



ISSN 1677-8464

## Diretrizes para Escolha de um Sistema de Gerenciamento de Conteúdos

Luciana Alvim Santos Romani<sup>1</sup>  
Sílvio Roberto Medeiros Evangelista<sup>2</sup>  
Felipe Kashiwagura<sup>3</sup>

Inúmeros *websites* têm sido construídos utilizando-se ferramentas manuais que geram páginas estáticas, cuja atualização depende de pessoal especializado. Esta atualização deve ser feita em cada página que compõe o *site* através de softwares para este fim como *FrontPage*, *DreamWeaver* ou diretamente nos arquivos HTML. Por exemplo, para modificar o conteúdo da área de Recursos Humanos (RH) de uma empresa que tenha sua *intranet* desenvolvida com páginas estáticas, é necessário que o pessoal do RH conheça ou seja treinado em alguma ferramenta para edição de páginas. Caso contrário, as alterações deverão ser feitas pelo *web designer* responsável. Neste caso, ele terá muito trabalho para manter o *site* e atender aos responsáveis por setores da empresa com dúvidas na manutenção das páginas.

O problema é ainda maior quando há necessidade de modificação do *design* do *site*, pois todas as páginas precisam ser refeitas. Como esta alteração é bastante trabalhosa, muitos *sites* passam anos com a mesma interface. *Sites* com conteúdos que precisam ser alterados constantemente também se tornam difíceis de serem mantidos em uma estrutura de páginas estáticas. Somado a esses fatores, está o fato dos usuários da internet terem, atualmente, expectativas maiores em relação ao conteúdo e tempo de atualização dos *sites*. Principalmente para aqueles que disponibilizam informação em tempo real, por exemplo,

portais de conteúdo como Yahoo, portais de previsão do tempo, etc. O aumento das expectativas em relação aos *sites* deve-se ao fato dos usuários a cada dia se tornarem mais especializados na navegação web, possuírem equipamentos melhores e possibilidades de acessos mais rápidos.

A fim de fornecer conteúdo atualizado de forma rápida e com qualidade a esses usuários finais, os *websites* passaram a adotar um novo paradigma de desenvolvimento. As antigas páginas estáticas desenvolvidas em HTML estão gradativamente sendo substituídas por sistemas de gerenciamento de conteúdo. Essas ferramentas também chamadas de CMS - Content Management System suportam a criação, gerenciamento, distribuição e publicação de conteúdo na internet (Pokorny, 2001; Robertson, 2003b). O poder de tais sistemas consiste no fato de que pessoas leigas em tecnologia da informação e computação sejam capazes de atualizar e publicar conteúdo, mesmo desconhecendo os detalhes da estrutura e *design*, mas mantendo a integridade das páginas.

O objetivo deste Comunicado Técnico é apresentar as diretrizes básicas para escolha de um CMS tomando por base a experiência adquirida com o desenvolvimento do Agritempo (Evangelista et al., 2003). O Agritempo é um sistema para web com atualização dinâmica de informações agrometeorológicas para todos os estados brasileiros.

<sup>1</sup> M.Sc. em Ciência da Computação, Pesquisadora da Embrapa Informática Agropecuária, Caixa Postal 6041, Barão Geraldo - 13083-970 - Campinas, SP. (e-mail: luciana@cnptia.embrapa.br)

<sup>2</sup> Dr. em Engenharia Elétrica, Técnico de Nível Superior da Embrapa Informática Agropecuária, Caixa Postal 6041, Barão Geraldo - 13083-970 - Campinas, SP. (e-mail: silvio@cnptia.embrapa.br)

<sup>3</sup> Graduando em Engenharia de Computação, Bolsista Projeto FINEP, Caixa Postal 6041, Barão Geraldo - 13083-970 - Campinas, SP. (e-mail: felipe@cpa.unicamp.br)

## Estrutura de um CMS

O objetivo de um CMS é permitir que a criação de um *site* seja feita de forma facilitada pelo uso de ferramentas simples desde a edição do conteúdo das páginas até a sua publicação. Além disso, o gerenciamento da estrutura, a aparência das páginas publicadas e a navegação disponível para os usuários finais também devem ser facilitados.

