

INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

*Uma Prospecção de Plataformas para o Desenvolvimento
do Sistema “Todos Nós em Rede”*

*Julio Cesar Dos Reis
Roberto Pereira
Heiko H. Hornung
Leonelo D. A. Almeida
Maria Cecília Calani Baranauskas*

Technical Report - IC-11-17 - Relatório Técnico

October - 2011 - Outubro

The contents of this report are the sole responsibility of the authors.
O conteúdo do presente relatório é de única responsabilidade dos autores.

Uma Prospeção de Plataformas para o Desenvolvimento do Sistema “Todos Nós em Rede”

*Julio Cesar Dos Reis, Roberto Pereira, Heiko H. Hornung,
Leonelo D. A. Almeida, M. Cecília C. Baranauskas*

Departamento de Sistemas de Informação, Instituto de Computação
Universidade Estadual de Campinas – UNICAMP
13083-970, Campinas, SP, Brasil

{julio.reis, roberto.pereira, leonelo.almeida, heiko.hornung, cecilia}@ic.unicamp.br

Resumo

O projeto “*Todos Nós em Rede*” (TNR) objetiva a formação continuada à distância de professores de Educação Especial dos sistemas de ensino público brasileiro, por meio da constituição de Redes Sociais Inclusivas (RSIs) desses profissionais. Este relatório técnico apresenta uma avaliação de tecnologias e a viabilidade técnica dessas para o desenvolvimento do sistema computacional do TNR. Este sistema se caracterizará como uma RSI nas quais valores como acessibilidade, autonomia e autoridade demonstram ser importantes. Este trabalho levanta, seleciona, apresenta, analisa e discute, a partir de uma pesquisa exploratória, possibilidades para a plataforma de desenvolvimento do sistema. Também é realizada uma análise comparativa que articula as características das diferentes plataformas e ilustra os aspectos positivos e negativos sob a luz dos requisitos essenciais para o sistema TNR.

Palavras-Chave: Redes Sociais *Online*; Todos Nós em Rede; AEE; CMS.

1 Introdução

O propósito do projeto TNR (Proesp/CAPES) [1] é propiciar condições para que Redes Sociais *Online* (RSOs) sejam formadas por professores da Educação Especial, que trabalham com o Atendimento Educacional Especializado (AEE) de alunos com deficiências em escolas regulares, como forma de assegurar, a partir de ações à distância nessa rede, encontros casuais e sistemáticos entre esses profissionais. Portanto, o projeto tem em vista principalmente: a) a busca de *expertise* e *experts* para a solução de problemas que professores encontram no cotidiano de seus trabalhos; b) o compartilhamento de idéias e a socialização de casos bem sucedidos, bem como de casos ainda não resolvidos; e também c) o estabelecimento de uma dinâmica continuada de estudos e respostas mediadas pela tecnologia de maneira autônoma [1].

No contexto desse projeto, o objetivo deste relatório técnico é efetuar uma prospecção técnica para o desenvolvimento do sistema computacional do TNR. O objetivo principal deste sistema será apoiar os professores na discussão de seus casos e práticas profissionais do dia-a-dia. Desta maneira, faz parte dos objetivos [1], do ponto de vista computacional em alto nível, estudar e desenvolver mecanismos para a identificação e atribuição de autoridade entre os participantes de uma RSO, estudar seus impactos, suas características e suas aplicações visando: i) a melhoria da qualidade na recuperação de informação; ii) a construção de uma rede social baseada na credibilidade, confiança, reputação e nas competências de seus membros; iii) a melhoria na confiança e na segurança das interações de seus membros na RSO.

De acordo com o contexto do projeto, há indícios de que uma solução computacional para apoiar professores do AEE na discussão de seus casos deve propiciar e considerar características de um sistema de RSO. Além disso, possibilidades de compartilhamento explícito de informação/objetos e a disponibilização de artefatos da Semiótica Organizacional [3] para apoiar a discussão de casos, também se apresentam como possíveis requisitos para o sistema, conforme investigações prévias no contexto do projeto. Como requisitos fundamentais que norteiam e motivam o desenvolvimento do projeto têm-se: acessibilidade e usabilidade, refletindo a natureza inclusiva do próprio projeto. Esses requisitos serão operacionalizados e refletidos em funcionalidades do sistema, exigindo uma base computacional comum. Adicionalmente, em um projeto acadêmico, as tecnologias envolvidas devem ser principalmente de código e de licença aberta para maximizar a disseminação do conhecimento criado ao desenvolver o sistema e minimizar o custo. Uma lista dos tipos de licenças abertas e suas descrições é disponível em [75]. A solução ainda deve atender a requisitos básicos de uma aplicação *Web* como ser multi-plataforma do lado cliente, possibilitar extensões, *etc.* A base computacional do sistema deve conseguir atender a estes requisitos e a outras necessidades tecnológicas do projeto (ainda sendo levantadas) visando experimentar soluções técnicas acopladas rapidamente a um núcleo comum de *software*. Finalmente, as tecnologias envolvidas devem ser de fácil aprendizado pelos membros da equipe de desenvolvimento que será composta por alunos de pós-graduação, pesquisadores e desenvolvedores *Web* tercerizados.

Fundamentado nestes requisitos iniciais, este relatório técnico visa desenvolver uma análise exploratória sobre possíveis soluções tecnológicas que possam atender a estes objetivos. Para isso, serão investigados *Content Management Systems* (CMSs) — Sistemas de Gerenciamento de Conteúdo — junto a suas características, e sistemas de RSOs, visando observar a viabilidade técnica de adaptação e instanciação destes sistemas para o contexto do TNR. A investigação de CMSs e de sistemas de RSOs foi abordada devido à facilidade e rapidez que estes sistemas podem trazer para o desenvolvimento de um novo sistema *Web*, além da possibilidade de utilizar recursos e *features* já disponíveis abertamente e que podem contribuir para o desenvolvimento do projeto. Até a publicação deste relatório técnico, o projeto TNR ainda havia chegado ao momento de elicitar requisitos dos demais *stakeholders* do projeto (*e.g.* professores, *experts*). Contudo, a investigação de CMSs parece pertinente por que esse tipo de sistema geralmente ou já incorpora ou facilita a extensão por funcionalidades de outros sistemas *Web*, como: fóruns, (*micro-*) *blogs*, *chat*, *etc.*

Este relatório técnico está organizado da seguinte maneira: a seção 2 apresenta o método adotado para a seleção e análise das tecnologias para o desenvolvimento do TNR; a seção 3 apresenta as opções levantadas junto às suas características, vantagens e desvantagens; a seção 4 faz uma discussão geral com base nos resultados obtidos; a seção 5 apresenta as considerações finais ressaltando as principais conclusões e apontando trabalhos futuros.

2 Método para Seleção e Análise das Plataformas

O objetivo de reutilizar soluções de CMSs de código aberto para reduzir o esforço necessário para o desenvolvimento de protótipos de trabalho já tem sido investigado na literatura [e.g. 4]. O uso de CMSs é uma possibilidade interessante no desenvolvimento de aplicações *Web*, pois estes são projetados para simplificar a publicação de conteúdos digitais, permitindo às pessoas criarem e disponibilizarem conteúdo sem necessitar de conhecimento técnico avançado para a utilização ou desenvolvimento de *software*. CMSs objetivam facilitar a criação, edição, publicação e distribuição de informação. Esse usualmente apresenta uma plataforma de uso facilitado, com soluções prontas e customizáveis reduzindo o tempo de implementação, o que permite a concentração de esforço em soluções específicas e na produção do *layout* [4].

Há diversos CMSs disponíveis atualmente, que incluem tanto soluções comerciais como também soluções livres. A análise e comparação entres CMSs tem sido descrita largamente na literatura, inclusive como relatórios anuais [e.g. 6]. Há centenas de soluções de CMS e muitos critérios de comparação. Contudo, seguindo critérios de instalação, configuração, administração e uso, uma comparação entre CMSs mais comumente usados não revela uma ordem representativa de diferença entre eles [4].

O presente trabalho não tem o objetivo de fazer uma análise profunda e extensiva de CMSs, devido primeiramente à quantidade disponível. Além disso, os critérios de avaliação muitas vezes devem considerar necessidades específicas relacionadas ao contexto de aplicação e utilização desses sistemas, o que impacta diretamente na viabilidade para adotá-los. Deste modo, selecionamos para análise um subconjunto de CMSs, mais comumente usados na *Web* e que são de código aberto.

Adicionalmente, selecionamos um subconjunto de “*Social Networking CMSs*” (SN-CMSs). SN-CMSs são uma categoria de CMSs recentemente surgida que podem ser caracterizados como CMSs com funcionalidades para RSOs, ou soluções com maior foco nestas aplicações. Devido à natureza do TNR, uma pesquisa exploratória mais profunda à procura de SN-CMSs foi realizada, uma vez que não há uma base de dados centralizadora que armazene características e possibilite uma comparação entre estes sistemas conforme existe para CMSs tradicionais [veja 7]. Fizemos a distinção entre SN-CMS e CMS tradicionais pois algumas aplicações se nomeiam explicitamente como SN-CMS. Muito dos CMS tradicionais também apresentam funcionalidades para redes sociais, mas estes tem um propósito mais geral. Desta forma, fizemos a distinção nestas duas categorias. Além dos

SN-CMSs, também é apresentado e avaliado o sistema de RSI [66] *VilanaRede* [8], por ser uma iniciativa de RSI e por disponibilizar recursos de múltiplas mídias. Uma rede social com recursos que promovam o acesso de todos, incluindo aqueles à margem da cultura digital, pode ser definida como uma RSI [66]. Finalmente, o sistema *LeMill* [9] também está incluído na análise por ser considerado uma RSO voltada ao compartilhamento, criação e colaboração em recursos educacionais. Estes dois sistemas foram pontualmente selecionados por suas características intrínsecas que se apresentam como compatíveis com os requisitos até então conhecidos do sistema TNR.

Os CMSs tradicionais escolhidos para análise foram: *WordPress* [9], *Drupal* [11], *Joomla* [12] e *Plone* [13]. Nessa análise são apresentadas as características básicas, tecnologias utilizadas, *etc.* Também foi realizada uma comparação entre eles mostrando vantagens e desvantagens; para isso foi utilizado o CMSMatrix [7]. Sobre os SN-CMSs, após uma pesquisa exploratória extensa, uma filtragem inicial foi realizada excluindo os sistemas comerciais. Os sistemas mais viáveis do ponto de vista técnico foram selecionados e apresentados. Como critérios de análise foram utilizadas e observadas questões de documentações disponíveis, se o sistema possui atividades recentes de atualização, uma comunidade de desenvolvedores e usuários ativa, possibilidades de modificação e personalização do *software*, entre outras. Ainda, pontos positivos e negativos foram apontados na apresentação das características dos SN-CMSs selecionados (*Elgg* [37], *Pligg* [33] e outros). As características técnicas são apresentadas e ressaltadas quando disponíveis. No *VilanaRede* também é apresentada uma entrevista com os desenvolvedores envolvidos no *design* do sistema. Na discussão geral, é desenvolvida uma análise técnica para mostrar as vantagens e desvantagens do ponto de vista dos aspectos técnicos de se utilizar tais sistemas como ponto de partida (base) para o sistema TNR.

