DESENVOLVIMENTO DE UMA FERRAMENTA PARA GERAR COMPONENTES CRUD NO CMS JOOMLA!

Renato Garcia de Campos, Francisco Assis da Silva, Leandro Luiz de Almeida, Marcelo Vinicius Creres Rosa

Faculdade de Informática – Universidade do Oeste Paulista (UNOESTE). Rua José Bongiovani, 700 – CEP 19050-920 – Presidente Prudente – SP – Brasil. renatogcampos@gmail.com, {chico,llalmeida,marcelorosa}@unoeste.br

RESUMO

Desenvolvido a partir do Mambo, o CMS JOOMLA! é escrito em PHP e roda em servidores Web Apache ou IIS, com o banco de dados MySQL. Nos dias de hoje, muitos usuários com conhecimento em desenvolvimento e programação já desenvolvem seus próprios componentes de acordo com suas necessidades para o JOOMLA!. Porém, a maioria que faz utilização do CMS não possui conhecimentos suficientes em tal parte e acabam ficando sem muitas opções para suas necessidades. A proposta de se criar uma ferramenta para o auxilio à criação de componentes para manipulações de dados, surgiu ao perceber benefícios como o baixo conhecimento, por exemplo, na utilização do JOOMLA! para a criação e administração de sites complexos. Ela foi desenvolvida para auxiliar o usuário na criação de componentes de acordo com suas necessidades, deixando assim, mais flexível a estrutura do CMS.
Palavras-chaves: JOOMLA; PHP; CMS; Componentes do JOOMLA.

DEVELOPING A TOOL TO GENERATE CRUD COMPONENTS IN THE CMS JOOMLA!

ABSTRACT

Developed from Mambo, the CMS JOOMLA! is written in PHP and is executed in the Apache Web Server or Internet Information Server, with MySQL database. Nowadays, many users with development and programming knowledge already make their own components according with their needs to JOOMLA!. However, the most people that use CMS don't have appropriate knowledge and don't have many options for their needs. The proposal to create a tool to help the creation of data manipulation components came to realize the benefits such as low knowledge, for example, in using the JOOMLA! for creation and administration of complex websites. It was developed to assist users in the creation of components according to their needs, making the CMS structure more flexible.
Keywords: JOOMLA; PHP; CMS; JOOMLA Components.

1. INTRODUÇÃO

A informática está intimamente ligada ao ser humano, seja em casa, no trabalho ou no lazer. A evolução da tecnologia vivida pela sociedade tem evidenciado o valor da informação. Uma das formas de se obter acesso a informação é por meio da internet.

A Internet trata-se de várias redes de computadores interligadas, capaz de interagir com computadores de todo o mundo através de outras redes interligadas. Milhões de pessoas utilizam desse sistema todos os dias, por meio de seu e-mail ou endereço eletrônico, uma espécie de CEP no qual seus computadores podem ser encontrados e passam a receber mensagens vindas de qualquer parte do mundo. Através da rede as pessoas podem trocar mensagens, fazer debates, realizar operações bancárias, copiar arquivos, trabalhar, participar de grupos de discussões, ler jornais e revistas e até mesmo namorar.

A Internet passou a fazer parte do dia a dia da sociedade, sendo cada vez maior o número de pessoas que acessam sites à procura de informações sobre os mais variados temas. A Web se tornou a plataforma universal mais rápida do que qualquer outra tecnologia na história. Ela permite às empresas de qualquer porte expor seus produtos ou serviços, efetuar pedidos, prestar suporte personalizado e ajudar clientes e fornecedores a manterem contato facilmente [Jassinek 2000].

As ferramentas que tornam esses processos de troca de mensagens, debates, operações bancárias, grupos de discussões entre outros realizáveis na Web são fundamentais para a administração e distribuição da informação. Uma dessas ferramentas é o CMS (Sistema de gerenciamento de conteúdo).

Desenvolver uma ferramenta que atue na criação de componentes para manipulações de dados, torna mais facilitado o poder de criação de novos recursos de acordo com as necessidades de cada usuário, e assim, tais componentes seriam criados em um curto período de tempo. Com a utilização dessa ferramenta, o uso do CMS JOOMLA! se torna mais viável, já que a mesma tem como fator principal a inexistência da necessidade de conhecimentos abrangentes sobre o funcionamento do CMS.

