

# Curso de Joomla 1.5

## (Administração e Programação)

### ÍNDICE

1 – Introdução	3
2 – Projeto	8
3 – Ambiente	16
3.1 – Para Linux	16
3.2 – Para Windows	19
4 – Instalação	21
4.1 – Pré-Requisitos	21
4.2 – Local	22
4.3 – No Servidor	24
5 – Administração	40
5.1 – Recursos nativos	40
5.2 – Administração do Joomla instalado sem conteúdo	42
5.3 – Inserindo Código HTML e JavaScript em Artigos	56
5.4 – Recursos de Terceiros	57
5.5 – Recursos extras	63
5.5.1. – Dicas Extras sobre Administração	63
5.5.2. – Alterando a pasta Administrator	65
6 – Segurança em sites web	66
6.1 – Cuidados Básicos na Administração do Joomla	66
6.2 – Dicas de Segurança	68
7 – Programação	69
7.1 – Ferramentas	69
7.1.1 – Extensões para o Firefox	69
7.1.2 – Extensões para o Joomla	74
7.1.3 – Editores de Programação	77
7.1.4 – IDEs	78
7.1.5 – Editores Gráficos	81
7.2 – Framework	87
7.2.1 – Boa Prática	87
7.2.2. – Convenções para Nomes	88
7.2.3 – Estrutura da URI do Joomla	93
7.2.4 – Application Message Queue	95
7.2.5 – Redirecionamento	97

---

7.2.6 - Metadatas . . . . .	98
7.2.7 - Texto de Dicas - Tooltips . . . . .	99
7.2.8 - Constantes Pré-definidas . . . . .	100
7.2.9 - Arquivos, Diretórios e Permissões . . . . .	101
7.2.10 - Fluxo Request/Response . . . . .	105
7.2.11 – Segurança na Programação . . . . .	106
7.2.12 - Trabalhando com Bancos de Dados . . . . .	114
7.2.13 - Manipulando Campos comuns . . . . .	133
7.2.14 - JTable . . . . .	136
7.2.15 - Incrementando Contador de Hits . . . . .	141
7.2.16 - Manipulando Erros com DBO . . . . .	142
7.2.17 - Conectar a um Banco de Dados Externo . . . . .	143
7.2.18 - Trabalhando com Usuários no Joomla . . . . .	144
7.2.19 - Envio de e-mail . . . . .	150
7.2.20 - Lista de Controle de Acesso - ACL . . . . .	152
7.2.21 - Funções para trabalhar com Navegadores . . . . .	161
7.2.22 - Tratamento de Erros . . . . .	163
7.3 – Criando Extensões . . . . .	165
7.3.1 – Criando Módulos para o Joomla . . . . .	166
7.3.2 - Criando um Módulo para adicionar o Logo do Site . . . . .	176
7.3.3 – Criando Componentes para o Joomla . . . . .	177
7.3.4 - Desenvolvendo Componentes usando MVC . . . . .	183
7.3.5 – Criando Plugins para o Joomla . . . . .	187
7.3.6 – Criando Templates Simples para o Joomla . . . . .	200
7.3.7 - Parâmetros nos Templates . . . . .	218
7.3.8 - Sobre o Favicon . . . . .	221
7.3.9 – Criação de Languages/Idiomas para o Joomla . . . . .	224
8 - Recursos Extras . . . . .	225
8.1 - Overrides - Sobrescrevendo Saídas . . . . .	225
8.2 - Sufixos de Classes . . . . .	226
9 - Extensões de Exemplo . . . . .	231
9.1 - Código Fonte . . . . .	231
9.2 - Componente de Exemplo – com_hoteiscrud2 . . . . .	232
9.3 - Módulo de Exemplo – mod_hoteiscrud . . . . .	234
9.4 - Template de Exemplo – tmp_green . . . . .	236
10 – Referências . . . . .	248

## 1 – Introdução

### O que é o Joomla?

É um software do tipo CMS (Content Management System - Sistema Gerenciador de Conteúdo), cujo objetivo é tornar mais prático e econômico o gerenciamento de conteúdo em grandes e pequenos sites, em especial do tipo portal. Ele facilita tanto essa tarefa que praticamente qualquer pessoa, mesmo sem conhecimento algum de informática, pode gerenciar o conteúdo do site.

Quando terminamos a instalação do Joomla praticamente temos a estrutura de um site completo. No caso o Joomla nos oferece uma estrutura básica de um portal funcional. Daí resta entender como funciona e customizar o site.

### O que significa a palavra Joomla?

É a pronúncia fonética para a palavra Swahili "Jumla", que significa "todos juntos" ou "como um todo".

### História

O nome Joomla! Vem do Swahili (idioma do leste da África) significa "todos juntos" ou "como um todo".

O Joomla nasceu de um outro CMS, o Mambo, que também era um CMS bastante premiado.

Trata-se do resultado da separação entre a equipe de desenvolvedores do Mambo e a empresa Miro, detentora dos direitos sobre o Mambo. A separação teve lugar uma vez que a Miro transferiu o controle do Mambo para uma fundação - a Mambo Foundation - onde os desenvolvedores teriam apenas uma participação passiva e pouco representativa. Esses desenvolvedores, preocupados com a integridade do projeto e com o futuro dos usuários, não aceitaram a transferência e, em 2005, criaram o "Joomla", também open source, a partir do código-fonte do Mambo 4.5.2.

- O Mambo foi criado pela empresa Miro no ano 2000.
- Em 2001 o Mambo foi liberado sob a licença GPL2
- Em 2004 o Mambo recebe o prêmio de melhor projeto de software livre da Linux Format
- Em 2005 ganha outros prêmios
- Em setembro de 2005 a equipe do Mambo se divide e nasce o Joomla
- O Joomla continua a tradição de ganhar prêmios do Mambo

### Licença

O Joomla é entregue sob a licença GPL2. Com isso ele garante que o programador que usar o Joomla deve entregar juntamente o código fonte, garantindo assim a liberdade de uso e alteração do site criado. Nada impede que se venda o serviço de instalação,

customização e inclusive extensões criadas, mas o que não é permitido é vender e impedir o acesso ao código fonte. Isso representa liberdade num sentido mais amplo.

### **Premiação**

A editora Packt Publishing (<http://www.packtpub.com/award>) patrocina anualmente um prêmio para os melhores CMS open source.

Em 2009 o primeiro lugar ficou novamente com o Drupal e o segundo com o Joomla.

O Joomla tirou o segundo lugar em 2008. O primeiro ficou com o Drupal.

Em 2007 tirou o primeiro lugar.

Em 2006 também tirou o primeiro lugar.

### **Algumas Características do Joomla**

- Atualmente conta com uma grande comunidade ao redor do mundo
- Já existe uma grande quantidade de extensões disponíveis, em especial catalogadas no site extensions
- É um CMS com uma grande quantidade de recursos
- É um dos CMS mais amigáveis para o usuário final e para o programador
- Desenvolvido em PHP com MySQL, que estão entre os softwares mais populares da web
- Praticamente todos os servidores de hospedagem atualmente suportam as exigências para instalar o Joomla
- Editor HTML amigável para adição e alteração dos artigos
- Registro de usuários com níveis de acesso ao conteúdo
- Pesquisa em todo o conteúdo do site
- Formulários de contato
- Enquetes
- Estatísticas no site e na administração
- Envio de e-mail
- Envio de artigos por e-mail
- Formatos de impressão dos artigos em HTML e PDF
- RSS
- Organização de conteúdo: seções, categorias e artigos
- Notícias
- Links
- Artigos do tipo padrão (site) ou blog
- Grande quantidade de extensões de terceiros gratuitas e comerciais
- Grande quantidade de templates gratuitos e comerciais

O Joomla gerencia o conteúdo e também permite a integração de aplicativos complexos:

- e-commerce
- forums
- redes sociais
- etc

**Dados Atuais sobre o Joomla (12/01/2010):**

- Ocorrências retornadas no Google - 89.500.000
- Total de usuários do Forum (<http://forum.joomla.org>) - 344.842
- Total de posts no forum - 1.912.368
- Total de extensões catalogadas no <http://extensions.joomla.org> - 3.934 (todas sob a GPL)
- Existe uma grande quantidade de templates gratuitos na internet (veja estes <http://joomlsos.de>, <http://joomla24.com>)
- Atualmente suporta 57 idiomas diferentes. Podem ser encontradas no Joomla Code ou no Extensions:  
[http://joomlancode.org/gf/project/jtranslation/frs/?action=&br\\_pkgrlssort\\_by=package\\_name&br\\_pkgrlssort\\_order=asc](http://joomlancode.org/gf/project/jtranslation/frs/?action=&br_pkgrlssort_by=package_name&br_pkgrlssort_order=asc)  
[http://extensions.joomla.org/index.php?option=com\\_mtree&task=listcats&cat\\_id=1837&Itemid=2](http://extensions.joomla.org/index.php?option=com_mtree&task=listcats&cat_id=1837&Itemid=2)

**Onde podemos utilizar o Joomla**

O Joomla pode ser utilizado em uma grande quantidade de tipos de sites

- Escolas e universidades (UFC e USP)
- Corporações
- E-commerce
- Organizações comunitárias e não governamentais
- Organizações governamentais (MEC)
- Intranets e extranets corporativas
- Páginas pessoais e familiares
- Jornais e revistas
- etc

**É muito complicado?**

É verdade que existem outros bons CMS mais simples de aprender que o Joomla, como é o caso do Wordpress, mas isso não deve ser suficiente para que faça opção pelo Wordpress ou por outro CMS ao invés do Joomla. Antes deve gastar algum tempo para entender seu funcionamento, experimentar seus recursos para então ter realmente uma opinião mais abalizada e tomar sua decisão. Nem sempre o mais fácil ou mais simples é o melhor. Experimente e só então decida.

Este tutorial tem também com o objetivo mostrar que é fácil e prazeroso trabalhar com Joomla na criação de sites.

**Ele faz tudo sem programação?**

Melhor dizendo ele faz muito sem nenhuma necessidade de conhecer programação. Mas não se iluda, para explorar os melhores recursos e para criar um site realmente profissional quanto mais conhecer de design e de programação PHP, melhor.

Com toda a facilidade oferecida pelo Joomla não tenha receio de perder seus clientes, pois ainda assim o cliente precisará de alguém experiente e a prova é que o mercado para Joomla está crescendo.

**Valorize-se, aprenda a usar Joomla**

Atualmente existe uma boa procura por profissionais que trabalham com Joomla, além de que geralmente o cliente nem toma conhecimento de qual ferramenta estamos usando para criar o site dele. Se entregamos rápido, se o site fica bonito e ainda por cima fica fácil de manter para o cliente está de bom tamanho e o Joomla permite tudo isso.

**Veja aqui a demanda em todo o Brasil:**

<http://www.google.com/insights/search/#q=joomla%2Cwordpress%2Cdrupal%2C&geo=BR&date=today%2012-m&cmpt=q>

**Aqui uma boa relação de site famosos criados com Joomla:**

<http://ribafs.org/portal/joomla/78-dicas/111-sites-famosos-em-joomla>

Após ler algumas declarações de que o Joomla é trabalhoso de aprender então resolvi escrever de maneira bem simples sobre o quanto é fácil criar sites com Joomla, pois é sinceramente é isso que acho, que é muito fácil criar sites com o Joomla.

Farei o possível para mostrar isso e acredito que não será algo trabalhoso pois o Joomla (ainda Mambo) é o software que mais me impressionou em todos esses anos que mexo com programação, exatamente pelos recursos que oferece ao usuário e ao programador.

Estarei assim melhorando minha contribuição com o Joomla e fazendo justiça à equipe de desenvolvimento, que em algumas situações é injustiçada.

Primeiro mostrarei que os requisitos necessários para se instalar o Joomla atualmente são comuns tanto nos servidores de hospedagem quando nos pacotes instaladores do Apache, PHP e MySQL. Meu instalador favorito é o Xampp, que existe em versões para o Linux, para o Windows e outros sistemas operacionais.

O instalador do Joomla é tão simples que é todo realizado com o mouse, sem precisar digitar uma linha de código. Além disso ele já detecta um dos 57 idiomas suportados e automaticamente traduz o instalador, no nosso caso, para o Português do Brasil.

Caso seja sua primeira instalação do Joomla instale conteúdo de exemplo para que ao final da instalação já tenha um portal completo, contendo várias extensões, categorias, conteúdo, enquete, banner, contato, menus, formulário de login e muito mais para que tenha uma idéia de como ficará seu portal. É frustrante instalar o software e ao final não vê nada funcional, como alguns CMS. Pensando nisso a equipe do Joomla colocou a opção de já instalar conteúdo de exemplo.

Agradeceria sinceramente receber um e-mail (ribafs@gmail.com) com críticas, sugestões e correções do português ou da parte técnica.

## Recursos Utilizados

Hardware – Netbook ASUS H1000

Sistema Operacional – Linux Ubuntu 9.10

Apache2, PHP5.2 e MySQL 5.2 (instalados pelos pacotes)

Navegador - Firefox 3.5.7

Processados de Textos – BrOffice 3.1

Editores Gráficos – Gimp 2.6 e Kolourpaint 4.3.4

CMS - Joomla 1.5.15

Extensões para o Firefox - Firebug 1.5.0, Measureit 0.3.92 e ColorZilla 2.0.2

Extensões para o Joomla – j!Dump e Jumi

Instrumento de Busca – Google

## 2 – Projeto

Um dos maiores objetivos do projeto é evitar retrabalho, ou seja, trabalho desnecessário no futuro.

Outro é que o resultado final seja de fato o mais próximo do esperado.

Uma prudente recomendação é a de antes de arregaçar as mangas para a criação do site ou portal, antes sentarmos e realizar um planejamento do que iremos fazer. Com uma boa descrição, falando dos detalhes.

Devemos iniciar o projeto antes de iniciar qualquer etapa de construção do portal.

O projeto é uma etapa do desenvolvimento web que tem grande importância, pois nessa fase descrevemos como será o projeto para que essa descrição nos sirva de rumo. A descrição deve ser tão mais elaborada quanto mais importante for o site ou aplicativo.

Simplificando devemos ter uma noção exata do que iremos fazer, para evitar retrabalho futuro.

Aqui abordaremos apenas pequenos cuidados que devemos tomar, com a intenção de ajudar a formar o hábito de se elaborar sempre um projeto antes de iniciar um site ou aplicativo. Caso esteja com um projeto importante, então procure por mais detalhadas informações e faça algo a altura do seu projeto, caso contrário poderá se arrepender bastante depois.

Também os itens de menus que corresponderão às seções, categorias ou artigos. Citar os módulos a serem usados, os componentes, plugins e templates entre outros.

O projeto deve levar em conta a hospedagem onde irá ficar o portal, quais os recursos e flexibilidades existentes e se de fato suporta bem o Joomla na versão atual; o domínio ou subdomínio que irá usar; o preço final de elaboração do portal (se for o caso) e as condições de pagamento; assim como o prazo estimado de conclusão.

Muitos outros cuidados que cada projeto pode exigir e que devemos ficar atentos.

Ao final desse trabalho é muito útil criar uma relação resumida das atividades e elaborar um diagrama (o OpenOffice Draw é uma boa alternativa) com as seções, categorias, artigos e menus e outros detalhes para que isso sirva de bússola durante a criação do portal.

Também vale lembrar que devemos criar ou usar uma logomarca existente e lembrar de encaixar bem no template.

Para isso uma boa ou várias conversas com o cliente que nos contratou é vital. Coleta de informações, visitas a sites existentes, documentos escritos, etc.

Lembre: um bom projeto evita retrabalho.



## Conteúdo do Site

Ao preparar o conteúdo do site devemos nos preocupar com sua coerência e devemos selecionar criteriosamente as palavras dos textos dos links.

## Recomendações do Rasmus Lerdorf

Rasmus Lerdorf, nada menos que o criador do PHP. Não é somente por ser o criador do PHP que devemos fazer tudo que ele recomenda, mas analise suas recomendações com senso crítico e veja se são coerentes. Algo importante que li em certa entrevista com ele foi a declaração de que um dos objetivos na criação do PHP foi de que deveria ser simples, fácil de entender. Acho que isso foi um dos motivos para a grande popularidade atual do PHP.

Veja abaixo algumas recomendações do criador do PHP, Rasmus Lerdorf.

- Usar programação procedural e OO somente quando claramente necessário.
- Idéia errada: tudo de procedural é ruim e tudo de OO é bom.
- Evite o uso de frameworks, pois tornam muito complexa a criação de aplicativos como também engessam a manutenção.
- Somente se você sempre construir exatamente o mesmo estilo de aplicativo é que então pode criar um framework.
- Seu tempo é melhor gasto criando uma infraestrutura básica ideal para qualquer tamanho de projeto e reusar pequenos componentes.
- Separe bem as camadas, HTML é HTML, PHP é PHP.
- Evite estruturas monolíticas, sempre modularizando sua programação, criando funções sempre que for o caso ou reusando pedaços de código.
- Tente limitar em 5 ou menos os includes de uma requisição
- Não exagere com OO, use somente quando necessário
- Use uma camada de abstração de bancos de dados
- Tudo tem um custo
- Fique de olho nas suas expressões regulares
- Caso use MVC, use com cuidado e critério, não tem problema com esse padrão de projeto, o problema é com os frameworks.
- Você deve quebrar grandes problemas em módulos pequenos.
- Mantenha o design simples e limpo:
- HTML deve parecer com HTML (evite templates complexos)
- Mantenha o código PHP em views extremamente simples: chamadas de funções, simples laços e substituição de variáveis deve ser tudo de que você precisa
- Segurança:
- Validação da entrada de dados
- Sempre que possível evite camadas e outras complexidades para tornar o código mais fácil de auditar
- Desempenho:
- Evite `include_once` e `require_once`

- Permaneça usando o estilo procedural e use OO somente quando realmente caracterizado

### **Evitar em Sites ou Aplicativos, especialmente em Sites:**

- Mudanças no layout que façam o visitante se perder
- Carga inesperada (pesado)
- Lentidão exagerada
- Falha de hardware ou no servidor: falta de espaço, banda ou banco fora
- DRY - Don't Repeat Yourself - Evitar repetição.

