

Software para integração de *feeds* e sistemas de gerenciamento de conteúdo

Marcelo Toledo Rafael Castilho

19 de janeiro de 2009

1 Introdução

Sistemas de gerenciamento de conteúdo (*CMS*), em particular os populares *blogs*, são ferramentas que permitem a publicação de conteúdo em um endereço *WEB*, e que exigem apenas o conhecimento do uso da *interface*.

Em geral, os *CMSs* permitem a visualização de itens de conteúdo publicado na forma parametrizada *RSS* por meio de canais alimentadores (*feeds*). Uma das ferramentas mais utilizadas pelos *blogueiros* é o agregador de *RSS*, capaz de coletar e organizar o conteúdo de diversos canais simultaneamente.

Em um *blog*, além de conteúdo próprio, é comum a citação de conteúdo de terceiros, realizado de forma "artesanal", através da cópia das informações disponíveis na página *WEB* ou no agregador e da colagem no editor de publicação do *CMS*.

O objetivo do software a ser desenvolvido é disponibilizar um agregador de canais que permita a publicação de itens de conteúdo no *CMS* do cliente de forma ágil, padronizada, e com base em suas preferências.

2 Especificação funcional

Esta especificação funcional tem como objetivo detalhar o funcionamento da aplicação, sem detalhar questões de usabilidade.

2.1 Cadastro de perfil de acesso

Para ter acesso ao software, o cliente deve efetuar a autenticação de um perfil de acesso a partir de um endereço de e-mail e senha cadastrados.

2.1.1 Novo perfil

O cadastro de um novo perfil de acesso é feito a partir do formulário de autenticação, através da opção "não cadastrado", que quando acionada, passa a exibir o formulário para um novo cadastro.

O diagrama mostra dois formulários lado a lado, separados por uma linha vertical. O formulário à esquerda, intitulado 'Autenticação', possui campos para 'E-mail:' e 'Senha:', um link 'não cadastrado' e um botão 'Enviar'. O formulário à direita, intitulado 'Novo cadastro', possui campos para 'E-mail:', 'Senha:' e 'Confirmar senha:', e um botão 'Enviar'. Uma seta preta aponta do formulário de autenticação para o de novo cadastro, indicando a transição de um usuário não cadastrado para o processo de criação de uma nova conta.

O procedimento para criação de um novo perfil a partir do formulário para um novo cadastro deve ser feito como segue:

1. Informar um endereço de e-mail, senha e confirmação de senha;
2. Se o endereço de e-mail não estiver cadastrado em um perfil de acesso, o cliente deverá receber uma mensagem de e-mail contendo uma *URL* de confirmação de cadastro;
3. Após a confirmação de cadastro, o cliente deverá receber uma mensagem de e-mail contendo uma notificação de cadastro confirmado e instruções para o primeiro acesso.
4. Se o endereço de e-mail informado já estiver cadastrado, o cliente deverá receber uma mensagem de e-mail contendo uma notificação da existência do perfil com instruções para acesso e alteração de senha (se necessário).

2.1.2 Atualização de perfil

Após efetuar o acesso, o cliente poderá alterar suas informações de acesso, preencher seus dados pessoais (nome, país, região, cidade, distrito, endereço, código postal, etc.), informar seu idioma, etc.

Para alteração do endereço de e-mail, o cliente deverá efetuar o seguinte procedimento:

1. Acionar o comando de alteração de endereço de e-mail, que enviará uma mensagem ao endereço atual com instruções para alteração.
2. A mensagem deve conter uma *URL* que permite acesso ao formulário de alteração de endereço de e-mail, onde o cliente deve informar o novo endereço.
3. Uma mensagem deve ser enviada ao antigo e novo endereços notificando a alteração.

2.1.3 Recuperação de acesso

O software deve permitir ao cliente recuperar o acesso quando não recordar sua senha, através do seguinte procedimento:

1. Informar o endereço de e-mail do perfil de acesso;
2. Se o endereço de e-mail estiver cadastrado em um perfil de acesso, o cliente receberá uma mensagem de e-mail com uma *URL* para o formulário de alteração de senha;
3. O formulário de alteração de senha deve permitir ao cliente digitar uma nova senha. Após a alteração, o software deverá enviar uma mensagem de e-mail ao cliente com a notificação de alteração.
4. Se o endereço de e-mail informado não estiver associado a um perfil de acesso, o cliente deverá receber uma mensagem de e-mail com a notificação da não existência do perfil de acesso e instruções para a criação de um novo.