De acordo com Robertson (2003b), a funcionalidade de um CMS pode ser subdividida em: criação e gerenciamento do conteúdo, publicação e apresentação. A criação do conteúdo é feita por meio de um ambiente de autoria com facilidades de edição, que permitem a criação de novas páginas ou atualização de conteúdo sem a necessidade de se conhecer linguagens técnicas como HTML, PHP ou outras. A maioria desses ambientes de autoria é baseada na web e possibilita a atualização do conteúdo, remotamente por meio de navegadores (Internet Explorer, Mozilla, etc.). De forma geral, é por meio de ferramentas de autoria que os CMSs possibilitam que vários usuários sejam responsáveis por módulos distintos e os atualizem simultaneamente das suas estações de trabalho.

O CMS utiliza um repositório para armazenar as informações a serem apresentadas em cada página. A existência deste repositório central contribui para que seja:

- mantido um histórico de todas as versões das páginas, quem as alterou e quando;
- possível controlar que um usuário somente altere as partes do *site* pelas quais é responsável;
- possível a integração com outras fontes de informação existentes e sistemas de tecnologia de informação.

Uma vez que o conteúdo final esteja no repositório, ele pode ser publicado. Os CMSs utilizam mecanismos para aplicação da aparência e *layout* das páginas ao conteúdo de forma automática durante a publicação. Este mecanismo é uma programação dinâmica interpretada ou executada no momento em que o usuário acessa o *site* (Pokorny, 2001), ou seja, o CMS busca no banco de dados o conteúdo a ser apresentado e aplica a ele a estrutura e aparência do *site* no momento do acesso, utilizando linguagens interpretadas como o PHP, por exemplo. Isto não significa que *websites* que utilizam o mesmo CMS tenham a mesma interface, pois a aparência pode ser diferente para cada um. Esta possibilidade refere-se à existência de um gerenciador de *templates* (design/aparência da página), que permite ao administrador do CMS, com um certo conhecimento em HTML, personalizar sua página de modo prático, sem a necessidade de alteração no código fonte do CMS. A navegação também é construída pelo CMS que busca a definição da estrutura no repositório de conteúdo.

Um outro aspecto importante dos CMSs é a facilidade de incorporação de novos recursos, desenvolvidos pela equipe responsável pelo CMS ou por colaboradores. Esta possibilidade de agregação de novos recursos deve-se ao fato destes sistemas serem modulares. Esta é uma característica básica dos gerenciadores de conteúdo que está diretamente relacionada com o princípio dos "blocos encaixe". Este princípio permite que os módulos sejam desenvolvidos

independentemente e anexados à estrutura do CMS quando necessário. Essa arquitetura permite que os CMSs sejam alterados de forma a atender ao objetivo e ao interesse de cada *site*. Além disso, possibilita um desenvolvimento posterior ou uma atualização de determinadas sessões sem que seja necessária uma atualização do núcleo do CMS, o que poderia causar um transtorno ou uma ruptura nos serviços fornecidos pelo *site*.

## Linguagens, banco de dados utilizados e sistemas existentes

Os CMSs podem ser escritos nas mais diversas linguagens de programação que tenham algum tipo de interface com a web. No entanto, a maioria deles é desenvolvida em PHP, Python, Java e Perl.

A partir de um sistema de publicação - Nuke - desenvolvido em Perl surgiu a idéia de criação de um dos primeiros sistemas de gerenciamento, o PHP-Nuke. Este software é um sistema de portal bastante completo com uma estrutura modular, que facilita a integração rápida de novas ferramentas (PHP-Nuke, 2003). Algumas variantes do PHP-Nuke que se destacam são o PostNuke (PostNuke.Com, 2003) e mais recentemente o Xoops (Xoops Projects, 2003), que parece ser a nova tendência em CMS (Broder, 2003).

