

Ficha Catalográfica

S725c

Criando Componentes Comerciais para Joomla! 1.5 e 1.7 / Ribamar
Ferreira de Sousa. - Fortaleza: Clube de Autores, 2010

128p.

1. Criação de Sites. 2. CMS Joomla!. Criação de Componentes. I.
Título.

Normalização Bibliográfica

Lúcia Maria Piacó Chaves – DNOCS-CGE/MD
Margarida Lúcia de Abreu Vieira – DNOCS-CGE/MD

RIBAMAR FERREIRA DE SOUSA



Ribamar Ferreira de Sousa é funcionário público lotado no DNOCS – Departamento Nacional de Obras Contra as Secas, em Fortaleza no Ceará, onde trabalhou como desenvolvedor web para a intranet, em PHP com PostgreSQL. Já tendo participado de diversas atividades no órgão: suporte, treinamento, administração do SGBD PostgreSQL, etc. Hoje (05/2011) trabalha na administração da rede com servidores Linux.

Concluiu o curso de engenharia civil e o de especialização em irrigação e drenagem, também tendo iniciado e quase concluído o curso de especialização em Java. Apenas iniciou (primeiro ano) o curso de psicologia na UFC. Concluiu diversos cursos na área de desenvolvimento web através do CDTC (<http://cursos.cdtc.org.br>) e em outros (online e presenciais). Ministrou treinamento em desenvolvimento web na UNIFOR e em algumas escolas de Fortaleza, como na SIGMA (ministrou curso de Joomla para os servidores das secretarias do governo do Estado do Ceará). Curso de Administração de Servidores Linux na 4Linux em São Paulo.

Interesses atuais: treinamentos em PHP, em Joomla e PostgreSQL, criação de sites (em Joomla). Além de implantação de pequenos servidores Linux e Virtualização de grandes servidores.

Autor do segundo livro brasileiro sobre Joomla (Curso de Joomla), publicado pela editora Clube de Autores (http://clubedeautores.com.br/book/12802—Curso_de_Joomla), pela Bubok e pela Lulu.

Participa ativamente da comunidade Joomla, especialmente em:

- <http://groups.yahoo.com/group/ajudajoomla>
- <http://groups.google.com/group/joomlaceara>
- <http://forum.joomla.org/viewforum.php?f=23&sid=040e5f737feb0680a18091a1a13e1aa4>
Site pessoal – <http://ribafs.org>

Fortaleza, Ceará, Brasil, 22 de novembro de 2011.

Agradecimentos

Gostaria de agradecer aqui à equipe de desenvolvimento do Joomla, que criou esta maravilha de software e a todos que de alguma forma colaboram para que o Joomla seja hoje o que é, através de livros, treinamentos, tutoriais, participação em listas, forums ou de outra forma.

Muito obrigado a todos vocês.

André

Um agradecimento especial ao colega André Coelho dos Santos por criar as capas deste livro e do livro Curso de Joomla.
Valeu André.

Margarida e Lúcia

Também aproveito para agradecer outra generosa colaboração, das também colegas de trabalho no DNOCS, Lúcia Maria Piancó Chaves e Margarida Lúcia de Abreu Vieira, que cuidaram da normalização deste livro e me ensinaram como fazer isso de hoje em diante. Embora eu não tenha cumprido a risca as recomendações das colegas mas estou muito agradecido.

Prefácio

Sobre homens e códigos

“A melhor maneira de dizer é fazer”

José Martí

“Nada existe de grandioso sem Paixão”

Hegel

As duas frases acima definem bem o Ribamar e sua relação com o software livre, quiçá com o mundo. É um apaixonado pelo conhecimento, pela liberdade de conhecimento e é um daqueles que dizem fazendo, com exemplos concretos.

Este livro é sem dúvida mais uma de suas valiosas contribuições para aqueles que utilizam o Joomla e desenvolvem em PHP. Muito provavelmente, se você chegou até aqui é porque já conhece o Joomla. Assim, não tenho intenção em me alongar nos aspectos técnicos, até porque o próprio Ribamar já detalha o conteúdo e a que se propõe o **“Criando Componentes Comerciais para Joomla 1.5”** na introdução do livro. Mas é necessário dizer que esta não é sua primeira contribuição e esperamos que não seja a última na área do desenvolvimento. O Riba, como é conhecido pelos amigos, já publicou diversos trabalhos tais como livros, wikibooks, tutoriais, cursos online etc. que versam sobre Joomla, Drupal, PostgreSQL, PHP, MySQL, desenvolvimento de aplicativos, frameworks, SOs... Enfim, uma longa lista. Não deixe de acessar o **www.ribafs.org**.

Gostaria de aproveitar este espaço que me foi tão gentilmente cedido, primeiro para prestar uma justa homenagem, na pessoa do Ribamar, a todos que contribuem de alguma forma com o software livre, impedindo que o conhecimento em uma área tão importante seja monopólio da cega lógica do lucro, que não visa o bem estar das pessoas e está levando o mundo à destruição.

Também gostaria de propor uma pequena reflexão: qual a responsabilidade social hoje que têm os desenvolvedores? Obviamente, na era das comunicações e quando vivemos o que Guy Debord qualificou de sociedade do espetáculo, não podemos

acreditar que nosso papel é somente escrever código ou utilizar despreocupadamente ferramentas de imenso poder, como é o próprio Joomla. Os códigos que escrevemos, os aplicativos, sites e portais que criamos, os dados que ajudamos a converter em informação, as informações que ajudamos a difundir, estão de alguma forma contribuindo para um mundo diferente do atual circo de horrores? Não sou ingênuo em acreditar que devemos agir como o lendário São Francisco. Vivemos em um mundo capitalista e precisamos, no mínimo, sobreviver. Mas virar as costas à nossa responsabilidade enquanto criadores e difusores de conhecimento, mais que isso, enquanto seres humanos, e simplesmente nos contentarmos em ser(mos)vis que se afogam nas marés do consumismo, do conformismo e da inconsciência, é demais. Não escrevemos código para máquinas, escrevemos para pessoas e, ao fazermos isso, devemos ter em conta o bem estar delas, que afinal nada mais são que nós próprios.

O livro que você tem em mãos (ou na tela) pode ser um bom exemplo de como isso é possível.

Como na maioria de suas empreitadas, o Ribamar neste livro, busca voar alto. Isso já fica patente em uma das frases da introdução:

Agora podemos então abrigar no CMS Joomla qualquer aplicativo em PHP. Mais um desafio. Vamos aceitá-lo?

Haroldo Barbosa

Jornalista, desenvolvedor web e usuário do Joomla

Criando Componentes Comerciais para Joomla 1.5 e Joomla 1.7

Índice

Introdução	11
Criando Componentes Comerciais	13
Exemplos de Componentes Comerciais	15
Cadastro de Clientes	15
Detalhando cada Arquivo do Cadastro de Clientes	19
Seção Admin	19
admin.clientes.php	19
admin.clientes.html.php	20
controller.php	23
toolbar.clientes.php	29
toolbar.clientes.html.php	29
install.sql	30
uninstall.sql	31
tables/clientes_users.php	31
clientes.xml	32
Seção site	34
clientes.php	34
clientes.html.php	43
page.inc.php	49
images	49
Controle de Estoque	51
estoque.php	52
estoque.html.php	57
clientes.php	59
clientes.html.php	67
Gerador de Componentes Comerciais	75
generator_admin.php	76
generator_site.php	89
xml.php	105
Versões para Joomla 1.7	108
Referências	110
Apêndice	113
Sobre a publicação deste livro	116

Introdução

O objetivo deste livro é o de mostrar a criação de dois componentes do tipo comercial, sendo um do tipo cadastro de clientes, apenas com uma tabela e outro tipo controle de estoque, com cinco tabelas. Dois componentes para Joomla 1.5 e dois para Joomla 1.7. Também veremos um gerador de componentes para componentes com uma única tabela somente para Joomla 1.5.

Empresas e instituições em geral tem seus aplicativos de cadastro e controle, como cadastro de funcionários, cadastro de clientes, cadastro de vendedores, cadastro de representantes, controle de estoque, etc.

São aplicativos muito importantes para elas.

Hoje contamos com um CMS que permite criar portais profissionais com grande facilidade e se pudéssemos criar também aplicativos integrados aos portais para termos intranets com a interface do Joomla seria muito interessante.

Bem, juntei minha experiência na criação de aplicativos em PHP com o conhecimento do Framework do Joomla e criei componentes que abrigam aplicativos do tipo comerciais no Joomla. Com isso podemos criar novos aplicativos para o Joomla e também migrar aplicativos existentes em PHP e em outras linguagens para usar com o Joomla bastando convertê-los em componentes, assim aproveitando toda a estrutura de templates, cadastro de usuários integrado entre outros recursos do Joomla.

Podemos então abrigar no CMS Joomla qualquer aplicativo em PHP.

Veja em seguida detalhes de como criar esses componentes.

Vale lembrar que aqui não estarei mostrando como criar componentes para o Joomla nem como usar bem o Framework do Joomla, mas especificamente como criar um tipo de componente, o componente comercial. É recomendado pelo menos um conhecimento inicial da linguagem PHP.

Vale lembrar também que este não é bem um livro, daqueles que irá ensinar teoria e prática sobre a criação de componentes comerciais, mas um guia eminentemente prático sobre a criação dos referidos componentes.

Fica também o alerta de que os componentes como estão aqui ainda precisam ser melhorados para ser colocados em produção. Melhorar o tratamento de erros. Se for o caso, melhorar o tratamento de acesso via session.

Analise e veja o que precisa melhorar, pois isso também depende de quem vai usar.

Sugestões serão realmente bem-vindas.

Criando Componentes Comerciais para Joomla 1.5

Aqui abordarei a criação de dois componentes, um componentes cadastro de clientes (com apenas uma tabela para o cadastro dos clientes e uma para o cadastro dos usuários autorizados a usar o componente na parte administrativa do Joomla 1.5) e um pequeno controle de estoque, com cinco tabelas para o controle mais uma para o cadastro dos usuários que terão acesso ao componente. Também mostrarei um gerador de componentes que trabalham com apenas uma tabela, onde você apenas cria a tabela e indica ao gerador e ele faz o resto.

Cadastro de Clientes para Joomla 1.5

Controle de Estoque para Joomla 1.5

Gerador de Componentes Comerciais para Joomla 1.5

A meu ver a seção de administração do Joomla não é indicada para acesso do usuário comum, mas apenas para os administradores.

Nem em termos de segurança, nem em termos de facilidade de uso, como também o fato de que o usuário terá acesso a muita coisa que talvez não deva.

Mesmo que o Joomla 1.7 nos traga um bom controle do usuário, é mais prático criar os aplicativos na seção do site e apenas controlar os usuários no admin.

Pensando nisso a parte principal dos componentes comerciais fica na seção do site, bastando que o usuário autorizado faça login para que seja identificado e o componente seja mostrado para ele.

Alguns recursos dos aplicativos criados:

- Operações básicas do CRUD com grid
- Busca
- Paginação
- Programação em duas camadas

Para o Joomla 1.5 criei código para controlar os usuários na seção admin.

A programação para o controle dos usuários no admin no Joomla 1.5 contam com:

- JTable
- Toolbar
- Paginação e os vários recursos padrões do admin do Joomla 1.5
- Uso do padrão de projeto MVC de forma simplificada.

Programação dos componentes na seção do site:

A programação em duas camadas organiza o código separando apresentação (correspondente ao V-view do MVC) da lógica, que engloba o M (model) e o C (controller) do MVC, ou seja, fica com a lógica em PHP e o manuseio do banco. Além de organizar, simplifica muito a codificação comparando com a programação que usa o padrão MVC no Joomla 1.5. Essa programação em duas camadas é indicada apenas para projetos de pequeno e médio porte.

Assim podemos usar o Joomla não somente para criar portais, mas com o conhecimento do Framework e seus recursos podemos usar o Joomla também em Intranets, abrigando tudo quanto é tipo de aplicativo.

Alerta sobre Relacionamentos

Cuidado ao criar tabelas relacionadas no MySQL, pois alguns servidores de hospedagem não tem o recurso de tabelas innodb habilitado.

Exemplos de Componentes Comerciais

Exemplo de Componente com uma única tabela

Cadastro de Clientes

O componente terá uma seção admin, onde serão cadastrados os usuários que terão acesso ao componente e uma seção no site, que conterá realmente o cadastro dos clientes, a parte principal do componente.

O cadastro conterá uma tabela com os campos:

```
jos_clientes
{
  id int(11) auto_increment not null primary key,
  cpf char(11),
  nome char(45),
  credito_liberado char(1),
  data_nasc date,
  email varchar(50)
};
```

No admin teremos uma tabela com apenas os campos:

```
jos_clientes_user
{
  id int(11) auto_increment not null primary key,
  login char(32),
  name char(45),
  published tinyint(1) unsigned NOT NULL DEFAULT '0'
};
```

Altere o prefixo jos_ se necessário.

Uma boa prática é que todas as tabelas do nosso componente contenham seu nome como prefixo, para que sejam identificadas com facilidade.

Observe que ambas as tabelas devem ter como campo chave id e

que este será chave primária e com `auto_increment`.

Isso para tirar vantagens do Framework do Joomla. Ou seja, existe um acordo entre o Framework do Joomla e o programador: o programador cumpre com algumas regras e o Framework facilita sua vida em alguns aspectos.

Todo o cadastro acontecerá na seção do site. Após o login o usuário identificado terá acesso ao componente Cadastro de Clientes.

Este componente será desenvolvido basicamente em duas camadas, uma representada pelo arquivo `clientes.php` e a outra pelo arquivo `clientes.html.php`. A primeira fica com a lógica e o acesso ao banco (M e C do MVC) e a última fica com a apresentação (V do MVC). Assim temos uma programação organizada e bem mais simples de codificar que no MVC usam estas ideias e informações.

Claro que esta programação tem suas limitações. Caso precise desenvolver um componente onde esse padrão não atenda a você então use o MVC.

Componente para Cadastro de Clientes

O componente será chamado `com_clientes` e terá a seguinte estrutura de arquivos:

admin/
site/
install.xml
leiametext
projeto.txt

install.xml - arquivo instalador, que contém informações sobre o tipo de extensão, a codificação de caracteres, o autor, a licença, a descrição, os arquivos, parâmetros, etc.

leiametext - contém informações para instalação e uso do

componente e outras que o autor achar necessárias.

projeto.txt - este arquivo, que contém informações sobre o projeto do componente, servirão para orientação durante a criação do componente.

admin/

language/

tables/

admin.clientes.html.php

admin.clientes.php (ponto de entrada do componente)

controller.php

index.html

install.sql - strings de criação da tabela quando o componente for instalado

toolbar.clientes.html.php

toolbar.clientes.php

uninstall.sql - strings SQL para remover a tabela quando da desinstalação

language/

en-GB/

en-GB.ini

pt-BR/

pt-BR.ini

tables/

clientes_users.php

site/

images/ - imagens da paginação

language/ - similar ao language do admin

clientes.html.php

clientes.php

index.html

page.inc.php - script de paginação

Detalhando cada Arquivo do Cadastro de Clientes

Não irei comentar todo o código, mas somente pontos que considera importantes e que tenham alguma particularidade relativa aos nossos componentes comerciais. A parte da administração basicamente não será comentada tendo em vista que é semelhante às de outros componentes para Joomla 1.5. A parte do site é que realmente é o código exclusivo dos nossos componentes comerciais.