3 Análise das Plataformas para o Desenvolvimento do Sistema TNR

Esta seção apresenta os sistemas selecionados para a análise. Primeiramente são apresentados os CMSs, em seguida os SN-CMSs, e logo os sistemas *VilanaRede* e *LeMill*.

3.1 CMSs

WordPress

O *WordPress* iniciou apenas como um sistema para a criação e gerência de *blogs*, mas tem evoluído para ser utilizado como um CMS completo, com muitos *plugins*, *widgets*, e temas. O *WordPress* possui a GNU *General Public License* (GPL), versão 2. É uma ferramenta de criação de *blogs* avançada e fornece um conjunto rico de recursos. Por exemplo, por meio dos painéis de administração pode-se definir opções para o comportamento e apresentação do *blog*. Via estes painéis, pode-se também facilmente compor um conteúdo no *blog*. Além das funcionalidades tradicionais de um CMS como, por exemplo: registro de usuário, senha, comentários, temas, suas principais funcionalidades incluem: (1) total conformidade com os padrões da *World Wide Web Consortium* [14]; (2) oferece controle de diferentes papéis e

privilégios, *e.g.* é configurável para se ter tipos de usuários que só podem postar rascunhos, não sendo permitido páginas públicas. Outro recurso interessante é o de “sutilezas tipográficas”, no qual o *WordPress* utiliza um recurso para converter, de forma inteligente, caracteres ASCII em entidades tipograficamente corretas em XHTML.

Do ponto de vista técnico, para executar o *WordPress* é necessário ter no mínimo a versão 4.3 do PHP, e a versão 4.1.2 (no mínimo) do servidor de banco de dados MySQL [15]. O servidor Apache [16] ou Nginx [17] são recomendados como servidores mais robustos e com mais recursos para a execução do *WordPress*. Este CMS possui *plugins* como o *Buddypress* [18] e *Mingle* [19] para incorporar funcionalidades de redes sociais na aplicação principal. Uma lista geral de funcionalidades pode ser encontrada em [20], e informações a respeito de acessibilidade no *WordPress* são descritas em [21].

Algumas desvantagens e pontos negativos podem ser observados no *WordPress*, tais como: segurança, performance e mecanismo de busca não muito preciso [22]. Adicionalmente, para o uso efetivo do *WordPress* como um servidor é necessário uma curva de aprendizagem para a equipe técnica, o que significa que não pode ser a melhor opção para usuários com pouca familiaridade na administração de CMSs. Entretanto, há ajuda e apoio disponíveis através da comunidade ativa de desenvolvedores *WordPress*, com numerosos vídeos e tutoriais *online* que facilitam a sua utilização por usuários novatos, assim como para usuários mais experientes.

Em síntese, o *WordPress* é um CMS mais simples, mas interessante; pode ser considerado mais adequado para publicação pessoal de *blogs*. Ele possui um grande número de temas e módulos disponíveis *online* e gratuitos. Dentre as suas vantagens pode-se observar principalmente que é um CMS relativamente fácil de usar, customizar e estender. Sobre as desvantagens, não possui funções mais sofisticadas para redes sociais, comércio eletrônico, fóruns, *wikis*, etc.

Drupal

Este é um CMS robusto e flexível para a construção de praticamente qualquer tipo de *Website*. O *Drupal* é distribuído sob licença GPL, versão 2. O *Drupal* tem alguns conceitos gerais como:

Módulos: é um *software* (código) que estende os recursos do *Drupal* e/ou sua funcionalidade.

Nó: Um nó no *Drupal* é o termo genérico para uma parte do conteúdo na aplicação. Cada nó tem um tipo de conteúdo. Ele também tem uma identificação, um título, uma data de criação, um autor (um usuário), um corpo e algumas outras propriedades. Há suporte para diversos tipos de conteúdo: página básica, artigo, página de *blog*, comentários, fórum, votação, etc.

Taxonomia: O *Drupal* possui um sistema de classificação de conteúdo, que é conhecido como taxonomia e é implementado no módulo de Taxonomia. Pode-se definir vocabulários próprios (grupos de termos para a taxonomia).

Tema: controla como o *site* é exibido, incluindo o seu aspecto gráfico, *layout* e cores.

Regiões, Blocos e Menus: Páginas no *Drupal* são estabelecidas em regiões, que podem incluir cabeçalho, rodapé, barras laterais, e seção de conteúdo principal.

Usuário, permissão e papel: São níveis de permissão para diferentes usuários.

De uma maneira geral, o *Drupal* trata a maioria dos tipos de conteúdo como variações do mesmo conceito, o nó. Estas são páginas estáticas com os diversos tipos de conteúdos possíveis. A estrutura de navegação da aplicação é projetada em separado pela edição de menus, “visões” (listas de conteúdos), e os blocos (conteúdos ao lado que muitas vezes têm links para seções diferentes do site). No *Drupal* os nós contêm a informação estruturada que pertence a um *post* (como o título, conteúdo, autor, data), enquanto o sistema de menus, assim como a taxonomia (identificação dos conteúdos) e as visões, criam a arquitetura de informação. Finalmente, o sistema de temas, juntamente com os módulos de exibição como os painéis controlam como tudo isso aparece visualmente. Uma vez que essas camadas são separadas, pode-se fornecer uma navegação completamente diferente e apresentação dos conteúdos para diferentes usuários com base em necessidades e papéis específicos. As páginas podem ser agrupadas de formas diferentes, priorizadas em uma ordem diferente, e as várias funções e conteúdos também podem ser mostrados ou ocultados conforme a necessidade. O *Drupal*, na versão mais recente ao escrever este relatório, o *Drupal 7*, tem alguns requisitos técnicos, como o Servidor *Web* recomendado - o Apache, a linguagem *PHP*, em sua versão mínima 5.2.4, o servidor de banco de dados como *MySQL* 4.1 ou *PostgreSQL* 7.4.

O *Drupal 7* apresenta algumas melhorias em relação às suas versões anteriores, tais como: i) gerenciamento de campos, na qual se pode definir uma estrutura de conteúdo próprio e adicionar campos personalizados para o conteúdo, os usuários, comentários, *etc.*; ii) as interfaces administrativas foram renovadas e estão mais acessíveis, o que pode tornar as tarefas diárias mais fáceis de encontrar e realizar; iii) a funcionalidade de adicionar imagens ao conteúdo foi incorporada; e iv) houve melhorias relacionadas à base de dados, como uma camada de abstração de banco de dados que visa permitir o uso de diferentes servidores de banco de dados com uma mesma base de código, provendo suporte a diversos servidores de banco de dados.

No *Drupal* há também uma lista de módulos comumente usados, especialmente para se criar funcionalidades de redes sociais (*e.g. friendlist, flagfriend, users' relationships, etc.*). Estes módulos ajudam a construir comunidades e a facilitar relações entre usuários. Além disso, há um grupo de desenvolvedores especialmente dedicado à discussão de ferramentas para redes sociais. O objetivo deste grupo é promover a comunicação entre desenvolvedores de módulos para redes sociais no *Drupal*, discutir estratégias, entre outras ações. Outro aspecto interessante no *Drupal* é a preocupação com usabilidade e acessibilidade [23, 24]. Sendo um CMS com diversos recursos mais complexos, ele pode demandar um custo maior de aprendizagem para entender sua organização interna, seu funcionamento, *etc.*

Em algumas circunstâncias o *Drupal* pode ser particularmente atraente, por exemplo, se for necessário ter uma aplicação com:

- Flexibilidade suficiente para evoluir em qualquer direção, no sentido de novas funcionalidades;

- Facilidade de configuração para interagir com outros *sites* ou com outras tecnologias;
- Facilidade para lidar com fluxos complexos de trabalho;
- Capacidade de criar tipos próprios de conteúdo, *e.g.*, adicionar um campo personalizado a uma página;
- Necessidade de vários módulos adicionais que atendam às necessidades específicas;
- Necessidade de desenvolver rapidamente uma funcionalidade personalizada;
- Possibilidade de gerar aplicações *web* e *mashups* usando *APIs* de terceiros;

De outra maneira, para certos usos limitados, o *Drupal* pode não ser a melhor escolha, como exemplo considere:

- Se o único requisito é o de desenvolver um *blog* pessoal, pode-se avaliar uma plataforma mais especializada em *blogs* como o *WordPress*. Embora o *Drupal* possa servir como uma plataforma para isso, um *software* específico pode ter uma interface de administração mais simples;
- Da mesma maneira, se o único requisito é criar uma *wiki*, deve-se considerar o uso de *software wiki* dedicado como o *MediaWiki* [25]. O *Drupal* pode ser configurado para que qualquer pessoa possa editar o conteúdo (e até mesmo permitir recursos avançados de *wiki* com a ajuda de módulos adicionais), mas pode ser mais simples usar uma solução mais especializada.
- Se o único requisito é possibilitar fóruns de discussão, pode-se considerar um sistema como o *PhpBB* [26], que contém um conjunto maduro de funcionalidades para fórum. No caso de necessitar de um fórum usual o *Drupal* dá suporte perfeitamente. Contudo, módulos de fórum com funcionalidades aprimoradas podem ser necessários para extensão.

Algumas desvantagens podem ser observadas no *Drupal*, como descrito em [22]: i) possui uma alta complexidade, o que o torna inadequado para usuários com pouca familiaridade na administração de CMSs, demandando um alto custo de aprendizagem; ii) o editor "*What You See Is What You Get*" (*WYSIWYG*) não faz parte do núcleo principal do sistema, mas sim é um *plugin* adicional; iii) o *Drupal* também exige um suporte computacional mais poderoso, com máquinas mais potentes; e iv) adicionalmente, alguns usuários acham complicada a interface de administração .

Joomla!