Original em inglês: <http://toys.lerdorf.com/archives/38-The-no-framework-PHP-MVC-framework.html>

### **Cinco Bons Hábitos de Programação em PHP**

Além de possibilitar que você construa código de forma mais produtiva, esses hábitos podem ajudá-lo a construir código sustentável para o tempo de vida de um aplicativo. Qualquer código escrito provavelmente passará a maior parte de seu tempo de vida em manutenção; a manutenção do aplicativo é uma grande despesa. Estabelecer bons hábitos de codificação aprimorarão os fatores de design, como modularidade, e seu código será mais fácil de entender e, portanto, mais fácil e mais barato de manter.

1. Use boa nomenclatura.
2. Faça blocos menores.
3. Documente seu código.
4. Trate das condições de erro.
5. Nunca copie e cole (ao invés crie uma função).

Fonte: <http://www.ibm.com/developerworks/br/library/os-php-5goodhabits/>

### **Outras Recomendações para o Projeto**

- Seções
- Categorias
- Artigos
- Menus e itens de menus
- Extensões internas
- Extensões de terceiros
- Temas
- Verifique os direitos autorais de todo o conteúdo do site:
- Textos
- Imagens (manipule o tamanho e a qualidade)
- Áudios e vídeos
- Não esqueça de citar o autor e seu site

- Mostre sempre a licença do seu conteúdo
- Use link para a index no logo do site
- Simplifique sempre que puder
- Mantenha o site com navegação intuitiva (use o breacrumbs e categorias)
- Mantenha um form de contato além de endereço, telefones e e-mails de contato
- Mantenha uma página sobre Você ou a Empresa
- Redirecionar mensagens de erro para página com mensagem de esclarecimento/desculpas e form de contato
- Atenção às meta-tags para maior visibilidade do seu site, assim como cada página deve ter um título com H1 ou H2
- Não abra automaticamente músicas ou vídeos, deixe um link ou botão ao invés
- Use um tamanho de página que atenda à maioria (1024x768 atualmente) e não use redimensionamento
- As páginas mais importantes devem ter mais cuidado com a acessibilidade
- Cuidado, pois alguns gráficos poderão roubar a festa desviando a atenção de algo mais importante
- Faça backup regularmente

Aadaptado do site <http://cursos.ribafs.org> - Aplicativos em PHP - Projeto

## Verificações a serem feitas antes de se publicar um novo site

### 1 - Favicon

Uma boa personalização para o site é adicionar um favicon bem representativo.

Para sites pessoais podemos adicionar a foto do dono do site ou blog.

No site abaixo entramos com uma foto e ele retorna o favicon.ico:

<http://www.degraeve.com/favicon/> ou então desenhe o seu favicon.

No caso de sites com Joomla que já tenham um favicon no template, basta sobrescrever o existente.

Para sites que ainda não tenham, adicione entre as tags <head>:

```
<link rel="icon" type="image/x-icon" href="/favicon.ico" />
```

### 2 - Título e Metadados

O título do site é um elemento importante em termos de SEO (Search Engine Optimization).

```
<title>RibaFS Portal</title>
```

Como também tags de descrição do site:

```
<meta name="description" content="Este é um site pessoal dedicado ao desenvolvimento web em PHP. Os focos são o CMS Joomla e PostgreSQL. Sobre o Joomla encontram-se tutoriais, dicas, cursos, módulos e componentes. Sobre o PostgreSQL tutoriais, dicas e cursos." />
```

### 3 - Checagem do site em vários navegadores

Como nosso site geralmente será visualizado por diversos navegadores é importante que ele seja visualizado corretamente pelo menos nos mais populares.

Checagem do ACID2 - <http://www.webstandards.org/files/acid2/test.html>

Checagem do CSS3 - <http://www.css3.info/selectors-test/test.html>

Download dos 8 navegadores mais populares - <http://www.xenocode.com/browsers/>

IETester - <http://www.my-debugbar.com/wiki/IETester/HomePage>

Testar o site em vários navegadores de vários sistemas operacionais - <http://browsershots.org/>. Entre com o site e aguarde a geração dos screenshots, que ao final poderá fazer download.

### 4 - Leitura completa e mais atenta

Leia com bastante atenção e procurando erros de grafia e concordância.

### 5 - Links

Clique em cada link para testar se estão corretos.

Certifique-se de que sua logomarca tem um link para seu próprio site (convenção comun). Não sublinhe texto que não é link pois isso confunde.

Site que checa por links - <http://validator.w3.org/checklink>

### 6 - Checar funcionalidade

Caso tenha um form de contato, teste se está funcionando.

Peça colegas de grupos para testar seu site e não somente parentes e amigos.

Áreas importantes para testar são: contato, login, busca, etc.

### 7 - Desabilitando recursos

Caso seu site possa funcionar sem JavaScript desabilite o JavaScript e teste como fica o seu site.

### 8 - Validações

Verifique as validações se funcionam adequadamente. Valide o CSS e XHTML.

Veja este artigo: 10 razões por que seu código não valida e como corrigir isso:

<http://net.tutsplus.com/articles/web-roundups/10-reasons-why-your-code-wont-validate-and-how-to-fix-it/>

Validação online - <http://validator.w3.org/> e <http://jigsaw.w3.org/css-validator/>

### 9 - Link para RSS

Este recurso é muito importante para quem visita o site e para atrair visitantes. Através do RSS o visitante pode voltar ao site.

**10 - Mapa do Site**

Adicione ou verifique o mapa do site. Quando o visitante não encontrar um recurso do site o mapa o ajudará.

**11 - Design defensivo**

Crie bonitas e eficientes páginas para o erro 404, quando o usuário solicitar uma página e não encontrar.

Nessa página personalizada deve ter um pedido de desculpa, seu e-mail e a orientação de verificar se o link está correto.

**12 - Otimize**

Aproveite para melhorar a performance do seu site, eliminando recursos desnecessários e melhorando outros:

imagens, flash, javascript, etc. Veja o artigo:

<http://ribafs.org/portal/joomla/78-dicas/125-melhores-praticas-para-um-site-mais-rapido>

**13 - Backup**

Lembre que o backup é muito importante para garantir as informações do seu site. Faça regularmente backup dos arquivos e do banco de dados.

**14 - CSS para Impressão**

Caso seu site não tenha procure adicionar recurso para que o visitante possa ter uma página de impressão para cada página que queira imprimir.

**15 - Estatísticas**

Caso queira estatísticas sobre os visitantes do site - <http://www.google.com/analytics/>

CheckList:

Aqui encontrará um PDF para que possa imprimir e realizar um controle mais preciso:

[http://www.boxuk.com/upload/website\\_launch\\_checklist\\_v1.pdf](http://www.boxuk.com/upload/website_launch_checklist_v1.pdf)

Original em inglês: <http://www.smashingmagazine.com/2009/04/07/15-essential-checks-before-launching-your-website/>

## Quanto Cobrar por um Site em Joomla?

Essa é uma questão que incomoda muito a quem inicia na criação de sites de uma maneira geral e em especial sites em Joomla.

Não existe uma tabela onde se possa tomar valores básicos e ainda piora quando se consulta os colegas em grupos e não se recebe respostas diretas.

Fala-se em cálculo de horas trabalhadas mas isso também nem é simples nem parece ser algo que de fato leve a um preço justo para a criação do site, pois assim todos os profissionais cobrarão o mesmo valor, o que não é algo justo. Cada profissional tem seu preço em consequência da sua qualidade de trabalho.

Também devemos levar em conta o cliente na hora de cobrar. Meus primeiros sites foram feitos para amigos e de graça. Quando fui ganhando confiança é que cheguei a cobrar pelos sites.

Quando cobramos devemos olhar para a capacidade de pagamento do cliente e também da finalidade do site (minha ótica). Quando crio um site para uma instituição filantrópica costumo fazer um preço mais camarada. Quando crio um site pessoal é um preço, quando crio um site para uma empresa pequena é um preço, para uma empresa maior é outro preço.

Claro que ninguém precisa seguir minhas regras mas as estou enumerando para servir de base, de algum rumo. O mais importante é ficar tranquilo e cobrar algo que considere justo pelo seu trabalho e também pela qualidade do site que está entregando.

Algo que também incomoda para quem está iniciando é o fato de o Joomla facilitar muito a criação do site. Como se isso tornasse o nosso trabalho algo com menor valor. Temos que olhar por outro lado, isso nos deixa com mais tempo para nos concentrar em localizar melhores extensões, realizar uma melhor customização e se for o caso para criar extensões que venham a tornar o site mais eficiente.

### Créditos para o Joomla

Uma questão que considero importante levantar é se mantemos os créditos para a equipe do Joomla no rodapé do site ou não.

Em geral acho muito justo manter o crédito, pois a equipe de desenvolvimento realizou um excelente trabalho e merece esse crédito. Outro fator que reforça isso é ter o aval de uma equipe como essa. Devemos explicar para o cliente, em caso de dúvida, que representa uma grande segurança para o site saber que em todos os problemas ocorridos a equipe prontamente procurará a correção e lançará essa correção. É muito diferente de um site que seja realmente construído somente por nós e em caso de problema nós seremos os únicos para corrigir.

Caso o cliente faça questão de ocultar os créditos tá tudo bem, você pode ocultar. Mas acredito que ainda assim deve explicar ao mesmo as vantagens de manter o crédito e que é justo fazer.

**Argumentando com o Cliente**

Joomla hoje é utilizado por muita gente boa: USP, UFC, prefeitura de Fortaleza, Ministério da Educação, Estado do Ceará, etc. Relação detalhada em:

<http://ribafs.org/portal/joomla/78-dicas/111-sites-famosos-em-joomla>

Mostre os recursos que acompanham um site em Joomla: form de contato, enquete, RSS, comentários, forum, exibição de notícias em PDF, formato de impressão e envio por e-mail, e vários outros que considere importantes.

Fale da grande quantidade de extensões de terceiros, tanto gratuitas quanto pagas.

Fale da grande quantidade de templates gratuitos e pagos e da facilidade de se trocar um template.

Fale da consolidação do Joomla, da equipe, da grande comunidade, da quantidade de extensões o que dá mais segurança para quem tem um site em Joomla.

## 3 – Ambiente Exigido para se instalar o Joomla

3.3 – Para Linux

3.4 – Para Windows

### 3.3 – Ambiente Para Linux

#### Instalação através dos Pacotes no Linux Ubuntu

Esta é a forma mais rica e flexível de se instalar o ambiente necessário ao Joomla no Linux. Assim selecionamos o que queremos e o que não queremos assim como a atualização será automática, juntamente com a atualização do sistema.

Aqui iremos ver como instalar o ambiente para o Joomla através da instalação com os pacotes do Linux Ubuntu. Isso deve funcionar também sem alteração para o Debian e outras distribuições derivadas dele.

Sistema - Administração - Gerenciador de Pacotes Synaptic

Editar - Marcar Pacotes por Tarefas...

Marque Lamp server

Clique em Marcar.

Depois acima em Aplicar e Aplicar novamente e aguarde.

Isso deve atender as necessidades de muitos softwares, inclusive as do Joomla. Caso falte algo então deverá instalar pelo próprio synaptic ou via console com "sudo apt-get install nomedopacote".

#### Instalação de Pacotes Adicionais

Após instalar, instale mais alguns pacotes importantes, como os abaixo e outros que considerar necessários com:

```
sudo apt-get install phpmyadmin php-pear php5-xsl php5-gd php5-curl php5-xdebug
```

Algo que se deve efetuar após é a ativação do mod\_rewrite do Apache, como também para trabalho local mais confortável, alterar o diretório web para o home do usuário e outros.

#### Ativando o mod\_rewrite no Ubuntu:

```
sudo a2enmod rewrite
```

Após reinicie o apache como sugerido.

Para maior conforto altere o documentRoot para seu home:

Crie a pasta /home/ribafs/htdocs

Troque ribafs pelo seu usuário.

E aponte para o htdocs nas duas linhas abaixo:

---

```
DocumentRoot /var/www
```

```
<Directory />
```

```
Options FollowSymLinks
```

```
AllowOverride All
```

```
</Directory>
```

```
<Directory /var/www/>
```



---

Onde tem `/var/www` troque por `/home/ribafs/htdocs` nas duas ocorrências.  
`sudo gedit /etc/apache2/sites-available/default`

**Alterar o DocumentRoot** e outras configurações que achar por bem, como por exemplo: `AllowOverride All`, para permitir a criação de `.htaccess` (visando SEO no Joomla, frameworks como CakePHP e ZendFramework)

Após alterações neste arquivo deverá **reiniciar o apache** com:  
`sudo /etc/init.d/apache2 restart`

Veja mais detalhes neste pequeno tutorial:

<http://ribafs.org/portal/colaboracoes/1-ribamar-fs/82-instalar-apache-php-mysql-e-postgresql-no-ubuntu>

## Instalação do Xampp no Linux

Aqui a distribuição usada foi o Ubuntu, mas em qualquer distribuição o processo é semelhante, pois sempre o Xampp deve ser instalado no diretório `/opt`.

### Download

<http://xampp.sf.net>

### Descompactar

Abra a console do Linux e acesse o diretório onde fez o download e execute :

```
sudo tar xzpf xampp.xxx.tar.gz -C /opt
```

Após descompactar será criada a pasta `/opt/lampp`

### Configurações

Para ambientes de teste é incômodo ficar tendo que usar o `sudo` para alterar as permissões, visto que o usuário comum não tem permissão para escrever em `/opt/lampp/htdocs`. Então é confortável alterar o `documentRoot` para o diretório do usuário, portanto criaremos o diretório `/home/ribafs/htdocs`:

Obs.: lembre de trocar `ribafs` pelo nome do seu usuário.  
`mkdir /home/ribafs/htdocs`

Copiar o `htdocs`:  
`sudo cp -ra /opt/lampp/htdocs/* /home/ribafs/htdocs`

Alterar a pasta de trabalho do Apache no script `httpd.conf`:  
`sudo gedit /opt/lampp/etc/httpd.conf`

Então, estando com o script aberto no `gedit`, tecla `CTRL+F` e digite `htdocs` e `Enter`.  
Feche a janelinha de busca e troque as duas ocorrências realçadas de `htdocs`. Trocar:  
`/opt/lampp/htdocs`

Por:  
/home/ribafs/htdocs

Onde tem ribafs deve ser o nome do seu usuário.

**Alterar as permissões do lang.tmp:**

```
sudo chmod 707 /home/ribafs/htdocs/xampp/lang.tmp
```

**Altere o dono dos arquivos de htdocs:**

```
sudo chown -R ribafs:ribafs /home/ribafs/htdocs
```

**Restartando**

```
sudo /opt/lampp/lampp start
```

Para ter maior conforto, estando no Linux altere as permissões da pasta htdocs para 777:

**Acesse a console e execute:**

```
sudo chmod -R 777 /home/ribafs/htdocs
```

Obs.: Lembre que isso, sendo rigoroso, ainda pode comprometer a segurança do seu sistema de arquivos. Portanto, só deve ser usado com tranquilidade em um ambiente seguro, sob um bom firewall ou fora da Internet.

**Testando**

Abra o navegador em:

<http://localhost>

**Tornando permanente a inicialização do Xampp**

```
sudo gedit /etc/rc.local
```

Adicione esta linha ao final:

```
sudo /opt/lampp/lampp startapache
```

```
sudo /opt/lampp/lampp startmysql
```

### 3.4 – Ambiente Para Windows

#### Instalação do Xampplite no Windows

Existem diversos instaladores do Apache, PHP e MySQL para Windows, mas o Xampp é o meu preferido. Caso tenha problemas com ele procure outro como o Wampp ou outro.

Para usar o Joomla antes temos que ter todo um ambiente instalado. Este ambiente é formado basicamente por Apache, PHP e MySQL além de alguns módulos. Veja detalhes no artigo Pré-requisitos da seção Instalação.

O Xampplite traz todos os recursos necessários para a instalação local do Joomla em computador com o Windows, ou seja, instalar o Joomla em nosso computador pessoal. O Xampplite traz o Apache, o PHP, o MySQL e vários outros recursos.

#### Download

<http://xampp.sf.net> ou em:

<http://www.apachefriends.org/en/xampp-windows.html#646>

Veja que a versão EXE é um arquivo RAR auto-extrativo, portanto bem menor.

#### Instalação

A instalação é um processo semelhante a qualquer instalação do Windows, exceto ao final quando devemos ativar os serviços.

Deixe a instalação correr normalmente instalando em [c:\xampplite](#) ou escolha outro diretório e quando aparecer a tela da console com a pergunta:

1) Should I add shortcut to the startmenu/desktop? (y/n): y

Apenas tecle ENTER

2) Should I locate the XAMPP paths corretly?

Should I proceed? (y/x=exit setup): y

Apenas tecle ENTER

3) Should I make a portable XAMPP without drive letters?

Your choice? (y/n): n

Digite "y" e tecle ENTER

4) XAMPP is ready to use

Press <Return> to continue:

Tecla ENTER

5) I have set the timezone ...

...