Como também não irei comentar o código PHP nem sua construção, pois aqui suponho que já seja conhecido. Apenas comentarei trechos que mostrem algo específico sobre os componentes comerciais no Joomla.

Por conta do controle de session obrigatoriamente o nível de acesso deve ser especial ou registrado, caso contrário o componente não aparecerá.

Index.html

Este arquivo tem como objetivo esconder a listagem de um diretório e deve estar em cada diretório do componente. Seu conteúdo é apenas a linha abaixo:

```
<html><body bgcolor="#FFFFFF"></body></html>
```

Seção admin

admin.clientes.php (ponto de entrada do componente no admin)

```
<?php
    defined( '_JEXEC' ) or die( 'Acesso restrito ao Joomla!');

    require_once( JApplicationHelper::getPath( 'admin_html' ) );
    require_once( JPATH_COMPONENT.DS.'controller.php' );

    JTable::addIncludePath(JPATH_COMPONENT.DS.'tables');
    $controller = new ClientesController( array('default_task' =>
```

```
'displayUsers') );

    $controller->execute( JRequest::getVar( 'task' ) );
    $controller->redirect();
?>
```

O script acima tem código semelhante ao de qualquer componente. Basicamente composto de includes para a parte admin do componente.

admin.clientes.html.php (view do arquivo acima)

```
<?php
defined( '_JEXEC' ) or die( 'Acesso restrito ao Joomla!');
class HTML_clientes {

    function editUsers($row, $option) {
        ?>
        <form action="index.php" method="post"
name="adminForm" id="adminForm">
            <fieldset class="adminform">
                <legend>Cadastro de Usuários</legend>
                <table>
                    <tr><td><label for="login">Login:</label></td>
                    <td><select class="select" name="login" id="login"
class="required"/>

                        <?php print ClientesController::combo();?>

                    </select>
                    <tr><td><label for="name">Name:</label></td>
                    <td><input class="text" type="text" name="name"
id="name" size="45" maxlength="45"
                        value="<?php echo $row->name; ?>"
class="required"/></td></tr>
                </table>
                <BR> <BR>
            </fieldset>
```

```

        <input type="hidden" name="id" value="<?php echo
$row->id; ?>" />
        <input type="hidden" name="option" value="<?php echo
$option; ?>" />
        <input type="hidden" name="task" value="" />
        <?php echo JHTML::_('form.token' ); ?>
    </form>
    <?php
    }
    function displayUsers( $option, &$rows, &$pageNav ) {
    ?>
        <form action="index.php" method="post"
name="adminForm">
            <table class="adminlist">
                <thead>
                    <tr>
                        <th width="20">
                            <input type="checkbox" name="toggle" value=""
onclick="checkAll(<?php echo count( $rows ); ?>);" />
                        </th>
                        <th class="title">Login</th>
                        <th class="title">Name</th>
                        <th width="15%">Published</th>
                    </tr>
                </thead>
                <?php

                jimport('joomla.filter.output');

                $k = 0;
                for ($i=0, $n=count( $rows ); $i < $n; $i++) {
                    $row = &$rows[$i];
                    $checked = JHTML::_('grid.id', $i, $row->id );
                    $published = JHTML::_('grid.published', $row, $i );
                    $link = JFilterOutput::ampReplace( 'index.php?
option=' . $option . '&task=edit&cid[]=' . $row->id );
                    ?>
                    <tr class="<?php echo "row$k"; ?>">
                        <td>

```

```

        <?php echo $checked; ?>
    </td>
    <td>
        <a href="<?php echo $link; ?>">
        <?php echo $row->login; ?></a>
    </td>
    <td>
        <?php echo $row->name; ?></a>
    </td>
    <td align="center">
        <?php echo $published; ?>
    </td>
</tr>
<?php
    $k = 1 - $k;
}
?>
</tfoot>
<td colspan="7"><?php echo $pageNav-
>getListFooter(); ?></td>
</tfoot>
</table>
<input type="hidden" name="option" value="<?php echo
$option; ?>" />
<input type="hidden" name="task" value="" />
<input type="hidden" name="checked" value="0" />
<?php echo JHTML::_('form.token' ); ?>
</form>
<?php
}
}
?>

```

Aqui tem uma particularidade. A chamada da função `combo()`. Essa função é definida no controller. Observaremos alguns detalhes quando analisarmos o arquivo `controller.php`. Ela popula o campo login do form com os logins dos usuários existentes.

controller.php (C do MVC)

```
<?php
defined( '_JEXEC' ) or die( 'Access denied!' );

jimport( 'joomla.application.component.controller' );

class ClientesController extends JController
{
    function __construct( $default = array() )
    {
        parent::__construct( $default );
        $this->registerTask( 'add', 'edit' );
        $this->registerTask( 'apply', 'save' );
        $this->registerTask( 'unpublish', 'publish' );
    }

    function combo($default_value=null)
    {
        $db =& JFactory::getDBO();
        $query='select username from #__users order by
username';
        $db->setQuery( $query);
        $db->query($query);
        $nr=$db->getNumRows();

        $n=0;
        while($n < $nr){
            $row=$db->loadResultArray();
            if($row[$n] == $default_value)
            {
                $combo .= "<option value=$row[$n]
selected='selected'>$row[$n]</option>";
            }else{
                $combo .= "<option
value=$row[$n]>$row[$n]</option>";
            }
            $n++;
        }
    }
}
```



```

    return $combo;
}

function edit() { // Edit or New
    global $option;
    $row =& jTable::getInstance('clientes_users', 'Table');
    $cid = JRequest::getVar( 'cid', array(0), "", 'array' );
    $id = $cid[0];

    if ($id != 0) { // Edit
        $row->load($id);
    } else { // New
        $row->login = JRequest::getVar( 'login', "", 'post', 'string',
JREQUEST_ALLOWRAW );
        $row->name = JRequest::getVar( 'name', "", 'post',
'string', JREQUEST_ALLOWRAW );
    }
    HTML_clientes::editUsers($row, $option);
}

function save() {
    global $option;
    global $mainframe;

    JRequest::checkToken() or die( 'Token Inválido' );

    $row =& jTable::getInstance('clientes_users', 'Table');

    if (!$row->bind(JRequest::get('post'))){
        echo "<script> alert('".$row->getError()."');
window.history.go(-1); </script>";
        exit();
    }

    $login = JRequest::getVar( 'login', "", 'post', 'string',
JREQUEST_ALLOWRAW );
    $name = JRequest::getVar( 'name', "", 'post', 'string',
JREQUEST_ALLOWRAW );

```

```

$row->login = $login;
$row->name = $name;

$erro = false;
if (trim($login) == "" || $login == null) {
    $erro = true;
    $msg = JText::_( 'Login is required!' );
    $mainframe->enqueueMessage($msg, 'error');
}

if (trim($name) == "" || $name == null) {
    $erro = true;
    $msg = JText::_( 'Name is required!' );
    $mainframe->enqueueMessage($msg, 'error');
}

if ($erro ) {
    return $this->execute('edit');
}

if (!$row->store()) {
    echo "<script> alert('".$row->getError()."');";
window.history.go(-1); </script>";
    exit();
}

switch ($this->_task) {
    case 'apply':
        $msg = 'Sucessufull changes!';
        $link = 'index.php?option=' . $option .
'&task=edit&cid[]='. $row->id;
        break;

    case 'save':
    default:
        $msg = 'Register writed';
        $link = 'index.php?option=' . $option;
        break;
}

```

```

    }

    $this->setRedirect($link, $msg);
}

function displayUsers()
{
    global $option, $mainframe;
    $limit = JRequest::getVar('limit', $mainframe-
>getCfg('list_limit'));
    $limitstart = JRequest::getVar('limitstart', 0);
    $db =& JFactory::getDBO();

    $query = 'SELECT count(*) FROM #__clientes_users';
    $db->setQuery( $query );
    $total = $db->loadResult();

    $query = 'SELECT * FROM #__clientes_users';
    $db->setQuery( $query, $limitstart, $limit );
    $rows = $db->loadObjectList();
    if ($db->getErrorNum()) {
        echo $db->stderr();
        return false;
    }
    jimport('joomla.html.pagination');
    $pageNav = new JPagination($total, $limitstart, $limit);
    HTML_clientes::displayUsers( $option, $rows, $pageNav );
}

function remove()
{
    global $option;
    $cid = JRequest::getVar( 'cid', array(), ", 'array' );
    $db =& JFactory::getDBO();
    if(count($cid))
    {
        JFactoryHelper::toInteger($cid);
        $cids = implode( ',', $cid );
    }
}

```

```

                $query = 'DELETE FROM #__clientes_users WHERE id
IN ( $cids );
                $db->setQuery( $query );
                if (!$db->query()) {
                    echo "<script> alert('".$db->getErrMsg()."');
window.history.go(-1); </script>";
                }
            }
            $this->setRedirect( 'index.php?option=' . $option );
        }

function publish()
{
    global $option;
    $cid = JRequest::getVar( 'cid', array(), "", 'array' );
    if( $this->_task == 'publish')
    {
        $publish = 1;
    }else{
        $publish = 0;
    }
    $clientesTable =& JTable::getInstance('clientes_users',
'Table');
    $clientesTable->publish($cid, $publish);
    $this->setRedirect( 'index.php?option=' . $option );
}
}
?>

```

Este é um controller semelhante aos controllers de outros componentes para Joomla 1.5, com exceção da função combo, que criei para ser populada pelos usuários no form. Veja também que ela retorna o username dos usuários, que pode ser substituído por outros campos da tabela. Veja que a tabela consultada não é a clientes_user mas a de usuários do Joomla. No caso queremos selecionar um entre todos os usuários cadastrados para ter acesso ao nosso componente.

toolbar.clientes.php

```
<?php
    defined( '_JEXEC' ) or die( 'Access denied!' );

    require_once( JApplicationHelper::getPath( 'toolbar_html' ) );

    switch($task) {
        case 'add':
        case 'edit':
        case 'save':
        case 'apply':
            toolbar_clientes::_NEW();
            break;
        default:
            toolbar_clientes::_DEFAULT();
            break;
    }
?>
```

toolbar.clientes.html.php

```
<?php
    defined( '_JEXEC' ) or die( 'Access denied!' );
    class toolbar_clientes {
        function _NEW() {
            JToolBarHelper::title( JText::_('New User'),'generic.png');
            JToolBarHelper::save();
            JToolBarHelper::apply();
            JToolBarHelper::cancel();
        }
        function _DEFAULT() {
            JToolBarHelper::title( JText::_('User
Management'),'generic.png');
            JToolBarHelper::publishList();
            JToolBarHelper::unpublishList();
            JToolBarHelper::editList();
            JToolBarHelper::deleteList();
        }
    }
}
```

```
        JToolBarHelper::addNew();
    }
}
?>
```

install.sql

```
DROP TABLE IF EXISTS `#__clientes_users`;
```

```
CREATE TABLE `#__clientes_users` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `login` char(32) NOT NULL,
  `name` char(45) NOT NULL,
  `published` tinyint(1) unsigned NOT NULL DEFAULT '0',
  PRIMARY KEY (`id`),
  UNIQUE KEY `login` (`login`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8
COLLATE=utf8_unicode_ci;
```

```
DROP TABLE IF EXISTS `#__clientes`;
```

```
CREATE TABLE `#__clientes` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `cpf` char(11) DEFAULT NULL,
  `nome` char(45) NOT NULL,
  `credito_liberado` char(1) NOT NULL,
  `data_nasc` date DEFAULT NULL,
  `email` varchar(50) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM AUTO_INCREMENT=101 DEFAULT
CHARSET=utf8;
```

```
INSERT INTO #__clientes VALUES("1","48672621397","Naomi B.
French","n","0000-00-00","aliquet@maurissagittis.ca");
INSERT INTO #__clientes VALUES("2","66424637906","Francesca
G. Walsh","n","0000-00-00","amet@acrisus.com");
```

Aqui mantive apenas dois registros para ilustrar, para evitar listar os 100 registros do exemplo.

uninstall.sql

```
DROP TABLE IF EXISTS `#__clientes`;  
DROP TABLE IF EXISTS `#__clientes_users`;
```

tables/clientes_users.php

```
<?php  
defined( '_JEXEC' ) or die ( 'Restrict access!' );  
  
class TableClientes_users extends JTable  
{  
    var $id = null;  
    var $login = null;  
    var $name = null;  
    var $published = null;  
  
    function __construct (&$db) {  
        parent::__construct('#__clientes_users','id',$db);  
    }  
}  
?>
```

Apenas repetindo que aqui cada campo da tabela clientes_users aparece como variável numa linha.

clientes.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<install type="component" version="1.5">
  <name>Clientes</name>
  <creationDate>Março de 2010</creationDate>
  <author>ComponentsGen</author>
  <authorEmail>ribafs@ribafs.org</authorEmail>
  <authorUrl>http://ribafs.org</authorUrl>
  <copyright>(c) 2010</copyright>
  <license>Livre</license>
  <version>1.0.0</version>
  <description>
    Componente gerado com o components_gen. Criado por
    Ribamar FS.
  </description>
  <filename>leiam.txt</filename>
  <filename>index.html</filename>
  <files folder = "site">
    <filename>clientes.php</filename>
    <filename>clientes.html.php</filename>
    <filename>page.inc.php</filename>
    <filename>index.html</filename>
    <filename>images/first.png</filename>
    <filename>images/next.png</filename>
    <filename>images/prev.png</filename>
    <filename>images/last.png</filename>
  </files>
  <install>
    <sql>
      <file driver="mysql" charset="utf8">install.sql</file>
    </sql>
  </install>
  <uninstall>
    <sql>
      <file driver="mysql" charset="utf8">uninstall.sql</file>
    </sql>
  </uninstall>
  <administration>
```

```
<menu>Clientes</menu>
<files folder="admin">
  <filename>index.html</filename>
  <filename>install.sql</filename>
  <filename>uninstall.sql</filename>
  <filename>admin.clientes.html.php</filename>
  <filename>admin.clientes.php</filename>
  <filename>controller.php</filename>
  <filename>toolbar.clientes.html.php</filename>
  <filename>toolbar.clientes.php</filename>
  <filename>tables/clientes_users.php</filename>
  <filename>tables/index.html</filename>
</files>
</administration>
</install>
```

Um detalhe interessante sobre este arquivo é que tanto podemos chamá-lo com o mesmo nome do componente quanto podemos chamá-lo de *install.xml*, o que a meu ver fica mais adequado.

Seção site

clientes.php (ponto de entrada deste componente)

```
<?php
/**
 * Componente com_clientes
 * Author: Ribamar FS
 * http://ribafs.org
 */

// no direct access
defined( '_JEXEC' ) or die( 'Restricted access' );

$user =& JFactory::getUser();
$username = $user->get('username');

$db =& JFactory::getDBO();

$query = "SELECT login FROM #__clientes_users WHERE
published = 1 AND login = '". $username. "'";

$db->setQuery( $query, 0);
$login = $db->loadResult();

if($login != $username) {
    JFactory::getApplication()->redirect('index.php', "", "");
}else{
    $session =& JFactory::getSession();
    $session->set('sessao', $login);
}