A idéia de se desenvolver manualmente um componente, não é uma tarefa muito fácil e também não muito rápida, dependendo do tipo de componente que se deseja (quanto mais difícil, maior a quantidade de tempo se gasta no seu desenvolvimento), é necessário conhecimento sobre o *framework* sobre o qual trabalha. Com a possibilidade de ter um mecanismo que construa componentes CRUD (*Create*, *Read*, *Update*, *Delete* – operações básicas utilizadas em bancos de dados relacionais), tornaria mais simples o uso do CMS e seria de grande valia para a comunidade de usuários JOOMLA!.

2. SISTEMA DE GERENCIAMENTO DE CONTEÚDO (CMS)

Um sistema de gerenciamento de conteúdo é um software que mantém o controle de cada parte do conteúdo do site, bem como uma biblioteca pública local que mantém registro de livros e os armazena. O conteúdo pode ser texto, fotos, músicas, vídeos, documentos, ou sobre qualquer coisa que se possa pensar. Uma grande vantagem da utilização de um CMS é que não exige quase nenhuma habilidade ou conhecimento técnico para gerenciá-lo [Sobre CMS 2009].

Existem CMS baseados em duas licenças diferentes: os baseados e distribuídos sob a licença GNU/GPL (Software Livre) ou licenças similares, e os comerciais, ou seja, softwares proprietários.

Um CMS baseado na licença GNU/GPL, tem muitas mais vantagens sobre um software proprietário. A independência em relação ao

fornecedor e a vantagem de realizar alterações sem custos é a mais importante vantagem para seus usuários, isso faz com que sempre escolham esta licença.

3. JOOMLA!

JOOMLA! é um premiado sistema de gerenciamento de conteúdo (CMS), que permite criar sites e poderosas aplicações online. Muitos aspectos, incluindo a sua facilidade de uso e extensibilidade, fizeram do JOOMLA! o CMS mais popular. E o melhor de tudo, JOOMLA! é uma solução de código aberto que está disponível gratuitamente a todos.

Possui um sistema de registro que permite ao usuário configurar as opções pessoais. A autenticação é uma parte importante da gestão de utilizadores. O JOOMLA! suporta vários protocolos, incluindo LDAP (protocolo para atualizar e pesquisar diretórios rodando sobre TCP/IP), OpenID (rede distribuída na qual a identidade do utilizador é dada por uma URL que pode ser verificada por qualquer servidor executando o protocolo), e até mesmo do Gmail. Isso permite aos usuários utilizarem das suas informações de conta existente para agilizarem o processo de registro.

Para fazer o uso do CMS JOOMLA! (versão utilizada 1.5.x), o local onde o portal ficará hospedado – ou no caso de testes, a máquina local – deverá possuir uma ferramenta que instale um servidor Web. Foi utilizado para este trabalho o APACHE [APACHE 2009], por ser um dos servidores mais bem sucedidos da Web livre que interpreta páginas PHP [PHP Group 2010][Soares 2008] (linguagem nativa do CMS JOOMLA!).

O PHP se trata de uma linguagem extremamente modularizada, que a torna ideal para instalação e uso em servidores Web. É muito parecida, em tipos de dados, sintaxe e funções, com a linguagem C ou C++. Dependendo das configurações do servidor, pode ser embarcada no código HTML. Existem versões do PHP disponíveis para os seguintes sistemas operacionais: Windows, Linux, FreeBSD, Mac OS, OS/2, AS/400, Novell Netware, RISC OS, AIX, IRIX e Solaris. Para executar o CMS JOOMLA!, este, necessita de suporte para SGBD (Sistema de Gerenciamento de Banco de Dados [Sobre SGBD 2009]) MySQL [MySQL 2009], que utiliza a linguagem SQL como interface, sendo atualmente um dos bancos de dados mais populares, este é o SGBD nativo do CMS.

Durante a fase de desenvolvimento da ferramenta deste trabalho, foi necessário instalar e fazer o JOOMLA! rodar em uma máquina local. A ferramenta utilizada para instalar e rodar o servidor local [Sobre Localhost 2009] foi o WampServer 2c [WampServer 2009].

A instalação do JOOMLA! pode ser realizada na língua que desejar (Inglês, Português, Alemão, etc.), a versão em inglês é a mais comum. Após sua instalação, é possível procurar nos sites das comunidades JOOMLA! a tradução para língua que desejar, tanto para atualizar o site principal como à área de administração do JOOMLA!. [JOOMLA! 2009].