Press <Return> to continue:

Tecla ENTER

6)

1. start XAMPP Control Panel
2. relocate XAMPP (current path: \xampplite)
3. disable HTTPS (SSL)
4. disable Server Side Includes (SSI)
5. enable Ipv4 only (current: Ipv4/6 (auto))

x Exit

Please choose (1-5/x):  
Digite "1" e tecla Enter

Será aberta a janela do Painel de Controle do Xampp  
Clique no check com Svc à esquerda de Apache e confirme  
Clique no check com Svc à esquerda de MySQL e confirme  
Clique no botão Start à direita de Apache  
Clique no botão Start à direita de MySQL

Pode fechar a janela do Painel de Controle do Xampp  
Na janela da console digite "x" e tecla ENTER.

Com isso seu computador está pronto para instalar o Joomla.

Veja agora o roteiro para instalar o Joomla sem conteúdo.

Para testar a instalação do Xampplite execute no navegador:  
<http://localhost>

## 4 – Instalação

- 4.1 – Pré-Requisitos
- 4.2 - Local
- 4.3 – No Servidor

### 4.1 – Pré-Requisitos

#### **Pré-requisitos para o Joomla 1.6** (próxima versão em desenvolvimento)

PHP 5.2+

MySQL 5.0.4+

#### **Pré-requisitos para o Joomla 1.5**

##### **Mínimo**

PHP 4.3.10 (mínimo),

MySQL 3.23.x ou superior (mínimo),

Apache 1.3 ou superior (mínimo),

MS IIS 6 (mínimo),

##### **Recomendado**

5.2.+ (recomendado)

4.1.x + (recomendado)

2.x + (recomendado)

7 (recomendado)

### **Módulos adicionais**

mod\_mysql

mod\_xml

mod\_zlib

Precisa se certificar de ter as funcionalidades MySQL, XML e Zlib habilitadas na instalação do PHP. Pode ser constatado no php.ini.

Não use as versões 4.3.9, 4.4.2 ou 5.0.4 do PHP, pois eles têm bugs conhecidos. Os melhores e mais consistentes resultados são obtidos com versões da série 5.x.

Para usar URLs amigáveis precisa ter a extensão mod\_rewrite instalada.

O Joomla foi testado em muitas variantes dos três principais sistemas operacionais: Windows, Linux e Apple Mac OSX. Também deve funcionar em outras plataformas mas ainda é pouco documentado.

Joomla ainda não é compatível com MySQL 6.x.

Testado nos principais navegadores: IE6 +, Firefox 1.5 +, Safari e Opera mas deve também funcionar em outros.

Existem problemas conhecidos com algumas configurações do CSS no IE6.

Joomla é otimizado para o Apache mas também roda no Microsoft IIS.

## 4.2 – Instalação Local do Joomla

### Instalação do Joomla 1.5

A versão que usaremos é a 1.5.15

### Instalação Local no Windows e no Linux

A instalação no Linux é semelhante à instalação no Windows, o que mudará basicamente serão as permissões dos diretórios e arquivos.

### Download

Faça o download no formato zip daqui:

<http://www.joomla.org/download.html>

### Instalação

1. Após o download crie o diretório **portal**, dentro do htdocs para abrigar o joomla, para que fique **htdocs/portal**.
2. Descompacte o Joomla que baixou em htdocs/portal
3. Chame no navegador <http://localhost/portal>
4. Primeira tela: Selecione o idioma. Nesta tela o nosso idioma geralmente já vem selecionado corretamente. Caso contrário selecione o idioma e clique em Próximo.
5. Tela de checagem de pré-requisitos: Verificação de pré-instalação.

Observe que existem dois grupos de informações, um acima e outro abaixo.

O primeiro grupo é de requisitos obrigatórios, que sem eles o Joomla não funcionará corretamente. O grupo inferior é de configurações recomendadas. Sua ausência poderá fazer falta, acarretando comportamento inesperado.

6) **Tela da Licença.** Fala da licença do Joomla, que no caso é a GNU/GPL em sua versão 2 e traz todo o texto da licença. Apenas clique em Próximo.

7) **Tela do Banco de dados** - Esta é talvez a tela mais importante

Nome do servidor – localhost

Nome do usuário – root

Senha – não digite nada aqui (caso usando o Xampp)

Banco – portal

8) Remover o diretório **Installation**

Agora remova o diretório **installation**

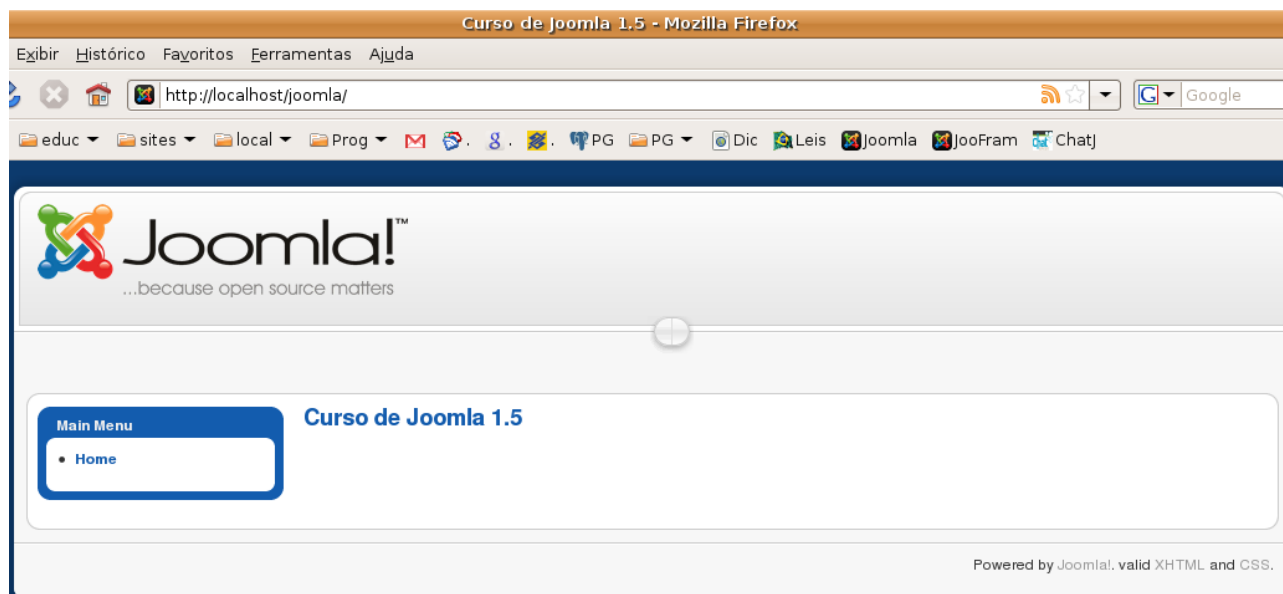
9) **Nome do site e dados do administrador**

Entre com o nome do site, por exemplo: **Portal do Ribamar**. Lembre que poderá alterar depois em:

Site – Configuração Global – Site – Nome do Site

Não clique no botão para adicionar conteúdo de exemplo, para que a instalação seja feita sem o conteúdo de exemplo.

Após a instalação clique em Site para visualizar o site instalado, como no exemplo abaixo:



Esta é uma instalação limpa, ou seja, quase "do zero", para que assim possamos conhecer melhor o funcionamento do Joomla e ao criar um portal termos menos trabalho, pois não precisaremos remover nada.

Vamos a ele. Instalamos o Joomla, agora vamos configurá-lo e então adicionar extensões e conteúdo.

Consulte agora o roteiro para administrar o Joomla.

## 4.3 – Instalação do Joomla no Servidor de Hospedagem

### Instalação do Joomla em um servidor de hospedagem com cPanel)

Download - <http://www.joomla.org>

#### Joomla! Announcements



Sunday, 04 October 2009  
Vote for Joomla!  
Joomla! is a finalist in the "Hall of Fame" and "Best Open Source PHP CMS"...

[click here now to](#)

#### Community News

Works as designed, but not necessarily as expected...  
When I was first approached to write an article for the Joomla! magazine, my first thoughts were that



Ao clicar no link DOWNLOAD DO JOOMLA então vamos para a página com:

### Download Joomla! 1.5.x

LANGUAGE	VERSION	DOWNLOAD
English (UK)	1.5.14 Full Package	<a href="#">ZIP</a>
English (UK)	1.5.13 to 1.5.14 Upgrade Package	<a href="#">ZIP</a>

[Download other Joomla 1.5.x packages »](#)

Aqui, devemos baixar geralmente a última versão, no caso a 1.5.14 Full Package, pacote ZIP.

O pacote abaixo é para quem tem a versão anterior (1.5.13) e deseja atualizar para a 1.5.14.

Caso queira atualizar de uma versão anterior à 1.5.13, então clique no link abaixo:



"Download other Joomla 1.5.x packages >>".

Clique neste link acima para darmos uma olhada.

### Versão Atual em três Formatos

Nessa página temos ao final três pacotes da versão atual em zip, tar.gz e bz2 (este último é o mais compactado).

Release Name	Release Date	Filename	File Size
Joomla1.5.14	2009-07-30 02:00:00-05	Joomla_1.5.14-Stable-Full_Package.zip	6 MB
		Joomla_1.5.14-Stable-Full_Package.tar.gz	5 MB
		Joomla_1.5.14-Stable-Full_Package.tar.bz2	4 MB

### Atualizações para a versão atual

Acima temos pequenos pacotes (patches) para atualizar desde o Joomla 1.5.0 até o Joomla 1.5.13 para a versão 1.5.14 (atual).

Release Name	Release Date	Filename	File Size
Joomla1.5.14updates	2009-07-30 02:00:00-05	Joomla_1.5.9_to_1.5.14-Stable-Patch_Package.tar.gz	1 MB
		Joomla_1.5.8_to_1.5.14-Stable-Patch_Package.zip	2 MB
		Joomla_1.5.10_to_1.5.14-Stable-Patch_Package.tar.gz	1 MB
		Joomla_1.5.10_to_1.5.14-Stable-Patch_Package.zip	1 MB
		Joomla_1.5.13_to_1.5.14-Stable-Patch_Package.zip	39 KB
		Joomla_1.5.8_to_1.5.14-Stable-Patch_Package.tar.gz	1 MB
		Joomla_1.5.7_to_1.5.14-Stable-Patch_Package.tar.gz	1 MB
		Joomla_1.5.7_to_1.5.14-Stable-Patch_Package.tar.bz2	1 MB

### Atualizando para a versão atual

Apenas identifique a sua versão existente (na seção administrador) e pegue o pacote respectivo.

Exemplo: sua versão atual é a 1.5.8. Então você deve baixar o pacote:

[Joomla\\_1.5.8\\_to\\_1.5.14-Stable-Patch\\_Package.zip](#)

Faça o download

Envie para o servidor direto para o raiz do seu portal Joomla

Então use os recursos do cPanel (gerenciador de arquivos) para descompactar o arquivo.

Prontinho. Seu portal está atualizado, geralmente é só esta mágica que temos que fazer. É que a equipe faz um trabalho bastante organizado e profissional, nos oferecendo apenas os arquivos que foram alterados.

**Recomendação:** Caso seu portal tenha conteúdo importante, então recomenda-se algumas precauções antes de atualizar:

- Baça backup do seu banco de dados
- Faça um backup completo dos arquivos
- Pare o site (use a opção do Administrator para colocá-lo fora do ar)
- Só então realize a atualização
- Faça alguns testes para saber se tudo correu bem
- Em caso afirmativo volte o site
- Guarde seus backups, pois mesmo não tendo percebido algum problema eles podem ocorrer posteriormente e então você restaura. Mais vale prevenir que remediar.

## Instalação

Supondo que baixamos o Joomla atual full (primeiro link).

Então devemos enviar para o servidor.

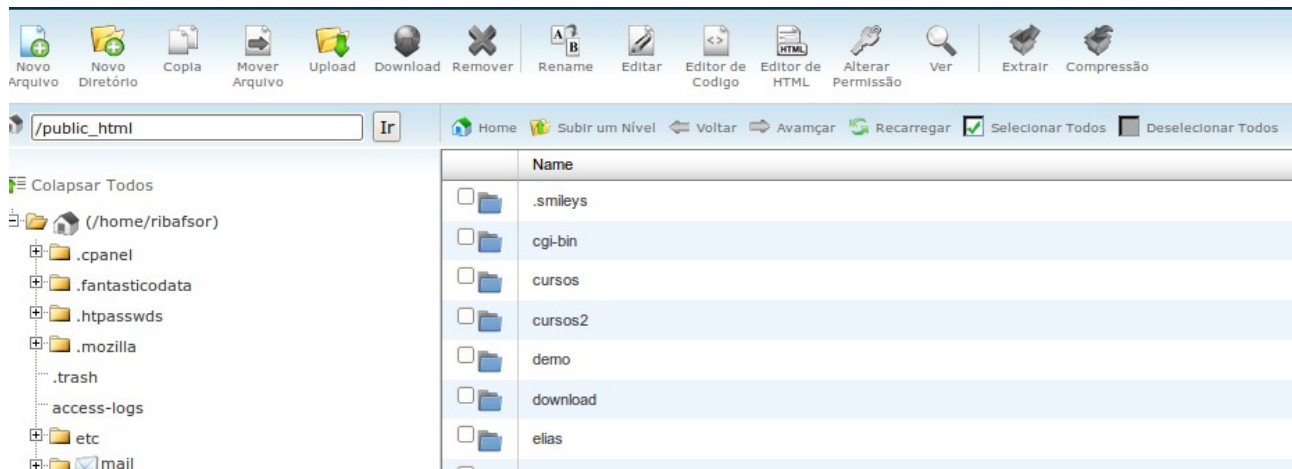
Como certa vez li que FTPs são sistemas inseguros eu sempre prefiro usar o gerenciador de arquivos do cPanel para essas operações.

Envio para o Servidor

Acesse sua área no servidor - <http://cpanel.dominio.org>

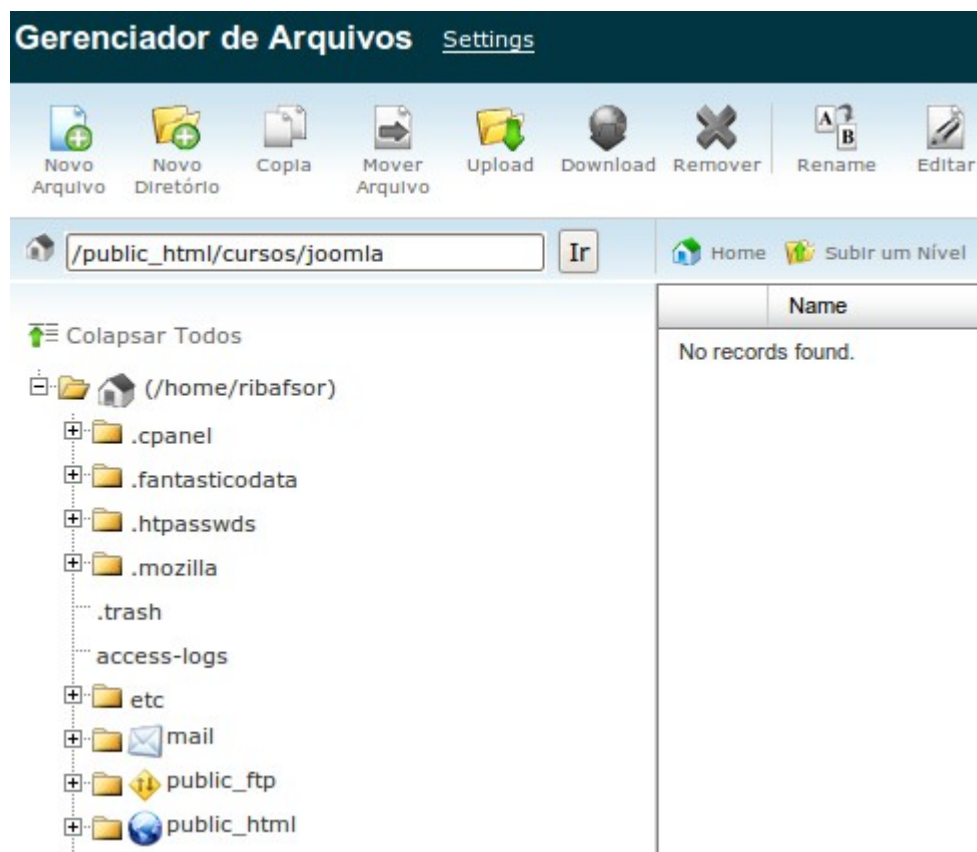


## Abra o Gerenciador de Arquivos



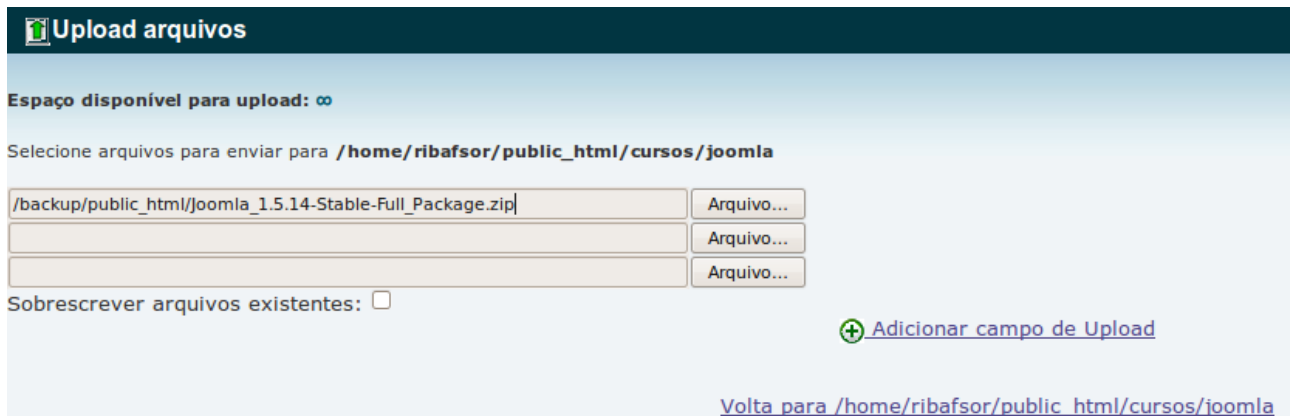
Crie um diretório (com nome de sua preferência) que abrigará os arquivos do Joomla. Aqui eu criei um diretório cursos e dentro um outro joomla, que abrigará o portal.