Este é um dos CMS mais usados na *Web* e possibilita construir *Websites* com poderosas aplicações. O *Joomla* é distribuído sob licença GPL versão 2. Adicionalmente, este tem uma das maiores comunidades de desenvolvedores de CMS se comparado principalmente ao *Drupal*. Além de muitos *plugins*, módulos e temas que podem estender e customizar uma instalação básica. O *Joomla* possui muitas funcionalidades e características que o tornam um CMS atrativo, tais como: possibilita uma total customização do *layout* das páginas, menus, *login*, seções, busca, conteúdo em arquivo, interfaces gráficas administrativas fáceis de usar, gerenciamento de conteúdo de usuários, internacionalização (suporte a múltiplas línguas),

gerenciamento de modelos, uma funcionalidade que apresenta quem está *online*, entre outras. Uma característica interessante é que possui suporte à mídia, por meio de uma ferramenta de gerência de mídia, cuja tarefa é gerenciar arquivos de mídia e pastas.

Do ponto de vista técnico, este CMS é desenvolvido utilizando as tecnologias PHP e MySQL e com métodos e padrões de desenvolvimento como: orientação a objeto, tem uma arquitetura do estilo *Model-view-controller* (MVC) e outros padrões de projeto. A clara separação no desenvolvimento entre a lógica e a apresentação ajuda a garantir a reutilização de código e manutenção. Usando o *Joomla*, os desenvolvedores podem aproveitar quase qualquer parte do códigos implementado de acordo com o padrão de projeto *observer* [74]. O *Joomla* possui ainda uma quantidade enorme de *plugins* para extensão, que podem agregar funcionalidades de: fórum, *wiki*, calendário de eventos, *blog*, mensagens privadas, editor de texto rico, adicionar comentários aos conteúdos, etc.

Na versão mais recente, ao escrever este relatório, o Joomla 1.6, apresentou melhorias de funcionalidades tanto para projetistas, desenvolvedores, administradores e *Webmasters*, como:

- Gerenciamento do processo de atualização de extensões de forma mais automática, rápida e fácil;
- Novos estilos de modelo, nos quais com estes os projetistas de interface podem fazer variações do mesmo modelo, que pode ser aplicado a seções específicas ou páginas da aplicação. Inclui também novos modelos disponíveis;
- Escrito em XHTML 1.0 e suporte a HTML 5;
- Editor com código colorido para editar HTML e CSS mais facilmente;
- Os *plugins* possuem um novo formato que têm a mesma estrutura comum de componentes e módulos;
- Um novo e coeso método para o desenvolvimento de componentes. A habilidade para os desenvolvedores “aprenderem pelo exemplo”; ao olharem os componentes do núcleo permite aos desenvolvedores integrarem facilmente as suas idéias ao CMS;
- Construção natural de consultas na base de dados, o que permite aos desenvolvedores criarem consultas programaticamente (em tempo de execução com parâmetros).

Dentre as vantagens deste CMS, pode-se citar que há um bom suporte de documentação como: guia de instalação, guia de início rápido, tutoriais, *wikis*, fórum de discussão de desenvolvedores, uma grande quantidade de desenvolvedores com uma comunidade ativa. Além disso, possui padrões profissionais e uma grande variedade e flexibilidade para customização. A respeito das desvantagens do Joomla, a principal é que há muitos produtos comerciais construídos sobre um desenvolvimento de código aberto. Muitos dos *plugins*, temas e serviços são pagos. Foram encontrados *plugins* para incorporar recursos e funcionalidades de redes sociais para no *Joomla*, e.g., *JoomlaPolis* [27] e o *JomSocial* [28], contudo estes *plugins* são comerciais. Outras desvantagens são apontadas em [22], tais como fraqueza na concepção do sistema e nenhuma prova da longevidade (continuação). Pelo motivo de sua complexidade, também há certo custo de aprendizagem para conseguir

dominar o desenvolvimento com este CMS. Também são apontados problemas a respeito do desempenho deste CMS [22].

Plone

O *Plone* é outro CMS robusto, livre e de código aberto. Este é desenvolvido com a linguagem de programação *Python* e executado sobre o servidor de aplicação *Zope*. O *Plone* é liberado sob a licença GPL versão 2 e é projetado para ser extensível. Este CMS dá suporte a vários tipos de conteúdo, incluindo textos, fotos e imagens. Estes podem existir em diversas formas como: documentos, notícias, eventos, vídeos, arquivos de áudio, ou quaisquer tipos de arquivos e dados que podem ser carregados. Pode-se criar pastas para armazenar conteúdo e criar uma estrutura de navegação. Dentre as funcionalidades básicas, o *Plone* oferece diversas [4], como:

- Possibilidade de criação de portais de *Internet* ou *Intranet*;
- Criação e manutenção de conteúdo realizado através de interface pelo navegador *Web*;
- Repositório de conteúdos como imagens, documentos e vídeos;
- Gerenciamento de usuários com níveis hierárquicos;
- Fluxo de publicação colaborativa de conteúdo (*workflow*);
- Geração automática de infra-estrutura de navegação;
- Criação de blog corporativo;
- Agendamento e divulgação de eventos;
- Galeria de imagens;
- Área integrada de notícias;
- Segurança e integridade dos dados.

O *Plone* é construído principalmente sobre o conceito de ações, que se referem a funcionalidades do usuário final associadas com objetos ou conteúdos. Por exemplo, ver, editar e compartilhar são tipos de ações; o mapa do site é uma ação, o formulário de contato também, etc. Estas ações são gerenciadas por “*portal_actions*” para ações genéricas, ou “*portal_types*” para visão, edição e todas as ações que estão ligadas a um determinado tipo de conteúdo e podem variar conforme o tipo. Como características gerais, usabilidade e acessibilidade são dois pontos principais deste CMS. Outros pontos fortes do *Plone* incluem seu fluxo de trabalho flexível e adaptável, a sua grande escalabilidade e segurança, ou seja, o suporte a diversos *sites* hospedados no servidor de instalação e a segurança dos dados e de acesso que este possibilita [4]. Outros pontos básicos incluem:

- Ampla gama de *plugins*;
- Adapta-se a qualquer sistema operacional;
- Apoio disponível na forma de uma grande equipe de desenvolvimento internacional e membros da comunidade *Plone*;
- Facilidade de instalação e uso;

- Motor de busca amigável, que inclui a opção de busca refinada (Mecanismo de busca avançada);
- Capaz de gerenciar conteúdo multilíngüe (Suporte a internacionalização);
- Funciona igualmente bem como uma ferramenta de *groupware* (Ferramentas colaborativas para *groupware*) e também um sistema de publicação de documentos.

A versão mais recente ao escrever este relatório, o *Plone 4*, inclui principalmente novas características e funcionalidades, tais como: i) em geral ele é cerca de 50% mais rápido do que o *Plone 3*, com grandes melhorias tanto para usuários anônimos como para registrados; ii) o *Plone 4* apresenta um tema padrão totalmente novo chamado de “*Sunburst*” que é mais elegante, minimalista, moderno e fácil de personalizar; iii) a interface de usuário foi “polida” e refinada para se tornar mais intuitiva; e iv) novas facilidades para a instalação e atualização de *plugins* foram disponibilizadas.

O *Plone*, com sua grande comunidade, possui documentos para desenvolvedores que falam de modelos, das tecnologias, entre outros tópicos de maneira bem organizada [29], além também de conter um manual do usuário bem completo [30]. Suas desvantagens [22] incluem principalmente o desempenho baixo quando o usuário está *logado*; o uso local é muitas vezes mais rápido quando o usuário não está conectado. Além disso, tem sido relatado que as eventuais alterações que o usuário faz parecem não ocorrer no sistema, o que pode ser devido ao armazenamento da página em *cache* [22]. Relacionado à funcionalidade para redes sociais, não foi encontrado para o *Plone* um *plugin* específico com funcionalidades nesta direção, apenas uma iniciativa tem sido cogitada [*e.g.* 31].

Comparação e Discussão sobre os CMSs

Uma comparação destes CMSs em diversos aspectos foi realizada usando o *CMSMatrix* [7]. Aspectos de requisitos de sistema, facilidade de uso, gerenciamento, flexibilidade e aplicações foram observados. É importante ressaltar que, nem todas as versões comparadas dos CMSs no *CMSMatrix* são as mais recentes.

Requisitos de sistema: Conforme ilustra a Figura 1, é possível observar que não há diferenças marcantes entre os CMSs neste quesito. A linguagem PHP assim como o servidor *Apache*, e o servidor de banco de dados MySQL são mais comumente usados entre eles. É interessante ressaltar que nas versões mais novas destes CMSs, outras e diversificadas opções de servidores de aplicações são suportados, o que parece ser uma tendência entre os CMSs.

System Requirements	WordPress 3.0.4	Plone 3.0	Joomla! 1.6.0	Drupal 6.10
<i>Application Server</i>	blank	Zope	CGI	Apache
<i>Approximate Cost</i>	Free	Free	Free	Free
<i>Database</i>	MySQL	Other	MySQL	MySQL
<i>License</i>	Open Source	Open Source	Open Source	Open Source
<i>Operating System</i>	Platform Independent	Platform Independent	Platform Independent	Platform Independent
<i>Programming Language</i>	PHP	Python	PHP	PHP
<i>Root Access</i>	No	No	No	No
<i>Shell Access</i>	No	Yes	No	No
<i>Web Server</i>	blank	Apache	Any	Apache

Figura 1: Comparação dos CMS sobre requisitos [7]

Facilidade de Uso: É possível observar (Figura 2) que a maioria das funcionalidades já faz parte do núcleo dos CMSs, ou é possível de ser adicionada gratuitamente. Pode-se destacar que o *Joomla* dá suporte à prototipação, mas não permite desfazer modificações, enquanto o *Drupal* não tem diversos níveis de interface de usuário, mas possui, mesmo que limitado, um assistente de configuração do site enquanto os outros não têm.

Ease of Use	WordPress 3.0.4	Plone 3.0	Joomla! 1.6.0	Drupal 6.10
<i>Drag-N-Drop Content</i>	Yes	Yes	No	Free Add On
<i>Email To Discussion</i>	Limited	Free Add On	Free Add On	Free Add On
<i>Friendly URLs</i>	Yes	Yes	Yes	Yes
<i>Image Resizing</i>	Yes	Yes	Yes	Free Add On
<i>Macro Language</i>	Free Add On	Yes	Yes	Free Add On
<i>Mass Upload</i>	Yes	Yes	Yes	Free Add On
<i>Prototyping</i>	Free Add On	Yes	Yes	Limited
<i>Server Page Language</i>	Yes	Yes	Yes	Yes
<i>Site Setup Wizard</i>	No	No	No	Limited
<i>Spell Checker</i>	Yes	Free Add On	Free Add On	Free Add On
<i>Style Wizard</i>	No	Free Add On	No	Limited
<i>Subscriptions</i>	Free Add On	Yes	Yes	Free Add On
<i>Template Language</i>	No	Yes	Yes	Limited
<i>UI Levels</i>	Yes	Yes	Yes	No
<i>Undo</i>	Limited	Yes	No	Limited
<i>WYSIWYG Editor</i>	Yes	Yes	Yes	Free Add On
<i>Zip Archives</i>	Free Add On	Free Add On	No	No

Figura 2: Comparação dos CMS sobre facilidade de uso [7]

Gerenciamento: O *Plone* possui uma funcionalidade relacionada a fluxo de trabalho, enquanto os outros são limitados ou não tem este quesito. O objetivo do fluxo de trabalho no *software* é facilitar e controlar as atividades de trabalho.