3.1. Estrutura do JOOMLA!

Depois de realizada a instalação do CMS, uma estrutura de pastas é criada conforme mostrado na Figura 1.

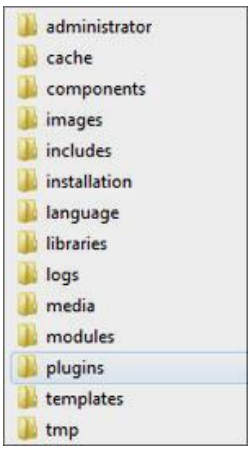


Figura 1. Estrutura de pastas JOOMLA!

Nas pastas “administrator” e “templates”, encontram-se basicamente o site de administração e o template do site. Nas demais, tem-se os módulos, componentes e outros recursos para exibição de conteúdo do site.

Quando o JOOMLA! for instalado em um servidor Web, automaticamente são instaladas duas áreas: Área de usuários (*Frontend*) – Figura 2 – e Área de administração (*Backend*) – Figura 3.

Estas áreas correspondem, respectivamente, ao site principal e a um site de administração do site principal.

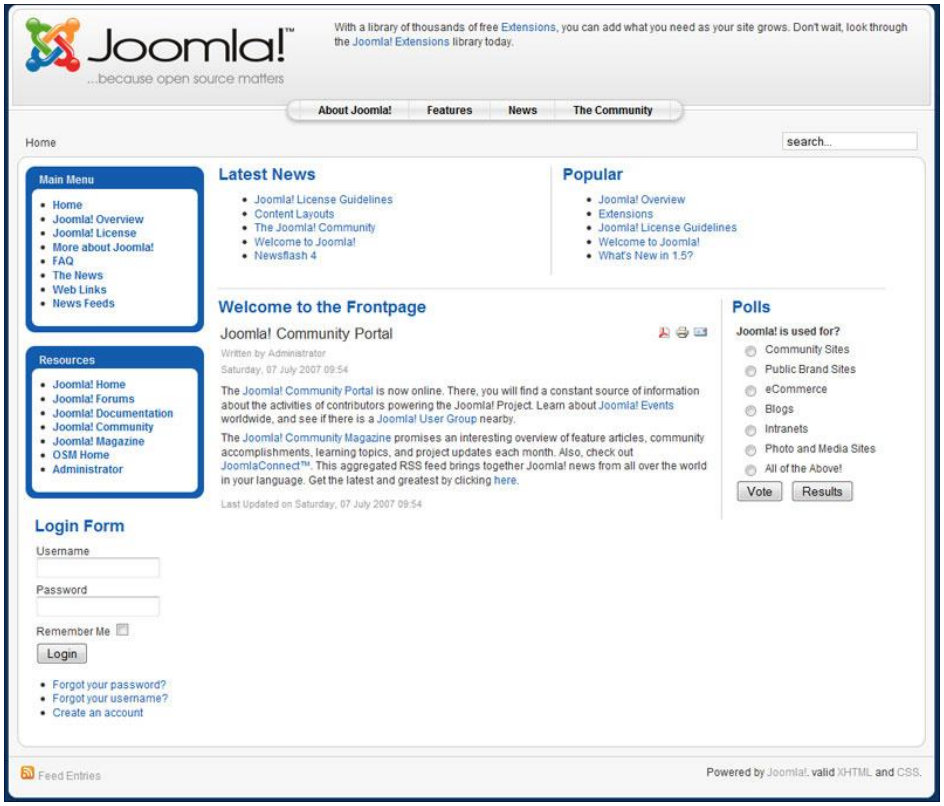


Figura 2. Área de usuários



Figura 3. Área de Administração

A Área de usuários é constituída do próprio site apresentado no navegador quando é digitado o seu endereço na barra de endereços, seu acesso é permitido a qualquer usuário conectado à internet. Já a Área de administração, permite a administração do site principal em relação ao seu conteúdo, nela é informado o que irá aparecer na parte principal (módulos, componentes, etc).

A estrutura do CMS JOOMLA! está baseada no modelo MVC (Modelo, Visão, Controle) que é um padrão de engenharia de software que visa organizar o código de forma que a lógica de trabalho fique separada da apresentação dos dados. Com isso tem-se mais facilidade para se realizar mudanças de implementação na lógica de negócio do projeto [Sobre MVC 2009]. Os componentes JOOMLA! implementam a arquitetura MVC visando uma melhor organização na arquitetura da aplicação.