## Envio dos arquivos por Upload pelo Gerenciador de Arquivos



Agora, com o diretório cursos/joomla aberto no painel direito, clicamos no botão Upload.

## Então veremos a janela para envio dos arquivos



The screenshot shows the Joomla! 'Upload arquivos' interface. At the top, it says 'Espaço disponível para upload: ∞'. Below that, it indicates the target directory: '/home/ribafsor/public\_html/cursos/joomla'. There are three input fields for file paths, each with an 'Arquivo...' button to its right. The first field contains the path '/backup/public\_html/joomla\_1.5.14-Stable-Full\_Package.zip'. Below the input fields, there is a checkbox labeled 'Sobrescrever arquivos existentes:' which is currently unchecked. To the right of the checkbox is a link '+ Adicionar campo de Upload'. At the bottom right, there is a link 'Volta para /home/ribafsor/public\_html/cursos/joomla'.

Podemos enviar mais de 3 arquivos por vêz assim como sobrescrever existentes. Quando finalizar o envio dos arquivos clique no link Volta para ...

No meu caso, como faço várias instalações do Joomla, então apenas Extraio para o novo diretório.

Após o envio vamos extrair o arquivo:

Selecione o arquivo e clique no botão Extrair (acima e à direita).

### Criação do Banco


Após a extração dos arquivos devemos criar o banco de dados que será usado no portal. O Joomla trabalha (até o momento) apenas com o SGBD MySQL.

Volte à Home do cPanel e clique no botão Bases de Dados MySQL, do grupo Bases de Dados.


Chamarei meu banco de joomlcurso

## Bases de Dados MySQL

*Bases de Dados MySQL permite você armazenar informações de uma maneira fácil só são difíceis de ler. Bases de Dados são solicitados por inúmeras aplicações, com usar o base de dados, você precisará criá-lo. Somente Usuários MySQL (diferente que tem permissão de acesso ao base de dados podem ler ou inserir dados no basi*

 Guia em Video

### Criar Nova Base de Dados

Nova Base de Dados:  

### Modificar Bases de Dados

Entre com o nome e clique em Criar Base de Dados.

Veja como ficará:



Clique então em Voltar

O nome ficou  
**ribafsor\_joomlcurso**

A regra é:  
Todo nome de banco ficará assim:

dominio + \_ + nomedado

### Criação de Usuário para o Banco de Dados

Role a tela para criar um usuário.

Entre com usuário, veja que o máximo é de 7 caracteres.  
Entre com uma senha. Veja que para ajudar ele testa se a senha tem cuidado com a segurança.  
Senhas fortes são consideradas as que combinam letras com algarismos e símbolos.

**Usuários de Mysql****Adicionar Novo Usuário**

Usuário:   \*Máximo sete caracteres

Senha:

Força da Senha:

Senha (Repetir):

Então clique em Criar Usuário.

Após o clique ele mostrará o usuário e a senha em texto claro.

Aqui também vale a regra anterior, o nome do usuário será precedido do domínio com \_.

**Associar Usuário ao Banco**

Role novamente a tela

Selecione o usuário e o banco

**Adicionar Usuário ao Base de Dados**


Usuário:  ▾

Base de Dados:  ▾

Então clique em Adicionar.

**Privilégios**

Dar os devidos privilégios ao usuário no Banco criado

 **Manutenção de Conta MySQL**
**Gerenciar Privilégios de Usuários**Usuário: **ribafsor\_joomlac**Base de Dados: **ribafsor\_joomlcurso**

<input checked="" type="checkbox"/> <b>TODOS PERMISSOES</b>	
<input checked="" type="checkbox"/> SELECIONAR	<input checked="" type="checkbox"/> CRIAR
<input checked="" type="checkbox"/> INSERIR	<input checked="" type="checkbox"/> ALTERAR
<input checked="" type="checkbox"/> ATUALIZAR	<input checked="" type="checkbox"/> DROP
<input checked="" type="checkbox"/> REMOVER	<input checked="" type="checkbox"/> TRANCAR TABELAS (LOCK)
<input checked="" type="checkbox"/> INDEX	<input checked="" type="checkbox"/> REFERENCIAS
<input checked="" type="checkbox"/> CRIAR TABELAS TEMPORÁRIAS	<input checked="" type="checkbox"/> CRIAR ROUTINA

[\[ Voltar \]](#)

Este momento é muito importante para quem está preocupado com segurança. Não efetuei testes mas alguns dos itens acima não são obrigatórios para todos os sistemas web.

Um exemplo radical de preocupação com segurança:

Se você tem um site onde os visitantes apenas fazem consultas ao banco de dados, então aqui bastaria selecionar apenas o primeiro item: SELECIONAR.

Podemos selecionar um a um ou clicar no checkbox com falta de concordância TODOS PERMISSÕES e então em Fazer Alterações.

Então clique em Voltar

Role a tela e veja que agora nosso banco está associado ao usuário:

ribafsor_joomlcurso	0.01 MB	ribafsor_joomlac 	<a href="#">Remover Base de Dados</a>
---------------------	---------	--	---------------------------------------

Se desejar remover o banco basta clicar no link à direita.

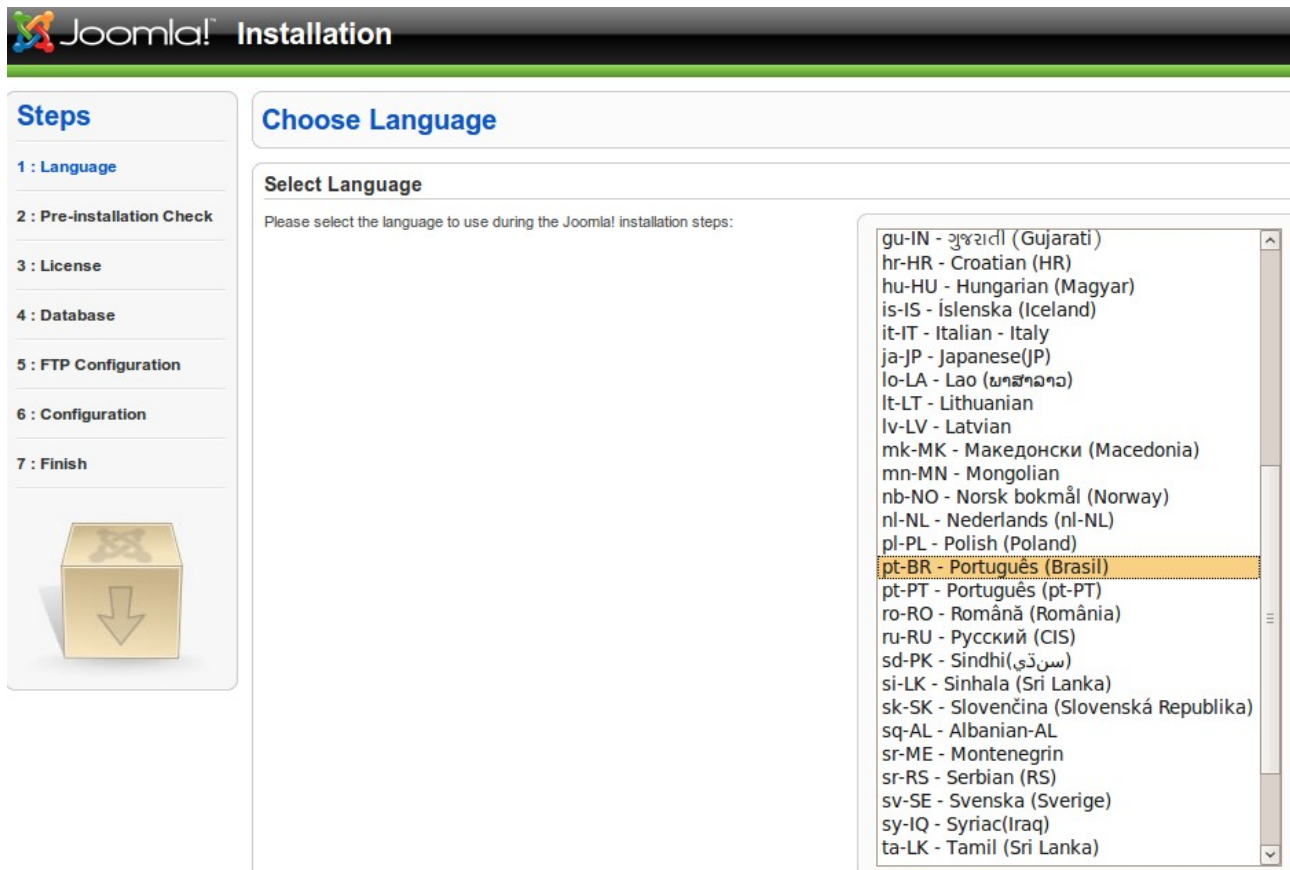
Agora estamos prontos para instalar o Joomla.

Chame no navegador a URL de onde instalou, no meu caso:

<http://ribafs.org/cursos/joomla>

## Seleção do Idioma

Após chamar no navegador



The screenshot shows the Joomla! Installation interface. On the left, a sidebar titled "Steps" lists seven steps: 1: Language, 2: Pre-Installation Check, 3: License, 4: Database, 5: FTP Configuration, 6: Configuration, and 7: Finish. Below the list is a yellow box with a downward arrow. The main content area is titled "Choose Language" and contains a "Select Language" section. It instructs the user to "Please select the language to use during the Joomla! installation steps:". A scrollable list of languages is displayed, with "pt-BR - Português (Brasil)" highlighted in orange. The list includes: gu-IN - ગુજરાતી (Gujarati), hr-HR - Croatian (HR), hu-HU - Hungarian (Magyar), is-IS - Íslenska (Iceland), it-IT - Italian - Italy, ja-JP - Japanese(JP), lo-LA - Lao (ລາວ), lt-LT - Lithuanian, lv-LV - Latvian, mk-MK - Македонски (Macedonia), mn-MN - Mongolian, nb-NO - Norsk bokmål (Norway), nl-NL - Nederlands (nl-NL), pl-PL - Polish (Poland), pt-BR - Português (Brasil), pt-PT - Português (pt-PT), ro-RO - Română (România), ru-RU - Русский (CIS), sd-PK - Sindhi(سنڌي), si-LK - Sinhala (Sri Lanka), sk-SK - Slovenčina (Slovenská Republika), sq-AL - Albanian-AL, sr-ME - Montenegrin, sr-RS - Serbian (RS), sv-SE - Svenska (Sverige), sy-IQ - Syriac(Iraq), and ta-LK - Tamil (Sri Lanka).

Caso o idioma não seja corretamente detectado selecione na lista e clique em Next.



## Verificações Antes de Instalar

**Passos**

- 1 : idioma
- 2 : verificação de pré-instalação
- 3 : licença
- 4 : banco de dados
- 5 : configuração do ftp
- 6 : configuração
- 7 : fim

**verificação de pré-instalação** Verificar Novamente Anterior Próximo

Verificação de pré-instalação para Joomla! 1.5.14 Stable [ Wojmamni Ama Naiki ] 30-July-2009 23:00 GMT:

Se algum destes itens não for suportado (destacado como **Não**), seu sistema não é compatível com os requisitos mínimos necessários. Por favor, providencie a correção destes erros. A persistência da falha pode fazer com que a instalação do Joomla! não funcione corretamente.

Versão PHP >= 4.3.10	Sim
- suporte à compressão Zlib	Sim
- Suporte XML	Sim
- Suporte MySQL	Sim
MB language é padrão	Sim
PHP mbstring function overload off	Sim
configuration.php editável	Sim

**Configurações Recomendadas:**

Estas configurações são recomendadas para o PHP, a fim de assegurar total compatibilidade com o Joomla!  
Contudo, o Joomla! ainda poderá funcionar, caso suas configurações, não estejam totalmente de acordo com o recomendado.

Diretiva	Recomendado	Atual
Safe Mode:	Off	Off
Exibir Erros:	Off	On
Envio de Arquivos:	On	On
Magic Quotes Runtime:	Off	Off
Register Globals:	Off	On
Output Buffering:	Off	Off
Session Auto Start:	Off	Off

Joomla! é um Software Livre liberado sob a licença GNU/GPL v2.0.

Aqui o Joomla verifica o sistema do servidor e nos dá alguns retornos sobre o mesmo. O que está OK aparece em verde. Se não atendeu aparece em vermelho.

Temos dois itens em vermelho:

Exibir erros - este não é vital e podemos deixar como está.

Register globals - Este é importante para a segurança e dependendo do servidor, podemos criar um arquivo php.ini, contendo apenas uma linha com:

```
register_globals = Off
```

e copiar para o raiz do Joomla que resolve.

Mas podemos continuar a instalação sem grandes problemas e após finalizar criar o arquivo acima, ou podemos dar uma pausa, criar o arquivo, teclar F5 para reiniciar e então veremos o item em verde.

Agora clique em Próximo.

### Licença

A próxima tela mostra a licença

Apenas clique em próximo ou role a tela para ver o conteúdo completo da licença.

## Configuração do Banco de Dados

### configuração do banco de dados

#### configurações da conexão:

Configurar o Joomla! para executar em seu servidor envolve 4 passos simples...

Neste Passo:

#### Configurações Básicas

Selecione o tipo de banco de dados na caixa de seleção. Este servidor normalmente é **mysql**.

Informe o nome do servidor de banco de dados onde Joomla! será instalado. **Atenção:** Este pode não ser necessariamente o mesmo servidor do servidor Web, então verifique com sua hospedagem para não deixar dúvidas.

Informe o nome de usuário, senha e nome do banco de dados do MySQL, os quais deseja que o Joomla! utilize. Já devem existir para o banco de dados que você irá utilizar.

#### Configurações Avançadas

Selecione como deseja lidar com tabelas existentes de instalações anteriores.

Informe o prefixo a ser usado pelas tabelas desta instalação do Joomla!.

#### Configurações Básicas

Tipo de Banco de Dados

mysql

Nome do Servidor

localhost

Nome de Usuário

ribafsor\_joomlac

Senha

●●●●●●●●

Nome do Banco de Dados

ribafsor\_joomlcurso

► Configurações Avançadas

Ao clicar em Configurações Avançadas teremos

▼ Configurações Avançadas

Remover Tabelas Existentes

Backup de Tabelas Antigas *Qualquer cópia de segurança das tabelas de uma instalação anterior do Joomla serão sobrescritas*

Prefixo da Tabela  *Não use 'bak\_' pois este prefixo é usado para a cópia de segurança de tabelas*

Para o caso de já existir uma tabela com mesmo nome e precisar sobrescrever e também se precisar mudar o prefixo padrão das tabelas.

Após preencher clique em Próximo.

## Configuração do FTP

Podemos deixar sem alterações.  
Apenas clique em Próximo.

## Configuração Principal

### configuração principal

Anterior Próximo

#### Nome do site:

Informe o nome do seu site Joomla!

Nome do site

#### Confirme o e-mail e a senha do administrador.

Informe seu endereço de e-mail, este será o endereço de e-mail do Super Administrador do site.  
Informe e confirme a senha nos campos a seguir. Esta será a senha que você precisará ao final do processo de instalação para acessar o Painel de Controle da Administração, através do nome de usuário **admin**.  
Se você estiver migrando o site pode ignorar esta seção porque suas configurações atuais serão automaticamente migradas.

Seu e-mail   
Senha do admin   
Confirme a senha do admin

#### Carregar exemplo de conteúdo, restaurar ou migrar backup de conteúdo

**IMPORTANTE:** É recomendado aos iniciantes que instalem os exemplos de conteúdo. Para isso, você precisa selecionar a opção adequada e clicar no botão antes de passar para a próxima etapa.

Antes de sair da instalação, você pode preencher o banco de dados. Há três modos para fazer isso:

**1. Exemplo de conteúdo** pode ser inserido. Para tal, selecione a primeira opção e logo em seguida clique no botão "Instalar exemplo de conteúdo".

**2.a Script SQL compatível com Joomla! 1.5** pode ser enviado do computador local e executado no site. Esta opção pode ser útil para instalar exemplo de conteúdos em seu idioma ou restaurar uma cópia de segurança do Joomla! 1.5. O script deverá ter os prefixos das tabelas corretos, deverá estar salvo com codificação UTF-8 e deverá obedecer ao esquema de banco de dados do Joomla! 1.5.x.

**2.b Migração de conteúdo de versões anteriores do Joomla!** Selecionando a opção "Carregar Script de Migração" ativa o suporte à migração do banco de dados do Joomla! 1.0.x para o novo Joomla! 1.5. As conversões necessárias são feitas automaticamente. O arquivo de script poderá ser criado automaticamente a partir do site antigo através do componente `com_migrator`. Para maiores informações sobre o componente e o procedimento necessário por favor, acesse o [Guia de Migração Joomla! 1.5](#).

O envio de arquivo suporta arquivos SQL não comprimidos, comprimidos em formato zip ou gz. Arquivos compactados devem conter apenas um único de script SQL. Se o arquivo for muito grande ou estiver causando erros de limite de memória ou tempo, envie o arquivo não comprimido para a pasta `/installation/sql/migration/`. Altere o nome para `migrate.sql` e então clique na opção "Eu já enviei o arquivo de migração para o servidor".