2.2 Cadastro de *CMS*

2.2.1 Novo *CMS*

Após efetuar acesso, o cliente poderá realizar o cadastro de um novo *CMS* através das seguintes etapas:

1. Informar o nome do *CMS*;
2. Informar a *URL* raiz do *CMS*. O software deverá detectar o tipo de *CMS* (*Wordpress*, *Joomla*, *Drupal*, etc.) a partir da *URL* informada;
3. Se for possível a detecção do tipo de *CMS*, o software deverá verificar a existência do gerenciador de publicações do *CMS* em um endereço padrão, de acordo com o tipo de *CMS* e da *URL* raiz;
4. Se o gerenciador de publicações do *CMS* não estiver disponível no endereço padrão, permitir que o cliente informe seu endereço através de uma *URL*. O software deverá verificar se o endereço informado é válido como gerenciador de publicações para o tipo de *CMS* escolhido;
5. Se a detecção do tipo de *CMS* não for bem sucedida, verificar a causa e informar o erro mais provável (endereço ou página não encontrados, erro no servidor, etc.);

6. Se o endereço estiver correto mas o tipo não for suportado, exibir notificação ao cliente e informar os tipos suportados. Permitir também que o cliente digite o tipo utilizado. Neste caso, o cadastro de *CMS* é descontinuado;
7. Após a determinação correta do tipo de *CMS* e endereço do gerenciador de publicações, solicitar ao cliente que informe o usuário e senha do gerenciador. O software deverá verificar se o usuário e senha permitem autenticação no gerenciador, caso contrário, notificar ao cliente que o usuário e senha não são válidos;
8. O software deve verificar de forma não invasiva (sem publicar conteúdo em produção) se o acesso ao gerenciador de publicações possui permissão para escrita de conteúdo (ex. *drafts* do *Wordpress*). Se não for possível tal verificação, o software poderá solicitar ao cliente se deseja realizar um teste de publicação em produção.

2.3 Cadastro de canal *RSS*

O cadastro de canais *RSS* deverá pertencer a um *CMS* já cadastrado.

Para adicionar um novo canal, o cliente deve informar a *URL* do alimentador ou um arquivo de importação *OPML*.

O título de um novo canal é definido pelo gerenciador de conteúdo. Posteriormente o cliente poderá informar o título de sua preferência.

Durante a importação de canais em um arquivo *OPML*, o título do canal será o mesmo definido no arquivo de importação.

2.4 *Dashboard*

O *dashboard* é a visualização padrão após a autenticação, composto por diversos painéis que permitem o gerenciamento das informações.

CMS		
Tecnologia		
Culinária		
Marcelo Toledo		

Basics of cooking items	
Vegan Barbecue, Easy Unusual BBQ Grilled Recipes	12/09/2008 10:02 PM
Vegan Easy Make Chilli Recipe With Eggplant	12/08/2008 08:50 AM
Easy Christmas Homemade Edible Food Gifts	12/05/2008 03:04 PM
Wartime Vegetarian Low-Cost Meals	12/02/2008 07:26 AM
Vegetarian Tortilla Soup Recipe	11/24/2008 09:07 PM

Culinária's Queue	
Easy Christmas Homemade Edible Food Gifts	12/09/2008 10:02 PM
Christmas diner starters	11/11/2008 04:02 PM
Recipes for Diabetics	11/08/2008 05:18 AM

RSS
Sushi of the day
Vegetarian cooking
Basics of cooking
George Foreman Blog

2.4.1 Painel de *CMS*

O painel de *CMS* contém a lista de *CMS*s cadastrados. A exibição das informações nos demais painéis depende do *CMS* selecionado.

2.4.2 Painel de canais *RSS*

O painel de *RSS* contém a lista de canais *RSS*s cadastrados para o *CMS* selecionado.

Novos canais são adicionados ao final da lista. Deve ser possível determinar a ordem dos canais através de *drag and drop* ou por ordem alfabética.

Um canal deve poder ser desativado ou removido.

2.4.3 Painel de itens de conteúdo

O painel de itens de conteúdo exibe a lista de itens de todos os canais ativos. Quando um canal é selecionado, exibe apenas os itens do respectivo canal.

O cliente poderá determinar a quantidade n de itens que devem aparecer na lista por vez. Serão carregados os últimos n itens a partir da data de publicação em ordem decrescente.

A relevância de um item de conteúdo é obtida através do número de vezes em que o item é publicado, envolvendo a estatística de todos os clientes que utilizam o software. Deve ser exibido na forma de um indicador no item.

Um item poderá ser marcado com uma estrela, para que possa ser filtrado posteriormente pelo método de alimentação.

Para visualizar um item, o cliente deve acionar o comando "visualizar", que exibirá o diálogo de visualização.

Um item pode ser adicionado ao *Queue* através de *drag and drop* ou através do comando "adicionar ao *Queue*". Um marcador deve indicar quando um item deste painel encontra-se disponível no painel de *Queue*.

O método de alimentação define as regras de automação na adição de itens deste painel ao painel de *Queue*.