Existem, ainda, CMSs que são desenvolvidos em Zope, um ambiente de desenvolvimento para web utilizado para construção de gerenciadores de conteúdo, portais e aplicações customizadas (Zope Corporation, 2003). Escrito em Python, Zope é considerado um software bastante poderoso, mas muito complicado. Os programadores podem utilizar tanto Python quanto uma linguagem de *script* do Zope para desenvolvimento das aplicações, além de um banco de dados próprio. Plone é o mais conhecido CMS escrito em Zope. Além de um sistema de gerenciamento de conteúdo, ele pode ser usado como um sistema para publicação de documentos, portal e ferramenta de *groupware* para colaboração entre grupos de localidades distantes (Limi et al., 2003).

O OpenCms, escrito em Java e usando a tecnologia XML, permite a criação e gerenciamento de *websites* complexos e possui um editor com uma interface gráfica semelhante às aplicações Office. Este tipo de interface WYSIWYG (*what you see is what you get*) auxilia os usuários na criação do conteúdo (Alkacon Software, 2003).

Todos estes CMSs estão incluídos na categoria de software livre. Dentre eles, os mais utilizados são os de código aberto (*open source*) que utilizam a licença GPL - General Public License. A maioria deles pode ser executada com ambiente composto por exclusivamente softwares livres (por exemplo, Linux, Apache, Tomcat, MySQL, etc.) como com sistemas comerciais (por exemplo, Windows NT, IIS, BEA Weblogic, Oracle, etc.).

Na maioria dos casos, os CMSs suportam inúmeros bancos de dados como o MySQL, PostgreSQL, MS SQL, Oracle, DB2 entre outros.

## Caso Agritempo

O Agritempo é um sistema de monitoramento agrometeorológico que permite a consulta a dados meteorológicos, mapas de monitoramento e previsão, zoneamento agrícola, publicações, glossário, notícias, etc. (Embrapa Informática Agropecuária, 2003). Grande parte destas informações é atualizada diariamente por vários usuários. Inicialmente o *site* foi desenvolvido com páginas estáticas em HTML e algumas funcionalidades utilizando a tecnologia de Java Servlets. Com o aumento da quantidade de dados a serem atualizados e de pessoas responsáveis por partes do *site*, houve a necessidade de migração para um CMS.

Devido ao prazo pequeno para desenvolvimento e implantação do novo *site* e dado que o grupo de programadores já tinha familiaridade com a linguagem PHP e o sistema de gerenciamento PHP-Nuke 6.5 com a base de dados MySQL, este foi o CMS escolhido. Alguns usuários estão cadastrados no sistema para que possam atualizar partes do *site*. A seqüência de figuras a seguir mostra a alteração do tópico Destaque que aparece na página principal do *site* Agritempo. Para atualizá-lo o usuário seleciona Mensagens no Menu de Administração, como ilustrado na Fig. 1.



Fig.1. Tela de administração do PHP-Nuke adaptada para o Agritempo.

Uma nova página é apresentada com um formulário contendo uma caixa de edição para preenchimento do título, do texto e para seleção da figura que serão mostrados na sessão Destaque, como pode ser visto na Fig. 2. Pode ser definido também o tempo que a notícia será exibida e quando será veiculada.

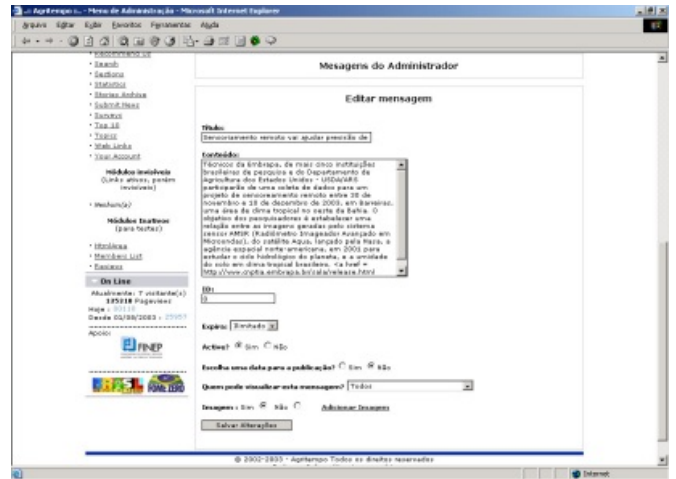


Fig.2. Formulário de entrada para a sessão Destaque.