3.2. Componentes e Módulos

Componentes são os elementos de conteúdo que funcionam como pequenas aplicações no JOOMLA!. Normalmente são mostrados no centro da página, na área de conteúdo principal do template, porém isto depende de como está organizado o template.

Durante a instalação, o JOOMLA! instala vários componentes por padrão, alguns deles são: *Contact*, *Login*, *FrontPage*, *Banners* e *Web Links*.

Alguns destes componentes fazem parte da própria arquitetura do JOOMLA!, um exemplo claro é o componente *FrontPage* – sendo um dos principais componentes, pois estrutura as informações juntamente com os estilos definidos para serem exibidos na primeira página – e não são manipuláveis através do menu.

Além do componente *FrontPage*, que no padrão do JOOMLA! é chamado pelo primeiro item do “Main menu”, qualquer *link* inserido como primeiro item do menu principal será considerado a página de abertura do seu site, desta forma, pode-se apresentar outras estruturas na página inicial, como um conteúdo estático ou qualquer outro tipo de componente.

A comunidade JOOMLA [JOOMLA! 2009], produz e atualiza constantemente centenas de novos componentes.

Os módulos estendem a capacidade do JOOMLA! com novas funcionalidades e permitem mostrar seus conteúdos em qualquer posição que o template permita. Estes são mais fáceis de serem instalados pela área de administração. Como padrão o JOOMLA! instala alguns módulos tais como: *Main Menu* (Menu Principal), *Polls* (Enquetes), *Banner*, *Feed Display*, *Footer*, *Latest News* e *Newsflash*.

A exibição destes módulos se dará conforme a posição definida na estrutura do template criado (*left*, *right*, *toolbar*, *top*, etc).

4. FERRAMENTA

Nos dias de hoje, alguns dos usuários, ou empresas especializadas do ramo, que utilizam CMS para gerenciamento de seus portais possuem conhecimentos em programação, sendo assim já desenvolvem seus próprios componentes de acordo com suas necessidades. Porém, a maioria que faz utilização do CMS não possui estes mesmos conhecimentos em tal parte e ficam sem muitas opções para suprir suas necessidades.

Já estão disponíveis na internet inúmeras quantidades de componentes prontos para o CMS que facilita a vida de diversos tipos de usuários, mas por ser tratar de componentes de modo genérico, podem não suprir algumas necessidades específicas dos próprios usuários. O grande problema é que o CMS se limita inicialmente a componentes que fazem apenas manipulação de documentos, não permitindo muita liberdade ao usuário.

A idéia de se desenvolver uma ferramenta que auxilia na criação de componentes para manipulações de dados não é nova, surgiu com o grande crescimento da utilização do CMS JOOMLA! para administração de sites complexos. Ela auxiliará o usuário na criação de componentes de acordo com suas necessidades, deixando assim, mais flexível a estrutura do CMS.

O *script* de criação do banco de dados mostrado como de exemplo na Figura 4 representa a fonte para alimentação da ferramenta. A Figura 5 ilustra o correspondente MER (Modelo Entidade Relacionamento) referente ao *script* de criação do banco de dados (Figura 4). Este modelo foi construído no Erwin 3.0, ferramenta de software utilizada para a modelagem de sistemas de informação e pertence ao conjunto de programas de apoio ao desenvolvimento de software.

```
CREATE TABLE cidade (
       cid_codigo            int NOT NULL,
       uf_codigo             int NULL,
       cid_nome              char(18) NULL,
        CONSTRAINT XPKcidade
              PRIMARY KEY (cid_codigo)
)
go
............
.........
.....
...
ALTER TABLE clientes
       ADD CONSTRAINT R_85
              FOREIGN KEY (cid_codigo)
                           REFERENCES cidade
go
```