$sessao = $session->get('sessao');
if(!isset($sessao)){
    JFactory::getApplication()->redirect('index.php', "", "");
}

jimport( 'joomla.application.helper' );
require_once( JFactory::getPath( 'front_html',
```

```
'com_clientes' ) );  
require_once(JPATH_SITE.DS.'components'.DS.'com_clientes'.DS.'page.inc.php');
```

```
// Check the task parameter and execute appropriate function  
switch( JRequest::getVar( 'task' )) {  
    case 'add':  
        add();  
        break;  
    case 'upd':  
        upd();  
        break;  
    case 'del':  
        del();  
        break;  
    case 'grid':  
        grid();  
        break;  
    case 'search':  
        search();  
        break;  
    default:  
        grid();  
        break;  
}
```

```
function date_db($date){ // Converte de d/m/Y para Y-m-d  
    $date = explode('/', $date);  
    $date = $date[2].'-'. $date[1].'-'. $date[0];  
    return $date;  
}
```

```
function add() {  
    OutputClientes::addClientes(); // Exibe o form  
    $submitnew = JRequest::getVar('SubmitNew') ;  
    $msg = "";  
  
    if(isset($submitnew)){
```

```

$cid = JRequest::getVar( 'id');
$cpf = JRequest::getVar( 'cpf');
$nome = JRequest::getVar( 'nome');
$credito_liberado = JRequest::getVar( 'credito_liberado');
$data_nasc = JRequest::getVar( 'data_nasc');
$email = JRequest::getVar( 'email');

$db =& JFactory::getDBO();

$query = 'SELECT * FROM #__clientes WHERE id='.$cid;
$db->setQuery( $query, 0);

$queryc = 'SELECT count(*) FROM #__clientes WHERE
id='.$cid;
$db->setQuery( $queryc, 0);
$nr = $db->loadResult();

if($nr > 0){
    JFactory::getApplication()->redirect('index.php?
option=com_clientes&task=add','Já existe um clientes com este
campo!','error');
}
/*
if($cpf==""){
    $msg .= 'CPF requerido! <br> ';
}

if($nome==""){
    $msg .= 'Nome requerido! <br> ';
}

if($msg) {
    JFactory::getApplication()->redirect('index.php?
option=com_clientes&task=add&id=$cid',$msg,'error');
}
*/

$cid = "".$db->getEscaped( $cid)."";
$cpf = "".$db->getEscaped( $cpf)."";

```

```

        $nome = "".$db->getEscaped( $nome)."";
        $credito_liberado = "".$db-
>getEscaped( $credito_liberado)."";
        $data_nasc = "".$db->getEscaped( $data_nasc)."";
        $email = "".$db->getEscaped( $email)."";

        $insertFields = "INSERT INTO #__clientes
(id,cpf,nome,credito_liberado,data_nasc,email) VALUES ($id,$cpf,
$nome,$credito_liberado,$data_nasc,$email)";

        $db->setQuery( $insertFields, 0);
        $db->query();

        if ($db->getErrorNum() != 0) {
            $error = $db->getErrorMsg().JFactory::getApplication()-
>redirect('index.php?option=com_clientes&task=grid','Erro na
consulta insertFields!', 'error');
        }else{
            JFactory::getApplication()->redirect('index.php?
option=com_clientes&task=grid','Clientes cadastrado com
sucesso!', 'message');
        }
    }
} // End function add_clientes

```

```

function upd()
{
    $subupd = JRequest::getVar('SubmitUpd') ;
    $db =& JFactory::getDBO();

    if(isset($subupd)){
        $id = JRequest::getVar( 'id');
        $cpf = JRequest::getVar( 'cpf');
        $nome = JRequest::getVar( 'nome');
        $credito_liberado = JRequest::getVar( 'credito_liberado');
        $data_nasc = JRequest::getVar( 'data_nasc');
        $email = JRequest::getVar( 'email');
    }
}

```

```

/*
if($nome==""){
    $msg .= JFactory::getApplication()->enqueueMessage('Nome
requerido! ','error');
}

if($msg) JFactory::getApplication()->enqueueMessage($msg.'<a
href="index.php?
option=com_clientes&task=upd&id=$id">Voltar</a>', 'error');
*/

// Adicionar a aspa simples (crase) para eescapar os
valores do Insert
$id = "".$db->getEscaped( $id)."";
$cpf = "".$db->getEscaped( $cpf)."";
$nome = "".$db->getEscaped( $nome)."";
$credito_liberado = "".$db-
>getEscaped( $credito_liberado)."";
$data_nasc = "".$db->getEscaped( $data_nasc)."";
$email = "".$db->getEscaped( $email)."";

$updateFields = "UPDATE #__clientes SET cpf = ".$cpf.", nome
= ".$nome.", credito_liberado = ".$credito_liberado.", data_nasc = ".
$data_nasc.", email = ".$email." WHERE id=".$id;

$db->setQuery( $updateFields, 0);
$rows = $db->loadObjectList();
$db->query();

if ($db->getErrorNum() != 0) {
    $error = $db->getErrorMsg().JFactory::getApplication()-
>enqueueMessage('Erro na consulta updateFields!', 'error');
} else {
    JFactory::getApplication()->redirect('index.php?
option=com_clientes&task=grid', 'Clientes atualizado com
sucesso!', 'message');
}
}

```

```

        $id = JRequest::getVar('id') ;

        $db =& JFactory::getDBO();
        $query = "SELECT * FROM #__clientes WHERE id ='" . $id . "'";
        $db->setQuery( $query, 0);
        $rows = $db->loadObjectList();

        OutputClientes::updClientes($rows); // Mostra form update
    } // end function upd()

function del() {
    $db =& JFactory::getDBO();

    // Get ID from form posting values
    $id = JRequest::getVar('id') ;
    $msg = "";

    if($id==""){
        $msg .= JFactory::getApplication()->enqueueMessage('ID
requerido!', 'error');
    }

    if($msg) JFactory::getApplication()-
>enqueueMessage($msg, 'error');

    $deleteFields = "DELETE FROM #__clientes WHERE id = ". $id;

    $db->setQuery( $deleteFields, 0);
    $db->query();

    if ($db->getErrorNum() != 0) {
        $error = $db->getErrorMsg().JFactory::getApplication()-
>enqueueMessage('Erro na consulta deleteFields!', 'error');
    }else{
        JFactory::getApplication()->redirect('index.php?
option=com_clientes','Clientes excluído com sucesso!', 'message');
    }
}

```



```

    }

    OutputClientes::delClientes(); // Executa rotina del do
    clientes.html.php

} // end del() function

function grid() {
    // Paging
    $config =& JFactory::getConfig();

    $host=$config->getValue('host');
    $db = $config->getValue('db');
    $pass = $config->getValue('password');
    $user = $config->getValue('user');
    $dbprefix = $config->getValue('dbprefix');

    mysql_connect($host, $user, $pass);
    mysql_select_db($db);

    $query = 'SELECT * FROM '.$dbprefix.'clientes ORDER BY id
ASC';
    $result=mysql_query($query);
    $total_rec=mysql_num_rows($result);

    if (mysql_error()) {
        $error = JFactory::getApplication()->enqueueMessage('Erro
na consulta rows!', 'error');
    }

    $page=new Page(); // creating new instance of Class Page

    $recpp = 10; // Registros por página
    $scroll = 20; // Visíveis

    $page->set_page_data($_SERVER['PHP_SELF'],$total_rec,
$recpp,$scroll,true,true,true);
    $result=mysql_query($page->get_limit_query($query));

```

```
        OutputClientes::gridClientes($result,$page,$total_rec);// Executa
rotina para exibir registros do clientes
    }
```

```
function search() {
```

```
    // Get ID from form posting values
    $id = JRequest::getVar('id') ;
    $nome = JRequest::getVar('nome') ;
```

```
    $db =& JFactory::getDBO();
```

```
    $query = "SELECT id, cpf, nome FROM #__clientes WHERE
nome LIKE '%" . $nome . "%' ORDER BY nome";
```

```
    $db->setQuery( $query, 0);    $rows = $db->loadObjectList();
```

```
        if ($db->getErrorNum() != 0) {
            $error = $db->getErrorMsg().JFactory::getApplication()-
            >enqueueMessage('Erro na consulta select!', 'error');
        }
```

```
        OutputClientes::searchClientes($rows);// Executa rotina para
exibir registros do clientes
    }
```

```
if($error != "") JFactory::getApplication()-
>enqueueMessage($error,'error');
```

```
?>
```

Como este é o ponto de entrada do componente, ou seja, o primeiro arquivo que o Joomla executa quando chamamos o componente, então aqui primeiro coloquei um filtro que irá listar o componente somente quando o usuário atual for igual a um dos usuários cadastrados na parte admin do componente.

Logo em seguida gravo uma variável no session para testar neste e no outro script se o usuário fez o login.

Depois incluí os arquivos: A parte view do componente (clientes.html.php) e a classe para a paginação (page.inc.php).

Em seguida adicionei um switch com as opções existentes: add, upd, del, grid e search.

De acordo com a opção escolhida pelo usuário será chamada a respectiva função.

Observe que algumas funções apenas executam procedimentos enquanto que outras executam alguns procedimentos e ao final chamam uma função da parte view do componente, para mostrar algo na tela e geralmente passam algum parâmetro. Como é o caso da função upd(), que ao final chama o método updClientes da classe OutputClientes passando \$rows como parâmetro:

```
OutputClientes::updClientes($rows);
```

Se observar o início do script clientes.php vai notar que o script clientes.html.php foi incluído logo no início e antes do código do clientes.php. Com isso quem primeiro será chamado será o clientes.html.php. Como o clientes.html.php é formado por uma classe contendo a apresentação do componente, forms e grid e como a grid é o default ela será aberta logo que o componente seja chamado.

Quando então alguém escolher uma das ações previstas no switch a função respectiva será chamada.

Agora vejamos as funções do clientes.php:

add() - adiciona o registro que veio do form de clientes.html.php

upd() - atualiza o registro

del() - exclui o registro selecionado

search() – efetua uma busca no banco

grid() - mostra a grid com paginação

clientes.html.php

É a parte da apresentação (view) do componente.

```
<?php
/**
 * @clientes.html.php
 * @Ribamar FS.
 * @http://ribafs.org
 */

// no direct access
defined( '_JEXEC' ) or die( 'Restricted access' );

class OutputClientes
{
    function addClientes(){
        ?>
        <body onLoad="document.form1.cpf.focus();">
        <h2 class="moduletable">Cadastro de Clientes</h2>
        <div align="left">
        <br>
        <form id="form1" name="form1" method="post"
action="index.php?option=com_clientes&task=add">
            <table border="0">
                <tr><td>Cpf</td><td><input type="text" name="cpf"
size="11" maxlength="11"></td></tr>
                <tr><td>Nome</td><td><input type="text" name="nome"
size="45" maxlength="45"></td></tr>
                <tr><td>Credito_liberado</td><td><input type="text"
name="credito_liberado" size="1" maxlength="1"></td></tr>
                <tr><td>Data_nasc</td><td><input type="text"
name="data_nasc" size="10" maxlength="10"></td></tr>
                <tr><td>Email</td><td><input type="text" name="email"
size="50" maxlength="50"></td></tr>
                <input type="hidden" name="id" value="<?php print $id;?>" />
                <tr><td>&nbsp;</td><td><input type="submit"
name="SubmitNew" value="Adicionar" /></td></tr>
            </table>
```

```

</form>
<br>
</div>
<br>
<?php
}

```

```
function updClientes($rows){
```

```
    if($rows) {
```

```
        foreach ($rows as $row){
```

```
            $id = $row->id;
```

```
            $cpf = $row->cpf;
```

```
            $nome = $row->nome;
```

```
            $credito_liberado = $row->credito_liberado;
```

```
            $data_nasc = $row->data_nasc;
```

```
            $email = $row->email;
```

```
        ?>
```

```
        <h2 class="moduletable">Cadastro de Clientes - Alteração</h2>
```

```
        <div align="left">
```

```
            <form id="form1" name="form1" method="post"
            action="index.php?option=com_clientes&task=upd">
```

```
                <br>
```

```
                <table border="0">
```

```
                    <tr><td>Cpf</td><td><input type="text" name="cpf"
                    size="11" maxlength="11" value="<?php print $cpf;?>"></td></tr>
```

```
                    <tr><td>Nome</td><td><input type="text" name="nome"
                    size="45" maxlength="45" value="<?php print $nome;?>"></td></tr>
```

```
                    <tr><td>Credito_liberado</td><td><input type="text"
                    name="credito_liberado" size="1" maxlength="1" value="<?php print
                    $credito_liberado;?>"></td></tr>
```

```
                    <tr><td>Data_nasc</td><td><input type="text"
                    name="data_nasc" size="10" maxlength="10" value="<?php print
                    $data_nasc;?>"></td></tr>
```

```
                    <tr><td>Email</td><td><input type="text" name="email"
                    size="50" maxlength="50" value="<?php print $email;?>"></td></tr>
```

```
                <input type="hidden" name="id" value="<?php print $id;?>
```

```

>" />
        <tr><td>&nbsp;</td><td><input type="submit"
name="SubmitUpd" value="Alterar" /></td></tr>
        </table>
    </form>
    <br>
</div>
<?php
    } // End foreach
} // End if

$subupdpf = JRequest::getVar('SubmitUpdpf');
if(isset($subupdpf)){
    JFactory::getApplication()->redirect('index.php?
option=com_clientes&task=grid',"','message');
}

} // End function

```

```

function delClientes() {
    $id = JRequest::getVar('id') ;
    JFactory::getApplication()->redirect("index.php?
option=com_clientes&task=grid&id=","','message');
}

```

```

function gridClientes($result,$page,$total_rec){
    ?>
        <h1 class="moduletable">Controle de Clientes</h1>

        <table border="0">
            <tr><td><b>ID</td><td><b>Cpf</td><td><b>Nome</td><td
colspan="2" align="center"><b>Ação</td></tr>
        <?php

            $fld2=$field2;
            if($result) {

```

```

while($row=mysql_fetch_array($result))
{
    $id = $row["id"];
    $cpf = $row['cpf'];
    $nome = $row['nome'];

    ?>
    <tr><td><?php print $id;?></td><td><?php print $cpf;?
>&nbsp;</td><td>&nbsp;<?php print $nome;?
>&nbsp;<td><a href="index.php?
option=com_clientes&task=upd&id=<?php print $id;?
">Alterar</a>&nbsp;&nbsp;&nbsp;</td>
    <td><a href="index.php?
option=com_clientes&task=del&id=<?php print $id;?>" onClick="if(!
confirm('Tem certeza de que deseja excluir o registro '+<?php print
$id;?>+' ?')) {return false;}">Excluir</a></td>
    </tr>

    <?php
} // End while
}

    $f = 'components/com_clientes/images/first.png';
    $n = 'components/com_clientes/images/next.png';
    $p = 'components/com_clientes/images/prev.png';
    $l = 'components/com_clientes/images/last.png';

    $comp = 'com_clientes';

    print '<tr><td colspan="5">';
    $page->get_first_page_nav("", $comp);
    $page->get_next_page_nav("", $comp);
    $page->get_number_page_nav("", $comp);
    $page->get_prev_page_nav("", $comp);
    $page->get_last_page_nav("</tr>';
    ?>
    <tr><td colspan="5" align="center"><a href="index.php?
option=com_clientes&task=add">Adicionar Novo</a></td></tr>