O *Plone* também tem uma área de transferência que tem como objetivo fornecer uma base para a ação de vários adaptadores para manipular ou usar as seleções da área de transferência. O *Drupal* e *Plone* têm uma funcionalidade que permite o conteúdo ter estágios, e apenas ser público quando permitido pelo autor (Figura 3).

Management	WordPress 3.0.4	Plone 3.0	Joomla! 1.6.0	Drupal 6.10
<i>Advertising Management</i>	No	Free Add On	Yes	Free Add On
<i>Asset Management</i>	Yes	Yes	Yes	Yes
<i>Clipboard</i>	No	Yes	No	No
<i>Content Scheduling</i>	Limited	Yes	Yes	Free Add On
<i>Content Staging</i>	No	Free Add On	No	Free Add On
<i>Inline Administration</i>	Free Add On	Yes	Yes	Yes
<i>Online Administration</i>	Yes	Yes	Yes	Yes
<i>Package Deployment</i>	No	Yes	No	No
<i>Sub-sites / Roots</i>	Yes	Yes	Yes	Yes
<i>Themes / Skins</i>	Yes	Yes	Yes	Yes
<i>Trash</i>	Yes	Free Add On	Yes	No
<i>Web Statistics</i>	Free Add On	Free Add On	Yes	Yes
<i>Web-based Style/Template Management</i>	Yes	Yes	Yes	Yes
<i>Web-based Translation Management</i>	Limited	Yes	Free Add On	Yes
<i>Workflow Engine</i>	No	Yes	No	Limited

Figura 3: Comparação dos CMS sobre gerenciamento [7]

Flexibilidade: Pode-se observar na Figura 4 que o *WordPress* é menos flexível que os outros CMSs, pois oferece menos recursos neste quesito, porém, funcionalidades como *deployment* de múltiplos sites, reuso de conteúdo e CGI são de baixa prioridade para o contexto do projeto TNR.

Flexibility	WordPress 3.0.4	Plone 3.0	Joomla! 1.6.0	Drupal 6.10
<i>CGI-mode Support</i>	No	Yes	Yes	Yes
<i>Content Reuse</i>	No	Yes	Yes	Limited
<i>Extensible User Profiles</i>	Free Add On	Yes	Yes	Yes
<i>Interface Localization</i>	Yes	Yes	Yes	Yes
<i>Metadata</i>	Yes	Yes	Yes	Yes
<i>Multi-lingual Content</i>	Free Add On	Yes	Free Add On	Yes
<i>Multi-lingual Content Integration</i>	Free Add On	Yes	Free Add On	Free Add On
<i>Multi-Site Deployment</i>	No	Yes	Free Add On	Yes
<i>URL Rewriting</i>	Yes	Yes	Yes	Yes

Figura 4: Comparação dos CMS sobre flexibilidade [7]

Aplicações: No aspecto de aplicações (Figura 5), todos os CMSs possuem por padrão algumas funcionalidades importantes, *e.g.* mecanismo de busca, ou possibilitam acrescentar outros aplicativos, como: *blog*, *bate-papo*, *etc.* É importante ressaltar que nestes CMSs analisados, há centenas de outros aplicativos (*plugins*) que permitem adicionar novas funcionalidades específicas.

Built-in Applications	WordPress 3.0.4	Plone 3.0	Joomla! 1.6.0	Drupal 6.10
<i>Blog</i>	Yes	Yes	Yes	Yes
<i>Chat</i>	Free Add On	Free Add On	Free Add On	Free Add On
<i>Data Entry</i>	Free Add On	Free Add On	Free Add On	Free Add On
<i>Discussion / Forum</i>	Free Add On	Free Add On	Free Add On	Yes
<i>Events Calendar</i>	Free Add On	Yes	Free Add On	Free Add On
<i>Groupware</i>	No	Free Add On	Free Add On	Free Add On
<i>Photo Gallery</i>	Yes	Yes	Free Add On	Free Add On
<i>Search Engine</i>	Yes	Yes	Yes	Yes
<i>User Contributions</i>	Yes	Yes	Yes	Yes
<i>Weather</i>	Free Add On	Free Add On	Free Add On	Free Add On
<i>Wiki</i>	Free Add On	Free Add On	Free Add On	Free Add On

Figura 5: Comparação dos CMS sobre aplicações [7]

Uma breve comparação entre estes CMSs é apontada em [32]. O *Joomla* oferece ao mesmo tempo poder e simplicidade aos usuários/desenvolvedores. A sua “espinha dorsal” é desenvolvida de uma forma a requisitar um mínimo de conhecimento técnico no seu uso para a efetiva configuração e manutenção do *Website*. Assim como o *Joomla*, o *Drupal* também foi desenvolvido utilizando PHP e oferece funcionalidades comuns ao *Joomla* e outras mais sofisticadas. Todavia, ele exige um pouco mais de conhecimento técnico do desenvolvedor para sua manipulação e customização. Para pessoas que não possuem um perfil técnico, pode demorar mais para dominar o *Drupal* do que o *Joomla*. Em compensação, encontra-se muito mais *plugins* não pagos para o *Drupal* do que para o *Joomla*. Já o *Plone* é apoiado por uma programação bem planejada, com uma plataforma poderosa e madura para o desenvolvimento de sistemas complexos. É relativamente fácil de usar, mas a sua aprendizagem pode demorar mais tempo em comparação com o *Joomla* e o *Drupal*. Adicionalmente, a sua linguagem de programação se mostra mais complexa.

Pode-se apontar alguns fatos sobre estes CMSs [32]:

- O *Joomla* é projetado para funcionar em ambientes compartilhados de hospedagem na *Web*, assim como o *Drupal*;
- O *Joomla* possui um instalador simples e mais fácil que dos outros;
- O *Drupal* tem ferramentas de edição de conteúdo poderosas tanto para os usuários comuns quanto para os desenvolvedores na criação de *Websites* sem se preocupar profundamente com a codificação;
- Similar ao *Joomla*, no *Drupal* as extensões são simples de entender para usuários não técnicos.
- Os *plugins* do *Drupal*, embora profundamente integrados, são menos poderosos em relação ao *Joomla* ou o *Plone*;
- O *Plone* possui requisitos de hospedagem especiais;
- O *Plone* desenvolve *Websites* sofisticados, mas pode não ser muito fácil de manusear por usuários pela primeira vez;
- O *Plone* não é tão adequado para projetos de pequeno porte, que podem ser bem atendidos e manuseados facilmente pelo *Joomla* ou o *Drupal*, até mesmo pelo *WordPress*, no caso de foco em *blogs*;
- A configuração do *Plone* é complexa e é feita em camadas diferentes, o que exige uma compreensão da estrutura de organização como um todo;
- O *Plone* é escrito usando a linguagem *Python*, que é uma poderosa linguagem, mas menos utilizada do que o PHP. Isso torna a criação de extensões e customizações para o *Plone* mais difícil, o que gera a necessidade de equipes mais especializadas.

3.2 SN-CMSs

Elgg [37]: Este é um *framework* para o desenvolvimento de redes sociais de código aberto sobre uma licença dual (GPL versão 2 e MIT). Ele fornece a funcionalidade necessária para permitir que se execute um *Website* de rede social. A maioria das funcionalidades da aplicação do usuário final em *Elgg* vem de *plugins*. O pacote completo do *Elgg* (gratuito) vem com uma seleção de *plugins* pré-instalados. Ele usa um *plugin* API para criar e adicionar outros recursos necessários. Funcionalidades interessantes relacionadas a redes sociais incluem: grafo social (relações entre usuários e demais usuários, objetos e *sites*), avatar, amigos, grupos, perfil, *dashboard*, *widgets*, mensagens privadas, etc. Sua principal desvantagem é que mesmo sendo um produto de código aberto, há diversas iniciativas comerciais para *plugins* e extensões, além de ser cobrada o suporte.

Pligg [33]: Este é um CMS de fonte aberta sobre licença GPL que fornece funcionalidades nas quais os usuários registrados podem adicionar conteúdos e se conectar com outros usuários. Há uma funcionalidade em que histórias podem ser criadas e votadas pelos membros. Ele provê funcionalidade de comunidades, como a criação de grupos, que permite aos usuários compartilharem artigos com outros membros que compartilham um interesse comum em um tópico. Ele prevê também o envio de mensagens privadas para efetuar a

comunicação entre os membros, e um sistema avançado de comentários, no qual não apenas se comenta um conteúdo, mas uma nota é dada ao usuário. Isso afeta a maneira como os comentários são exibidos, colocando comentários de alta qualidade acima dos demais. Também disponibiliza recursos de edição de perfil do usuário. *Pligg* é um clone da aplicação *Digg* [34], e fornece principalmente funcionalidades para *Websites* de *bookmarking* social. Entre as desvantagens pode-se citar a comunidade de desenvolvedores pelo suporte limitado, e o fato dos módulos e *templates* adicionais serem pagos.

Hotaru CMS [43]: Este é um CMS de código aberto, uma plataforma desenvolvida em PHP para criar *Websites*. O uso mais popular para Hotaru CMS é um estilo *Digg* de *Website* para *bookmarking* social. Este utiliza um *plugin bookmarking* como base, e inclui votação e comentários. Ao adicionar *plugins* adicionais, pode-se incluir funcionalidades de rede social, como mensagens, seguidores, etc. O *Hotaru* possui características semelhantes a de muitos outros CMS, como ser customizável, baseado em *plugins* para extensão, etc. O CMS *Hotaru* pode ser usado sob os termos da GPL versão 3.

LiveStreetCMS [35]: é uma plataforma livre para o desenvolvimento de *blogs* e redes sociais. Dentre as funcionalidades mais diferenciadas percebidas pode-se apontar um sistema de votação e *ranking* para *blogs*, tópicos, comentários e usuários. O projeto tem sido implementado no paradigma de orientação a objeto, em PHP5 com MySQL5, e é construído sobre o princípio do modelo MVC, o que torna possível alterar a lógica da aplicação e a aparência com facilidade. Módulos e *templates* adicionais são pagos.