Após salvar as alterações o PHP-Nuke une as informações que foram digitadas no formulário com a estrutura pré-definida de apresentação e monta a página que será exibida para o usuário final como ilustrada na Fig. 3.



Fig 3. Página principal do *site* Agritempo.

O fato do PHP-Nuke possuir código aberto permitiu que alterações pudessem ser feitas no sistema. Além disso, com o tempo reduzido para implantar o novo *site*, era aconselhável usar um sistema bastante difundido pois em caso de dúvidas ou problemas sempre haveria listas, grupos e documentação disponível para buscar soluções. A instalação e configuração do CMS, servidores, PHP e banco MySQL foram feitas rapidamente de forma satisfatória.

A primeira dificuldade surgiu no momento de incorporar o design (*template*) proposto para o Agritempo ao PHP-Nuke. Este CMS utiliza uma estrutura (*framework*) simétrica, ou seja, os elementos do lado esquerdo do *site* devem ter a mesma forma daqueles apresentados do lado direito. Como na proposta de design do Agritempo, esta simetria inexistia foram feitas modificações estruturais no núcleo do PHP-Nuke, mais especificamente no gerenciador de *templates*, a fim de permitir a quebra dessa simetria de colunas imposta pelo mesmo.

Foi necessário também o desenvolvimento de dois novos módulos - ajuda e mapas - para atender às necessidades do

projeto. Além disso, foram retirados códigos excessivos e inseridas verificações de segurança de módulos e blocos ativos. Para minimizar as possibilidades de ataque ao PHP-Nuke, códigos referentes aos módulos de e-mail e fórum que faziam parte do núcleo do sistema foram omitidos. Adicionalmente, foram utilizadas variáveis para o upload de arquivos, de forma a impossibilitar que arquivos não permitidos sejam enviados ao servidor. Todas as variáveis que o PHP-Nuke recebe pelos métodos GET e POST também são verificadas com expressões regulares, o que permite filtrar possíveis ataques por meio de códigos maliciosos enviados via URL ou formulário para o PHP-NUKE.

## Diretrizes para escolha de um CMS

A partir da experiência com a implantação de um CMS para gerenciar o *site* Agritempo, pôde-se verificar que é necessário o estabelecimento de critérios para a escolha do gerenciador. Além disso, é importante priorizar estes critérios de acordo com as necessidades da instituição e do projeto.

Alguns critérios enumerados a seguir são fundamentais para a escolha de um CMS:

- **Tipo de conteúdo:** deve-se verificar se as páginas são grandes ou não, altamente estruturadas ou desestruturadas, complexas ou simples no que se refere ao *layout* e o uso de figuras. O tipo de conteúdo influi na escolha do CMS pois existem ferramentas muito complexas como por exemplo o Plone que não são adequadas para gerenciar *sites* mais simples.
- **Natureza dos autores:** quais as habilidades e conhecimentos os autores que serão responsáveis pela atualização do conteúdo do *site* devem ter. Também é importante contabilizar o número de autores que serão responsáveis por sessões do *site* e se os mesmos têm disponibilidade de tempo para aprender a usar o CMS e para atualizar o *site*. Se os autores forem leigos em informática é recomendável a escolha de uma ferramenta cuja interface tenha boa usabilidade não exigindo treinamento para sua operacionalização.
- **Uso final:** deve-se considerar em qual contexto o conteúdo será usado e publicado. Isto inclui os formatos que serão publicados (*web*, *wireless*, etc.) bem como qualquer necessidade de reuso do conteúdo. A observância deste critério evita contratemplos futuros ao se descobrir que há a necessidade de publicação de um determinado formato não suportado pelo CMS adotado.
- **Recursos:** é preciso verificar se o CMS pode ser usado com pacotes de software livre (servidores e banco de dados) ou se apenas com produtos comerciais. O custo de adoção e manutenção de um CMS depende também de quais recursos serão acoplados a ele para o seu perfeito funcionamento.
- **Suporte:** é fundamental ter um suporte ativo, com fórum ou listas de discussão, na qual os usuários possam tirar suas dúvidas e haja monitoramento de *bugs* e falhas de segurança. Nestes casos, em geral, o tamanho da comunidade de usuários é bastante significativo. A falta de suporte pode levar a uma demora na solução de problemas e no caso de *sites* a rapidez na correção de falhas é fundamental, evitando danos causados por falhas de segurança.

