabilidade do sistema ao todo.

Com base em todas as informações alimentadas no módulo servidor, cabe ao *software* cliente simplesmente sincronizar o que foi definido. A aplicação em execução pode ser vista com um ícone junto a bandeja do sistema, com algumas poucas opções de configuração (Figura 14).

Figura 14 - Módulo cliente



Fonte: Autoria Própria

4. Considerações finais

Com o desenvolvimento deste estudo, pode-se verificar que:

- A gerência de documentos apresentada pode ser realizada incluindo processos da Engenharia *Web*, por meio de uma ferramenta *web* colaborativa;
- Por meio do desenvolvimento do protótipo, diferentemente de *softwares* tradicionais, verificou-se que devem ser levados em conta, aspectos relacionados ao uso do sistema via ambiente *web*, tais como comunicação entre os diferentes módulos (i.e.: cliente – servidor);
- O mecanismo de comparação com a documentação do lattes permite identificar rapidamente se o usuário tem a documentação comprobatória de todos os itens cadastrados em seu currículo;
- A interação proporcionada pelo uso da ferramenta permite concentrar informações da equipe, fornecendo um canal de comunicação e evitando o armazenamento de versões desatualizadas por parte de algum autor.

Referências

BORBA, Gilmar. 2010. **Sistemas colaborativos: introdução**. Acesso em 09 Dez. 2012. Disponível em <http://gilmarborba.com.br>

BORGES, Marcello Wanderley. **A influência dos sistemas colaborativos aplicados a cadeia de suprimentos no setor de serviços**. Revista Eletrônica Tecnologias e Produção, Vol. 1, No 1, 2012.

JACYNTHO, Mark Douglas de Azevedo. **Processos para Desenvolvimento de Aplicações Web 2009**

(Monografias em Ciência da Computação, Departamento de Informática, PUC-Rio).

LAUDON, K Kenneth. C.; LAUDON, Jane.P. - **Sistemas de Informação Gerenciais**. São Paulo: Pearson/Prentice Hall, 5.ed., 2004.

MARCHIORI, Patrícia Zeni; BETTONI, Eduardo Michelotti; APPEL, Andre Luiz; TABORDA, Carlos Alexandre Lourenço. **Aspectos estruturais e motivacionais e possíveis zoneamentos discursivos em software social acadêmico**. Liinc em revista, Rio de Janeiro, v. 6, n. 2, p. 355-369, 2010.

REZENDE, D. A., & ABREU, A. F. (2003). **Tecnologia da informação aplicada a sistemas de informação**

empresariais (3ª ed.). São Paulo: Atlas.

YAMAKAWA, Eduardo Kazumi et al. **Comparativo dos softwares de gerenciamento de referências bibliográficas: Mendeley, EndNote e Zotero**. Transinformação, Campinas, v. 26, n. 2, p. 167-176, Aug. 2014.

Artigo submetido em: 15.08.2015

Artigo aprovado para publicação em: 14.03.2016