    <form method="post" action="index.php?
option=com_clientes&task=search">
    <tr><td colspan="4"><input type="text" name="<?php print
nome;?>" size="45" value="Digite o nome ou parte"
onClick="this.value="""></td>
    <td><input type="submit" value="Localizar"></td></tr>
    </form>

</table>
<?php
} // End function grid

function searchClientes($rows) {
?>
    <h1 class="moduletable">Localização de Clientes</h1>

    <table border="0">
    <tr><td><b>Cpf</b>
        <td><b>Nome</b>
        <td colspan="2" align="center"><b>Ação</b>
    </tr>
<?php
    if($rows) {
        foreach ($rows as $row){
            // Encode text to HTML formatting

            ?>
            <tr><td>&nbsp;<?php print $field1;?>&nbsp;</td>
                <td>&nbsp;<?php print $field2;?
            &nbsp;&nbsp;&nbsp;&nbsp;</td>

                <td><a href='index.php?
option=com_clientes&task=upd&id=<?php print $id;?

```



```

>'>Alterar</a></td>
        <td><a href='index.php?
option=com_clientes&task=del&id=<?php print $id;?>' onClick="if(!
confirm('Tem certeza de que deseja excluir '+<?php print $field2;?>
+ '?')) {return false;}">Excluir</a></td>
        </tr>

        <?php
        }// End foreach
    }// End if ($rows)

    print '</table><br><br><br>';
}// End function grid
}// End class

function dateBr($date){// Converte de Y-m-d para d/m/Y
    $date = explode('-', $date);
    $date = $date[2].'/'.$date[1].'/'.$date[0];
    return $date;
}

?>

```

Este script é composto de uma classe que tem funções/métodos correspondentes aos existentes no script clientes.php.

Logo no início é feito o teste para ver se a variável de session está setada..

Mesmo que o clientes.php seja a porta de entrada do componente, ele chama este script em seu início, conseqüentemente clientes.html.php é o primeiro script a aparecer quando nosso componente for chamado.

Como a grid() é a função default em clientes.php nosso componente abre com a grid para que o usuário veja os registros existentes e possa executar uma das ações previstas: add, upd, del e search.

Observe que cada método da classe basicamente mostra algo na

tela. Quando muito temos um pequeno trecho em PHP, nada de acesso ao banco aqui. Fiz questão de separar as camadas. Todo o código pesado em PHP e todo acesso ao banco ficou no script `clientes.php`. Aqui ficou a camada de apresentação (view) somente. Mesmo não sendo um código em três camadas ou usando MVC, ainda assim fiz questão de separar bem em duas camadas.

Gostaria de comentar sobre o método `gridClientes()`, que mostra a grid na tela. Esta forma de trabalhar é muito amigável para o usuário, visto que ele tem acesso a todas as ações numa única tela. Fica assim um código além de amigável também muito produtivo.

page.inc.php

Este é um script do Harish Chauhan, que é responsável pela paginação do componente. Apenas fiz algumas adaptações de forma que recebesse parâmetros e usasse as imagens.

Um ótimo script para paginação.

images

As imagens usadas na paginação.
Observe que adicionei dicas quando o usuário move o mouse sobre as imagens.

Controle de Estoque

Este é o segundo exemplo de componente comercial. Agora contendo várias tabelas.

Controle de estoque simples, contendo as tabelas: clientes, funcionarios, produtos, pedidos e itens de pedidos além da tabela de usuários. A idéia não é a de criar um controle de estoque completo mas apenas a de mostrar como funciona para várias tabelas a ideia dos componentes comerciais no Joomla 1.5. Pelo mesmo motivo também não listarei o xml.

É bom ressaltar que este componente não é um controle de estoque, onde se abate o estoque ao efetuar o pedido e outras características de controle de estoque.

Para este nem vou mostrar a parte do admin, que é semelhante à do componente cadastro de clientes. Vamos ver apenas a parte do site, que é a mais importante aqui.

Por conta do controle de session obrigatoriamente o nível de acesso deve ser especial ou registrado, caso contrário o componente não aparecerá.

estoque.php (ponto de entrada do componente)

```
<?php
/**
 * @version      $Id: estoque.php 1 2010-03-20$
 * @author-name Ribamar FS
 * @copyright    Copyright (C) 2010 Ribamar FS.
 * @license      GNU/GPL, see http://www.gnu.org/licenses/old-licenses/gpl-2.0.txt
 * JInstaller is free and open source software. This version may have
 been modified
 * pursuant to the GNU General Public License, and as distributed it
 includes or is
 * derivative of works licensed under the GNU General Public
 License or other free or
 * open source software licenses.
 */

// no direct access
defined( '_JEXEC' ) or die( 'Restricted access' );

$user =& JFactory::getUser();
$username = $user->get('username');

// $usertype = $user->get('usertype');
// $aid = $user->get('aid'); // 0 - guest, 1 - registered and 2 - special

$db =& JFactory::getDBO();

$query = "SELECT login FROM #__estoque_users WHERE
published = 1 AND login = ".$username."";

$db->setQuery( $query, 0);
$login = $db->loadResult();

// If is not authenticated login or group is not special redirection to
idnex.php
if($login != $username) {
    JFactory::getApplication()->redirect('index.php', "", "");
}
```

```

}else{
    $session =& JFactory::getSession();
    $session->set('sessao', $login);
}

$sessao = $session->get('sessao');
if(!isset($sessao)){
    JFactory::getApplication()->redirect('index.php', "", "");
}

jimport( 'joomla.application.helper' );
require_once( JApplicationHelper::getPath( 'front_html',
'com_estoque' ) );
require_once(JPATH_SITE.DS.'components'.DS.'com_estoque'.DS.'
clientes.php');
require_once(JPATH_SITE.DS.'components'.DS.'com_estoque'.DS.'f
uncionarios.php');
require_once(JPATH_SITE.DS.'components'.DS.'com_estoque'.DS.'
produtos.php');
require_once(JPATH_SITE.DS.'components'.DS.'com_estoque'.DS.'
pedidos.php');
require_once(JPATH_SITE.DS.'components'.DS.'com_estoque'.DS.'
pedido_itens.php');

function dropdown($fld_store, $fld_view, $val_default=null, $table,
$fld_order, $select_name, $method='asc') {

    $combo = "<select name=\"\$select_name\">\n";
    $combo .= "<option value=\"NULL\">Select Value</option>\n";

    $query = "select $fld_store, $fld_view from $table order by
$fld_order $method";

    $db =& JFactory::getDBO();
    $db->setQuery($query);
    $db->query($query);

    $rows = $db->loadAssocList();

```

```

foreach($rows as $row) {
    $store = $row[$fld_store];
    $view = $row[$fld_view];

    if($store == $val_default){
        $combo .= "<option value=\"\$store\" selected=\"selected\"
>$view</option>\n";
    }else{
        $combo .= "<option value=\"\$store\">$view</option>\n";
    }
}
$combo .= "</select>";
return $combo;
}
//print dropdown('id', 'nome', $cliente_fk,'jos_estoque_clientes',
'nome','cliente_fk', $method='asc');

```

```

// Check the task parameter and execute appropriate function
switch( JRequest::getVar( 'task' )) {
    case 'addcli':
        add_cli();
        break;
    case 'updcli':
        upd_cli();
        break;
    case 'delcli':
        del_cli();
        break;
    case 'gridcli':
        grid_cli();
        break;
    case 'searchcli':
        search_cli();
        break;
    case 'addfun':
        add_fun();
        break;
    case 'updfun':

```

```
    upd_fun();
    break;
case 'delfun':
    del_fun();
    break;
case 'gridfun':
    grid_fun();
    break;
case 'searchfun':
    search_fun();
    break;
case 'addpro':
    add_pro();
    break;
case 'updpro':
    upd_pro();
    break;
case 'delpro':
    del_pro();
    break;
case 'gridpro':
    grid_pro();
    break;
case 'searchpro':
    search_pro();
    break;
case 'addped':
    add_ped();
    break;
case 'updped':
    upd_ped();
    break;
case 'delped':
    del_ped();
    break;
case 'gridped':
    grid_ped();
    break;
case 'searchped':
```



```

        search_ped();
    break;
case 'addite':
    add_ite();
    break;
case 'updite':
    upd_ite();
    break;
case 'delite':
    del_ite();
    break;
case 'gridite':
    grid_ite();
    break;
case 'searchite':
    search_ite();
    break;
}
?>

```

Veja que de forma semelhante ao componente com_clientes, aqui também são incluídos no início todos os scripts da camada de apresentação de cada tabela.

Logo em seguida é feito o teste para saber se o usuário atual tem acesso ao componente. Caso tenha, o componente é mostrado.

Em seguida aparece a função dropdown() que é uma função interessante, usada na apresentação para mostrar o conteúdo de uma tabela. Ela também tem uma característica muito útil, que é mostrar nas atualizações a combo com o valor default como o valor do registro exibido.

Para cada uma das tabelas: clientes, funcionarios, produtos, pedidos e pedido_itens teremos um grid que facilitará as operações de CRUD, contando também busca e paginação.

estoque.html.php

```
<?php
/**
 * @version      $Id: estoque.html.php 1 2010-03-20$
 * @author-name Ribamar FS
 * @copyright    Copyright (C) 2010 Ribamar FS.
 * @license      GNU/GPL, see http://www.gnu.org/licenses/old-
licenses/gpl-2.0.txt
 * JInstaller is free and open source software. This version may have
been modified
 * pursuant to the GNU General Public License, and as distributed it
includes or is
 * derivative of works licensed under the GNU General Public
License or other free or
 * open source software licenses.
 */

// no direct access
defined( '_JEXEC' ) or die( 'Restricted access' );

$sessao = $session->get('sessao');
if(!isset($sessao)){
    JFactory::getApplication()->redirect('index.php', "", "");
}

?>

<body>
<h2 class="moduletable">Controle de Estoque</h2>
  <table border="0"><tr>
    <td><a href="index.php?
option=com_estoque&task=gridcli">Clientes</a></td><td>&nbsp;</t
d>
    <td><a href="index.php?
option=com_estoque&task=gridfun">Funcionários</a></td><td>&nb
sp;</td>
    <td><a href="index.php?
option=com_estoque&task=gridpro">Produtos</a></td><td>&nbsp;</td><td>&nbsp;</td></tr>
</table>
</body>
```

```
/td>
  <td><a href="index.php?
option=com_estoque&task=gridped">Pedidos</a></td><td>&nbsp;</
td>
  <td><a href="index.php?
option=com_estoque&task=gridite">Itens de Pedidos</a></td>
</tr></table>
<br>
```

Este script representa a parte da apresentação do componente.

É um menu que chama o grid de cada uma das tabelas para as operações sobre a mesma.

clientes.php

```
<?php
/**
 * @version      $Id: clientes.php 1 2010-03-20$
 * @author-name Ribamar FS
 * @copyright    Copyright (C) 2010 Ribamar FS.
 * @license      GNU/GPL, see http://www.gnu.org/licenses/old-licenses/gpl-2.0.txt
 * JInstaller is free and open source software. This version may have
 * been modified
 * pursuant to the GNU General Public License, and as distributed it
 * includes or is
 * derivative of works licensed under the GNU General Public
 * License or other free or
 * open source software licenses.
 */

// no direct access
defined( '_JEXEC' ) or die( 'Restricted access' );

$sessao = $session->get('sessao');
if(!isset($sessao)){
    JFactory::getApplication()->redirect('index.php', "", );
}

jimport( 'joomla.application.helper' );
require_once(JPATH_SITE.DS.'components'.DS.'com_estoque'.DS.'
clientes.html.php');
require_once(JPATH_SITE.DS.'components'.DS.'com_estoque'.DS.'
page.inc.php');

function date_db($date){ // Converte de d/m/Y para Y-m-d
    $date = explode('/', $date);
    $date = $date[2].'-'. $date[1].'-'. $date[0];
    return $date;
}
```

```

function add_cli() {
    OutputClientes::updCli($rows);
    $submitnew = JRequest::getVar('SubmitNew') ;
    $db =& JFactory::getDBO();

    if(isset($submitnew)){
        $sid = JRequest::getVar( 'id' );
        $cpf = JRequest::getVar( 'cpf' );
        $nome = JRequest::getVar( 'nome' );
        $credito_liberado = JRequest::getVar( 'credito_liberado' );
        $data_nasc = JRequest::getVar( 'data_nasc' );
        $email = JRequest::getVar( 'email' );

        $data_nasc=date_db($data_nasc);

        $query = "SELECT * FROM #__estoque_clientes WHERE
id=".$sid;
        $db->setQuery( $query, 0);

        $queryc = "SELECT count(*) FROM #__estoque_clientes
WHERE id=".$sid;
        $db->setQuery( $queryc, 0);
        $nr = $db->loadResult();

        if($nr > 0){
            JFactory::getApplication()->redirect('index.php?
option=com_estoque&task=addcli','Já existe um cliente com este
CPF!', 'error');
        }

        if($cpf==""){
            $msg .= "CPF requerido! <br> ";
        }

        if($nome==""){
            $msg .= "Nome requerido! <br> ";
        }

        if($msg) {

```

```

        JFactory::getApplication()->redirect("index.php?
option=com_estoque&task=addcli&cpf=$cpf",$msg,'error');
    }

    // Adicionar a aspa simples (crase) para eescapar os
valores do Insert
    $cpf = "" . $db->getEscaped($cpf) . "";
    $nome = "" . $db->getEscaped($nome) . "";
    $credito_liberado = "" . $db-
>getEscaped($credito_liberado) . "";
    $data_nasc = "" . $db->getEscaped($data_nasc) . "";
    $email = "" . $db->getEscaped($email) . "";

    $insertFields = "INSERT INTO #__estoque_clientes " .
"(cpf, nome, credito_liberado, data_nasc, email)" . " VALUES (" .
$cpf . "," . $nome . "," . $credito_liberado . "," . $data_nasc . "," .
$email . ")";

    $db->setQuery( $insertFields, 0);
    $db->query();

    if ($db->getErrorNum() != 0) {
        $error = $db->getErrorMessage().JFactory::getApplication()-
>redirect('index.php?option=com_estoque&task=gridcli','Erro na
consulta insertFields!', 'error');
    }else{
        JFactory::getApplication()->redirect('index.php?
option=com_estoque&task=gridcli','Cliente cadastrado com
sucesso!', 'message');
    }
}
} // End function add

function upd_cli()
{
    $subupd = JRequest::getVar('SubmitUpd') ;
    $db =& JFactory::getDBO();

```

```

if(isset($subupd)){
    $id = JRequest::getVar( 'id');
    $cpf = JRequest::getVar( 'cpf');
    $nome = JRequest::getVar( 'nome');
    $credito_liberado = JRequest::getVar( 'credito_liberado');
    $data_nasc = JRequest::getVar( 'data_nasc' );
    $email = JRequest::getVar( 'email' );