Liferay [41]: Essa é uma das plataformas de *software* de código aberto de CMS disponíveis para redes sociais, e também para o desenvolvimento de portais, *intranets*, *extranets* corporativas, fóruns e publicação na *Web* em geral. O *Liferay* é composto de duas seções principais: o “*journal*” e a solução de “colaboração”. O primeiro é o sistema de gerenciamento de conteúdo do portal, permitindo aos editores publicar conteúdo, gerenciar páginas, criar menus hierárquicos e a estrutura do *Website* em geral. O segundo é o aspecto de rede social do portal, que inclui fóruns, colaboração, *wiki*, *blogs*, *email*, calendários e *tagging*. Ambos podem ser estendidos por “*portlets*” de código aberto, que adicionam mais funcionalidades ao sistema. O *Liferay* possui licença MIT. Entre suas principais vantagens pode-se citar principalmente os aspectos de colaboração, com fórum, *blogs*, *wikis*, calendário, mensagens instantâneas. Contudo, mesmo dizendo que possibilita uma lista dinâmica de amigos, a formação de grupos ou comunidades não é comentada. Entre as desvantagens, a solução está mais intimamente ligada a uma visão comercial, corporativa do que a maioria dos projetos de código aberto. Do ponto de vista técnico, esta solução mistura um conjunto grande de tecnologias e linguagens de programação, de *scripts* e também padrões. Ele é compatível com diversos sistemas operacionais, servidores de aplicação e banco de dados.

SocialCMS [42]: Esse é um *software* para o desenvolvimento de *Websites*. É construído em plataforma PHP/MySQL/Linux. A versão de código aberto inclui apenas funcionalidades básicas de um CMS. A versão empresarial inclui diversas melhorias como *plugins*, gerenciamento de *banner*, gerenciamento de participantes, documentação *online*, etc. Apenas

nesta versão as funcionalidades a respeito de redes sociais são incluídas. Há também uma variada opção de *plugins* que fazem a integração com aplicações como: *Youtube*, *Twitter*, *Amazon*, etc. A principal desvantagem desta solução é que ela está mais intimamente ligada a uma visão comercial, corporativa do que a maioria dos projetos de código aberto.

Anahita Social Engine [36]: Esta é uma plataforma de fonte aberta e um *framework* para o desenvolvimento de serviços de redes sociais. Em Anahita tudo e todos são representados como nós. Isso significa que pessoas, grupos, eventos, *posts* em *blogs*, tópicos de discussão, fotos, álbuns entre outros são todos nós. A relação entre os nós é mantida por grafos. Esse é construído sobre a plataforma *Joomla* junto a outros *frameworks*. As suas funcionalidades mais interessantes são as essenciais de rede social, que incluem perfis de pessoas, *feeds*, gráficos sociais flexíveis, gestão de privacidade, etc. As atividades de um nó de ator são compiladas como um histórico de lista de itens; os nós podem receber comentários, e pessoas são habilitadas a bloquear pessoas indesejadas, etc.

Clonesumating [40]: Esse é uma plataforma de rede social desenvolvido com base no “consumating.com”; *i.e.* uma versão de fonte aberta do código que o executa. Essa plataforma possui funcionalidades de rede social, incluindo perfis de usuário, *tagging* de usuário, correspondência e descoberta com base em combinações de *tags* peculiares, atividades em grupo, tais como concursos semanais de fotos, calendário de eventos, RSS, etc. Este software é escrito principalmente em Perl.

Applesed [38]: Esse é outra plataforma de rede social distribuído e descentralizado de código aberto. Se você é um usuário de uma rede social *Appleseed*, você pode conectar-se com qualquer usuário em qualquer outro *Website Appleseed*, enviar mensagens, compartilhar mídias e juntar-se a discussões. Semelhante à forma como funciona o *email*, redes sociais distribuídas não se limitam a apenas uma aplicação específica. Pode-se mudar para uma nova plataforma e manter os seus amigos. Do ponto de vista técnico, o *framework* social *Appleseed* é um componente baseado em estrutura MVC em PHP. É possível introduzir temas e é projetado para agrupar componentes provenientes de vários servidores em uma mesma página. Os desenvolvedores podem usar a estrutura para a construção do lado do servidor de plataformas sociais que podem se conectar e ampliar o grafo social do *Appleseed*. Há uma documentação básica disponível para usuários e desenvolvedores.

OxWall [45]: Este uma plataforma de rede social de código fonte aberto. Foi desenvolvido usando PHP/MySQL para criar comunidades *online*. Este também possui diversos *plugins* que possibilitam flexibilizar e estender a maneira como ele funciona. Estes *plugins* são unidades completas de funcionalidade que podem ser utilizadas para diversos fins. Além de funcionalidades características de CMS, como gerenciamento de pessoas e conteúdo, moderação, gerenciamento de permissões, o *OxWall* tem funcionalidades que suportam compartilhamento de fotos e vídeos, *blogs*, fóruns, *wiki*, grupos, eventos, amigos, comentários, *tagging*, avaliação de conteúdo, customização de perfis, e outras funcionalidades como bate-papo, que podem ser adicionadas através de *plugins*. Muitos destes *plugins* estão sob licença BSD. O *OxWall* tem um exemplo de serviço de hospedagem, que é uma hospedagem gratuita de criador de redes sociais, a *wall.fm*.

PHP Social [46]: Essa é uma aplicação de rede social de código aberto. Além de funcionalidades básicas de um CMS, oferece suporte à internacionalização, mecanismos de busca com *URLs* amigáveis, verificação de usuário, entre outros. A respeito de funcionalidades específicas para redes sociais, a aplicação PHPSocial tem algumas interessantes como: níveis de requisição de amigos, tipos de amizade, privacidade de alguns campos, níveis de privacidade, separação de grupos, de redes em sub-redes. Dentre os pontos negativos da aplicação está a falta de documentação, poucos *plugins* e estes são pagos.

Social Network Server [44]: É uma plataforma específica para desenvolver aplicações de redes sociais, incluindo inclusive funcionalidades relacionadas a mídias, como *download* e *upload* de música, fotos, *etc.* A plataforma foi criada para duas versões do *Website*, uma para a *Web*, e outra para celulares. Este sistema permite lançar uma rede social própria sem ter que desenvolver qualquer programação e pouca customização. As principais funcionalidades não são tão diferentes das de outras plataformas, e ainda possui deficiências básicas como a falta de suporte a outras línguas além do inglês. Além disso, a plataforma ainda não está disponível para licenciamento/utilização e, conforme indicado, o uso deve envolver um custo.

Spree [39]: Esse visa criar uma comunidade ativa *online*, na qual cada usuário pode fazer perguntas ou atuar como autoridade, ajudando os companheiros. Nesse, todos participantes compartilham conhecimento e se beneficiam a partir do conhecimento dos outros usuários. *Spree* pode ser considerado um motor de busca social, na qual os usuários fazem perguntas para encontrar outros usuários participantes, que tenham conhecimentos na área e estejam dispostos a ajudar. Ele identifica os especialistas mais qualificados para uma determinada questão de forma automática e em tempo real, facilitando a comunicação em tempo real entre usuários e especialistas através de uma interface de bate-papo. É uma maneira de interagir com colegas de profissão e obter respostas e conselhos de fontes conhecedoras. Esse sistema fornece um portal para comunidade *Web* para a procura especializada automatizada de autoridades. As principais funcionalidades incluem, por exemplo: a classificação automática de perguntas em linguagem natural, a procura de especialistas, bate-papo, histórico de interação, gerenciamento de usuários, *etc.* Quando se faz uma pergunta em linguagem natural usando o *Spree*, este a compara com as pessoas que são bem conceituadas sobre o assunto e as recomenda. As perguntas de usuário automaticamente são encaminhadas para outros usuários competentes que estão dispostos a ajudar. Um protótipo foi lançado sob a licença GNU *Affero General Public License* (AGPL). Assim, pode-se criar uma instância de uma própria comunidade de autoridades, ou incluir esta tecnologia a outros projetos de *software*.

Insoshi [47]: Uma plataforma de rede social de fonte aberta desenvolvida em *Ruby on Rails* [64]. O código fonte é livre e disponível sob a licença MIT. Há poucas informações disponíveis sobre este sistema.

Além destes, na mesma categoria, há diversos outros. Alguns já apontados como descontinuados, como exemplo *LovdByLess* [48], *PHPizabi* [49], *Barnraiser* [50]. Além disso,

há diversas opções comerciais como: *WebNetwork* [51], *Kootali* [52], *Boonex* [53], *Blogtronix* [54], *Sparta* [56], *Web Crossing* [56], *JCow* [57], *Sharetronix* [58], *Social Engine* [59], *People Agregator* [60]. Alguns *plugins* para funcionalidades de rede social para outras plataformas também foram encontrados como: *Yogurt* [61], para o CMS *Xoops* [62], *Community Engine* [63], para aplicações *Ruby on Rails*; e scripts para criar redes sociais privadas, e.g. *MyMates* [65].

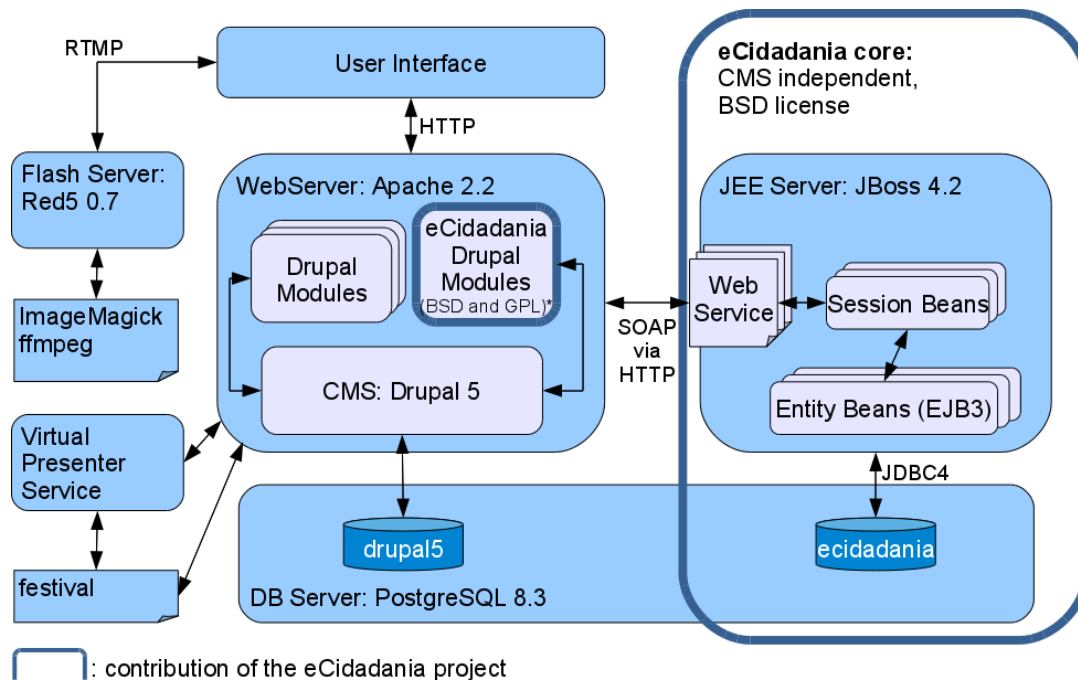
Uma análise articulada dessas plataformas face as outras opções aqui levantadas são sintetizadas e discutidas na seção 4.