**3. Instalação Limpa** Se você deseja um instalação limpa, informe o nome do site, e-mail e senha do admin. Então, vá para a etapa final clicando em **Próximo**.

**Instalar exemplo de conteúdo** *Instalação deste é recomenda a iniciantes. Isto irá instalar o exemplo de conteúdo que está incluso no pacote de instalação do Joomla!*

**Carregar script de migração** *O script de migração precisa ser gerado no site antigo, através do componente `com_migrator`. Digite o prefixo das tabelas do site antigo e a codificação utilizado no mesmo (veja a configuração de `_ISO` no arquivo de idiomas ou pelo navegador em `info/encoding/source`). Os scripts SQL migração do Joomla! 1.5 precisam ser compatíveis com Joomla! 1.5 e precisam ter o prefixo para as tabelas apropriadas.*

**Tamanho Máximo para Envio** O tamanho máximo arquivo que pode ser enviado para seu servidor é 50.00MB.

Exemplo de conteúdo foi instalado com sucesso

Prefixo das tabelas antigas

Encoding do site antigo

Script de Migração

Eu já enviei o script de migração para o servidor (ex. via FTP/SCP)

Este script é um script de migração para Joomla! 1.0.

Agora entramos com o nome/título do site, e-mail e senha do administrador. Ele criará um usuário administrador com login "admin".

### Conteúdo de Exemplo

Logo abaixo temos a opção de instalar conteúdo de exemplo. Bastante recomendada para quem está instalando pela primeira vez ou ainda não está acostumado com a estrutura interna de configurações do Joomla. Clique no botão.

### Script de Migração

Caso esteja migrando de Joomla 1.0 clique em Carregar script de migração e indique o script que você já deve ter em algum diretório local.

Observe as demais informações que deve entrar abaixo.

Clique em Próximo.

## Final

fim

Site  Admin 

**Parabéns! O Joomla! está instalado.**

Clique no botão "Site" para acessar seu site Joomla! ou em "Admin" para ser levado à entrada do Painel de Administração.

Na página de ajuda do Joomla! para seu idioma, você irá encontrar uma ajuda passo-a-passo fácil, para você instalar seu próprio pacote de idiomas no Joomla! Lá você encontrará uma lista de pacotes de idioma disponíveis.

Clique no botão à direita para abrir o site de ajuda em uma nova janela.

**LEMBRE-SE DE REMOVER  
O DIRETÓRIO INSTALLATION  
COMPLETAMENTE. Você não poderá passar deste ponto, até que o diretório  
installation seja removido. Esta é uma medida de segurança do Joomla!.**

**Detalhes da Sessão do Administrador**

Nome de Usuário: admin

[Joomla! em seu idioma?](#)

Visite o site de ajuda do Joomla! para maiores informações e downloads.

Aqui somos obrigados a excluir o diretório "installation" do Joomla antes de continuar, ou melhor, antes de finalizar.

Acesse o Gerenciador de Arquivos e remova o diretório installation.

Então pode ir ao site ou ao admin.

Vejamos o site como está:

The one thing about a Web site, it always changes! Joomla! makes it easy to add Articles, content, images, videos, and more. Site administrators can edit and manage content 'in-context' by clicking the 'Edit' link. Webmasters can also edit content through a graphical Control Panel that gives you complete control over your site.

Home

**Main Menu**

- Home
- Joomla! Overview
- Joomla! License
- More about Joomla!
- FAQ
- The News
- Web Links
- News Feeds

**Latest News**

- Joomla! License Guidelines
- Content Layouts
- The Joomla! Community
- Welcome to Joomla!
- Newsflash 4

**Popular**

- Joomla! Overview
- Extensions
- Joomla! License Guidelines
- Welcome to Joomla!
- What's New in 1.5?

**Welcome to the Frontpage**

**Joomla! Community Portal**

Written by Administrator  
Saturday, 07 July 2007 09:54

The Joomla! Community Portal is now online. There, you will find a constant source of information about the activities of contributors powering the Joomla! Project. Learn about [Joomla! Events](#) worldwide, and see if there is a [Joomla! User Group](#) nearby.

The Joomla! Community Magazine promises an interesting overview of feature articles, community accomplishments, learning topics, and project updates each month. Also, check out [JoomlaConnect™](#). This aggregated RSS feed brings together Joomla! news from all over the world in your language. Get the latest and greatest by clicking [here](#).

Last Updated on Saturday, 07 July 2007 09:54

**We are Volunteers**

Written by Administrator  
Saturday, 07 July 2007 09:54

The Joomla Core Team and Working Group members are volunteer developers, designers, administrators and managers who have worked together to take Joomla! to new heights in its relatively short life. Joomla! has some wonderfully talented people taking Open Source concepts to the forefront of industry standards. Joomla! 1.5 is a major leap forward and represents the most exciting Joomla! release in the history of the project.

**Joomla! Security Strike Team**

Written by Administrator  
Saturday, 07 July 2007 09:54

The Joomla! Project has assembled a top-notch team of experts to form the new Joomla! Security Strike Team. This new team will solely focus on investigating and resolving security issues. Instead of working in relative secrecy, the JSST will have a strong public-facing presence at the [Joomla! Security Center](#).

**Polls**

**Joomla! is used for?**

Community Sites

Public Brand Sites

eCommerce

Blogs

Intranets

Photo and Media Sites

All of the Above!

**Who's Online**

We have 1 guest online

**Advertisement**

Featured Links:

[Joomla!](#)  
Joomla! The most popular and widely used Open Source CMS Project in the world.

[JoomlaCode](#)  
JoomlaCode, development and distribution made easy.

**Login Form**

Username

Password

Veja que a equipe já nos brinda com um portal completo e funcional.

Aqui apenas precisamos efetuar configurações, personalizações e nosso portal está valendo.

Com um design profissional e com muitos recursos, com a qualidade dos melhores portais existentes.

## Seção Administrator

Para acessar a seção de administração podemos clicar no link Administrator ou pelo link:

<http://ribafs.org/cursos/joomla/administrator>



### Joomla! Administration Login

Use a valid username and password to gain access to the Administrator Back-end.

[Return to site Home Page](#)



Username

Password

Language

Login

Lembre que o login é "admin".

Recomenda-se que troque o login para o seu login ou um outro.

## Administração do Joomla

**Joomla! Curso de Joomla 1.5** Version 1.5.14

Site Menus Content Components Extensions Tools Help Preview 0 1 Logout

**Welcome to Joomla!**

Congratulations on choosing Joomla! as your content management system. To help you get started, check out these excellent resources for securing your server and pointers to documentation and other helpful resources.

**Security**

On the Internet, security is always a concern. For that reason, you are encouraged to subscribe to the [Joomla! Security Announcements](#) for the latest information on new Joomla! releases, emailed to you automatically.

If this is one of your first Web sites, security considerations may seem complicated and intimidating. There are three simple steps that go a long way towards securing a Web site: (1) regular backups; (2) prompt updates to the [latest Joomla! release](#); and (3) a [good Web host](#). There are many other important security considerations that you can learn about by reading the [Joomla! Security Checklist](#).

If you believe your Web site was attacked, or you think you have discovered a security issue in Joomla!, please do not post it in the Joomla! forums. Publishing this information could put other Web sites at risk. Instead, report possible security vulnerabilities to the [Joomla! Security Task Force](#).

**Learning Joomla!**

A good place to start learning Joomla! is the "[Absolute Beginner's Guide to Joomla!](#)" There, you will find a [Quick Start to Joomla! guide](#) and [video](#), amongst many other tutorials. The [Joomla! Community Magazine](#) also has [articles for new learners](#) and experienced users, alike. A great place to look for answers is the [Frequently Asked Questions \(FAQ\)](#). If you are stuck on a particular screen in the Administrator (which is where you are now), try clicking the [Help toolbar button](#) to get assistance specific to that page.

If you still have questions, please feel free to use the [Joomla! Forums](#). The forums are an incredibly valuable resource for all levels of Joomla! users. Before you post a question, though, use the [forum search](#) (located at the top of each forum page) to see if the question has been asked and answered.

**Getting Involved**

If you want to help make Joomla! better, consider getting involved. There are [many ways you can make a positive difference](#). Have fun using Joomla!

▶ Logged In Users

▶ Popular

▶ Recent added Articles

▶ Menu Stats

▶ Joomla! Security Newsfeed

Aqui está a seção de administração do Joomla. Aqui poderemos controlar e personalizar muita coisa no portal.

Sempre saia desta seção clicando em Logout/Desconectar e evite navegar paralelamente por outros sites enquanto usa esta seção.

## 5 – Administração

- 5.1– Recursos nativos
- 5.2 - Recursos de terceiros
- 5.3- Recursos Extras

### 5.1 – Administrando os recursos nativos do Joomla

Após a instalação do Joomla, mesmo que seja uma instalação sem conteúdo de exemplo, ele já vem com uma boa quantidade de recursos nativos úteis, como componentes, menus, módulos, plugins e templates.

## Administração do Joomla instalado com conteúdo de exemplo

Recomendado para as primeiras instalações do Joomla.

### Instalação Resumida

Instalação do Joomla 1.5 com conteúdo de exemplo:

- Faça o download da última versão do Joomla 1.5 de <http://www.joomla.org/download.html>
- Crie uma pasta no htdocs. Algo como htdocs/portal
- Descompacte o Joomla dentro da pasta htdocs/portal
- Chame pelo navegador com <http://localhost/portal>
- A instalação é simples e em português. Apenas pode gerar alguma dúvida na configuração do banco de dados que deve ser:
  - Servidor - localhost
  - Usuário - root
  - Senha – (em branco, se estiver usando o Xampp)
  - Banco - portal
- Clique no botão Instalar Exemplo de Conteúdo
- Ao final exclua a pasta installation

### Administração do Joomla 1.5

Esta é a instalação recomendada para quem está instalando o Joomla nas primeiras vezes.

Após instalar acesse a seção Administrator com:

<http://localhost/portal/administrator>



## Recursos Nativos do Portal

Veja que já tem um portal completo e funcional. Observe que com essa instalação temos uma grande quantidade de recursos no nosso portal:

Menu principal, de topo e outros, enquete, usuários online, RSS, rodapé, banner, busca por todo o conteúdo do site, flash de notícias, últimas novidades (artigos de categorias), populares (artigos mais visitados de categorias), logomarca, opções para exibir conteúdo como PDF, imprimir ou enviar por e-mail, form de login, artigos publicados em forma de blog.

Agora resta ajustar o site para ficar como queremos:

- ajustar idioma,
- ajustar a tradução de alguns termos
- criar novo conteúdo
- desabilitar alguns módulos.
- Instalar alguns e desinstalar outros.
- Entre outros.

## 5.2 - Administração do Joomla instalado sem conteúdo de exemplo

Quando já adquirimos experiência com a instalação e a administração do Joomla é produtivo instalar o Joomla sem conteúdo de exemplo. Com isso não precisaremos remover nada nem desabilitar, apenas adicionaremos o que vamos usar.

Em caso de dúvidas, por ainda não lembrar como adicionar um recurso é útil fazer uma instalação paralela do Joomla com conteúdo para dar uma olhada como o recurso foi implementado e então implementar aqui na instalação sem conteúdo.

Agora vamos administrar o Joomla após uma instalação sem conteúdo.

### Aumentar tempo da sessão

Sugestão útil para trabalho local com mais conforto.

Site – Configuração Global – Sistema – Configurações da Sessão – Tempo da Sessão - 90

### Instalar Extensão

Todas as extensões do Joomla são instaladas da mesma form: componentes, módulos, plugins, languages e templates . Para ser instalada a extensão deve estar em um arquivo compactado.

- Extensões
- Instalar/Desinstalar
- Clique em Enviar arquivo...
- Selecione a extensão e clique em Enviar arquivo & Instalar

### Desinstalar Extensão

- Extensões
- Instalar/Desinstalar
- Clicar na aba abaixo da extensão que deseja desinstalar
- Clicar no botão à esquerda do nome da extensão para selecioná-la
- Clicar no botão acima Desinstalar

### Habilitar e Desabilitar Extensões

Temos um módulo (Ex.: Estatísticas) funcionando e ativo (habilitado) e não desejamos desinstalar mas apenas desabilitar, então acesse:

Extensões – Administrar Módulo

Selecione o Módulo Estatísticas

Clique no tique verde na coluna Habilitado para que se transforme em um círculo vermelho. Se clicar no círculo vermelho habilita o módulo  
Caso queira habilitar ou desabilitar Desabilitar.

## 1 - Trocar login do admin

<http://localhost/portal/administrator>

Após a instalação, na primeira vez que acessar o administrator troque o usuário admin por um outro login (por questão de segurança).

Clique em Site - Administrar Usuário – admin e altere para o seu login

**Obs.: Este passo é importante apenas quando mandar o site para o servidor. Se estiver trabalhando localmente em seu computador não precisa fazer isso.**

## 2 - Instalar Idioma pt-BR

Baixar de - <http://joomlancode.org/gf/project/portuguesebr/frs/> e tornar pt-BR o padrão do site e do admin:

Instale como sugerido acima

Extensions – Install/Uninstall

Indique o arquivo baixado

Torne o idioma instalado o padrão do Site e do Administrator

Extensions – Language Manager

Selecione Português(Brasil) e clique no botão Default acima e à esquerda

Clique na aba Administrator (à esquerda de Site)

Selecione novamente Português(Brasil) e clique no botão Default acima e à esquerda

## 3 – Algumas Traduções Manuais

Após instalar o idioma é interessante traduzir alguns termos que continuam em inglês.

Clique em Menus – Main menu\*

Traduzir o título do item de menu **Home** para **Início** do Main menu\* e também traduzir o Apelido de home para inicio (sem acento).

Traduzir o título do módulo Main menu para Principal que aparece no site.

Clique em Extensões – Administrar Módulo – Main menu e altere para Principal

Traduzir o título do menu Main menu, que aparece no menu Menus:

Clique em Menus – Administrar Menu – Main menu

Mude Main menu para Principal

## 4 - Configurar Componentes Internos

### Banner

Geralmente utilizado para divulgar outros sites através da troca de banners, onde recebemos uma imagem do banner, criamos uma categoria e um cliente e um banner para a divulgação. Veja os passos abaixo.

4.1-**Banner**: Crie uma nova categoria chamada **farmacias**, um cliente chamado **João Brito** e um banner chamado **Farmácia Central** (criar também um novo módulo tipo Banner)

Clique no menu Componentes – Banners – Categorias – Novo

Então preencha os dados do banner:

-Nome (Farmácia Central)

-Categoria (Farmácias)

-Cliente (João Brito)

-URL de destino (farmaciacentral.com.br)

-Selecione a imagem do banner em Seletor de imagem do Banner (Caso não tenha criado uma escolha uma das existentes como exercício)

4.2-**Contatos**: Criar uma nova categoria chamada **farmacias**, um contato chamado **Pedro**

Criar item de menu no menu Principal, do tipo Contatos para exibir o contato, com título **Fale comigo**.


Clique no menu Componentes – Contatos – Categorias – Novo

Selecione a categoria e o usuário que receberá os e-mails.

Dica: evite exibir o e-mail.

Veja como faço quando é apenas um site pessoal:

Os dados de quem vai receber as mensagens de contato.

 **Contato: [ Editar ]**

---

**Detalhes**

Nome:	Pedro
Apelido:	pedro
Publicado:	<input type="radio"/> Não <input checked="" type="radio"/> Sim
Categoria:	farmacias
Associado ao Usuário:	Administrator
Ordenar:	1 (Pedro)
Nível de Acesso:	Público Registrado Especial
ID:	1

---

**Informação**

Cargo:	
E-mail:	ribafs@gmail.com
Rua:	
Cidade/Bairro:	
Estado/País:	Ceará
CEP:	
País:	Brasil
Telefone:	
Celular:	
Fax:	
URL do Site:	http://ribafs.org
Informações Adicionais:	

O que exibir e o que ocultar:

**Parâmetro**

▼ Parâmetros do Contato	
Nome	<input type="radio"/> Ocultar <input checked="" type="radio"/> Exibir
Cargo	<input checked="" type="radio"/> Ocultar <input type="radio"/> Exibir
E-mail	<input checked="" type="radio"/> Ocultar <input type="radio"/> Exibir
Rua	<input checked="" type="radio"/> Ocultar <input type="radio"/> Exibir
Cidade/Bairro	<input checked="" type="radio"/> Ocultar <input type="radio"/> Exibir
Estado/País	<input type="radio"/> Ocultar <input checked="" type="radio"/> Exibir
CEP	<input checked="" type="radio"/> Ocultar <input type="radio"/> Exibir
País	<input type="radio"/> Ocultar <input checked="" type="radio"/> Exibir
Telefone	<input checked="" type="radio"/> Ocultar <input type="radio"/> Exibir
Celular	<input checked="" type="radio"/> Ocultar <input type="radio"/> Exibir
Fax	<input checked="" type="radio"/> Ocultar <input type="radio"/> Exibir
URL do Site	<input type="radio"/> Ocultar <input checked="" type="radio"/> Exibir
Informações adicionais	<input checked="" type="radio"/> Ocultar <input type="radio"/> Exibir
Imagem do Contato	<input checked="" type="radio"/> Ocultar <input type="radio"/> Exibir
vCard	<input checked="" type="radio"/> Ocultar <input type="radio"/> Exibir

► Parâmetros Avançados

► Parâmetros do e-mail

Para sites pessoais evite divulgar telefones e para sites de empresas divulgue telefones, e-mails, endereços, FAX, etc.