    $data_nasc=date_db($data_nasc);

    if($cpf==""){
        $msg .= JFactory::getApplication()->enqueueMessage("CPF
requerido! ",'error');
    }

    if($nome==""){
        $msg .= JFactory::getApplication()->enqueueMessage("Nome
requerido! ",'error');
    }

    if($msg) JFactory::getApplication()->enqueueMessage($msg."<a
href=\"index.php?
option=com_estoque&task=updccli&id=$id\">Voltar</a>",'error');
    // Adicionar a aspa simples (crase) para eescapar os
valores do Insert
    $id = "" . $db->getEscaped($id) . "";
    $cpf = "" . $db->getEscaped($cpf) . "";
    $nome = "" . $db->getEscaped($nome) . "";
    $credito_liberado = "" . $db-
>getEscaped($credito_liberado) . "";
    $data_nasc = "" . $db->getEscaped($data_nasc) . "";
    $email = "" . $db->getEscaped($email) . "";

    $updateFields = "UPDATE #__estoque_clientes SET cpf=".$cpf.",
nome=".$nome.", credito_liberado=".$credito_liberado.",
data_nasc=".$data_nasc.", email=".$email." WHERE id=".$id;

    $db->setQuery( $updateFields, 0);
    $rows = $db->loadObjectList();

```

```

$db->query();

    if ($db->getErrorNum() != 0) {
        $error = $db->getErrMsg().JFactory::getApplication()-
>enqueueMessage('Erro na consulta updateFields!','error');
    }else{
        JFactory::getApplication()->redirect('index.php?
option=com_estoque&task=gridcli','Cliente atualizado com
sucesso!','message');
    }
}

    $id = JRequest::getVar( 'id');
    $query = "SELECT * FROM #__estoque_clientes WHERE id=".$id;
$db->setQuery( $query, 0);
$rows = $db->loadObjectList();
    OutputClientes::updCli($rows); // Mostra form update
} // final função upd()

```

```

function del_cli() {

    $db =& JFactory::getDBO();

    // Get ID from form posting values
    $id = JRequest::getVar('id') ;
    $msg = "";

    if($id==""){
        $msg .= JFactory::getApplication()->enqueueMessage('ID
requerido!','error');
    }

    if($msg) JFactory::getApplication()-
>enqueueMessage($msg,'error');

    $deleteFields = "DELETE FROM #__estoque_clientes WHERE id

```



```

= ". $id;
$db->setQuery( $deleteFields, 0);
$db->query();

    if ($db->getErrorNum() != 0) {
        $error = $db->getErrMsg().JFactory::getApplication()-
>enqueueMessage('Talvez tenha antes que excluir todos os pedidos
deste cliente para excluí-lo!', 'error');
    }else{
        JFactory::getApplication()->redirect('index.php?
option=com_estoque?task=gridcli','Cliente excluído com
sucesso!', 'message');
    }
    OutputClientes::delCli(); // Executa rotina del do
clientes.html.php

} // final função del()

// Mostrar a lista de entradas de cliente
function grid_cli() {
    // Paginação de resultados
    $config =& JFactory::getConfig();

    $host=$config->getValue('host');
    $db = $config->getValue('db');
    $pass = $config->getValue('password');
    $user = $config->getValue('user');
    $dbprefix = $config->getValue('dbprefix');

    mysql_connect($host, $user, $pass);
    mysql_select_db($db);

    $query = "SELECT * FROM ".$dbprefix."estoque_clientes
ORDER BY id ASC";
    $result=mysql_query($query);
    $total_rec=mysql_num_rows($result);

    if (mysql_error()) {

```

```
        $error = JFactory::getApplication()->enqueueMessage('Erro
na consulta rows!','error');
    }
```

```
    $page=new Page(); // creating new instance of Class Page
```

```
    $recpp = 10; // Registros por página
    $scroll = 20; // Visíveis
```

```
    $page->set_page_data($_SERVER['PHP_SELF'],$total_rec,
    $recpp,$scroll,true,true,true);
    $result=mysql_query($page->get_limit_query($query));
```

```
    OutputClientes::gridCli($result,$page,$total_rec);// Executa
rotina para exibir registros do clientes
}
```

```
// Procurar Cliente
```

```
function search_cli() {
    $nome = JRequest::getVar('nome') ;
    $db =& JFactory::getDBO();
```

```
    $query = "SELECT id, cpf, nome FROM #__estoque_clientes
WHERE nome LIKE '%" . $nome . "%' ORDER BY nome";
    $db->setQuery( $query, 0);    $rows = $db->loadObjectList();
```

```
    if ($db->getErrorNum() != 0) {
        $error = $db->getErrorMsg().JFactory::getApplication()-
>enqueueMessage('Erro na consulta selectf!','error');
    }
```

```
    OutputClientes::searchCli($rows);// Executa rotina para exibir
registros do clientes
}
```

```
if($error != "") JFactory::getApplication()-
>enqueueMessage($error,'error');
```

```
?>
```


Este é o script correspondente à tabela clientes, que em seu início testa a variável de session, insere o script clientes.html.php e o page.inc.php.

Então traz a definição de cada uma das funções: add(), upd(), del(), grid() e search().

Essas funções são semelhantes às funções do componente com_clientes.

clientes.html.php

```
<?php
/**
 * @version      $Id: clientes.html.php 1 2010-03-20$
 * @author-name Ribamar FS
 * @copyright    Copyright (C) 2010 Ribamar FS.
 * @license      GNU/GPL, see http://www.gnu.org/licenses/old-licenses/gpl-2.0.txt
 * JInstaller is free and open source software. This version may have
 * been modified
 * pursuant to the GNU General Public License, and as distributed it
 * includes or is
 * derivative of works licensed under the GNU General Public
 * License or other free or
 * open source software licenses.
 */

// no direct access
defined( '_JEXEC' ) or die( 'Restricted access' );

$sessao = $session->get('sessao');
if(!isset($sessao)){
    JFactory::getApplication()->redirect('index.php',"");
}
}
```

```

class OutputClientes
{

function updCli($rows){
    if($rows) {
        foreach ($rows as $row){
            $id = $row->id;
            $cpf = $row->cpf;
            $nome = $row->nome;
            $credito_liberado = $row->credito_liberado;
            $data_nasc = $row->data_nasc;
            $email = $row->email;

            $data_nasc = dateBr($data_nasc);
        }

        ?>
        <h2 class="moduletable">Cadastro de Clientes -
        Alteração</h2>
        <div align="left">

            <form id="form1" name="form1" method="post"
            action="index.php?option=com_estoque&task=updccli">
            <br>
            <table border="0">
                <tr><td>CPF</td><td><input name="cpf" id="cpf" size="20"
                maxlength="20" value="<?php print $cpf;?>" ></td></tr>
                <tr><td>Nome</td><td><input name="nome" id="nome"
                size="45" maxlength="45" value="<?php print $nome;?>" ></td></tr>
                <tr><td>Crédito Liberado?</td><td><input
                name="credito_liberado" size="1" maxlength="1" value="<?php print
                $credito_liberado;?>" ></td></tr>
                <tr><td>Data de Nascimento</td><td><input
                name="data_nasc" id="data_nasc" size="10" maxlength="10"
                value="<?php print $data_nasc;?>" ></td></tr>
                <tr><td>E-mail</td><td><input name="email" id="email"
                size="40" maxlength="100" value="<?php print $email;?>"
                ></td></tr>
            <tr><td><input type="hidden" name="id" value="<?php print $id;?
            >" />
            <tr><td>&nbsp;</td><td><input type="submit"

```

```

name="SubmitUpd" value="Alterar" /></td></tr>
</table>
</form>
<br>
</div>
<?php
    } //End if
    }else{// End foreach
?>
    <body onLoad="document.form1.cpf.focus();">
    <h2 class="moduletable">Cadastro de Clientes</h2>
    <form id="form1" name="form1" method="post"
    action="index.php?option=com_estoque&task=addcli">
    <table border="0">
        <tr><td>CPF</td><td><input name="cpf" id="cpf" size="11"
    maxlength="11"></td></tr>
        <tr><td>Nome</td><td><input name="nome" id="nome"
    size="45" maxlength="45"></td></tr>
        <tr><td>Crédito Liberado?</td><td><input
    name="credito_liberado" size="1" maxlength="1"></td></tr>
        <tr><td>Data de Nascimento</td><td><input
    name="data_nasc" id="data_nasc" size="10"
    maxlength="10"></td></tr>
        <tr><td>E-mail</td><td><input name="email" id="email"
    size="40" maxlength="100"></td></tr>
        <tr><td>&nbsp;</td><td><input type="submit"
    name="SubmitNew" value="Adicionar" /></td></tr>
    </table>
    </form>
    <br>
<?php
    }
    $subupdpf = JRequest::getVar('SubmitUpdpf');
    if(isset($subupdpf)){
        JFactory::getApplication()->redirect('index.php?
    option=com_estoque&task=gridcli',"message');
    }

} // End function

```

```
function delCli() {
    JFactory::getApplication()->redirect('index.php?
option=com_estoque&task=gridcli',"message');
}
```

```
function gridCli($result,$page,$total_rec){
?>
```

```
<h1 class="moduletable">Cadastro de Clientes</h1>
```

```
<table border="0">
```

```
<tr><td><b>Código</td><td><b>CPF</td><td><b>Nome</td><td
colspan="2" align="center"><b>Ação</td></tr>
<?php
```

```
if($result) {
    while($row=mysql_fetch_array($result))
    {
        $id = $row['id'];
        $cpf = $row['cpf'];
        $nome = $row['nome'];
        ?>
        <tr><td><?php print $id;?></td><td><?php print $cpf;?
>&nbsp;</td><td>&nbsp;<?php print $nome;?
>&nbsp;&nbsp;&nbsp;</td>
        <td><a href="index.php?
option=com_estoque&task=updcli&id=<?php print $id;?
">Alterar</a>&nbsp;&nbsp;&nbsp;</td>
        <td><a href="index.php?
option=com_estoque&task=delcli&id=<?php print $id;?>"
onClick='if(!confirm("Tem certeza de que deseja excluir
o cliente "+"<?php print $nome;?>"+" ?")) {return
false;}'>Excluir</a></td></tr>

<?php
} // End while
} // End if ($result)
```

```

    $f = 'components/com_estoque/images/first.png';
    $n = 'components/com_estoque/images/next.png';
    $p = 'components/com_estoque/images/prev.png';
    $l = 'components/com_estoque/images/last.png';

    $comp = 'com_estoque';
    $tsk = 'cli';

    print "<tr><td colspan='5'>";
    $page->get_first_page_nav("<img src='". $f. "'
title='Primeiro'>", $comp, $tsk);
    $page->get_next_page_nav("<img src='". $n. "'
title='Próximo'>", $comp, $tsk);
    $page->get_number_page_nav("", $comp, $tsk);
    $page->get_prev_page_nav("<img src='". $p. "'
title='Anterior'>", $comp, $tsk);
    $page->get_last_page_nav("<img src='". $l. "' title='Último
(total de $total_rec registros)'" , $comp, $tsk);
    print '</td></tr>';
    ?>
    <tr><td colspan="5" align="center"><a href="index.php?
option=com_estoque&task=updccli">Adicionar Novo</a></td></tr>

    <form method="post" action="index.php?
option=com_estoque&task=searchcli">
    <tr><td colspan="4"><input type="text" name="nome" size="33"
value="Digite o nome ou parte" onClick="this.value=""></td>
    <td><input type="submit" value="Localizar"></td></tr>
    </form>

    </table>
    <?php
} // End function grid

function searchCli($rows) {
    ?>
    <h1 class="moduletable">Localização de Clientes</h1>

```



```

<table border="0">
<tr><td><b>CPF</b></td>
    <td><b>Nome</b></td>
    <td colspan="2" align="center"><b>Ação</b></td>
</tr>
<?php
if($rows) {
    foreach ($rows as $row){
        // Encode text to HTML formatting
        $id = $row->id;
        $cpf = $row->cpf;
        $nome = $row->nome;

        ?>
        <tr><td>&nbsp;<?php print $cpf;?>&nbsp;</td>
            <td>&nbsp;<?php print $nome;?
            >&nbsp;&nbsp;&nbsp;&nbsp;</td>

                <td><a href="index.php?
option=com_estoque&task=updccli&id=<?php print $id;?
>">Alterar</a></td>
                <td><a href="index.php?
option=com_estoque&task=delcli&id=<?php print $id;?>"
onClick="if(!confirm('Tem certeza de que deseja excluir o cliente '+<?
php print $nome;?> + '?')) {return false;}">Excluir</a></td>
            </tr>

            <?php
            //Linha horizontal
            //echo "<hr />";
            }// End foreach
        }// End if ($rows)

        print "</table><br><br><br>";
    }// End function grid
}// End class

function dateBr($date){// Converte de Y-m-d para d/m/Y
    $date = explode('-', $date);

```

```
$date = $date[2].'/'.$date[1].'/'.$date[0];  
return $date;  
}  
?>
```

Corresponde a apresentação de clientes no componente estoque.

De forma semelhante teremos o código dos demais scripts:

```
funcionarios.php  
funcionarios.htmlphp  
pedidos.php  
pedidos.html.php  
pedido_itens.php  
pedido_itens.html.php  
produtos.php  
produtos.html.php
```

Obs.: A aplicação do session em cada um dos scripts impede que alguém grave a URL de um componente e faça o acesso direto sem fazer login.

Exemplo: tente fazer login direto em uma seção assim:

[http://localhost/joomla/index.php?
option=com_estoque&task=updccli&id=4](http://localhost/joomla/index.php?option=com_estoque&task=updccli&id=4)

Será direcionado para a index caso não tenha o devido acesso.

Gerador de Componentes Comerciais para o Joomla 1.5

Além dos dois componentes também veremos a criação de um gerador que apenas recebe a tabela e devolve um componente completo.

Impressiona o que se pode fazer com uma linguagem de programação como o PHP, em especial com os recursos para trabalhar com strings e arquivos, pastas e permissões. Esses recursos permitem automatizar muitas tarefas repetitivas e quase (passando parâmetros, variáveis, etc).

Ao identificar uma rotina que é utilizada várias vezes podemos criar uma função ou método para minimizar o esforço de programação e minimizar as possibilidades de erro.

Quando eu percebi que um componente para cadastro de clientes, funcionários, vendedores, etc, que cada um testes difere do outro apenas em alguns pontos, então me veio a ideia de criar um gerador que me faça praticamente todo o trabalho na criação do componente. Com a experiência anterior na criação de algo semelhante (Framework Ágil), um framework que automatiza a criação de aplicativos com muitos e bons recursos.

Dáí me veio a ideia de criar um gerador para componentes comerciais no Joomla 1.5.

Como funciona um gerador de aplicativos como este?

Ele guarda as partes fixas em variáveis ou constantes ou então faz cópias destas para a nova aplicação a ser gerada.

As partes variáveis devem ser adicionadas no momento da geração. Parecido com os parâmetros de funções.

Ao final juntamos as partes fixas com as variáveis e temos o resultado final.

Geralmente tem bem mais envolvido, mas simplificando é mais ou menos isto. Para detalhes veja o código do gerador.

Veja que o gerador tem um diretório onde existem vários arquivos. Existe um diretório images para as imagens da paginação. Um arquivo para a conexão, um para as funções usadas. A index, o script com a classe de paginação e os arquivos responsáveis pela geração de código:

```
generator_admin.php  
generator_site.php  
xml.php
```

generator_admin.php

```
<?php  
/**  
 * @version      $Id: install.jinstaller.php 1 2010-03-20$  
 * @author-name Ribamar FS  
 * @copyright    Copyright (C) 2010 Ribamar FS.  
 * @license      GNU/GPL, see http://www.gnu.org/licenses/old-licenses/gpl-2.0.txt  
 * JInstaller is free and open source software. This version may have  
 been modified  
 * pursuant to the GNU General Public License, and as distributed it  
 includes or is  
 * derivative of works licensed under the GNU General Public  
 License or other free or  
 * open source software licenses.  
 */  
  
include("../connection.inc.php");  
include_once("../functions.inc.php");  
  
$table = $_POST["table"]; // site  
define(DS, DIRECTORY_SEPARATOR);  
  
$prefix = explode('_', $table);  
$pref = $prefix[0];  
  
if($pref != 'jos'){
```

```

    print "<script>alert('The table must have jos
prprefix!');history.back();</script>";
    exit();
}
$postfix = substr($table, strlen($pref)+1, strlen($table)-strlen($pref)
+1);

$file_name=$postfix.'_users.php';

$index_str='<html><body bgcolor="#FFFFFF"></body></html>';

// component
if(!file_exists('com_.$postfix)){
    mkdir('com_.$postfix);
}
// index.html
if(!file_exists('com_.$postfix.DS.'index.html')){
    write_string('com_.$postfix.DS.'index.html',$index_str);
}
//admin
if(!file_exists('com_.$postfix.DS.'admin')){
    mkdir('com_.$postfix.DS.'admin');
}
if(!file_exists('com_.$postfix.DS.'admin'.DS.'index.html')){
    write_string('com_.$postfix.DS.'admin'.DS.'index.html',
$index_str);
}
//site
if(!file_exists('com_.$postfix.DS.'site')){
    mkdir('com_.$postfix.DS.'site');
}
if(!file_exists('com_.$postfix.DS.'site'.DS.'index.html')){
    write_string('com_.$postfix.DS.'site'.DS.'index.html',$index_str);
}
//tables
if(!file_exists('com_.$postfix.DS.'admin'.DS.'tables')){
    mkdir('com_.$postfix.DS.'admin'.DS.'tables');
}
if(!file_exists('com_.$postfix.DS.'admin'.DS.'tables'.DS.'index.html')){

```

```

        write_string('com_'.
$postfix.DS.'admin'.DS.'tables'.DS.'index.html',$index_str);
    }

// jtable content
$jtable_str ="<?php
defined( '_JEXEC' ) or die ( 'Restrict access!' );

class Table".ucfirst($postfix)." _users extends JTable
{
    var \$$id = null;
    var \$$login = null;
    var \$$name = null;
    var \$$published = null;

    function __construct (&\$db) {
        parent::__construct('#__".$postfix."_users','id',\$db);
    }
}
?>";

// JTable file
write_string('com_'. $postfix.DS.'admin'.DS.'tables'.DS.$file_name,
$jtable_str);

// install.sql

$install_str ="DROP TABLE IF EXISTS `#__".$postfix."_users`;

CREATE TABLE `#__".$postfix."_users` (
`id` int(11) NOT NULL AUTO_INCREMENT,
`login` char(32) NOT NULL,
`name` char(45) NOT NULL,
`published` tinyint(1) unsigned NOT NULL DEFAULT '0',
PRIMARY KEY (`id`),
UNIQUE KEY `login` (`login`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8
COLLATE=utf8_unicode_ci;\n\n";

```

```

$install_str2 = script_table($table);
$install_str2 = str_ireplace('jos_', '#__', $install_str2);

$install_str .= $install_str2;

write_string('com_'. $postfix .DS. 'admin'.DS. 'install.sql', $install_str);

// uninstall.sql
$uninstall_str= "DROP TABLE IF EXISTS `#__`$postfix`; \nDROP
TABLE IF EXISTS `#__`.$postfix.`_users`;";
write_string('com_'. $postfix .DS. 'admin'.DS. 'uninstall.sql',
$uninstall_str);

// admin.nomecomponente.php
$file_admin = 'admin.'. $postfix .'.php';

$file_admin_str ="<?php
    defined( '_JEXEC' ) or die( 'Acesso restrito ao Joomla!');

    require_once( JApplicationHelper::getPath( 'admin_html' ) );
    require_once( JPATH_COMPONENT.DS.'controller.php' );

    JTable::addIncludePath(JPATH_COMPONENT.DS.'tables');
    \ $controller = new
" .ucfirst($postfix). "Controller( array('default_task' => 'displayUsers' ) );

    \ $controller->execute( JRequest::getVar( 'task' ) );
    \ $controller->redirect();
?>
";
write_string('com_'. $postfix .DS. 'admin'.DS. $file_admin,
$file_admin_str);

// admin.nomecomponente.html.php
$file_adminhtml = 'admin.'. $postfix .'.html.php';

$file_adminhtml_str="<?php
    defined( '_JEXEC' ) or die( 'Acesso restrito ao Joomla!');
    class HTML_" . $postfix . " {

```



```

function editUsers(\$row, \$option) {
    ?>
    <form action="index.php" method="post"
name="adminForm" id="adminForm">
        <fieldset class="adminform">
            <legend>Cadastro de Usuários</legend>
            <table>
                <tr><td><label for="login">Login:</label></td>
                <td><select class="select" name="login" id="login"
class="required"/>

                    <?php print ".ucfirst(\$postfix)."Controller::combo();?>

                </select>
                <tr><td><label for="name">Name:</label></td>
                <td><input class="text" type="text" name="name"
id="name" size="45" maxlength="45"
                    value="<?php echo \$row->name; ?>"
class="required"/></td></tr>
            </table>
            <BR> <BR>
            </fieldset>
            <input type="hidden" name="id" value="<?php echo \$
$row->id; ?>" />
            <input type="hidden" name="option" value="<?php
echo \$option; ?>" />
            <input type="hidden" name="task" value="" />
            <?php echo JHTML::_('form.token' ); ?>
        </form>
    <?php
    }

function displayUsers( \$option, & \$rows, & \$pageNav ) {
    ?>
    <form action="index.php" method="post"
name="adminForm">
        <table class="adminlist">
            <thead>

```

```

<tr>
<th width="20">
<input type="checkbox" name="toggle" value=""
onclick="checkAll(<?php echo count( \$rows ); ?>);" />
</th>
<th class="title">Login</th>
<th class="title">Name</th>
<th width="15%">Published</th>
</tr>
</thead>
<?php

jimport('joomla.filter.output');

\$k = 0;
for (\$i=0, \$n=count( \$rows ); \$i < \$n; \$i++) {
    \$row = &\$rows[\$i];
    \$checked = JHTML::_('grid.id', \$i, \$row->id );
    \$published = JHTML::_('grid.published', \$row, \$i );
    \$link = JFilterOutput::ampReplace( 'index.php?
option=' . \$option . '&task=edit&cid[]=' . \$row->id );
    ?>
<tr class="<?php echo "row\$k"; ?>">
<td>
<?php echo \$checked; ?>
</td>
<td>
<a href="<?php echo \$link; ?>">
<?php echo \$row->login; ?></a>
</td>
<td>
<?php echo \$row->name; ?></a>
</td>
<td align="center">
<?php echo \$published; ?>
</td>
</tr>
<?php
\$k = 1 - \$k;

```

```

    }
    ?>
    <tfoot>
    <td colspan="7"><?php echo \$pageNav-
>getListFooter(); ?></td>
    </tfoot>
    </table>
    <input type="hidden" name="option" value="<?php
echo \$option;?>" />
    <input type="hidden" name="task" value="" />
    <input type="hidden" name="checked" value="0"
/>
    <?php echo JHTML::_( 'form.token' ); ?>
    </form>
    <?php
    }
}
?>
";
write_string('com_'.$postfix.DS.'admin'.DS.$file_adminhtml,
$file_adminhtml_str);

// File controller.php
$file_controller_str = "<?php
defined( '_JEXEC' ) or die( 'Access denied!' );

jimport( 'joomla.application.component.controller' );

class ".ucfirst($postfix)."Controller extends JController
{
    function __construct( \$default = array() )
    {
        parent::__construct( \$default );
        \$this->registerTask( 'add', 'edit' );
        \$this->registerTask( 'apply', 'save' );
        \$this->registerTask( 'unpublish', 'publish' );
    }
}

```

```

function combo(\$default_value=null)
{
    \$db =& JFactory::getDBO();
    \$query='select username from #__users order by
username';
    \$db->setQuery (\$query);
    \$db->query(\$query);
    \$nr=\$db->getNumRows();

    \$n=0;
    while(\$n < \$nr){
        \$row=\$db->loadResultArray();
        if(\$row[\$n] == \$default_value)
        {
            \$combo .= "<option value=\$row[\$n]
selected='selected'>\$row[\$n]</option>\";
        }else{
            \$combo .= "<option value=\$row[\$n]>\$row[\$n]</option>\";
        }
        \$n++;
    }
    return \$combo;
}

```

```

function edit() { // Edit or New
    global \$option;
    \$row =& JTable::getInstance( ".\$postfix."_users', 'Table');
    \$cid = JRequest::getVar( 'cid', array(0), ", 'array' );
    \$id = \$cid[0];

    if (\$id != 0) { // Edit
        \$row->load(\$id);
    } else { // New
        \$row->login = JRequest::getVar( 'login', "", 'post', 'string',
JREQUEST_ALLOWRAW );
        \$row->name = JRequest::getVar( 'name', "", 'post',
'string', JREQUEST_ALLOWRAW );
    }
}

```

```

    }
    HTML_".$postfix."::editUsers(\$row, \$option);
}

function save() {
    global \$option;
    global \$mainframe;

    JRequest::checkToken() or die( 'Token Inválido ' );

    \$row =& jTable::getInstance( "$postfix._users", 'Table' );

    if (!\$row->bind(JRequest::get('post'))) {
        echo "<script> alert('\$.\$row->getError().\");
window.history.go(-1); </script>\";
        exit();
    }

    \$login = JRequest::getVar( 'login', "", 'post', 'string',
JREQUEST_ALLOWRAW );
    \$name = JRequest::getVar( 'name', "", 'post', 'string',
JREQUEST_ALLOWRAW );

    \$row->login = \$login;
    \$row->name = \$name;

    \$erro = false;
    if (trim(\$login) == "" || \$login == null) {
        \$erro = true;
        \$msg = JText::_( 'Login is required!' );
        \$mainframe->enqueueMessage(\$msg, 'error');
    }

    if (trim(\$name) == "" || \$name == null) {
        \$erro = true;
        \$msg = JText::_( 'Name is required!' );
        \$mainframe->enqueueMessage(\$msg, 'error');
    }
}

```

```

        if (\$erro ) {
            return \$this->execute('edit');
        }

        if (!\$row->store()) {
            echo "\<script> alert(\".\$row->getError().\");
window.history.go(-1); </script>\";
            exit();
        }

        switch (\$this->_task) {
            case 'apply':
                \$msg = 'Sucessfull changes!';
                \$link = 'index.php?option=' . \$option .
                '&task=edit&cid[]=' . \$row->id;
                break;

            case 'save':
            default:
                \$msg = 'Register writed!';
                \$link = 'index.php?option=' . \$option;
                break;
        }

        \$this->setRedirect(\$link, \$msg);
    }

function displayUsers()
{
    global \$option, \$mainframe;
    \$limit = JRequest::getVar('limit', \$mainframe-
>getConfig('list_limit'));
    \$limitstart = JRequest::getVar('limitstart', 0);
    \$db =& JFactory::getDBO();

    \$query = 'SELECT count(*) FROM #__'.$postfix.'_users';
    \$db->setQuery( \$query );
    \$total = \$db->loadResult();
}

```

```

        \ $query = 'SELECT * FROM #__".$postfix."_users';
        \ $db->setQuery( \ $query, \ $limitstart, \ $limit );
        \ $rows = \ $db->loadObjectList();
        if ( \ $db->getErrorNum() ) {
            echo \ $db->stderr();
            return false;
        }
        jimport('joomla.html.pagination');
        \ $pageNav = new JPagination( \ $total, \ $limitstart, \ $limit);
        HTML_" . $postfix . " :: displayUsers( \ $option, \ $rows, \
$pageNav );
    }

function remove()
{
    global \ $option;
    \ $cid = JRequest::getVar( 'cid', array(), "", 'array' );
    \ $db =& JFactory::getDBO();
    if(count( \ $cid))
    {
        JArrayHelper::toInteger( \ $cid);
        \ $cids = implode( ',', \ $cid );
        \ $query = 'DELETE FROM #__".$postfix."_users
WHERE id IN ( \ $cids );
        \ $db->setQuery( \ $query );
        if ( ! \ $db->query() ) {
            echo "\<script> alert("\ . \ $db->getErrorMessage() . "\");
window.history.go(-1); </script>";
        }
    }
    \ $this->setRedirect( 'index.php?option=' . \ $option );
}

function publish()
{
    global \ $option;
    \ $cid = JRequest::getVar( 'cid', array(), "", 'array' );
    if ( \ $this->_task == 'publish' )

```

```

        {
            \ $publish = 1;
        }else{
            \ $publish = 0;
        }
        \ $. $postfix. "Table =& jTable::getInstance(" .
$postfix. "_users", 'Table');
        \ $. $postfix. "Table->publish(\ $cid, \ $publish);
        \ $this->setRedirect( 'index.php?option=' . \ $option );
    }
}
?>";
write_string('com_'. $postfix. DS. 'admin'. DS. 'controller.php',
$file_controller_str);

// toolbar.nocomponente.html.php
$toolbarhtml_str ="<?php
    defined( '_JEXEC' ) or die( 'Access denied!' );
    class toolbar_". $postfix. " {
        function _NEW() {
            JToolBarHelper::title( JText::_('New User'),'generic.png');
            JToolBarHelper::save();
            JToolBarHelper::apply();
            JToolBarHelper::cancel();
        }
        function _DEFAULT() {
            JToolBarHelper::title( JText::_('User
Management'),'generic.png');
            JToolBarHelper::publishList();
            JToolBarHelper::unpublishList();
            JToolBarHelper::editList();
            JToolBarHelper::deleteList();
            JToolBarHelper::addNew();
        }
    }
?>";
$toolbarhtml_name = 'toolbar.'. $postfix. '.html.php';
write_string('com_'. $postfix. DS. 'admin'. DS. $toolbarhtml_name,
$toolbarhtml_str);

```



```

// toolbar.nomecomponente.php
$toolbar_name = 'toolbar.'.$postfix.'.php';

$toolbar_str="<?php
    defined( '_JEXEC' ) or die( 'Access denied!' );

    require_once( JApplicationHelper::getPath( 'toolbar_html' ) );

    switch(\$task) {
        case 'add':
        case 'edit':
        case 'save':
        case 'apply':
            toolbar_".$postfix."::_NEW();
            break;
        default:
            toolbar_".$postfix."::_DEFAULT();
            break;
    }
?>";
write_string('com_'. $postfix .DS. 'admin'.DS. $toolbar_name,
$toolbar_str);

include_once('./generator_site.php');
include_once('./xml.php');
?>

```

Este script contém as partes que formarão os arquivos e diretórios da parte admin do componente.

Veja que criamos as variáveis com o conteúdo dos arquivos e depois criamos os arquivos e seus diretórios.

generator_site.php