3.3 *VilanaRede*

O sistema de RSI *VilanaRede* foi desenvolvido no contexto do projeto e-Cidadania [66]. O *VilanaRede* é uma RSI construída com, por e para a comunidade por meio de oficinas semio-participativas que envolveram representantes da comunidade alvo na Vila União, em Campinas, SP. Os usuários no *VilanaRede* anunciam produtos, serviços e idéias. Diversas funcionalidades foram desenvolvidas para contemplar os requisitos de uma RSI como: mecanismos de meta-comunicação para auxiliar na ajuda com o sistema; apresentador virtual para auxiliar com leitura de anúncios; uso de interfaces flexíveis para ajustar a interface a diferentes necessidades de usuários; uso de multimídia transversal ao sistema, *i.e.* as pessoas podem fazer anúncios, colaborar e comentar anúncios usando áudio ou vídeo; uso de recursos de acessibilidade, como ajuste do tamanho e contraste da tela; mecanismos de *awareness* que habilitam os usuários a saber quem está *online*, iniciar conversas contextualizadas e ver grafo de conversas entre os usuários; mecanismo de busca semântica, entre outros recursos.

Do ponto de vista técnico, a RSI *VilanaRede* usa o CMS *Drupal* 5 que é executado em um servidor *Apache* com um servidor de banco de dados *PostgreSQL*. Este CMS é utilizado para gerenciar aspectos básicos e padrões do sistema como: conta de usuários, tipos de conteúdo e outras funcionalidades gerais. Todas as outras funcionalidades, que vão além ou que não são providas pelo CMS de maneira a satisfazer os requisitos definidos no projeto, são providas pelo “e-cidadania Core” como serviços *Web* que rodam em um servidor JEE. O servidor de aplicação *Jboss* 4.2 gerencia funcionalidades específicas da RSI *VilanaRede* como as mencionadas anteriormente. A Figura 6 apresenta uma perspectiva técnica simplificada da arquitetura do sistema. Uma perspectiva conceitual é descrita em Baranauskas *et al.* [69].

Os serviços *Web* providos pelo núcleo do projeto e-Cidadania são acessados via *Simple Object Access Protocol* (SOAP). Estes serviços são independentes do CMS e publicados com licença BSD. Para acessá-los, um módulo e PHP *scriptlets* foi implementado no lado do CMS. O sistema também utiliza outros recursos para multimídia como *Red5* e *ImageMagick*.



* because of the Drupal license, few parts of the eCidadania Drupal extensions had to be published under the GPL license

Figura 6: Perspectiva técnica da arquitetura do sistema [67]

Visando encontrar respostas mais exatas sobre a arquitetura e as tecnologias envolvidas no *VilanaRede* e uma opinião sobre as decisões de projeto do *software* desenvolvido, uma entrevista foi conduzida com os principais desenvolvedores envolvidos nas decisões técnicas e no desenvolvimento do sistema. Nesta entrevista, os quatro principais desenvolvedores foram convidados a responder algumas questões sobre os aspectos técnicos do *VilanaRede*. As perguntas elaboradas foram:

1. Você acha que seria possível replicar a implementação do *VilanaRede* com suas características arquiteturais facilmente para outro sistema de RSI? Quais principais dificuldades estariam envolvidas?
2. Você acha que o uso do CMS *Drupal* foi uma boa opção de projeto?
3. Você usaria a mesma arquitetura caso não existisse restrições a respeito de licença? Por quê?
4. Qual(is) aspecto(s) você acha que poderia(m) melhorar no projeto do *VilanaRede* para viabilizar extensões de novas funcionalidades de *software* mais rapidamente e com menos custo em tempo/aprendizagem?

A respeito da questão 1, os desenvolvedores apontaram que seria possível efetuar uma adaptação do sistema e reaproveitar o código, mas indicaram algumas restrições como: conhecer bem a arquitetura e principalmente o esquema de trabalho do *Drupal*; outro

aspecto é que as tecnologias envolvidas podem dificultar a participação de pessoas no desenvolvimento, uma vez que requerem conhecimento significativo delas. Atualizar as tecnologias envolvidas foi outro ponto levantado, como exemplo efetuar a migração para uso do *Drupal* 7 e *JBoss* 6, que poderia requerer mais recursos em termos de tempo. Um dos desenvolvedores descreveu: “O *VilanaRede*, apesar da preocupação com a modularização de funcionalidades específicas da rede social, tem sua implementação bastante amarrada no código. Isso implica que, para criar uma instância da arquitetura seria necessário a recodificação do sistema (não há parametrização suficiente para instanciar sem recodificação)”. Além de um custo alto, dificuldades envolveriam a recodificação de código em diversas linguagens.

Sobre a segunda questão, de maneira unânime, os desenvolvedores apontaram o uso do *Drupal* como uma ótima opção, devido principalmente à sua completude e por prover a funcionalidade básica do sistema. O seu potencial, por ter uma comunidade de desenvolvimento ativa com suporte significativamente melhor que outros CMSs, além de uma boa documentação e também por ser relativamente fácil de aprender (usa linguagem PHP) são outros pontos favoráveis.

Sobre a terceira questão, há argumentos prós e contras a respeito da arquitetura. Há opiniões que indicam aspectos positivos, que a arquitetura é robusta e permite adicionar novos componentes de um jeito fácil e também porque facilita o desenvolvimento de soluções em diferentes linguagens (isso pode aumentar a participação de outros desenvolvedores no projeto). Isso também pode garantir uma certa independência do CMS, o que facilita a migração entre 2 CMSs ou 2 versões do mesmo CMS, *e.g.*, a atualização do *Drupal* de 5 para 7 provavelmente seria mais difícil em uma outra arquitetura. Por outro lado, um dos desenvolvedores apontou que tal característica arquitetural pode dificultar a contratação de desenvolvedores terceirizados, a manutenção é dificultada, e o fato de ser fácil de aprender as linguagens e de atrair outros desenvolvedores para a equipe se mostrou menos evidente que o esperado.

Sobre a quarta questão, os desenvolvedores indicaram aspectos que poderiam melhorar no projeto do *VilanaRede* para viabilizar extensões de novas funcionalidades de *software* mais rapidamente e com menos custo em tempo/aprendizagem. Eles apontaram principalmente: (1) pode-se melhorar primeiramente uma qualificação e estudo do CMS para que toda a equipe de desenvolvimento conheça a fundo todas as características da aplicação, para que eles aproveitem bem suas funções; (2) prospectar uma arquitetura mais simples que possa gerar incrementos contínuos e menores, além de respeitar a arquitetura do desenvolvimento (desenvolver e testar localmente, teste de integração no sistema, efetuar *deployment* apenas de versões testadas de código); (3) outro fator interessante apontado é a necessidade de se realmente fazer desenvolvimento ágil, contudo isso requer participação maior dos desenvolvedores.

De maneira geral, os desenvolvedores comentaram que, no *VilanaRede*, é exigido grande conhecimento do desenvolvedor para entender quais funcionalidades do *VilanaRede* são do CMS, do Java, do PHP, ou de outras tecnologias. Citaram que o ideal seria uma redução das tecnologias utilizadas e uma clara identificação da procedência de cada recurso do sistema e dos passos para a manutenção. Fundamentado nas características funcionais, não funcionais

e técnicas do *VilanaRede*, além da entrevista com os desenvolvedores, uma sumarização das principais vantagens e desvantagens identificadas foi sintetizada na Tabela 1.

Tabela 1: Sumarização da Análise sobre o *VilanaRede*

Vantagens	Desvantagens
Sistema foi projetado para ser uma RSI, considerando aspectos necessários e úteis para uma aplicação desta natureza	Muitas tecnologias envolvidas na aplicação
Amplas funcionalidades interessantes para uma RSI	Requer a aprendizagem de diversas tecnologias pelos desenvolvedores
Arquitetura permite adicionar novas funcionalidades independentemente de CMSs	Dificuldades para a manutenção e identificação das funcionalidades nos códigos
Foco em inclusão e acessibilidade com funcionalidades projetadas para que faça sentido à comunidade de usuários	Não há uma comunidade de desenvolvimento dedicada

3.4 *LeMill*

O *LeMill* [9] é uma comunidade *Web* para procurar, criar e compartilhar recursos de aprendizagem. Essa plataforma foi desenvolvida pelo Grupo de Pesquisa em Ambientes de Aprendizagem [70] do Laboratório de Mídias da Universidade Helsinki de Artes e *Design*, na Finlândia. O *LeMill* teve seu desenvolvimento iniciado no contexto do projeto *CALIBRATE* [71]. O *LeMill* pode ser usado principalmente por pessoas interessadas em recursos de aprendizagem para encontrar principalmente recursos, técnicas pedagógicas, ferramentas e dicas para o uso que encorajem abordagens de aprendizagem colaborativa. Os usuários organizados em comunidades são principalmente encorajados com ferramentas de suporte para usar, criar e combinar materiais de aprendizagem. Além disso, ele é usado para compartilhar experiências e criar recursos de aprendizagem colaborativamente, propiciar desenvolvimento profissional e meta-aprendizagem.

As funcionalidades [72] do *LeMill* permitem às pessoas principalmente:

- encontrarem conteúdos abertos para enriquecerem seus materiais de aprendizagem;
- navegarem por materiais didáticos destinados a áreas de estudo, ensino de língua e faixa etária do estudante;
- navegarem por atividades pedagogicamente significativas com relação à teoria de fundamento;

- terem contato com outros professores que ensinam a mesma área temática, ou na mesma língua, ou da mesma faixa etária;
- fazerem coleção de materiais bons ou promissores e atividades;
- criarem planos de aula com todas as descrições do conteúdo e da atividade de que necessitam para uma aula ou curso;
- usarem material pedagógico em seus próprios ambientes virtuais de aprendizagem;
- compartilharem experiências que tem tido com alguns conteúdos de aprendizagem e atividades;
- criarem e compartilharem material de aprendizado a partir do material que eles têm em seus próprios computadores, e também construídos a partir de pedaços de conteúdo de aprendizagem abertos e disponíveis;
- traduziream e melhorarem materiais feitos por outros em outras línguas;
- criarem novos materiais baseados em material feito por outros;
- criarem e se juntarem em grupos em certos temas;
- seguirem e comentarem atividades de grupo;

Os principais conceitos e artefatos de interação que podem ser encontrados no ambiente *LeMill* são:

Conteúdo: Contém pedaços (ou ativos) e material de aprendizagem (ou objetos). Os pedaços podem ser usados para construir material de aprendizagem, e o material pode ser combinado em entidades maiores, tais como material didático de um único curso de estudo. Ao efetuar a construção do material, os membros usam os recursos que eles têm colecionado.