Após criar o componente vamos criar um item de menu para divulgar este componente.

Criar item de menu no Menu de Topo para Contatos do tipo Layout de Contato Padrão com título **Fale Comigo**

#### 4.3-Enquete

Clique em Componentes – Enquete – Novo

Preencha os dados da enquete com uma pergunta e algumas respostas.

Ao final lembre de publicar a enquete.

Crie também um módulo para exibir a enquete no site.

Extensões – Administrar Módulo - Novo – Votações/Enquete

Entre com o título **Enquete** e selecione à esquerda a Enquete que você criou e clique em

Salvar.

#### 4.4-WebLinks

Criar uma nova categoria chamada **farmacias** e um novo link chamado **Farmácia Central**

Clique no menu Componentes – WebLinks – Categorias – Novo

Com título **farmacias**

Clique em Links e crie um novo link

Entre com o nome e selecione a categoria.

#### 5-Conteúdo

A criação de conteúdo usando seção e categoria traz grandes vantagens.

5.1-Criar Seção com nome **Farmácias**,

contendo as categorias **Farmácia Central**, **Farmácia Montese** e **Farmácia Parangaba**.

Em cada uma das categorias adicionar os artigos: **Melhoral**, **Anador** e **Paracetamol**.

Ou seja, a categoria Farmácia Central conterà 3 artigos, a categoria Farmácia Montese também conterà 3 artigos e a Farmácia Parangaba também conterà 3 artigos. Os artigos Melhoral, Anador e Paracetamol estará nas três categorias.

Adicione em cada artigo o seguinte conteúdo:

Quantidade de Anador - 250

Preço unitário do Anador – R\$ 0,10.

#### Criar uma Seção

Conteúdo – Administrar Seção – Novo

Apenas o título é obrigatório - **Farmácias**

#### Criar Categoria

Conteúdo – Administrar Categorias – Novo

As três categorias devem pertencer à seção Farmácias.

Apenas a seção (**Farmácias**) o título (**Farmácia Central**) são obrigatórios.

## Criar Artigo

Conteúdo – Administrar Artigos – Novo

Aqui o título, a seção, a categoria e o texto são obrigatórios.

Exemplo: Artigo Melhoral

## Melhoral

Quantidade de Melhoral - 350

Preço unitário do Melhoral – R\$ 0,20.

### Dica:

Crie os 3 artigos da categoria Farmácia Central.

Selecione os 3

Clique no botão acima chamado Copiar e copie para a categoria Farmácia Montese

Repita os passos para copiar para a categoria Farmácia Parangaba.

Após as cópias ajuste se necessário o conteúdo de cada artigo.

Criar um item de menu no menu Principal para a Seção Farmácias

Do tipo layout padrão da Seção com título "Farmácias". Veja detalhes abaixo.

### Detalhes sobre a Criação de Item de Menu

Menus – Principal\* - Novo

Artigos – Seção – Layout Padrão de Seção

Título – Farmácias (Seção)

Parâmetros (Básico) – Selecionar a Seção Farmácias

Criar outro item de menu com título **Farmácias Blog** também no menu Principal para a Seção Farmácias do tipo **Layout do blog de Seção**.

Então experimente acessar o site e clicar nos dois itens de menu acima para ver a diferença de categorias para blogs.



## 6 - Criar Módulos Internos

Vou relatar apenas algumas propriedades, as que eu não relatar podem ficar com os valores default.

Para criar cada um dos módulos abaixo clique no menu:

Extensões – Administrar Módulo - Novo

### 6.1- **Banners:**

Exibir título – Não

Posição – footer

Título – Farmácia Central

### 6.2 - **Votação/Enquete:**

Posição – right

Selecione à direita e acima a esquete que você criou no componente criada

### 6.3 - **Quem está Online**

Título – Online

Posição – right

### 6.4 - **Breadcrumbs**

Exibir título – Não

Posição – breadcrumbs

Texto para o elemento de Início – Início

Caso ainda não saiba o que o Breadcrumb faz veja essa captura de um artigo dentro de uma categoria (mostra para o usuário onde ele encontra-se):

[Início](#) >> [Farmácias](#) >> [Farmácia Montese](#) >> [Paracetamol](#)

### 6.5 – **Pesquisa**

Exibir título – Não

Posição – user4

### 6.6 - **Últimas notícias**

Título – **Novidades**

Posição - user1

### 6.7 - **Conteúdo mais lido**

Título – **Populares**

Posição – user2

### 6.8 - **Estatísticas**

Marcar somente Contador de acessos

## 6.9 - Login

Não precisa alterar nada

## 6.10 - Rodapé

Exibir título – Não

Posição – footer

## 6.11 - Notícias em Destaque

Título - **Flash de notícias**

Categoria - **Farmácia Central**

Exibir título – Não

Posição - top

## 6.12-Imagem Aleatória

Criar uma pasta chamada **acaso** em imagens e entrar com seu caminho assim. Depois copie para esta pasta algumas imagens.

Título – Imagem Aleatória

Pasta da imagem - images/acaso

Tipo da imagem – jpg (também pode ser outro se sua imagem for de outro tipo, mas só pode ser de um tipo e todas as imagens devem ser do mesmo tamanho, de preferência).

Link – <http://farmaciamentese.com.br> (somente se desejar que ao clicar em qualquer imagem o visitante seja redirecionado para o site do link).

6.13 – Crie um segundo módulo tipo HTML Personalizado, mas este para o site e coloque nele o Tradutor do Google

Extensões – Administrar Módulo – Novo – HTML Personalizado

Título – Tradutor

Posição – right

Exibir Título - Não

Acesse - <http://www.google.com/webmasters/gadgets/foryourpage/index.html>

Selecione o Google Translator ou vá no link:

<http://www.google.com/ig/directory?synd=open&hl=pt-BR&gl=BR&url=http%3A%2F%2Fwww.google.com%2Ffig%2Fmodules%2Ftranslatemypage.xml>

Clique em Obter código e selecione o código para colar na região do Editor do Módulo. Selecione a ferramenta HTML e cole o código recebido, que é algo como:

```
<script src="http://www.gmodules.com/ig/ifr?url=http://www.google.com/ig/modules/translatemypage.xml&up_source_language=en&synd=open&w=160&h=60&title=Google+Translate+My+Page&lang=pt-BR&country=ALL&border=%23ffffff%7C3px%2C1px+solid+%23999999&output=js"></script>
```

## 7 - Criar Menus

**Menu do Usuário** (só deve aparecer quando se efetua login, ou seja, será do grupo registrado)

Clique no menu Menus – Administrar Menus – Novo e crie o menu Usuário

### 7.1- Usuário

Nome único – usermenu, em todos os outros digite Usuário.

Habilitar o módulo Usuário e alterar Nível de acesso para Registrado.

Adicionar em Parâmetros Avançados - Sufixo de Classe do Módulo: `_menu`

#### 7.1.1- Seus dados

Adicionar item para usuário Alterar seus dados.

Novo item de menu – Usuário – Layout de formulário de usuário.

Nível de acesso – Registrado. Título – Seus dados

#### 7.1.2 - Enviar Artigo

Adicionar item para envio de artigo.

Novo item de menu – Artigos – Artigo - Layout de envio de artigo.

Nível de acesso – Especial. Título – Enviar Artigo

#### 7.1.3-Enviar Link

Adicionar item para envio de link.

Novo item de menu – Weblinks - Weblink – Layout de envio de Weblinks.

Nível de acesso – Especial. Título – Enviar Link

## Menu de Topo

Clique no menu Menus – Administrar Menus – Novo e crie o menu abaixo:

### 7.2- Menu de Topo

Nome único – topmenu e demais digitar Menu de Topo.

Clique em Extensões – Administrar Módulo – Menu de Topo

Habilitar o módulo Menu de Topo, ocultar o título, posição – user3,

7.2.1-Mudar o item de menu Contato (Fale comigo) do menu Principal para o menu topmenu que acabamos de criar

## Novo menu chamado Conteúdo

### 7.3-Criar menu chamado Conteúdo


Clique no menu Menus – Administrar Menus – Novo e crie o menu Conteúdo (da mesma forma como criou os anteriores)

Obs.: Ao criar novos menus sempre adicionar ao Módulo, em Parâmetros Avançados - Sufixo de Classe do Módulo "`_menu`".

## 8-URLs Amigáveis

Clique no menu Site – Configuração Global – Site e marque URLs Amigáveis e Usar *mod\_rewrite* do Apache, como mostra a figura abaixo:

**Configurações de SEO**

URLs Amigáveis (Search Engine Friendly)	<input type="radio"/> Não	<input checked="" type="radio"/> Sim
Usar <i>mod_rewrite</i> do Apache	<input type="radio"/> Não	<input checked="" type="radio"/> Sim 
Adicionar sufixo às URLs	<input checked="" type="radio"/> Não	<input type="radio"/> Sim

Depois disso renomeie o arquivo htaccess.txt que existe no raiz do site htdocs/portal para “.htaccess”. De forma que fique assim htdocs/portal/.htaccess

Com isso seu site estará agora com URLs amigáveis. Veja a vantagem.

Antes, ao clicar em FAQ o visitante ia para a página:

[http://localhost/portal/index.php?option=com\\_content&view=section&id=3&Itemid=41](http://localhost/portal/index.php?option=com_content&view=section&id=3&Itemid=41)

Agora ele vai para:

<http://localhost/portal/faq>

## 9-Editor

Instalar um novo menu (JCE) e trocar o atual por este

9.1-Instalar novo editor – JCE (componente e plugin):

Faça o download daqui:

<http://www.joomlacontenteditor.net/downloads/editor/179-joomla15x>

Instale e troque para que este seja seu editor atual do Joomla, assim:

Clique em Site – Configuração Global – Site - Editor WYSIWYG Padrão

Selecione Editor JCE e salve.

9.2-Outra alternativa é ampliar os recursos do editor default do Joomla, que é o TinyMCE, através do plugin

Clique em Extensões – Administrar Plugin - Editor – TinyMCE.

Nos parâmetros do plugin em Funcionalidade selecione Estendido.

## 10-Diversos

10.1-Alterar paginação para 30 – Configuração Global – Site – Tamanho da lista

10.2-Alterar a mensagem de Site em manutenção, adicionando o e-mail do responsável.  
Site – Configuração Global – Site – Mensagem para Site em Manutenção

10.3-Para melhor organização de mídias criar os diretórios:  
**farmaciacentral, farmaciamentese e farmaciaparangaba.**

Clique no menu Site – Administrar Mídia e crie os diretórios.

10.4-Altere o fuso horário para -03:00 (menos 3, que é o do Brasil)  
Clique no menu Site - Configuração Global – Servidor – Configuração da localidade – Fuso horário

### **11-Usuários e Privilégios**

Criar um usuário de cada grupo existente: registrado, autor, editor, redator, gerente, administrador e super administrador.

Clique no menu Site – Administrar Usuários – Novo e crie cada um dos usuários abaixo. Observe que cada um é de um grupo diferente.

11.1 – Criar um usuário do grupo Registrado chamado "registrado".

11.2 – Criar um usuário do grupo Autor chamado "autor".

11.3 – Criar um usuário do grupo Redator chamado "editor".

11.4 – Criar um usuário do grupo Editor chamado "redator".

11.5 – Criar um usuário do grupo Gerente chamado "gerente".

11.6 – Criar um usuário do grupo Administrador chamado "administrador".

11.7 – Criar um usuário do grupo Super Administrador chamado "superadmin".

**Registrado** – visitante normal que se registrou no site. Pode visualizar e alterar seus dados mas não pode editar nem enviar artigos ou links.

**Autor** – Pode enviar artigos ou links para serem aprovados e somente acessa o frontend. Não pode editar artigos existentes.

**Editor** – Pode enviar artigos ou links. Pode editar artigos existentes. Acessa somente pelo frontend.

Tanto autor quanto editor precisam que um publisher/redator ou superior publique seus artigos.

**Publisher/Redator** – Pode enviar, editar e publicar artigos mas somente pelo frontend.

**Gerente, Administrador e Super Administrador** – Podem fazer tudo que os anteriores fazem e também efetuar login pelo backend e com maiores privilégios.

11.8-Trocar editor, fuso horário e idioma do usuário **redator**. Observe que o Registrado não tem editor em suas propriedades.

11.9-Efetue login com cada um dos usuários e observe os recursos disponíveis para o mesmo, pelo frontend e pelo backend (quando for o caso). Veja que alguns grupos de usuários não têm privilégio de se conectar pelo backend.

## 12-Templates

Clique em Extensões – Instalar/Desinstalar

12.1-Instalar alguns templates selecionados

### **Tornar o template instalado o template padrão do site**

Extensões

Administrar Tema

Selecione o template à esquerda

Clique no botão Padrão

12.2-Instalar um módulo para visualização de templates pelo site

## 13-Favicon

Criar um favicon.ico que represente o site e copiar para o raiz do template.

Converte imagem em favicon e cria online - <http://www.favicon.cc/>

Apenas crie um arquivo com 16x16 pixels e renomeie para favicon.ico.

## 14 - Banner

Criar banner representando o site, que pode ser animado (o Gimp é uma ótima opção). Para criar um gif animado no Gimp:

- Crie as várias imagens com as dimensões do banner (468x60pixels)
- E abra o Gimp,
- Então clique em Arquivo – Abrir como camadas ...
- E salve como GIF.
- No momento de salvar salve como animação e adicione o intervalo entre as imagens de uns 3000 milisegundos.

## Backup e Migração do Portal

Instalar o Gerenciador de backup dos arquivos e do banco JoomlaPack

joomlaPack (<http://joomlapack.net/download/JoomlaPack-Components-download.html>)

Muitos recursos interessantes, backup do banco completo, backup de todos os arquivos do site, backup de todos os bancos do servidor (MySQL). A opção default é todos os arquivos e o banco do site.

Também pode enviar por e-mail para o superadministrador ou outro user indicado.

Veja que existe uma grande quantidade de configurações mas as configurações default já atendem para quem deseja fazer backup do banco e de todos os arquivos.

Apenas clique em Componentes – JoomlaPack – Backup Now

Uma grande vantagem é gerar o pacote completo (arquivos e banco) e ao restaurar pode ser onde você desejar, pois o pacote reconstituiu um instalador mais simples mas que funciona.

No caso podemos redistribuir um site depois de todo configurado ou transferir com facilidade um site local para o servidor ou do servidor para local e de um para qualquer outro computador.

No site **Joomla – Portal Modelo** - <http://modelo.ribafs.org/> instalei tudo que está aqui e várias extensões de terceiros. Alguns itens tem nomes diferentes mas está parecido.

### 5.3 - Inserindo Código HTML e JavaScript em Artigos

Os artigos publicados no Joomla podem ser personalizados utilizando código HTML. É possível inserir estilos e gadgets, porém independente do seu site está ou não usando um editor WYSIWYG, ao salvar o artigo, todo código inserido será removido. No tutorial abaixo veremos como evitar que essa remoção ocorra.

Para evitar a remoção do código HTML:

1. Acesse o seu Painel de Controle. Veja como acessar o Painel de Controle;
2. Clique em "Article Manager";
3. Selecione a opção "Parameters";  
Parameters
4. Em "Filtering options" selecione:

Filter groups: selecione o grupo que terá permissão para inserir HTML no conteúdo.

Exemplo: Super Administrator.

Filter type: Whitelist

Filtering options

5. Role a barra de rolagem para cima e clique no botão "Save";

Para inserir HTML no artigo basta clicar em "Edit HTML Source".

Edit HTML Source

Dica do site: <http://www.1fish.info/> - Free Joomla Website, Hosting and Template



## 5.4 – Recursos de Terceiros

### Administrando Recursos de Terceiros no Joomla

Atualmente existe bem perto de 4000 extensões para Joomla somente as catalogadas no site <http://extensions.jooml.org>.

Frente a essa grande quantidade de extensões é produtivo ter nossa relação de extensões preferidas. Abaixo enumero a minha relação, que, claro, pode ser diferente da sua e aparece aqui apenas para servir de referência, embora possa ser útil também.

**Obs.:** Sempre visite o site do autor para receber a última versão.

#### Menus

- Módulo (horizontal) KwickSliding\_menu\_J15 -

[http://www.camp26.biz/component/option,com\\_repository/Itemid,7/func,fileinfo/id,10/](http://www.camp26.biz/component/option,com_repository/Itemid,7/func,fileinfo/id,10/)

- Vertical -

[http://www.camp26.biz/component/option,com\\_repository/Itemid,7/func,fileinfo/id,15/](http://www.camp26.biz/component/option,com_repository/Itemid,7/func,fileinfo/id,15/)

#### Tools

- JoomlaPack - <http://joomlapack.net/download/itemlist/category/51-joomlapack-j15x.html>

- Advanced Modules - <http://www.nonumber.nl/extensions/advancedmodulemanager?tab=download>

System - Plugin que ativa mais propriedades nos módulos, inclusive um ícone. Configurações no plugin.