```
<?php
/**
 * @version      $Id: install.jinstaller.php 1 2010-03-20$
 * @author-name Ribamar FS
 * @copyright    Copyright (C) 2010 Ribamar FS.
 * @license      GNU/GPL, see http://www.gnu.org/licenses/old-licenses/gpl-2.0.txt
 * JInstaller is free and open source software. This version may have
 been modified
 * pursuant to the GNU General Public License, and as distributed it
 includes or is
 * derivative of works licensed under the GNU General Public
 License or other free or
 * open source software licenses.
 */

recurse_copy('images','com_'. $postfix.DS.'site'.DS.'images');
copy('page.inc.php','com_'. $postfix.DS.'site'.DS.'page.inc.php');

$file_name=$postfix.'.php';

$res = mysql_query("select * from $table");
$fields = mysql_num_fields($res);

$add_escape = "";
$insert_str = "";
$values = ' VALUES (';
$fields .= '(';
$update_str = "\"UPDATE #__postfix SET ";

for ($i=0; $i < $fields; $i++) {
    $name = mysql_field_name($res, $i);

    if($i==0){
        $add_request .= "$$name = JRequest::getVar( '$name');\n";
    }else{
        $add_request .= "\t\t$$name =
```

```

JRequest::getVar( '$name');\n";
    }

    if($i==0){
        $add_escape .= "$$name = \"\".\$db->getEscaped( $
$name).\\"";\n";
    }else{
        $add_escape .= "\t\t$$name = \"\".\$db->getEscaped( $
$name).\\"";\n";
    }

    if($i<$fields-1){
        $flds .= $name .';';
    }else{
        $flds .= $name;
    }

    if($i<$fields-1){
        $vls .= "$$name" .';';
    }else{
        $vls .= "$".$name;
    }

    if($i>0){
        if($i<$fields-1){
            $update_str .= "$name = \".$$name.\\"";
        }else{
            $update_str .= "$name = \"\".\\"";
        }
    }
    if($name='id') $pk=$name;
}
$insert_str = "\"INSERT INTO #__postfix $flds) $vls);\\"";

$update_str = $update_str . " WHERE id=\\"$. $pk.'";

$file_str="<?php
/**
* Componente com_".$postfix."

```

```

* Author: Ribamar FS
* http://ribafs.org
*/

// no direct access
defined( '_JEXEC' ) or die( 'Restricted access' );

jimport( 'joomla.application.helper' );
require_once( JApplicationHelper::getPath( 'front_html', 'com_'.
$postfix." " ) );
require_once(JPATH_SITE.DS.'components'.DS.'com_'.
$postfix.".DS.'page.inc.php');

$user =& JFactory::getUser();
$username = $user->get('username');
/$usertype = $user->get('usertype');
$aid = $user->get('aid'); // 0 - guest, 1 - registered and 2 - special

$db =& JFactory::getDBO();

$query = "SELECT login FROM #__".$postfix."_users WHERE
published = 1 AND login = '". $username. "'";

$db->setQuery( $query, 0);
$login = $db->loadResult();

// If is not authenticated login or group is not special redirection to
index.php
if($login != $username || $aid != 2) JFactory::getApplication()-
>redirect('index.php',"");

// Check the task parameter and execute appropriate function
switch( JRequest::getVar( 'task' )) {
    case 'add':
        add();
        break;
    case 'upd':
        upd();
}

```

```

        break;
    case 'del':
        del();
        break;
    case 'grid':
        grid();
        break;
    case 'search':
        search();
        break;
    default:
        grid();
        break;
}

```

```

function date_db(\$date){ // Converte de d/m/Y para Y-m-d
    \$date = explode('/',\$date);
    \$date = \$date[2].'-'.\$date[1].'-'.\$date[0];
    return \$date;
}

```

```

function add() {
    Output".ucfirst(\$postfix).":add".ucfirst(\$postfix)."; // Exibe o
form
    \$submitnew = JRequest::getVar('SubmitNew') ;
    \$msg = "";

    if(isset(\$submitnew)){
        ".$add_request."

        \$db =& JFactory::getDBO();

        \$query = 'SELECT * FROM #__".\$postfix.'" WHERE id='\.
\$id;
        \$db->setQuery( \$query, 0);

        \$queryc = 'SELECT count(*) FROM #__".\$postfix.'"

```

```

WHERE id='\$id;
    \$db->setQuery( \$queryc, 0);
    \$nr = \$db->loadResult();

    if(\$nr > 0){
        JFactory::getApplication()->redirect('index.php?
option=com_".\$postfix."&task=add','Já existe um ".\$postfix." com
este campo!','error');
    }
/*
    if(\$cpf==""){
        \$msg .= 'CPF requerido! <br> ';
    }

    if(\$nome==""){
        \$msg .= 'Nome requerido! <br> ';
    }

    if(\$msg) {
        JFactory::getApplication()->redirect('index.php?
option=com_".\$postfix."&task=add&id=\$id',\$msg,'error');
    }
*/
    ".$add_escape."

    \$insertFields = ".$insert_str."

    \$db->setQuery( \$insertFields, 0);
    \$db->query();

    if (\$db->getErrorNum() != 0) {
        \$error = \$db->getErrorMessage().JFactory::getApplication()-
>redirect('index.php?option=com_".\$postfix."&task=grid','Erro na
consulta insertFields','error');
    }else{
        JFactory::getApplication()->redirect('index.php?
option=com_".\$postfix."&task=grid','".ucfirst(\$postfix)." cadastrado
com sucesso!','message');
    }

```

```

    }
} // End function add_".$postfix."

function upd()
{
    \ $subupd = JRequest::getVar('SubmitUpd') ;
    \ $db =& JFactory::getDBO();

    if(isset(\ $subupd)){
        ". $add_request."
    /*
    if(\ $nome==""){
        \ $msg .= JFactory::getApplication()->enqueueMessage('Nome
requerido! ', 'error');
    }

    if(\ $msg) JFactory::getApplication()->enqueueMessage(\
    $msg.'<a href="index.php?option=com_". $postfix."&task=upd&id=
    $sid">Voltar</a>', 'error');
    */
        // Adicionar a aspa simples (crase) para eescapar os
valores do Insert
        ". $add_escape."

        \ $updateFields = ". $update_str."

        \ $db->setQuery( \ $updateFields, 0);
        \ $rows = \ $db->loadObjectList();
        \ $db->query();

        if (\ $db->getErrorNum() != 0) {
            \ $error = \ $db-
>getErrorMessage().JFactory::getApplication()->enqueueMessage('Erro
na consulta updateFields!', 'error');
        }else{
            JFactory::getApplication()->redirect('index.php?
option=com_". $postfix."&task=grid', "" .ucfirst($postfix)." atualizado
com sucesso!', 'message');

```

```

    }
}

\ $id = JRequest::getVar('id') ;

\ $db =& JFactory::getDBO();
\ $query = \"SELECT * FROM #__\".$postfix.\" WHERE id =\".\
$id.\"\"\";
\ $db->setQuery( \ $query, 0);
\ $rows = \ $db->loadObjectList();

    Output\".ucfirst($postfix).\"::upd\".ucfirst($postfix).\"(\ $rows); //
Mostra form update

} // end function upd()

function del() {
    \ $db =& JFactory::getDBO();

    // Get ID from form posting values
    \ $id = JRequest::getVar('id') ;
    \ $msg = \";

    if(\ $id=="){
        \ $msg .= JFactory::getApplication()->enqueueMessage('ID
requerido!', 'error');
    }

    if(\ $msg) JFactory::getApplication()->enqueueMessage(\
$msg, 'error');

    \ $deleteFields = \"DELETE FROM #__\"$postfix \" WHERE id = \".\
$id;

    \ $db->setQuery( \ $deleteFields, 0);
    \ $db->query();

    if (\ $db->getErrorNum() != 0) {

```



```

        \$error = \$db->getErrorMsg().JFactory::getApplication()-
>enqueueMessage('Erro na consulta deleteFields!', 'error');
    }else{
        JFactory::getApplication()->redirect('index.php?
option=com_".$postfix."'."."ucfirst($postfix)." excluído com
sucesso!', 'message');
    }

```

```

    Output".ucfirst($postfix)."::del".ucfirst($postfix)."; // Executa
rotina del do ".$postfix.".html.php

```

```

} // end del() function

```

```

function grid() {
    // Paging
    \ $config =& JFactory::getConfig();

    \ $host=\ $config->getValue('host');
    \ $db = \ $config->getValue('db');
    \ $pass = \ $config->getValue('password');
    \ $user = \ $config->getValue('user');
    \ $dbprefix = \ $config->getValue('dbprefix');

    mysql_connect(\ $host, \ $user, \ $pass);
    mysql_select_db(\ $db);

    \ $query = 'SELECT * FROM '.\ $dbprefix.'".".".$postfix." ORDER BY
id ASC';
    \ $result=mysql_query(\ $query);
    \ $total_rec=mysql_num_rows(\ $result);

    if (mysql_error()) {
        \ $error = JFactory::getApplication()->enqueueMessage('Erro
na consulta rows!', 'error');
    }

    \ $page=new Page(); // creating new instance of Class Page

```

```

        \ $recpp = 10; // Registros por página
        \ $scroll = 20; // Visíveis

        \ $page->set_page_data(\$_SERVER['PHP_SELF'],\ $total_rec,\
        $recpp,\ $scroll,true,true,true);
        \ $result=mysql_query(\ $page->get_limit_query(\ $query));

        Output".ucfirst($postfix)."::grid".ucfirst($postfix)."\ $result,\
        $page,\ $total_rec);// Executa rotina para exibir registros do ".
        $postfix."
    }

function search() {

    // Get ID from form posting values
    \ $id = JRequest::getVar('id') ;
    \ $nome = JRequest::getVar('nome') ;

    \ $db =& JFactory::getDBO();

    \ $query = "\ SELECT id, cpf, nome FROM #__". $postfix." WHERE
    nome LIKE '%\ ". \ $nome. \ "%' ORDER BY nome\ ";
    \ $db->setQuery( \ $query, 0); \ $rows = \ $db->loadObjectList();

    if (\ $db->getErrorNum() != 0) {
        \ $error = \ $db->getErrorMsg().JFactory::getApplication()-
        >enqueueMessage('Erro na consulta selectf!', 'error');
    }

    Output".ucfirst($postfix)."::search".ucfirst($postfix)."\ $rows);//
    Executa rotina para exibir registros do ". $postfix."
}

if(\ $error != "") JFactory::getApplication()->enqueueMessage(\
$error,'error');

?>";

```

```

// JTable file class
write_string('com_'. $postfix.DS.'site'.DS.$file_name,$file_str);

$filehtml_name=$postfix.'.html.php';

// form add
for ($i=0; $i < $fields; $i++) {
    $name = mysql_field_name($res, $i);
    if($i==0){
        $len = mysql_field_len($res, $i);
    }else{
        $len = mysql_field_len($res, $i);
    }

    if($name !='id'){
        if($i < $fields -1){
            $form_add .= "<tr><td>".ucfirst($name)."</td><td><input
type='text'" name=\"$name\" size=\"$len\"
maxlength=\"$len\"></td></tr>\n\t\t";
        }else{
            $form_add .= "<tr><td>".ucfirst($name)."</td><td><input
type='text'" name=\"$name\" size=\"$len\"
maxlength=\"$len\"></td></tr>";
        }
    }

    if($i ==0){
        $loop_row .= "$$name = \$row->$name;\n";
    }else{
        if($i < $fields -1){
            $loop_row .= "\t\t\t\t$$name = \$row->$name;\n";
        }else{
            $loop_row .= "\t\t\t\t$$name = \$row->$name;";
        }
    }
}

if($i==1) $field1=$name;
if($i==2) {
    $field2=$name;
}

```

```

        $len2 = $len;
    }

    if($name !='id'){
        if($i ==1){
            $form_upd .= "<tr><td>".ucfirst($name)."</td><td><input
type='\"text\"' name='\"$name\"' size='\"$len\"' maxlength='\"$len\"'
value='\"<?php print $$name;?>\"></td></tr>\n";
        }else{
            $form_upd .= "\t\t<tr><td>".ucfirst($name)."</td><td><input
type='\"text\"' name='\"$name\"' size='\"$len\"' maxlength='\"$len\"'
value='\"<?php print $$name;?>\"></td></tr>\n";
        }
    }
}

$filehtml_str ="<?php
/**
 * @$. $postfix. ".html.php
 * @Ribamar FS.
 * @http://ribafs.org
 */

// no direct access
defined( '_JEXEC' ) or die( 'Restricted access' );

class Output".ucfirst($postfix)."
{
    function add".ucfirst($postfix)."{
        ?>
        <body onLoad='\"document.form1.\".$field1.\".focus();\">
        <h2 class='\"moduletable\">Cadastro de ".ucfirst($postfix)."</h2>
        <div align='\"left\">
        <br>
        <form id='\"form1\"' name='\"form1\"' method='\"post\"'
action='\"index.php?option=com_\".$postfix.\"&task=add\">
        <table border='\"0\">
            ".$form_add."

```

```

        <input type="hidden" name="id" value="<?php print \$id;?
>" />
        <tr><td>&nbsp;</td><td><input type="submit"
name="SubmitNew" value="Adicionar" /></td></tr>
    </table>
</form>
<br>
</div>
<br>
<?php
}

function upd".ucfirst($postfix)."(\\$rows){

    if(\\$rows) {
        foreach (\\$rows as \\$row){
            ".\$loop_row."
        ?>
        <h2 class="moduletable">Cadastro de ".ucfirst($postfix)." -
Alteração</h2>
        <div align="left">

            <form id="form1" name="form1" method="post"
action="index.php?option=com_".\$postfix."&task=upd" >
                <br>
                <table border="0">
                    ".\$form_upd."
                    <input type="hidden" name="id" value="<?php print \$id;?
>" />
                    <tr><td>&nbsp;</td><td><input type="submit"
name="SubmitUpd" value="Alterar" /></td></tr>
                </table>
            </form>
            <br>
        </div>
        <?php
        } // End foreach
    } //End if

```

```

        \$_subupdpf = JRequest::getVar('SubmitUpdpf');
        if(isset(\$_subupdpf)){
            JFactory::getApplication()->redirect("index.php?option=com_".\$postfix."&task=grid","message");
        }
    }
}

```

// End function

```

function del".ucfirst(\$postfix)."( ) {
    \$id = JRequest::getVar('id') ;
    JFactory::getApplication()->redirect("index.php?option=com_".\$postfix."&task=grid&id=".\$id.".","message");
}

```

```

function grid".ucfirst(\$postfix)."( \$result, \$page, \$total_rec){
?>
    <h1 class="moduletable">Controle de ".ucfirst(\$postfix)."</h1>

    <table border="0">

<tr><td><b>ID</td><td><b>".ucfirst(\$field1)."</td><td><b>".ucfirst(\$field2)."</td><td colspan="2" align="center"><b>Ação</td></tr>
<?php

```

```

        \$fld2= \$field2;
        if(\$result) {
            while(\$row=mysql_fetch_array(\$result))
            {
                \$id = \$row["id"];
                \$".\$field1." = \$row[".\$field1."];
                \$".\$field2." = \$row[".\$field2."];

                ?>
                <tr><td><?php print \$id;?></td><td><?php print \$".\$field1.";?>&nbsp;</td><td>&nbsp;<?php print \$".\$field2.";?>&nbsp;<td>&nbsp;<?php print "&nbsp;</td></tr>
            }
        }
    }
}

```