Atividades: Contém informações e descrições de várias técnicas e atividades que podem ser usadas em situações de aprendizagem. As descrições podem ser acompanhadas de ligações para mais informações relacionadas. Uma descrição da atividade também pode conter ligações para “Conteúdos” com os quais ela é usada, assim como para “Ferramentas” que podem ser usadas na atividade.

Ferramentas: Contém informações e descrições de diversas ferramentas digitais e físicas necessárias em situações de aprendizagem. As descrições podem ser acompanhadas de ligações para *Websites* com mais informações, e por ferramentas digitais reais (se disponíveis *online*). Uma descrição de uma ferramenta também pode conter ligações para “Conteúdos” nos quais ela é usada, assim como para “Atividades” que podem ser usadas com a ferramenta.

Comunidade: Contém os membros registrados de um grupo. Cada membro da comunidade mostra algumas informações sobre si mesmo, como país onde reside, idioma, habilidades e interesses. Os membros podem criar suas coleções pessoais dos recursos que consideram interessantes ou úteis e outros membros podem ver essas coleções. Os recursos encontrados ao navegar ou pesquisar nos repositórios podem ser adicionados às coleções para uso posterior. As coleções podem ser usadas para criar planos de aula; assim, as pessoas têm todos os conteúdos, atividades e ferramentas em um único local conveniente. Cada grupo da comunidade pode ter seu próprio *blog*, onde os membros podem postar conteúdos.

A página do grupo também mostra as ações que os membros do grupo têm feito recentemente, para que as pessoas possam acompanhar o progresso do grupo enquanto eles trabalham com recursos de aprendizagem. Qualquer pessoa está autorizada a comentar o trabalho do grupo.

Histórias: Ao planejar ou após a realização de um evento de aprendizagem ou um curso completo, os professores são incentivados a escrever um relato descrevendo suas experiências com os conteúdos, atividades e ferramentas utilizadas. Essas histórias serão mostradas sempre que alguém estiver vendo qualquer dos recursos mencionados na história. Desta maneira, as experiências dos professores são facilmente disponíveis para os demais, e eles podem ver como outros professores têm usado um determinado conteúdo, ou uma determinada atividade.

Do ponto de vista técnico, o *LeMill* é um produto *Plone* usando o servidor de aplicações *Zope*. Essa plataforma é escrito usando a linguagem de programação *Python* com um ambiente de execução. Todos estes *softwares* são livres. O *LeMill* não é um produto *Plone* normal que pode ser apenas ligado em qualquer instância do *Plone*. Contudo, este é um perfil de extensão, o que significa que personaliza todos os aspectos de uma instância do *Plone*. Não se pode executar o *LeMill* como parte de uma instância do *Plone* normal, mas é necessário uma instância exclusiva para o *LeMill*. A customização deste ambiente é possível, mas requer um programador experiente na linguagem de programação e também no código e arquiteturas de *software* subjacentes. A Tabela 2 apresenta uma sumarização da análise do *LeMill*. Outra aplicação da mesma categoria do *LeMill* é o *Mahara* [73].

Tabela 2: Sumarização da Análise sobre o *LeMill*

Vantagens	Desvantagens
Possui conceitos e artefatos de interação interessantes para o contexto de ensino e aprendizagem e comunidades de aprendizagem	Dificuldade de Customização devido a complexidade da aplicação
Possui funcionalidades focadas para o contexto de educação e para aprendizagem colaborativa	Dificuldade para estender e adicionar novas funcionalidades na aplicação
Ferramentas focadas para a colaboração entre os membros	Mesmo sendo uma solução de código livre, há um grupo fechado e limitado de desenvolvedores e colaboradores do projeto

4 Discussão

A escolha de uma solução ou plataforma base para o desenvolvimento de um *software* específico pode depender de diversas variáveis: desde o conhecimento técnico do usuário, quanto ao tamanho do projeto, as necessidades específicas, restrições e requisitos não funcionais como segurança e desempenho. Para o contexto do TNR, mesmo no estágio atual de elicitação de seus requisitos específicos, algumas características devem ser suportadas, possibilitando vislumbrar alguns caminhos a serem tomados para o desenvolvimento do sistema computacional.

Segundo a análise dos 4 principais CMSs, pode-se perceber que não há diferenças tão marcantes entre eles. A respeito das soluções de SN-CMS, pode-se notar que há muitos projetos comerciais; as soluções de código aberto mais evoluídas têm um apelo comercial, principalmente nos *plugins*. Pode-se também notar que as soluções livres e de código aberto das SN-CMSs são projetos ainda pequenos, que não tem o suporte de uma grande comunidade e uma documentação bem organizada e completa. Os projetos mais avançados limitam as soluções de código aberto e tem o objetivo de vender as soluções empresariais ou todas as extensões e customizações tem um preço embutido.

Adicionalmente, observou-se que as soluções de funcionalidades específicas de SN-CMSs não diferem radicalmente de *plugins* existentes dos CMSs tradicionais. Em alguns casos específicos, seria interessante fazer um estudo para agregar funcionalidades diferenciadas, *e.g.* do *Spree* com uma nova funcionalidade de um CMS. Logo, o uso de *APIs* para isso seria necessário. A vantagem de se usar um CMS altamente utilizado e difundido na *Web* é principalmente por que ele é mais robusto, uma vez que foi mais testado e possui um suporte mais efetivo da comunidade.

O *Wordpress* mostrou-se, apesar de sua potencialidade, ser mais limitado e voltado para *blogs*. O *Plone*, apesar de sua potência, robustez e segurança, requer alta complexidade para estender suas funcionalidades, além de não possuir soluções de extensão para redes sociais. As soluções *Joomla* e *Drupal* são ótimas candidatas ao desenvolvimento pretendido, segundo as características e vantagens ilustradas. Eles principalmente têm um grande suporte de uma comunidade plenamente ativa, com documentos completos e organizados. O *Joomla* perde pontos devido a diversos de seus *plugins* serem comerciais, inclusive os relacionados a redes sociais. É importante notar que todos as plataformas demandarão um custo de aprendizagem para dominar plenamente seu uso e possibilitar customizações e extensões.

O *VilanaRede* apresenta funcionalidades muito relevantes, que realmente caracterizam uma RSI, e que a princípio vão ao encontro dos requisitos mais fundamentais do TNR. Contudo, mesmo apresentando uma arquitetura mais flexível para adicionar novas funcionalidades, devido à natureza desta solução, que agregada à complexidade do *Drupal* em gerenciar e manter adequadamente ambas as partes da arquitetura proposta, ele pode não ser a melhor opção para o TNR. Esta solução adiciona complicadores, como por exemplo, a quantidade de tecnologias envolvidas e exige uma melhor organização e experiência da equipe de desenvolvimento. É importante ressaltar que se deve analisar possibilidades de transportar muitas das funcionalidades presentes no *VilanaRede* para a solução do TNR. Para

não ser necessário realizar uma recodificação completa, uma possível saída seria gerar uma nova instância do *VilanaRede*. Contudo, algumas ações seriam importantes como considerar uma versão mais atual do *Drupal*, incluindo recursos de redes sociais, simplificar a arquitetura, diminuir o número de tecnologias envolvidas e “modularizar” o desenvolvimento para facilitar a manutenção. Além de organizar melhor o trabalho da equipe de desenvolvimento, realizando testes locais, integração de código, *etc.*

O *LeMill*, tendo uma natureza de sistema de *software* direcionado a conteúdo educacional, também apresenta funcionalidades muito interessantes com relação ao objetivo do TNR. É possível dizer que estas poderiam suportar de maneira inteligente os requisitos do TNR, como suporte a colaboração, comunidades, *etc.* No entanto, uma vez construído usando como base o *Plone*, o *LeMill* herda toda a sua complexidade de desenvolvimento e customização. O estudo da adequação de algumas funcionalidades disponíveis no *LeMill* para a solução do sistema TNR é recomendável.

Relacionando os CMSs com os SN-CMSs, pode-se perceber que estes últimos não são tão testados e usados quanto o *Drupal* ou *Joomla*, por exemplo. Estes CMSs, em uma perspectiva de desenvolvimento, proporcionam uma série de oportunidades: suas comunidades de desenvolvimento são numerosas e ativas, principalmente as do *Drupal*. A criação de módulos tem o suporte de uma grande comunidade de desenvolvedores. O *Drupal* pode ser o mais indicado principalmente devido a sua facilidade de uso, o grande número de módulos disponíveis sem custo, além do suporte da comunidade e de documentos. Todavia, é um aplicativo complexo, que necessita de um custo de aprendizagem considerável. É também importante ressaltar que todas estas plataformas não tem foco explícito em aspectos relacionados a valores como credibilidade, confiança, reputação, *etc.*, e não tem suporte específico para lidar com eles. Logo, é necessário o estudo de como adicionar novas funcionalidades para considerar estes aspectos no TNR.

Adicionalmente, ressalta-se a importância de realizar testes pilotos rápidos usando as últimas versões destes CMSs, observando o tempo necessário para o desenvolvimento de novas funcionalidades para ambos no contexto do TNR. Assim, seria possível responder, fundamentando na presente pesquisa, qual solução, além de suas características, vantagens e desvantagens intrínsecas permitiria efetuar extensões mais facilmente e rapidamente para o contexto do TNR.

5 Conclusão

O projeto TNR visa entender e criar soluções de *software* para redes sociais que atendam e suportem as necessidades dos professores que discutem os casos no contexto do AEE. Este relatório efetuou uma investigação das principais possibilidades de plataformas para se construir o sistema TNR. A principal dificuldade enfrentada foi lidar com uma grande opção de soluções possíveis, frente a uma análise fundamentada apenas em requisitos de alto nível do projeto. Uma melhor decisão deve prover uma ampla diversidade de recursos úteis e necessários para o projeto (*e.g.* multimídia, acessibilidade, modularização, extensões,

linguagem). No mesmo sentido, uma boa decisão com uma escolha mais adequada pode reduzir significativamente a necessidade de codificação e, especialmente, a necessidade da adoção de outras tecnologias e servidores para complementar o solução a ser desenvolvida.