Figura 4. *Script* de alimentação

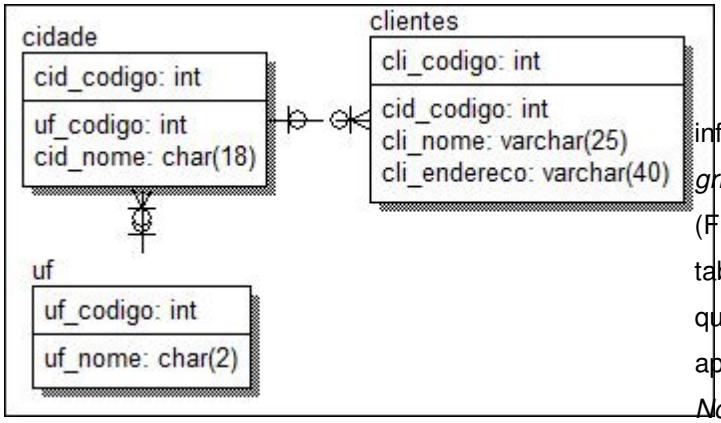


Figura 5. Exemplo de modelagem

Embora os componentes do JOOMLA! sejam baseados no modelo MVC, é possível desenvolver componentes sem seguir este padrão. Foi partido do princípio de utilizar uma forma de geração de código fonte aceito pelo CMS e o mesmo realize as funções desejadas com as limitações impostas.

O *script* de criação do banco de dados é inserido na ferramenta, ela fará a leitura do mesmo e mostrará na tela os nomes das tabelas, campos de cada tabela e ainda o tipo de cada campo.

Na primeira *grid* da tela da ferramenta desenvolvida (Figura 6) é mostrada uma coluna com os nomes das tabelas, uma coluna em branco chamada *Label* que é utilizada para nomear os nomes das tabelas que irão aparecer na aplicação gerada (Figura 6, item 1) e as operações possíveis de serem realizadas:

- Criar: possibilita a inserção de registros na tabela referente;
- Ler: possibilita ver os registros cadastrados na tabela referente;
- Remover: possibilita a remoção de registros cadastrados na tabela referente;
- Alterar: possibilita a alteração de registros cadastrados na tabela referente;
- Movimentação: possibilita informar qual a tabela principal de uma movimentação.

As demais colunas da *grid* são informações referentes a cada tabela. A segunda *grid* também na tela da ferramenta desenvolvida (Figura 6) possui uma coluna com os nomes das tabelas, uma coluna em branco chamada *Label* que é utilizada para nomear os campos que irão aparecer na aplicação gerada, um campo *NotNULL* que deverá ser marcado com um 'S' caso aquele campo necessite da validação de

campo vazio em tela da aplicação gerada. As demais colunas são informações sobre cada campo na respectiva tabela. Abaixo das duas *grids*, existem alguns campos que necessitam ser preenchidos:

- Nome do componente: preenchido somente com valores alfanuméricos;
- Autor: nome do autor do componente;
- Caption Componente: será o "nome" que aparecerá na tela em forma de título para informar qual componente estará sendo usado;
- Versão: versão compatível com o CMS (1.5.x);
- Descrição: descrição do componente.

A Figura 6 ilustra a ferramenta com o *script* SQL inserido e os campos devidamente preenchidos.