- **Módulos:** deve-se verificar se o CMS possui os módulos necessários para o desenvolvimento do *site* ou se existem módulos/blocos extras disponíveis para *download* na internet. Além disso, é aconselhável verificar a viabilidade e dificuldades envolvidas com o desenvolvimento de novos módulos.
- **Software livre com código aberto:** possibilita a alteração ou desenvolvimento de novos módulos e núcleos pela equipe de desenvolvimento ou por terceiros.
- **Template e idioma:** é importante que o CMS seja versátil permitindo o desenvolvimento de novos *templates*, já que os *sites* costumam alterar a aparência com frequência. Além disso, é sempre bom verificar se existe uma versão traduzida para o idioma desejado, pois sua tradução pode ser bastante trabalhosa.
- **Segurança:** não é conveniente colocar em produção versões betas dos CMSs pois essas são passíveis de erros. É aconselhável que o administrador do *site* faça parte de um grupo de discussão do CMS que utiliza ou que venha a utilizar, para atualizar-se sobre *bugs* e tipos de ataques que ocorrem nesse tipo de sistema. Deve-se atentar também para a segurança do servidor, utilizando as últimas versões estabilizadas dos servidores web e de Banco de Dados.

A equipe de desenvolvimento pode enumerar os critérios mais importantes a serem considerados. Além disso, incluir outros como por exemplo prazo de desenvolvimento e linguagem de programação.

## Considerações finais

Existem muitas opções diferentes de CMSs, cada um com vantagens e desvantagens. É fundamental determinar os requisitos relativos ao tipo de conteúdo e grupo de autores, além de outros critérios para a escolha do CMS mais adequado às necessidades do *website*. Os critérios para escolha de um CMS enumerados neste trabalho são gerais e atendem aos requisitos básicos para a maioria dos *sites*. No entanto, a equipe de desenvolvimento pode analisar outros aspectos relevantes ao *website* e estabelecer novos critérios. Pode ainda optar por um conjunto reduzido dos mesmos.

Da mesma forma, como o que ocorreu no caso do Agritempo, mudanças e novas implementações podem ser necessárias. Os desenvolvedores devem estar cientes dessa possibilidade, já que em certas situações pode não existir um CMS que atenda todos os requisitos completamente.

Com os avanços tecnológicos e novos CMSs sendo desenvolvidos e colocados disponíveis, pode ser interessante migrar de uma plataforma para outra com mais e melhores recursos. Neste sentido, a equipe do Agritempo já estuda a possibilidade de utilização de um novo CMS, que reduza o tempo de resposta do sistema e que não imponha restrições construtivas que dificultem a implementação dos *templates* e da arquitetura da informação para os mais diferentes navegadores.

Vale ressaltar que os critérios para a escolha de um CMS enumerados neste trabalho são resultado de estudos realizados a partir da literatura e da experiência no desenvolvimento do Agritempo. No entanto, ainda é necessária uma validação dos mesmos aplicando-os na

escolha de um outro CMS que substitua o atual no gerenciamento do *site* Agritempo.