Este trabalho apresentou uma análise dos CMSs mais utilizados na *Web* e buscou soluções de CMS com foco em redes sociais. Além destes, consideramos e analisamos, do ponto de vista técnico, soluções específicas de *software* para RSIs no contexto educacional. Fundamentado em uma análise das principais plataformas, uma discussão foi desenvolvida apontando pontos negativos e positivos dos diversos caminhos possíveis frente às soluções analisadas. A partir desta investigação, recomenda-se que se façam testes práticos e rápidos com as versões mais atuais dos CMSs *Drupal* e *Joomla*, que se mostraram mais maduros e adequados para o contexto do TNR. Assim, será possível concluir com mais precisão a melhor plataforma base para se desenvolver o sistema.

Agradecimentos

Este trabalho é financiado pela PROESP/CAPES (processos nro. #23038.01457/2009-11 e 383536/2011-1). Os autores também agradecem aos amigos do InterHAD, IC/UNICAMP e NIED.

Referências Bibliográficas

1. MANTOAN, M.T.E & BARANAUSKAS, M. C. C. 2009. **Redes Sociais e Autonomia Profissional: novos rumos para formação continuada a distância de professores de AEE**. Proposta de Projeto CAPES/PROESP.
2. BOYD, D. M & ELLISON, N. B. 2008. **Social Network Sites: Definition, History, and Scholarship**. Journal of Computer-Mediated Communication, 13. International Communication Association. p. 210–230.
3. LIU, K. 2000. **Semiotics in information systems engineering**. Cambridge University Press.
4. ALMEIDA, L.D.A; SANTANA, V.F.; BISPO, D. M.; BARANAUSKAS, M. C. C. 2008. **An Architecture to Integrate Content Management Systems and Web Applications**. Technical Report (IC-08-08), Institute of Computing, University of Campinas, Brazil. Available at www.ic.unicamp.br/~reltech
5. UFRGS. **O que é o Plone?** Disponível em: <paginas.ufrgs.br/tutorial/dicas-iniciais/sobre-o-plone>. Acessado em Abril de 2011.
6. Water and Stone. **Open Source Cms Market Share Report. 2010**. Disponível em: <waterandstone.com/book/2010-open-source-cms-market-share-report>. Acessado em Maio de 2011.
7. Csmatrix. Disponível em: <csmatrix.org>. Acessado em Fevereiro de 2011.
8. VilanaRede. Disponível em: <vilanarede.org>. Acessado em Março de 2011.
9. LeMill. Disponível em: <lemill.net>. Acessado em Abril de 2011.
10. Wordpress. Disponível em: <wordpress.org>. Acessado em Abril de 2011.

11. *Drupal*. Disponível em: <drupal.org>. Acessado em Abril de 2011.
12. Joomla. Disponível em: <joomla.org>. Acessado em Março de 2011.
13. Plone. Disponível em: <plone.org>. Acessado em Abril de 2011.
14. W3C. Disponível em: <w3.org>. Acessado em Maio de 2011.
15. MySQL. Disponível em: <mysql.com>. Acessado em Maio de 2011.
16. Apache. Disponível em: <apache.org>. Acessado em Maio de 2011.
17. Nginx. Disponível em: <nginx.net>. Acessado em Maio de 2011.
18. BuddyPress. Disponível em: <buddypress.org>. Acessado em Abril de 2011.
19. Mingle. Disponível em: <wordpress.org/extend/plugins/mingle>. Acessado em Abril de 2011.
20. Wordpress. **Features**. Disponível em: <codex.wordpress.org/WordPress_Features>. Acessado em Abril de 2011.
21. Wordpress. **Accessibility**. Disponível em: <codex.wordpress.org/Accessibility>. Acessado em Abril de 2011.
22. CMSCOUK. **Open Source CMS**. Disponível em: <cms.co.uk/open-source>. Acessado em Maio de 2011.
23. *Drupal*. **Accessibility**. Disponível em: <Drupal.org/about/accessibility>. Acessado em Maio de 2011.
24. *Drupal*. **HTML and CSS techniques and tools**. Disponível em: <Drupal.org/node/37156>. Acessado em Maio de 2011.
25. MediaWiki. Disponível em: <mediawiki.org/wiki/MediaWiki>. Acessado em Maio de 2011.
26. PhpBB. Disponível em: <phpbb.com>. Acessado em Maio de 2011.
27. JoomlaPolis. Disponível em: <joomlapolis.com>. Acessado em Maio de 2011.
28. JooSocial. Disponível em: <jomsocial.com>. Acessado em Maio de 2011.
29. Plone. **Plone Community Developer Documentation**. Disponível em: <plone.org/documentation/manual/plone-community-developer-documentation>. Acessado em Maio de 2011.
30. Plone. **User manual**. Disponível em: <plone.org/documentation/manual/plone-4-user-manual>. Acessado em Março de 2011.
31. Plone Social Networking. Disponível em: <groups.google.com/group/plone-social-networking>. Acessado em Março de 2011.
32. JV-Extensions Blog. Disponível em: <jv-extensions.com/blogs/sniranjan/7-advantages-of-joomla-over-other-content-management-systems-like-drupal-or-plone.html>. Acessado em Março de 2011.
33. Pligg. Disponível em: <pligg.com>. Acessado em Abril de 2011.
34. Digg. Disponível em: <digg.com>. Acessado em Abril de 2011.
35. LiveStreetCMS. Disponível em: <livestreetcms.com>. Acessado em Maio de 2011.
36. Anita Social Engine. Disponível em: <anahitapolis.com/download/anahita-social-engine>. Acessado em Maio de 2011.
37. Elgg. Disponível em: <elgg.org>. Acessado em Abril de 2011.

38. Appleaseed. Disponível em: <opensource.appleaseedproject.org>. Acessado em Abril de 2011.
39. Spree. Disponível em: <project.askspre.de>. Acessado em Maio de 2011.
40. Clonesumating. Disponível em: <code.google.com/p/clonesumating>. Acessado em Maio de 2011.
41. LifeRay. Disponível em: <liferay.com>. Acessado em Abril de 2011.
42. SocialCMS. Disponível em: <socialcms.com>. Acessado em Maio de 2011.
43. Hotarucms. Disponível em: <hotarucms.org>. Acessado em Abril de 2011.
44. Social Network server. Disponível em: <socialnetworkserver.com>. Acessado em Maio de 2011.
45. Oxwall. Disponível em: <oxwall.org>. Acessado em Maio de 2011.
46. Phpsocial. Disponível em: <phpsocial.net>. Acessado em Abril de 2011.
47. Insoshi. Disponível em: <dogfood.insoshi.com>. Acessado em Abril de 2011.
48. Lovdbyless. Disponível em: <lovdbyless.com>. Acessado em Abril de 2011.
49. Phpizabi. Disponível em: <phpizabi.net>. Acessado em Abril de 2011.
50. Barnraiser. Disponível em: <barnraiser.org>. Acessado em Maio de 2011.
51. Webnetwork. Disponível em: <webscribble.com/products/webnetwork>. Acessado em Maio de 2011.
52. Kootali. Disponível em: <kootali.demo.agriya.com>. Acessado em Abril de 2011.
53. Dolphin. Disponível em: <boonex.com/dolphin>. Acessado em Abril de 2011.
54. Blogtronix. Disponível em: <blogtronix.com>. Acessado em Abril de 2011.
55. Sparta social networks. Disponível em: <spartasocialnetworks.com>. Acessado em Maio de 2011.
56. Webcrossing. Disponível em: <webcrossing.com>. Acessado em Abril de 2011.
57. Jcow. Disponível em: <jcow.net>. Acessado em Abril de 2011.
58. Sharetronix. Disponível em: <sharetronix.com>. Acessado em Abril de 2011.
59. Social engine. Disponível em: <socialengine.net>. Acessado em Abril de 2011.
60. People Agregator. Disponível em: <broadbandmechanics.com>. Acessado em Maio de 2011.
61. Yogurt Social Network. Disponível em: <xoops.org/modules/repository/singlefile.php?cid=97&lid=1668>. Acessado em Abril de 2011.
62. Xoops. Disponível em: <xoops.org>. Acessado em Maio de 2011.
63. Community engine. Disponível em: <communityengine.org>. Acessado em Maio de 2011.
64. Ruby on rails. Disponível em: <rubyonrails.org>. Acessado em Abril de 2011.
65. Mymates. Disponível em: <sourceforge.net/projects/mymates>. Acessado em Abril de 2011.
66. NERIS, V. P. A.; ALMEIDA, L. D. ; MIRANDA, L. C. ; HAYASHI, E. ; BARANAUSKAS, M. C. C. 2009. **Towards a Socially-constructed Meaning for Inclusive Social Network Systems**. In: International Conference on Informatics and Semiotics in Organisations. Beijing. pp. 247-254.

67. BARANAUSKAS, M.C.C. 2007. **e-Cidadania: Systems and Methods for the Constitution of a Culture mediated by Information and Communication Technology**. Research Proposal for the Microsoft Research-FAPESP Institute.
68. BARANAUSKAS, M.C.C. et al. 2010. **e-Cidadania: Systems and Methods for the Constitution of a Culture mediated by Information and Communication Technology**. Research Report n. 5 Complement B. System Architecture, Software Platform, Installation and Configuration. FAPESP Process N. 2007/54564-1
69. BARANAUSKAS, M.C.C. et al. 2008. **e-Cidadania: Systems and Methods for the Constitution of a Culture mediated by Information and Communication Technology**. Research Report n. 2. FAPESP Process N. 2007/54564-1
70. le group. Learning Environments research group – Media Lab Helsinki – Aalto University. Disponível em: <legroup.uiah.fi>. Acessado em Março de 2011.
71. Calibrate Project. Disponível em: <calibrate.eun.org/ww/en/pub/calibrate_project/home_page.htm>. Acessado em Março de 2011.
72. Plone. **Functionality**. Disponível em: <goedel.uiah.fi/projects/calibrate/wiki/LeMill#Functionality>. Acessado em Março de 2011.
73. Mahara. Disponível em: <mahara.org>. Acessado em Maio de 2011.
74. GAMMA, ERICH; HELM, RICHARD; JOHNSON, RALPH; VLISSIDES, JOHN. 1994. **Design Patterns: Elements of Reusable Object-Oriented**. Addison Wesley ISBN 978-0201633610.
75. Open Software Licenses. Disponível em: <www.opensource.org/licenses/alphabetical>. Acessado em Outubro de 2011.