2.4.4 Painel de *Queue*

O painel de *Queue* contém a lista de itens de conteúdo selecionados para publicação.

A publicação de itens manualmente é feita através do diálogo de publicação, acionado pelo cliente através da seleção de um item do *Queue* e do comando "abrir diálogo de publicação".

Novos itens são adicionados ao final da lista. A ordem dos itens pode ser alterada através de *drag and drop*.

O método de publicação define as regras de automação na publicação de conteúdo no *CMS*.

2.4.5 Diálogo de visualização

O diálogo de visualização exibe todas as informações disponíveis para um item de conteúdo (título, data de publicação, fonte, etc.), além de um *link* para visualização do conteúdo no endereço de origem.

2.4.6 Método de alimentação

O método de alimentação é um conjunto de configurações que determinam a forma como um item da lista de itens de conteúdo é adicionado ao painel de *Queue*.

A primeira configuração define o estado de alimentação, que pode assumir os valores "manual" ou "automático". No estado manual, os itens de conteúdo são adicionados ao *Queue* somente através da ação do cliente.

O estado automático atribui ao software a responsabilidade na adição de itens ao *Queue*, e para isso, solicita ao cliente as seguintes configurações:

- Com estrela: Somente itens marcados com estrela no painel de itens de conteúdo serão utilizados para adição. Padrão: Todos;
- Palavras-chave: A escolha de itens é feita com base em uma ou mais palavras-chave. Padrão: Em branco;
- Ordenação: Define a ordem de adição de itens a partir da data de publicação, relevância ou randômico. Padrão: Data de publicação;
- Máximo: Quantidade máxima de itens que podem ser adicionados ao *Queue*. Enquanto o número de itens do *Queue* não for menor ou igual ao número de itens definido por esta configuração, novos itens não poderão ser adicionados ao *Queue*. Padrão: 5 itens. Máximo: 25 itens;

A adição de itens ao *Queue* manualmente sempre deve estar disponível, sem considerar as configurações de estado.

2.4.7 Método de publicação

O método de publicação é um conjunto de configurações que determinam a forma como um item de conteúdo do painel de *Queue* é publicado no *CMS*.

De forma análoga ao método de alimentação, o método de publicação pode assumir os valores de estado "manual" ou "automático". No estado manual, um item somente é publicado através da ação do cliente.

O estado automático atribui ao software a responsabilidade na publicação de itens no *CMS*, e para isso, solicita ao cliente as seguintes configurações:

- Frequência: Define o intervalo de tempo entre as publicações (ex: 5min 15min, 30min, 1hora, 2hrs, 3hrs, 6hrs, 12hrs, 24hrs). Padrão: 30min;

No estado automático, a ordem dos itens no *Queue* determina a ordem de publicação. Os itens do início da lista são publicados em primeiro.

A publicação de itens no estado automático é gerenciada pelo gerenciador de publicação.

A publicação de itens ao *Queue* manualmente sempre deve estar disponível, sem considerar as configurações de estado.

2.4.8 Diálogo de publicação

O diálogo de publicação exibe as informações de um item de conteúdo (título, data de publicação, fonte, etc.) e permite a alteração dos valores, além de um campo para comentários adicionais.

Deve estar disponível uma opção para indicar a fonte do conteúdo, que quando selecionada, inclui no final do conteúdo um *link* para a notícia no endereço de origem.

A publicação do conteúdo é feita através do comando "publicar".

É possível armazenar as alterações feitas em um item através do comando "salvar", sem necessariamente publicá-lo.

Informações adicionais podem ser incluídas na publicação de conteúdo (Categorias, *Tags*, etc.), dependendo somente da implementação dos recursos necessários no *plugin* do tipo de *CMS* utilizado.

2.5 *Back-end*

O *back-end* é o conjunto de ferramentas do software responsável por gerenciar os recursos de forma independente ao cliente.

2.5.1 Tipo de *CMS*

O tipo de *CMS* define qual software é utilizado pelo cliente. Os tipos disponíveis serão adicionados ao software na forma de *plugins*, e devem implementar uma interface determinada pelo software.

Um *plugin* de *CMS* deve ser capaz de gerenciar as diferentes versões de um tipo de *CMS*.

2.5.2 Gerenciador de conteúdo

Os itens de conteúdo de um canal podem ser compartilhados por todos os clientes do software, afim de eliminar redundância desnecessária de informações.

Quando um novo canal é adicionado, o sistema obtém as informações sobre o canal (título, descrição, etc.) e os itens de conteúdo disponíveis.

Um *daemon* deverá obter novos itens de conteúdo para um canal cadastrado, com uma frequência em função da periodicidade de atualização do conteúdo no alimentador deste canal.

2.5.3 Gerenciador de publicação

O gerenciador de publicação possui um *daemon* responsável pela publicação de conteúdo de um *Queue* em um *CMS* quando o estado do método de publicação está definido como automático.