- EasyCreator - <http://joomlancode.org/gf/project/easyjoomla/frs/>

j!Dump - <http://code.joomla.org/gf/project/jdump/frs> (componente e plugin)

- Jumi – Boa extensão para ajudar na criação de componentes, módulos e plugins. <http://jumi.vedeme.cz/>

#### Calendários

- JEvents - <http://www.jevents.net/jevents-download>

#### Banners

- Hot image slider (módulo). Bom para promoções, portfólio, grandes avisos ou troca de banner

<http://www.hotjoomlatemplates.com/free-joomla-extensions/20-hot-image-slider-free-jquery-image-slider.html>

#### Chats

- JPFChat (simples) - <http://www.jpchat.com/index.php/downloads>

- Skype me

#### Forum

Simplest Forum - [http://simplestforum.org/index.php?option=com\\_content&view=article&id=50&Itemid=56](http://simplestforum.org/index.php?option=com_content&view=article&id=50&Itemid=56)

joo!BB - <http://sourceforge.net/projects/joobb/files/>

ninjaBoard - [http://ninjaforge.com/index.php?option=com\\_ninjacentral&page=show\\_package&id=70&Itemid=245](http://ninjaforge.com/index.php?option=com_ninjacentral&page=show_package&id=70&Itemid=245)  
Kunena - <http://joomlancode.org/gf/project/kunena/frs/> (componente)  
Kunena discuss (plugin) - [http://joomlancode.org/gf/project/plg\\_kdiscuss/frs/](http://joomlancode.org/gf/project/plg_kdiscuss/frs/)

Caríssimos. O Kunena é uma evolução do tão consagrado FireBoard do Joomla!  
Um componente para Fórum completíssima e recheada de diversos recursos, um deles é o que vou falar aqui.

O Plugin Kunena Discuss. Com ele você pode discutir um artigo diretamente no fórum, ou seja, você cria um artigo e automaticamente um tópico é gerado na categoria que você define dentro do fórum para ser debatido.

Existe a opção "Quick Post" onde as pessoas discutem diretamente dentro do artigo (com uma estrutura semelhante ao JComments).

Até agora eu não achei tradução pt\_Br para o Plugin, mas o arquivo .INI é pequeno e não dará muito trabalho a quem se aventurar na tradução.

Sinceramente estou satisfeitíssimo com o Componente e seu Plugin.  
Quem quiser testar, pode baixar a última versão do Kunena, no Joomla Clube  
O Plugin Kunena Discuss você encontra no JED.  
Pedro Pisandelli na lista joomlaceara do Google.

### **Traduções**

- JoomFish - <http://www.joomfish.net/en/downloads>  
- Multilingual Registration Approval - Este plugin obriga um registo a ser aprovado por um administrador. Primeiro o email do usuário tem de ser verificado por ele. Depois o administrador aprova o registo. Nota: o tema tem de incluir uma posição para a mensagem ( <jdoc:include type = "message" /> ) Se utilizar o JoomFish, a configuração global para cada idioma tem de ser parametrizada.

### **Comentários**

- jComments - <http://www.joomlatune.com/jcomments.html>

### **Comunidades**

- Community Builder - <http://www.joomlapolis.com/content/section/5/37/>

### **Conteúdo**

- K2 - <http://k2.joomlaworks.gr/>  
- Sobi2 -  
[http://www.sigsiu.net/download/components/sigsiu\\_online\\_business\\_index\\_2.html](http://www.sigsiu.net/download/components/sigsiu_online_business_index_2.html)  
- FLEXIcontent - <http://www.flexicontent.org/>  
- Sourcerer - <http://www.nonumber.nl/extensions/sourcerer?tab=download> (escrever PHP ou JS no conteúdo)

### **Ajuste Tamanho de Fontes**

- mod\_fhw-fontadjust - <http://www.floridahostweb.com/downloads/category/1-joomla-modules-category> (Posição breadcrumbs)

### Aniversário

plgContentAnniversary - Estou hoje com {anniversary 1956-08-03} anos.

<http://joomlancode.org/gf/project/pasamioprojects/frs/>

### Player de Vídeo e áudio

AllVideos Reloaded - <http://joomlancode.org/gf/project/allvideos15/frs/> (ver link Documentação no plugin para ajuda). Audio em images/stories/audio e Vídeos em images/stories/videos

Este componente também instala plugins e um módulo.

**ALERTA:** não copie e cole da documentação, pois pode dar problema.

plg\_allvieos - <http://joomlancode.org/gf/project/allvideos15/frs/>

AllVideos - <http://www.joomlaworks.gr/>

denVideo (vídeo, áudio, imagens: mp3, divx, mp4, wmv, wma, H264, images(JPG, GIF, PNG):

<http://joomlancode.org/gf/project/denvideo/frs/>

{denvideo URL WIDTH} = auto set size based on Width

{denvideo URL WIDTH HEIGHT} = set custom Width and Height

{denvideo URL WIDTH HEIGHT AUTOSTART} = customize Width, Height and set if the video starts automatically (true/false)

so im thinking there might be a way to pass along the flashVars like

```
{denvideo http://poorvida.com/FLAM/flam-player.swf 300 315 ovr_color=0x75855C  
ovr_langage=en ovr_playlist=mc+astro}
```

or simplified

```
{denvideo http://poorvida.com/FLAM/flam-player.swf 300 315 0x75855C en mc+astro}
```

##### Fast Friendly Suport

[url]<http://forum.joomla.org/viewtopic.php?f=262&t=308308&p=1352096#p1352096>[/url]

Alerta: computadores sob firewall devem liberar os sites tipo Youtube, caso sejam bloqueados para que os vídeos sejam exibidos.

Para reproduzir vídeos do Windows no Firefox tipo wmv precisamos da extensão:

<https://addons.mozilla.org/en-US/firefox/addon/446> (MediaPlayerConnectivity) ou do MoonLight - <http://go-mono.com/moonlight/>

Uma boa idéia é converter os arquivos do Windows para FLV.

### Downloads

- JotLoader - <http://www.kanich.net/radio/site/jotloader/jotloader-download>

- PhocaDownloads - <http://www.phoca.cz/download/section/3-phoca-download>

- Remository - <http://remository.com/downloads/softwarephp5/joomla-1.5-software/>  
(componente)

### **E-commerce**

Virtuemart - <http://virtuemart.net/downloads> (instalar componente, módulos, plugins (do pacote Complete e habilitar todos) e languages: descompactar pacote Languages e copiar tudo para o instalado)

### **Editores**

JCE (componente e plugin) - <http://www.joomlacontenteditor.net/downloads/editor/179-joomla15x>

### **Templates**

<http://joomlaos.de> e <http://joomla24.com>

Módulo para visualização de templates pelo site – Template Switch -  
[http://www.mybp3.com/templateswitcher/cat\\_view/2-downloads](http://www.mybp3.com/templateswitcher/cat_view/2-downloads)

### **Livro de Visitas**

Phoca - <http://www.phoca.cz/download/section/4-phoca-guestbook>

JamBook - [http://joomlancode.org/gf/project/jambook/frs/?action=FrsReleaseView&release\\_id=3636](http://joomlancode.org/gf/project/jambook/frs/?action=FrsReleaseView&release_id=3636)

EasyBook (bom) - [http://joomlancode.org/gf/project/easyjoomla/frs/?action=FrsReleaseBrowse&frs\\_package\\_id=225](http://joomlancode.org/gf/project/easyjoomla/frs/?action=FrsReleaseBrowse&frs_package_id=225) (Em Manage Entries selecionar nas Preferências)

### **Captcha**

<http://groups.google.com/group/joomlacaptcha> (captcha em todos os forms)

Qcontacts - <http://www.latenight-coding.com/joomla-addons/qcontacts/download-qcontacts.html>

### **Mapas do Site**

Xmap - [http://joomla.vargas.co.cr/downloads/cat\\_view/1-xmap](http://joomla.vargas.co.cr/downloads/cat_view/1-xmap)

Joomap - <http://koder.de/projekte/joomap/>

### **RSS**

Ninja RSS Syndicator - [http://ninjaforge.com/index.php?option=com\\_ninjacentral&page=show\\_package&id=74&Itemid=236](http://ninjaforge.com/index.php?option=com_ninjacentral&page=show_package&id=74&Itemid=236) (na posição syndication)

### **Upload**

Simple Upload - <http://ribafs.org/portal/joomla/69-modulos/96-simple-upload>

### **Gerenciador de Arquivos**

NinjaXplorer - [http://ninjaforge.com/index.php?option=com\\_ninjacentral&page=show\\_package&id=44&Itemid=235](http://ninjaforge.com/index.php?option=com_ninjacentral&page=show_package&id=44&Itemid=235)

### **Galerias de Imagens**

Phora Gallery - <http://www.phoca.cz/download/category/1-phoca-gallery-component>  
MorfeoShow - <http://morfeoshow.joomlaitalia.com/>

### Gerador de Formulário

JForms (envia por e-mail e/ou guarda no banco) - [http://jforms.mosmar.com/index.php?option=com\\_content&view=article&id=3&Itemid=4](http://jforms.mosmar.com/index.php?option=com_content&view=article&id=3&Itemid=4)

### Seguidor do Tweeter

Módulo tweetfollower - <http://www.joomla-labs.com/download/5-twitter-follower-counter-module.html>

### Google Maps

googleDirection - <http://www.kksou.com/php-gtk2/Joomla-Gadgets/googleDirections-plugin.php#download>

Também requer o plugin - <http://www.kksou.com/php-gtk2/Joomla-Gadgets/googleMaps-plugin.php>

phoca maps - <http://www.phoca.cz/download/category/37-phoca-maps-component>

Requer Google Map key (exclusivo do site) adquirida aqui:  
<http://code.google.com/intl/pt-BR/apis/maps/signup.html>

ABQIAAAAlvQ5AlfcJUqSIGo23xoguxTcrb04Ta7esAXVyDkWMbrqOKR80hSRWUIA1221u9jgSXsjTxSqQUvi dA

Latitudes e Longitudes - <http://itouchmap.com/latlong.html>  
Basta digitar o endereço e receber lat e long.

```
{googleDir width=680 height=400 dir_on_right=1 hide_direction_text=1  
from="Stanford University, CA 94305"  
to="456 University Ave. Palo Alto, CA 94301  
to: 1600 Amphitheatre Parkway, Mountain View CA 94043,  
to: 701 First Avenue, Sunnyvale, CA 94089"}
```

### Gerenciamento do Banco de Dados

easySQL - <http://lurm.net/?other/easysql>

### Doações

mod\_paypal - [http://www.joomlaspan.com/index.php?option=com\\_sobi2&sobi2Task=sobi2Details&catid=14&sobi2Id=13&Itemid=3](http://www.joomlaspan.com/index.php?option=com_sobi2&sobi2Task=sobi2Details&catid=14&sobi2Id=13&Itemid=3)  
donate - <http://joomla.org/gf/project/donation/frs/>

### Metadados

Missing Metadata - <http://www.joomlajumpstart.com/download-missing-metadata>

**News**

Vemod News Mailer - [http://vemod.unimatrix.net/index.php?option=com\\_phocadownload&view=category&id=1:vemod-news-mailer-for-joomla-1.5&Itemid=83](http://vemod.unimatrix.net/index.php?option=com_phocadownload&view=category&id=1:vemod-news-mailer-for-joomla-1.5&Itemid=83)

**Social**

The Ultimate Social Bookmarking Plugin - <http://joomladigger.com/joomla-1.5-extensions/plugins/view-category.html> (requer registro)

Um ótimo site para encontrar boas extensões e traduzidas para português do Brasil é o JoomlaClube:

<http://joomlaclube.com.br/>

## 5.5 - Recursos extras

**Administrando no Joomla alguns recursos não contemplados acima.**

### 5.5.1 - Dicas Extras sobre Administração do Joomla

#### Quebra de Página

Um bom recurso para artigos é a Quebra de páginas. O Joomla cria um índice com os nomes dados às páginas quebradas.

#### Para Inserir Quebras de Página em artigos:

Clicar no texto, onde deseja a quebra

Clicar no botão Quebra de Página1

**Página inicial** – Este é um item importante do site, pois com ele podemos dizer que um artigo será automaticamente aberto sempre na página inicial.

#### Inserindo Imagem no Artigo

##### Quando a imagem já se encontra no servidor

- Clicar no botão **Imagem** abaixo
- Selecionar a imagem
- Clicar no botão **Inserir** acima

Estas imagens acima devem ser inseridas através do media manager ou de outra forma, como FTP NinjaXplorer, cPanel, etc.

##### Quando a imagem está em nossa máquina local

- Clicar no botão Imagem (abaixo)
- Clicar em Arquivo e selecionar a imagem
- Clicar em Iniciar Envio para fazer o upload
- Após finalizar clique na imagem e então
- Clicar no botão inserir acima

A imagem estando posicionada no editor podemos arrastar e redimensioná-la.

#### Artigo Default

No Joomla 1.5 podemos escolher qual será o artigo default:

- Menus – Main Menu \*
- Selecionar o item e Padrão

Vídeo ensinando a Criar um Artigo no Joomla 1.5:

<http://joomlaforbeginners.com/joomla-1.5-video-tutorials/creating-an-article-in-joomla-1.5.html>

**Créditos dos Artigos**

Menu Conteúdo – Administrar Artigos

Botão Preferências

Então exibe-se/oculta-se o que bem entender.

**Alerta**

Ao escrever um artigo, crie offline num editor HTML, como o **kompozer** (<http://kompozer.net>) e só quando finalizar abra o Joomla e cole o conteúdo. Acontece muito de o timeout expirar ou haver uma queda de energia ou outro problema e perdermos tudo que digitamos. Podemos aumentar o timeout mas em termos de segurança não é recomendado.

Idealmente copie e cole partes de um artigo grande então clique em Aplicar.

Assim faça até concluir.

**Mensagens Privadas no Backend**

Quando algum autor ou qualquer outro que escreva artigo tenha escrito um artigo usando o frontend o backend exibe uma mensagem acusando o novo artigo. Logo que um usuário acessa o backend ele pode perceber o aviso acima e à direita, com o número de mensagens. Então clicamos na mensagem para saber detalhe e publicar o artigo ou eliminar, se for o caso.

Podemos configurar o gerenciador de mensagens para nos enviar essas mensagens por e-mail:

Ferramentas – Ler mensagens – Configurações

Marcar "Avisar por e-mail quando chegar nova mensagem:".



## 5.5.2 - Alterando a pasta Administrator

1) Adicione na primeira linha do administrator/index.php

```
if ($_COOKIE['JoomlaAdminSession'] != "1234567890")
{
header("Location: ../index.php");
}
```

2) Crie o diretório que será o novo admin

ad1948mc

Bom exemplo em termos de segurança

3) Dentro dele crie um arquivo ad1948mc/.htaccess contendo:

```
RewriteEngine On
RewriteCond %{REQUEST_URI} ^/administrator
RewriteCond %{HTTP_COOKIE} !JoomlaAdminSession=1234567890
RewriteRule .* - [L,F]
```

4) Crie um ad1948mc/index.php, contendo:

```
<?php
$admin_cookie_code="1234567890";
setcookie("JoomlaAdminSession",$admin_cookie_code,0,"");
header("Location: /portal/administrator/index.php");
?>
```

Original em inglês:

<http://www.joomlahackers.net/joomla-tutorials/change-joomla-admin-name-or-path.html>

## 6 – Segurança em sites web

### 6.1 - Cuidados Básicos na Administração do Joomla

#### Ataque

Caso o seu site sofra invasão tome muito cuidado para que não aconteça novamente. Remova todo o conteúdo e restaure um backup confiável. Faça um rastreamento e tente descobrir a falha que permitiu o ataque e procure eliminar reforçando a segurança. Se possível use um servidor dedicado, onde terá mais segurança pelo fato de ser apenas você no computador.

Não divulgue que seu site foi atacado, muito menos detalhes sobre seu site e que extensões está usando.

#### Atualizações

Esteja sempre muito atento para atualizar o seu site sempre que sair uma nova versão. Não aplique o patch simplesmente, mas antes pare o site, faça um backup completo de todos os arquivos e de todo o banco (JoomlaPack) e só então aplique o patch. Em caso de problema poderá restaurar o backup. Boa chance de ocorrer problema aqui é quando você está usando o template default e faz alterações nele ou faz alguma alteração em uma das extensões nativas do Joomla, o que deve ser evitado.

Uma boa pedida para se manter atualizado é cadastrar seu e-mail na seção de downloads do Joomla ou assinar o RSS e consultar seu e-mail ou o leitor de RSS pelo menos uma vez ao dia.

Sempre antes de instalar uma extensão de terceiro consulte o site sobre vulnerabilidades das extensões abaixo:

[http://docs.joomla.org/Vulnerable\\_Extensions\\_List](http://docs.joomla.org/Vulnerable_Extensions_List)

Também lembre de estar sempre atualizando as extensões de terceiros que tem no site. Existem algumas extensões que cuidam disso, veja no <http://extensions.joomla.org>.

#### Servidor de Teste

Tenha um servidor de teste local, que simule ao máximo o site do servidor, versão do Joomla, extensões instaladas, etc. Um servidor de teste instalado localmente é mais ágil e nele podemos reverter os possíveis problemas de uma instalação de extensão sem traumas, o que representa maior segurança do que fazer testes no site que está valendo no servidor.

#### Login e Senha

Sua primeira providência ao instalar o site em Joomla deve ser a de alterar o login default, que é admin, para um de sua preferência.

Maior cuidado deve ter com a senha, usando senha forte contendo letras, números e símbolos e ainda por cima os escolhendo de forma bem aleatória e diferente do habitual. A senha de um site importante deve ser diferente de todas as senhas que se utiliza: dos e-mails, de outros sites, etc.

**Configurações do Joomla e do PHP**

Seja exigente com as configurações de segurança do php.ini e do Joomla. Tudo que puder fazer para melhorar a segurança, faça e faça rápido e bem feito.

**Backups**

Efetue backups com a regularidade que considera adequada para seu site, mas nunca menospreze essa medida, pois ela o salvará em caso de problema. Nunca deixe essa medida somente por conta do servidor de hospedagem, faça você mesmo.

**Servidor de Hospedagem**

Seja criterioso ao escolher o servidor de hospedagem para seu site ou para o cliente. Observe há quantos anos ele está no mercado, peça informações de colegas que já tem sites hospedados. Veja se os recursos oferecidos pelo servidor atendem suas necessidades. Principalmente veja a estabilidade do servidor, se vive caindo, ou se o suporte é eficiente.