```

        <td><a href="\index.php?option=com_".
$postfix."&task=upd&id=<?php print \$id;?
>\>">Alterar</a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td>
        <td><a href="\index.php?option=com_".
$postfix."&task=del&id=<?php print \$id;?>" onClick="\if(!
confirm('Tem certeza de que deseja excluir o registro '+<?php print \
$id;?>+' ?')) {return false;}\">Excluir</a></td>
    </tr>

    <?php
} // End while
}

    \$f = 'components/com_".\$postfix."/images/first.png';
    \$n = 'components/com_".\$postfix."/images/next.png';
    \$p = 'components/com_".\$postfix."/images/prev.png';
    \$l = 'components/com_".\$postfix."/images/last.png';

    \$comp = 'com_".\$postfix."";

    print '<tr><td colspan=\"5\">';
        \$page->get_first_page_nav("\<img src=\".\$f.\"
title='Primeiro'\>\",\$comp);
        \$page->get_next_page_nav("\<img src=\".\$n.\"
title='Próximo'\>\",\$comp);
        \$page->get_number_page_nav("\",\$comp);
        \$page->get_prev_page_nav("\<img src=\".\$p.\"
title='Anterior'\>\",\$comp);
        \$page->get_last_page_nav("\<img src=\".\$l.\" title='Último
(total de \$total_rec registros)\",\$comp);
    print '</td></tr>';
    ?>
    <tr><td colspan=\"5\" align=\"center\"><a href=\"index.php?
option=com_".\$postfix."&task=add\">Adicionar Novo</a></td></tr>

    <form method=\"post\" action=\"index.php?option=com_".
$postfix."&task=search\">
    <tr><td colspan=\"4\"><input type=\"text\" name=\"<?php print
$field2;?>" size=\"\$len2\" value=\"Digite o nome ou parte\"
onClick=\"this.value=\"></td>

```

```

        <td><input type="submit" value="\Localizar"></td></tr>
    </form>

</table>
<?php
}// End function grid

function search".ucfirst($postfix)."\($rows) {
?>
    <h1 class="moduletable">Localização de
".ucfirst($postfix)."</h1>

    <table border="0">
    <tr><td><b>".ucfirst($field1)."</td>
        <td><b>".ucfirst($field2)."</td>
        <td colspan='2' align='center'><b>Ação</td>
    </tr>
<?php
    if(\($rows) {
        foreach (\$rows as \$row){
            // Encode text to HTML formatting
            ".$loop_row3."
            ?>
            <tr><td>&nbsp;<?php print \$field1;?>&nbsp;</td>
                <td>&nbsp;<?php print \$field2;?
            >&nbsp;&nbsp;&nbsp;</td>

                <td><a href='index.php?option=com_".
$postfix."&task=upd&id=<?php print \$id;?>'>Alterar</a></td>
                <td><a href='index.php?option=com_".
$postfix."&task=del&id=<?php print \$id;?>' onClic="if(!
confirm('Tem certeza de que deseja excluir '+<?php print \$field2;?>
+ '?')) {return false;}\">Excluir</a></td>
            </tr>

            <?php
        }// End foreach
    }// End if (\$rows)

```



```

        print '</table><br><br><br>';
    }// End function grid
}// End class

function dateBr(\$date){// Converte de Y-m-d para d/m/Y
    \$date = explode('-',\$date);
    \$date = \$date[2].'/'.$date[1].'/'.$date[0];
    return \$date;
}

?>;
write_string('com_'.\$postfix.DS.'site'.DS.\$filehtml_name,
\$filehtml_str);

?>

```

Este gera os arquivos e diretórios da parte referente ao site do componente.

xml.php

```
<?php
/**
 * @version      $Id: install.jinstaller.php 1 2010-03-20$
 * @author-name Ribamar FS
 * @copyright    Copyright (C) 2010 Ribamar FS.
 * @license      GNU/GPL, see http://www.gnu.org/licenses/old-licenses/gpl-2.0.txt
 * JInstaller is free and open source software. This version may have
 * been modified
 * pursuant to the GNU General Public License, and as distributed it
 * includes or is
 * derivative of works licensed under the GNU General Public
 * License or other free or
 * open source software licenses.
 */
```

```
$xml_str = "<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE install
SYSTEM \"http://dev.joomla.org/xml/1.5/component-install.dtd\">
<install type='component' version='1.5'>
  <name>\".ucfirst($postfix).\"</name>
  <creationDate>Março de 2010</creationDate>
  <author>ComponentsGen</author>
  <authorEmail>ribafs@ribafs.org</authorEmail>
  <authorUrl>http://ribafs.org</authorUrl>
  <copyright>(c) 2010</copyright>
  <license>Livres</license>
  <version>1.0.0</version>
  <description>
    Component generated with components_gen. Created by
    Ribamar FS.
  </description>
  <filename>leiname.txt</filename>
  <filename>index.html</filename>
  <files folder = \"site\">
    <filename>\".$postfix.\".php</filename>
    <filename>\".$postfix.\".html.php</filename>
```

```

        <filename>page.inc.php</filename>
        <filename>index.html</filename>
        <filename>images/first.png</filename>
        <filename>images/next.png</filename>
        <filename>images/prev.png</filename>
        <filename>images/last.png</filename>
    </files>
    <install>
        <sql>
            <file driver="mysql" charset="utf8">install.sql</file>
        </sql>
    </install>
    <uninstall>
        <sql>
            <file driver="mysql" charset="utf8">uninstall.sql</file>
        </sql>
    </uninstall>
    <administration>
        <menu>".ucfirst($postfix)."</menu>
        <files folder="admin">
            <filename>index.html</filename>
            <filename>install.sql</filename>
            <filename>uninstall.sql</filename>
            <filename>admin.".$postfix.".html.php</filename>
            <filename>admin.".$postfix.".php</filename>
            <filename>controller.php</filename>
            <filename>toolbar.".$postfix.".html.php</filename>
            <filename>toolbar.".$postfix.".php</filename>
            <filename>tables/".$postfix."_users.php</filename>
            <filename>tables/index.html</filename>
        </files>
    </administration>
</install>";

// XML write
write_string('com_' . $postfix . DS . $postfix . '.xml', $xml_str);
$path = dirname(__FILE__) . DS . 'com_' . $postfix;

print "Component generated. <h3>". $path . "</h3>Now package it and

```

```
install!";  
?>
```

Este arquivo cria o instalador (xml) do componente.

Como ele cria de forma genérica talvez seja interessante editar o arquivo gerado a pós a geração para ajustes.

Versões para Joomla 1.7

Os mesmos componentes clientes e estoque com versão para Joomla 1.7

Como são muito semelhantes apenas falarei do que mudou até agora no Joomla 1.7.

Ressaltar:

- Diferenças no XML
- Diferenças no item de menu para o componente
- Diferenças no suporte a languages/traduições.
- Também houve diferença no path interno dos arquivos. Veja por exemplo, que no componente cadastro de clientes o path das imagens na versão 1.7 precisou ser alterado para:

```
$f = '../components/com_clientes/imagens/first.png';
```

Enquanto que na versão para 1.5 era assim:

```
$f = 'components/com_clientes/imagens/first.png';
```

Adicionei ../ no início de cada imagem, indicando que o path agora está na pasta components.

O que mudou nos arquivos de tradução

Usando JavaScript na tradução, com a função script, por exemplo:

```
JText::script('com_helloworld_HelloWorld_Error_Some_Values_Are_Unacceptable');
```

As strings agora terão o format o:

```
COM_HELLOWORLD_ADMINISTRATION=HelloWorld
```

```
administration
```

```
COM_HELLOWORLD_CATEGORIES=Categories
```

```
COM_HELLOWORLD_MANAGER=HelloWorld manager
```

```
COM_HELLOWORLD_MESSAGES=Messages
```

```
COM_HELLOWORLD_HELLOWORLD_CATEGORY=Category
```

```
COM_HELLOWORLD_HELLOWORLD_CATEGORY_DESC=Category of the message
```

```
COM_HELLOWORLD_HELLOWORLD_CREATING=Creating
```

COM_HELLOWORLD_HELLOWORLD_DETAILS=Details
COM_HELLOWORLD_HELLOWORLD_OPTIONS=Options
COM_HELLOWORLD_HELLOWORLD_EDITING=Editing
COM_HELLOWORLD_HELLOWORLD_ERROR_SOME_VALUES_ ARE_UNACCEPTABLE=Some values are unacceptable
COM_HELLOWORLD_HELLOWORLD_GREETING_DESC=Messa ge to be displayed

Melhora no Template

Para que nossos componentes funcionem na versão 1.7 alpha2 do Joomla devemos fazer alguns pequenos ajustes:

- Como o Joomla 1.7 tem um bom controle de usuários e acesso aos componentes, podemos remover os arquivos da parte admin do componente, deixando apenas o install.sql, uninstall.sql e index.html e fazer os devidos ajustes no xml.

- Mais um ajuste, e este necessário, é criar um diretório views na parte do site que será o responsável pelo menu no admin do Joomla 1.7. Será assim para o caso do componente cadastro de clientes: site/views/cliente/tmpl/default.xml

O conteúdo do default.xml deve ser:

```
<?xml version="1.0" encoding="utf-8"?>
<metadata>
<layout title="com_clientes_Clientes_View_Default_Title">
<message>com_clientes_Clientes_View_Default_Desc</
message>
</layout>
<fieldset name="request">
<fields
group="request"
array="true"
>
<field
id="id"
name="id"
type="list"
label="com_clientes_Clientes_View_Default_Ms
```

```
g_Label"
description="clientes_Clientes_View_Default_Ms
g_Desc"
default="1"
>
<option value="1">Cadastro de Clientes!
</option>
</field>
</fields>
</fieldset>
</metadata>
Ou somente assim:
<?xml version="1.0" encoding="utf-8" ?>
<metadata>
<layout title="com_estoque_Estoque_Default_Title">
<message>com_estoque_Estoque_Default_Title</messag
e>
</layout>
</metadata>
```

Referências

Livros

- Learning Joomla! 1.5 Extension Development do Joseph LeBlanc - Editora Packt Publishing
- Mastering Joomla! 1.5 Extension and Framework Development do James Kennard - Editora Packt Publishing
- Joomla Avançado de Ricardo Lima Caratti e Leonardo Mafra Silva - Editora Novatec - <http://www.novateceditora.com.br/>
- Professional Joomla! do Dan Rahmel - Editora Wiley Publishing, Inc. - <http://www.wiley.com>
- Joomla! 1.5 Development Cookbook do James Kennard - Editora Packt Publishing
- Joomla! 1.5 Template Design do Tessa Blakeley Silver - Editora Packt Publishing
- Joomla! E-Commerce with VirtueMart do Suhreed Sarkar - Editora Packt Publishing
- Joomla! Web Security do Tom Canavan - Editora Packt Publishing
- Joomla! Bible do Ric Shreves – Editora Wiley Publishing Inc.

Dica: Geralmente as editoras oferecem o código fonte dos livros gratuitamente, como é o caso da Packt Publishing <http://www.packtpub.com/support>.

Sites

Joomla

<http://www.joomla.org/>
<http://extensions.joomla.org/>
<http://www.joomla24.com/> (Tempates)
<http://joomlaos.de/> (Templates)
[http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/CMSs_\(Gerenciadores_de_Conte%C3%BAdo\)/Joomla_\(Portais\)](http://pt.wikibooks.org/wiki/Aplicativos_em_PHP/CMSs_(Gerenciadores_de_Conte%C3%BAdo)/Joomla_(Portais))

Language pt_BR

<http://joomla-code.org/gf/project/portuguesebr/frs/>

Extensões diversas

<http://extensions.joomla.org/>

<http://extensoes.ribafs.org>

Template Builder Online

<http://www.joomlatemplatebuilder.com/template-builder.php>

Apêndice

Para melhorar a aparência do Joomla estando no template default da versão 1.5 como também da versão 1.7 e acredito que também em outros templates (não testei), adicionei ao arquivo template.css do template em uso o seguinte trecho:

```
/* Adicionei */

* {
    font-family:Georgia, Arial, Verdana;
    font-size:12px;
    line-height:18px;
    padding:0;
    margin:0;
    color:#000000;
    list-style:none;
    text-align:left;
}

h1,h2,h3,h4h5,h6,p,blockquote,form,label,ul,ol,dl,fieldset,address {
    margin: 0.5em 0;
}

table {
background-color: #FFFFFF;
}

td, th {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 11px;
    line-height: 24px;
    color: #0C3A6D;
}

/* Forms Controls*/

form {
    margin-left: 0;
```

```
background-color: #FFFFFF;
text-align: left;
}
```

```
input {
background-color: #E9E9E9;
border-style: 1px solid;
border-color: blue gray blue;
color: #003366;
font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 11px;
}
```

```
input:focus {
background-color: #E9E9E9;
border-style: 1px solid;
border-color: red red red red;
color: green;
font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 11px;
}
```

```
select {
background-color: #B5F7A2;
font:12px verdana, arial, helvetica, sans-serif;
color:black;
}
```

```
select:focus {
background-color: #276527;
font:12px verdana, arial, helvetica, sans-serif;
color:white;
}
```

```
li,dd {
margin-left:0em;
}
```

```
fieldset {
```

```
padding:0.5em;  
}  
/*\Adicionei */
```

Ambiente de Criação e Teste

Este livro e os componentes foram criados em um ambiente com:

- Linux Ubuntu 9.10 e 10.04
- Open Office 3.2
- Firefox 3.6.3
- Apache 2
- PHP 5.3
- gEdit
- Joomla 1.5.17 e Joomla 1.7 Beta1

Código Fonte

Para efetuar o download do código fonte acesse:

<http://ribafs.org/livros/ccomerciais>

Entre com

login – ccomerciais
senha – livroccjoomla

Inclusive também encontrará o PDF do livro.

Sobre a publicação deste livro

Aproveitarei este espaço para falar sobre a forma como este livro foi publicado. Publicar um livro em editora convencional é um processo bastante burocrático, que tem vantagens e desvantagens. Você assina um contrato de fidelidade sobre o assunto publicado que deve durar enquanto houver pelo menos um exemplar ou por cinco anos (no caso do contrato ser de 5 anos).

Sem contar as várias exigências além. Quando fui escrever o meu primeiro livro (Curso de Joomla), procurei várias editoras convencionais, mas como eu também queria compartilhar todo o conteúdo do livro em um site não encontrei nenhuma que aceitasse tal forma de escrever. Resumindo, foi quando encontrei algumas pequenas editoras que aceitam a publicação sem qualquer exigência nem interferência no seu processo de publicação. Elas nem tomam conhecimento se eu publico ou não em outro site o conteúdo. Além de que o processo de publicação é muito democrático e desburocratizado. Simplesmente você entra no site, preenche o cadastro e envia o PDF do seu livro e quando termina o livro está publicado.

Um alerta: como é muito simples o processo corremos o risco de publicar algo que não foi devidamente corrigido (como foi o caso do meu primeiro livro, que teve que ser corrigido em seguida). Agora vou passar o site das três editoras deste tipo que encontrei:

<http://www.clubedeautores.com.br>

<http://www.bubok.com>

<http://www.lulu.com/>

A primeira brasileira, a segunda portuguesa e a última em inglês (Canadá e Estados Unidos) mas todas publicam sem problema livros em português.