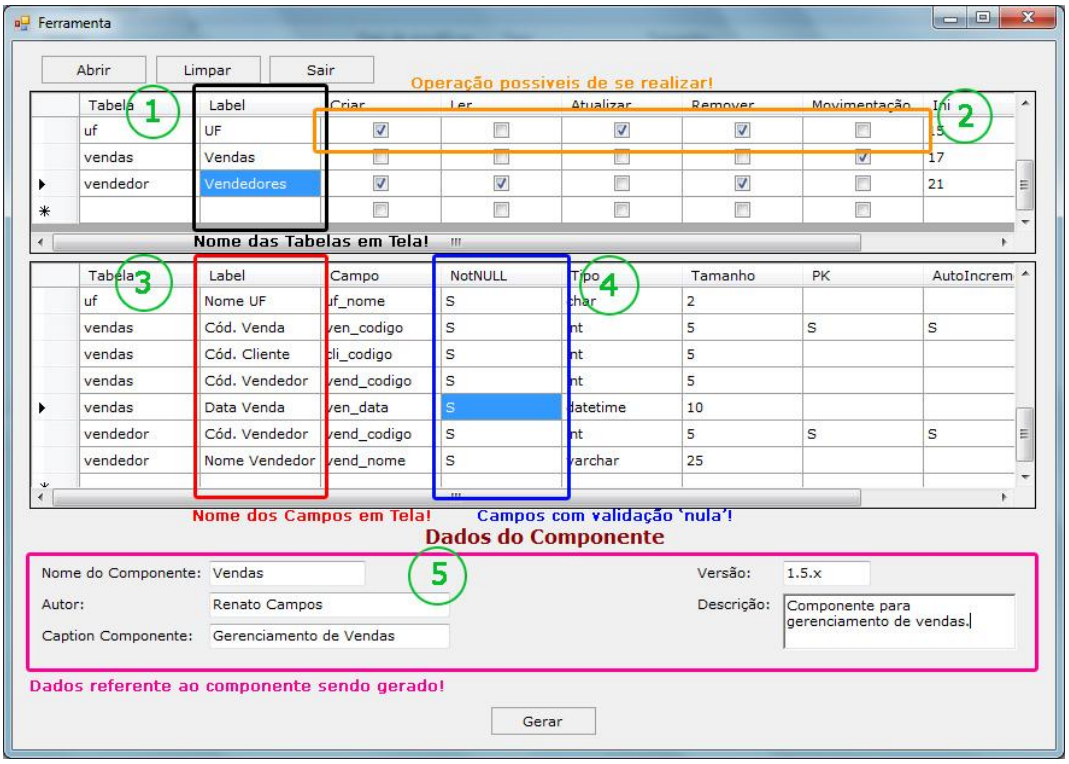


Figura 6. Tela da aplicação com o *script* inserido

4.1. Resultados da Geração do Componente de Vendas

O padrão seguido de geração pela ferramenta é o aceito pelo CMS. Os arquivos com extensão PHP e XML devem possuir o nome a partir do nome dado ao componente na ferramenta. Quando o botão "Gerar" for acionado, será realizada a geração do componente desejado, uma pasta será criada no local escolhido com o nome "com_" + o campo "Nome do Componente" com base na Figura 6, item 5. Para o exemplo utilizado no artigo, o nome da pasta criada será "com_vendas".

Na pasta "com_vendas" haverá uma estrutura como da Figura 7, que conterá os seguintes arquivos: "install.sql", "unistall.sql", "vendas.php" e "vendas.xml" (o nome dado ao componente no exemplo do artigo é vendas).

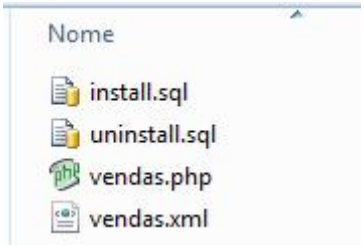


Figura 7. Estrutura do Componente

No arquivo “install.sql”, contém as criações das tabelas que o script SQL original possui, com as sintaxes aceitas pelo banco de dados nativo do JOOMLA!. Já o arquivo de “uninstall.sql”, contém as exclusões de todas as tabelas criadas pelo arquivo “install.sql”. Os dois arquivos são gerados nos padrões aceitos pelo CMS.

O arquivo XML (Figura 8) contém dados referentes à instalação do componente, como nome do autor, nome do componente, nome dos arquivos de criação e exclusão de tabelas do banco. E, no arquivo PHP, contém toda a lógica de desenvolvimento do componente em si, com base nas informações fornecidas pela ferramenta. Tanto o arquivo XML como o PHP são gerados também nos padrões aceitos pelo CMS.

```
<?xml version="1.0" encoding="utf-8"?>
<install version="1.5.0" type="component">
  <name>VENDAS</name>
  <author>RENATO CAMPOS</author>
  <version>1.5.X</version>

  ..............
</install>
```

Figura 8. Arquivo XML

4.2. Instalando o Componente de Vendas

Com o componente já gerado, a sua instalação se torna simples. Deve-se acessar a Área Administrativa do CMS, ir no menu “Extensions”, e logo após em “Install/Uninstall”. Como foi gerada uma pasta com os arquivos de instalação, é necessária (não obrigatória) a opção “Install from Directory” (Figura 9). Deve-se informar o caminho de onde se localiza a pasta gerada e acionar o botão “Install”. Logo em seguida uma mensagem de sucesso é informada.