## Referências bibliográficas

ALKACON SOFTWARE. OpenCMS. Org the open source content management systems. Disponível em: < <http://www.opencms.org>> . Acesso em: ago. 2003.

BRODER, T. Conheça as melhores tecnologias open source para criar *sites* dinâmicos. Geek, v. 4, n. 31, p. 20-23, 2003.

EMBRAPA INFORMÁTICA AGROPECUÁRIA. Agritempo sistema de monitoramento agrometeorológico. [Campinas]: Embrapa Informática Agropecuária: Cepagri/Unicamp, 2002. Disponível em: <http://www.agritempo.gov.br>. Acesso em set. 2003.

EVANGELISTA, S. R. M.; TERNES S.; SANTOS, E. H. dos; ASSAD, E. D.; ROMANI, L. A. S.; FRANZONI, A. Agroclima sistema de monitoramento agroclimatológico. In: CONGRESSO BRASILEIRO DE AGROMETEOROLOGIA, 13., 2003, Santa Maria. Situação atual e perspectivas da agrometeorologia: anais... Santa Maria: Unifra, SBA, UFSM, 2003. v. 1, p. 603-604.

LIMI, A.; RUNYAN, A.; ANDERSEN, V. The plone book. Disponível em: < <http://www.plone.org/documentation/book>> . Acesso em: nov. 2003.

PHP-NUKE. PHP-Nuke - advanced content management system. Disponível em: < <http://www.phpnuke.org/>> . Acesso em: jul. 2003.

POKORNY, J. Static pages are dead: how a modular approach is changing interaction design. Interactions, New York, v. 8, n. 5, p.19-24, 2001.

POSTNUKE. COM. Postinuke: navegação. Disponível em: < <http://www.postnuke.com>> . Acesso em: jul 2003.

ROBERTSON, J. Choosing the right CMS authoring tools. Sydney: Step Two Designs, 2003a. Disponível em: < [http://www.steptwo.com.au/papers/kmc\\_authoringtools/](http://www.steptwo.com.au/papers/kmc_authoringtools/)> . Acesso em: out. 2003.

ROBERTSON, J. So, what is a content management system?. Sydney: Step Two Designs, 2003b. Disponível em: < [http://www.steptwo.com.au/papers/kmc\\_what/](http://www.steptwo.com.au/papers/kmc_what/)> . Acesso em: out. 2003.

XOOPS PROJECTS. Xoops official site news. Disponível em: < <http://www.xoops.org>> . Acesso em: ago. 2003.

ZOPE CORPORATION. Zope.org. Disponível em: < <http://www.zope.org>> . Acesso em: ago. 2003.

### Comunicado Técnico, 53

Ministério da Agricultura, Pecuária e Abastecimento

Governo Federal

Embrapa Informática Agropecuária  
Área de Comunicação e Negócios (ACN)  
Endereço: Caixa Postal 6041 - Barão Geraldo  
13083-970 - Campinas, SP  
Fone: (19) 3789-5743  
Fax: (19) 3289-9594  
e-mail: [sac@cnptia.embrapa.com.br](mailto:sac@cnptia.embrapa.com.br)

1ª edição on-line - 2003

ÓTodos os direitos reservados.

### Comitê de Publicações

Presidente: *Luciana Alvim Santos Romani*  
Membros Efetivos: *Carla Geovana Macário, José Ruy Porto de Carvalho, Marcia Izabel Fugisawa Souza, Marcos Lordello Chaim, Suzilei Almeida Carneiro.*  
Suplentes: *Carlos Alberto Alves Meira, Eduardo Delgado Assad, Maria Angelica Andrade Leite, Maria Fernanda Moura, Maria Goretti Gurgel Praxedis.*

### Expediente

Supervisor editorial: *Ivanilde Dispató*  
Normalização bibliográfica: *Maria Goretti Gurgel Praxedis*  
Editoração eletrônica: *Área de Comunicação e Negócios*