3 Notas de Implementação

A linguagem escolhida para o desenvolvimento da aplicação foi o **PHP**. O *core* foi inspirado no *Zend Framework*, que segue uma estrutura no padrão **MVC**, confirme a seguir:

- **application**: arquivos da aplicação no padrão **MVC**;
 - controller**: controle de ações;
 - model**: classes de **ORM**;
 - view**: visualização;
 - helper**: *scripts* auxiliares de visualização;
 - layout**: diagramação geral (*layout*);
 - template**: diagramação de ações;
- **config**: arquivos de configuração;
- **db**: arquivos de base de dados (**SQL**);
- **library**: arquivos de bibliotecas;
 - AB**: *core MVC*;
 - Zend**: *Zend Framework*;
- **log**: arquivos de histórico;
- **public**: *bootstrap*, deve ser visível para *WEB*;

4 Instalação

4.1 Ambiente

O sistema operacional deve ser compatível com *LINUX* ou *UNIX*, deve ter instalado o servidor de *HTTP Apache 2* com *mod_php* e servidor de base de dados não mandatório, porém recomendável, *PostgreSQL*.

A configuração do apache deve ser feita de forma semelhante ao exemplo à seguir:

```
Listen 127.0.0.1:80
ServerName 127.0.0.1
<VirtualHost 127.0.0.1:80>
    DocumentRoot /var/www/autoblog/public
    <Directory /var/www/autoblog/public>
        AllowOverride All
        allow from all
    </Directory>
    ErrorLog /var/log/apache2/autoblog.error.log
    LogLevel warn
</VirtualHost>
```

Neste caso, a raiz da aplicação encontra-se no diretório `/var/www/autoblog` e somente a pasta (*DocumentRoot*) `./public` estará visível para a *WEB*.

4.2 Configuração

A configuração da aplicação é feita a partir do arquivo `./config/environment.php`, onde os seguintes parâmetros deverão ser ajustados:

- **BASE_PATH**: Localização da raiz da aplicação no disco. Ex: `/var/www/autoblog`;
- **BASE_URL**: Localização da raiz da aplicação na *WEB*. Ex: `http://www.autoblog.com`;
- **database->driver**: *Driver PDO* da base de dados. Padrão: `pgsql` (*PostgreSQL*). Para outros drivers, verifique em <http://br2.php.net/pdo>;
- **database->host**: Endereço *IP* ou *hostname* do servidor de base de dados. Ex: `localhost`;
- **database->username**: Nome de usuário. Ex: `postgres`;

- **database->password:** Senha. (Quando não houver senha, deve conter o valor do tipo *string* e vazio);
- **database->db:** Nome da base de dados. Ex: `autoblog`;

Todos os caminhos acima devem suprimir o separador `"/` no final.

4.3 Base de dados

A base de dados deverá estar de acordo com o definido na configuração. Em seguida, deve-se carregá-la com as definições de estrutura e informações padrão:

```
$ cd ./db
$ php configure.php schema.sql
$ php configure.php data.sql
```

Caso a utilização do *script* `configure` acima retorne algum erro, o processo de carga deve ser feita manualmente, respeitando a ordem dos arquivos acima.

4.4 Permissões

Deve-se ajustar a permissão dos arquivos abaixo, conforme segue:

```
$ chmod a+w ./log/error_log
```

A Observações

A.1 *Add-ons*

Add-ons são recursos adicionais que podem ser incluídos em versões futuras, incluídos neste documento apenas como idéias gerais sem a preocupação de incluir detalhes de implementação.

É possível unificar (*merge*) dois ou mais itens do *Queue*. Para isto, é necessário selecionar os itens correspondentes e acionar o "comando de unificação de itens", gerando um novo item unificado. Um diálogo deverá ser exibido solicitando o título deste novo item unificado. Por padrão o título sugerido é o primeiro item selecionado. O novo item deve conter uma marcação de unificado. Deve ser possível desfazer a ação através do comando "separar itens unificados".

Quando um tipo de *CMS* não for suportado, e o cliente informar o tipo de *CMS* utilizado, o *backend* poderá verificar se houve falha na detecção e solicitar ajuste no desenvolvimento, e informar ao cliente sobre a resolução do problema. Se o tipo não for suportado, informar ao cliente que o software não é suportado. É possível também verificar quais são as tendências para implementações de novos tipos.

A.2 Segurança

O software deve limitar a quantidade de mensagens de e-mail enviadas durante a solicitação de um novo cadastro de perfil ou durante a recuperação de acesso a um perfil de acesso, afim de evitar uso mal intencionado do envio de mensagens.

O software deve conter algum tipo de proteção para que as publicações em um *CMS* não sejam feitas por seu próprio alimentador automaticamente e indefinidamente (autófago).