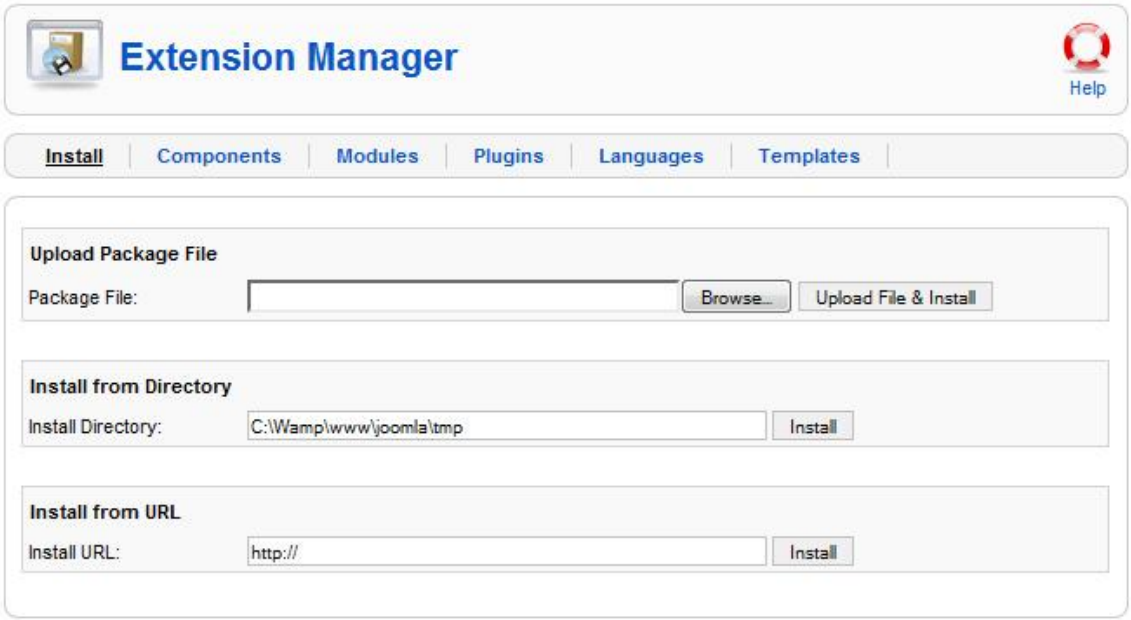


Figura 9. Instalação do Componente

Com o componente devidamente instalado, seu acesso ainda está limitado a criação de um menu para ele. A criação de um menu vinculado ao componente também é realizada pela Área Administrativa, no menu “Menus→Menu Manager”, logo em seguida deve-se escolher em qual menu deseja-se adicionar o *link* para o componente gerado e deve-se acionar “Edit Menu Item(s)”, e logo em seguida “New”,

para poder vincular um menu ao componente instalado.

5. CONCLUSÕES

A existência de uma ferramenta que atua na criação de componentes CRUD facilitará o trabalho de muitos usuários que utilizam o CMS JOOMLA! devido ao menor tempo despendido para desenvolvimento de componentes para atender as diversas necessidades.

Além do tempo, outros fatores podem ser incluídos como despesas com desenvolvedores e despesas para adquirir conhecimentos necessários para o desenvolvimento.

A ferramenta foi desenvolvida no intuito de ajudar usuários com poucos conhecimentos tanto no CMS, quanto em lógica de programação. O usuário então, não possui contato com nenhuma parte interna do CMS que exige conhecimento do seu *framework*. Com isso o proveito tirado do CMS pode ser ainda maior, com menor conhecimento e tempo.

REFERÊNCIAS

Documentation and About WampServer, disponível em http://www.wampserver.com/en/, 2009.

Documentation and About JOOMLA!. Disponível em http://www.joomla.org. Acessado em 20 de março de 2009.

Documentation and About APACHE. Disponível em http://apache.org. Acessado em 18 de março de 2009.

Documentation and About MySQL. Disponível em http://www.mysql.com. Acessado em 22 de março de 2009.

Documentation and About PHP. Disponível em http://php.net. Acessado em 18 de março de 2009.

Jassinek, R. A Influência da Internet em Negócios Empresariais: Identificação e Caracterização de Elementos para Análise de Sites. Dissertação (Mestrado em Administração) 163 p. Programa de Pós-Graduação em Administração, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2000.

Sobre CMS. Disponível em http://pt.wikipedia.org/wiki/Cms. Acessado em 18 de março de 2009.

Sobre Localhost. Disponível em http://pt.wikipedia.org/wiki/Localhost. Acessado em 4 de abril de 2009.

Sobre MVC. Disponível em http://pt.wikipedia.org/wiki/MVC. Acessado em 14 de maio de 2009.

Sobre SGBD. Disponível em http://pt.wikipedia.org/wiki/Sgbd. Acessado em 21 de março de 2009.

Soares, W. PHP 5: conceitos, programação e integração com banco de dados. São Paulo, Editora Érica, 2008.