Veja aqui algumas recomendações para a contratação de um servidor de hospedagem:

<http://ribafs.org/portal/colaboracoes/1-ribamar-fs/97-boa-hospedagem>

Referência: <http://coisasdewebloucos.blogspot.com/2009/04/os-10-mais-erros-mais-estupidos-do.html>

## 6.2 - Dicas de Segurança

Alguns cuidados básicos para reforçar a segurança do site.

### Ocultar Mensagens de Erro

Evite que as mensagens de erro mostrem informações importantes, como nome do servidor, versão dos softwares, etc. Isso deve ser feito no código quando está usando um aplicativo que você criou. No caso do Joomla ele já vem configurado por padrão para ocultar essas mensagens. Mas caso queira verificar vá em

Site – Configuração Global – Sistema – Configuração de Depuração. Desative ambos.

### URLs Amigáveis

Ative também as URLs amigáveis. Para isso acesse:

Site – Configuração Global – Site – Configurações de SEO

Marque como sim tanto a opção URLs Amigáveis quanto Usar mod\_rewrite.

Depois renomeie o arquivo “htaccess.txt” que existe no raiz do Joomla para “.htaccess”.

### Extensões de Terceiros

Mantenha uma relação de todas as extensões de terceiros que tem instaladas no site. Todas as extensões que exibem seus nomes e versão no site devem ter essas informações removidas.

Algumas basta ir nas configurações, outras requerem que editemos alguns scripts.

### Valores Default

Alguns valores default devem ser evitados, como o prefixo do banco de dados (durante a instalação). Em termos de segurança, valor default não deve ser usado.

## 7 – Programação

### 7.1 - Ferramentas

- 7.1.1- Extensões para o Firefox
- 7.1.2 - Extensões para o Joomla
- 7.1.3 - Editores de Programação
- 7.1.4 - IDEs
- 7.1.5 - Editores Gráficos

### 7.2 - Framework

### 7.3 - Criando Extensões

- 7.3.1 - Módulos
- 7.3.2 - Componentes
- 7.3.3 - Plugins
- 7.3.4 - Templates
- 7.3.5 - Languages

### 7.4- Extensões de Exemplo

## 7.1 – Ferramentas

- 7.1.1- Extensões para o Firefox
- 7.1.2 - Extensões para o Joomla
- 7.1.3 - Editores de Programação
- 7.1.4 - IDEs
- 7.1.5 - Editores Gráficos

### 7.1.1 – Extensões para o Firefox

#### Extensões do Firefox para Trabalhar com Desenvolvimento Web

Firebug  
Measureit  
ColorZilla

### Firebug

O Firebug é uma extensão revolucionária para trabalho especialmente com CSS. Com ela se trabalha com o CSS vendo onde se localiza na página dentro do HTML. Também se trabalha com JavaScript.

O trabalho com desenvolvimento de templates Joomla é bem mais facilitado com o uso do Firebug.

## Instalação

Abra o Firefox  
Clique no menu Ferramentas – Complementos  
Clique na aba Adicionar  
Digite firebug e tecele ENTER

Quando aparecer clique em **firebug**  
Então clique em Adicionar ao Firefox  
Clique em Instalar agora  
Ao final clique no botão Reiniciar o Firefox

Dessa forma se instala qualquer instensão do Firefox encontrada no repositório da Mozilla.

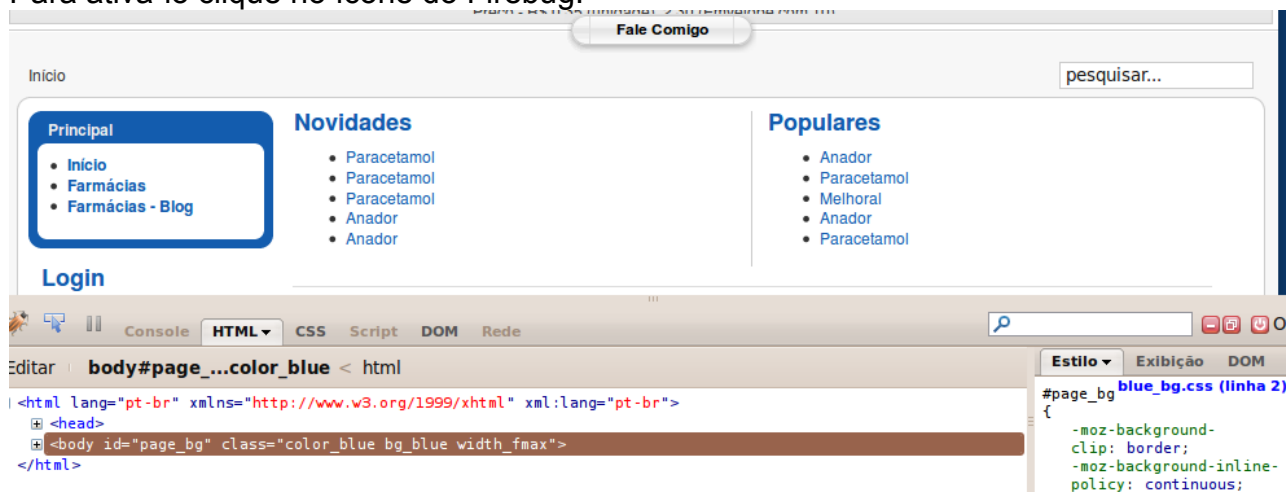
## Usando

Após a instalação o Firebug adiciona um ícone na barra de tarefas:



É o ícone do inseto (barata).

Para ativá-lo clique no ícone do Firebug.



Veja que a tela fica dividida em dois painéis. Acima fica a página e abaixo o painel doo Firebug.

## Painel do Firebug:



O item selecionado por default é o HTML, que mostra os respectivos IDs e classes e à direita o código do CSS respectivo.

### **Inspeção de Elementos**

O principal recurso que utilizo é o segundo ícone da esquerda para a direita, o inspetor de elementos:



Ao clicar neste ícone e mover o mouse sobre os elementos da página, o Firebug irá mostrar no painel abaixo detalhes importantes sobre o CSS e o HTML de cada elemento.

Quando quiser examinar um elemento apenas clique sobre o mesmo na página. Veja que abaixo e à esquerda aparece o HTML com o ID ou classe e à direita o código do CSS inclusive com o nome do arquivo e seu caminho.

Explore esse recurso que é bastante útil.

Existem muitos tutoriais sobre o Firebug na Internet inclusive no Youtube.

## Measureit

Esta extensão destina-se a medir a tela ou qualquer elemento (em pixel).

Instale da mesma forma que instalou o Firebug.

### Usando

Apenas clique no ícone da barra de tarefas (abaixo e à esquerda):



Clique no início e arraste até o final de onde deseja medir.



## ColorZilla

Esta extensão é muito útil para indicar a cor de qualquer elemento de uma página em hexadecimal.

Após instalada adiciona um ícone na barra de tarefas:



### Usando

#### **Capturando Cores**

Apenas clique no ícone e mova o mouse pela página.

Ao passar o ponteiro sobre um elemento o respectivo número da cor em hexadecimal e RGB aparece na barra de status juntamente com as coordenadas X e Y. Captura cores inclusive de imagens.

Para desativar apenas clique com o botão direito em uma área livre da tela.

#### **Seletor de Cores**

Clique com o botão direito do mouse sobre uma área livre da página – ColorZilla – Seletor de Cores

## 7.1.2 – Extensões para o Joomla

Aqui algumas poucas extensões para Joomla que ajudam na criação de outras extensões.

**Jumi** - <http://jumi.vedeme.cz/>

Jumi é o kit de extensões para Joomla que facilita a criação de extensões para o próprio Joomla.

Atualmente o Jumi tem um plugin que permite adicionar código HTML ou JavaScript diretamente em artigos. Veja os passos:

- Abrir o artigo no editor
- Clicar no botão do Jumi abaixo no editor
- Clicar no ponto onde deseja inserir o código e enter com o código
- Clicar em "encode"
- Colar o código no editor e Salvar

Existem várias outras ferramentas úteis com o Jumi.

### EasyCreator

Download:

[http://joomlancode.org/gf/download/frsrelease/8995/43596/com\\_easycreator\\_0.0.11.zip](http://joomlancode.org/gf/download/frsrelease/8995/43596/com_easycreator_0.0.11.zip)

Esta extensão destina-se a criar um protótipo de extensão para o Joomla 1.5.

Também tem uma outra finalidade interessante, empacotar qualquer extensão instalada de terceiros ou mesmo internas do Joomla.

Muito útil para o caso de não termos o original por perto ou querer ver uma das extensões internas do Joomla.

Após clicar no menu Componentes – EasyCreator

The screenshot shows the EasyCreator Joomla! interface. At the top, there's a 'Project' dropdown menu and a '+ More...' button. The main heading asks 'What do you want to Create today.. =;)' with a 'I USE IT' button. Below, there are sections for 'Registered projects' (showing JoomlaPack) and 'Unregistered Projects' (showing com\_easycreator). A table lists available components like Admin, Authentication, Content, Editors, Site, and Temas Admin, with some marked as 'NONE FOUND'.

Veja que oferece recursos para a criação de Componentes, Módulos, Plugins e Templates.

### Criando um Componente

Clique no Select e selecione New Project para exibir a tela abaixo:

The screenshot shows the 'Extension type' selection screen in EasyCreator. It has a progress indicator '1 / 5' and a title 'Extension type'. Below the title, it says 'Choose a component type from a predefined template'. There are four main columns: Components, Modules, Plugins, and Temas. Each column lists various templates with radio buttons for selection. The 'Components' column is currently selected.

Aqui apenas clicamos no tipo de projeto que desejamos criar, que pode um dos vários tipos. Veja que inclusive podemos criar componentes que usem o padrão MVC, aqui podendo criar cada uma das partes. Para a criação de templates ele apenas oferece recurso de cópia dos existentes.

Vale observar que após iniciar um projeto, após Build podemos clicar em Configure para ter acesso ao editor de código e selecionar um dos arquivos à esquerda para alterar.

Ao final nossa extensão será empacotada e pronta para ser distribuída. O autor lembra que ainda encontra-se em estágio alpha e não faz muito mas já é útil.

## J!Dump

Esta extensão é bastante útil para o desenvolvimento de extensões do Joomla 1.5 via código.

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

Ela conta com algumas funções úteis para depurar o código. A principal é a função `dump()`. Esta função recebe dois parâmetros, uma variável e uma string com o nome da variável.

Exemplo:

```
dump($user, 'usuario');
```

Com isso o `jdump` mostrará o valor da \$variável, mesmo que ela seja um array ou um objeto e ainda mostra detalhes como chave e valor.

### 7.1.3 – Editores de Programação

Embora exista muitos editores especializados em programação é bom escolher um ou dois de nossa preferência. Aqui falarei rapidamente dos meus dois preferidos, um para Linux e outro para Windows.

#### Linux

##### **gEdit**

Para programação no Linux meu editor favorito é o gEdit, mas existem vários outros como o Kate, o vim, emacs, etc.

O gedit acompanha as principais distribuições Linux.

Os principais recursos do gEdit são:

- Leve
- Colorir sintaxe
- Suporte a UTF-8
- Numeração de linhas
- Indentação de várias linhas
  - Selecione várias linhas e pressione Tab que todas receberão uma indentação

#### Windows

##### **PSPad**

Bom editor for Windows com suporte a UTF-8.

Recursos do editor:

- Colorir sintaxe
- Completa funções
- Numeração de linhas
- Gerador de Hashs MD5 e outros
- Seletor de Cores
- Mapa de caracteres
- Gerador de Loren Ipsum
- Conversor de bases numéricas: decimal, hexadecimal, romana, binário, octal, etc
- Verificador de código HTML

Obs.: funciona no Linux através do Wine.

### 7.1.4 – IDEs

As IDEs são softwares com muitos recursos para programação. Algumas são especializadas em apenas uma única linguagem mas outras são modularizáveis, podendo funcionar com várias linguagens, dependendo do módulo usado.

Vou abordar as duas IDEs livres mais populares, que são o NetBeans e o Eclipse. O NetBeans é mais indicado para quem está iniciando, mas algumas vezes vai depender do perfil, portanto experimente estas duas e outras que achar por bem para então eleger a sua preferida.

- NetBeans for PHP
- Eclipse PDT

Ambas as IDEs acima têm versões para Windows e Linux.

#### NetBeans for PHP

Download – <http://netbeans.org>

Instalação típica no Windows.

No Linux apenas dê permissão de execução para instalar.

#### Novo Projeto com Aplicativo Existente

Arquivo – Novo Projeto

Em Categorias clique em PHP

Em Projetos clique em Aplicativo PHP com código-fonte existente e clique em Próximo

Pasta de códigos-fonte – Clique em Procurar e indique a pasta do portal Joomla (htdocs/portal)

Nome do Projeto – Entre um nome para o projeto e Clique em Próximo

Na tela Executar configuração confira e clique em Finalizar.

Agora já pode editar o código aproveitando os recursos do NetBeans.

F6 – Para abrir o projeto no navegador.

#### Recursos interessantes

-Colorir Sintaxe

-Auto-completar código (logo que iniciamos a digitação do nome de uma função existente aparece uma lista com nomes que iniciam com as letras que digitamos.

-Debug (talvez este seja um dos recursos mais úteis, pois ajuda a evitar erros. Logo que

cometemos um erro o mesmo é acusado e geralmente assim que cometemos e no máximo ao teclar ENTER).

-Busca em todos os arquivos de todos os projetos. E a indicação é simultânea com nossa digitação.

Tecla de atalho – CTRL+F

-Criação de testes PHPUnit

-Kenai, CVS, Mercurial e Subversion

-Integração com XDebug

E muito mais

Tutoriais no site do Netbeans - <http://netbeans.org/kb/>

Com PHP - <http://netbeans.org/kb/trails/php.html>

Criando um aplicativo em PHP com MySQL - <http://netbeans.org/kb/61/php/wish-list-tutorial-main-page.html>

## Eclipse PDT

Outra ótima IDE para programação.  
Muitos bons recursos e esta versão é exclusiva para trabalhar com PHP.

**Download** – <http://eclipse.org/pdt>

## Instalação

Apenas instale antes o Java JRE 6.  
Após o download descompacte o arquivo baixado.  
Acesse a pasta e execute o arquivo eclipse.

## Tutoriais

[http://video.google.com.br/videosearch?hl=pt-BR&num=100&newwindow=1&resnum=0&q=eclipse+pdt+tutorial&um=1&ie=UTF-8&ei=vUdKS\\_QSh7W4B5PH1YAC&sa=X&oi=video\\_result\\_group&ct=title&resnum=1&ved=0CA4QqwQwAA#](http://video.google.com.br/videosearch?hl=pt-BR&num=100&newwindow=1&resnum=0&q=eclipse+pdt+tutorial&um=1&ie=UTF-8&ei=vUdKS_QSh7W4B5PH1YAC&sa=X&oi=video_result_group&ct=title&resnum=1&ved=0CA4QqwQwAA#)

[http://2tbsp.com/content/getting\\_started\\_eclipse\\_php\\_development\\_tools\\_%28pdt%29](http://2tbsp.com/content/getting_started_eclipse_php_development_tools_%28pdt%29)

<http://www.vivaolinux.com.br/artigos/impressora.php?codigo=8069>

<http://www.codejacked.com/php-development-with-eclipse-pdt/>



## 7.1.5 – Editores Gráficos

Quem trabalha com desenvolvimento web precisa ter por perto um bom editor de imagens. De preferência um simples e ágil e um segundo mais completo.

Abaixo estou indicando dois que utilizo mas claro que deve usar aquele com o qual você mais se sentir à vontade.

### For Linux

Gimp (Linux e Windows)  
Kolourpaint

### For Windows

PhotoFiltre  
MSPaint

## Gimp

Este é um programa específico para tratamento de imagens, que conta com bons e profissionais recursos.

O Gimp tem versões para Linux e Windoows.

Downloads - <http://www.gimp.org/downloads/>

### Algumas teclas de atalho:

CTRL+N - Abre uma nova imagem  
CTRL+S - Salvar imagem atual  
CTRL+O - Abrir imagem  
CTRL+W - Fechar imagem atual  
CTRL+Z - Desfazer  
CRTL+R - Refazer  
CTRL+A - Selecionar tudo

CTRL+I - Inverter a seleção  
CTRL+D - Duplicar imagem (em nova janela)  
CTRL+L - Abre diálogo de camadas  
CTRL+G - Abre diálogo de gradientes

CTRL+C - copiar seleção  
CTRL+V - colar copiado ou recortado  
CTRL+X - recortar  
SHIFT+CTRL+C - copiar seleção ou imagem atual em nova camada nomeadaa  
SHIFT+CTRL+V - colar copiado ou recortado da camada na memória

SHIFT+CTRL+X - recortar imagem ou seleção da camada atual

CTRL+, - Preencher a seleção com a cor de fundo atual

CTRL+. - Preencher a seleção com a cor de frente atual

### **Outras Teclas de Atalho**

r - seleção retângulo

e - seleção elíptica

o - seleção de cores (conta-gotas)

f - seleção livre (free)

u - seleção contínua

t - seleciona ferramenta texto

m - mover seleção

l - preencher com gradiente

i - seleção com tesoura

b - curva de berzier

z - zoom

SHIFT+M - medir distâncias e ângulos

x - alterna as cores de fundo e frente

q - alinhamento

SHIFT+B - preenchimento

SHIFT+T - redimensionar

SHIFT+R - rotacionar

SHIFT+S - inclinar

SHIFT+D - clarear um pouco

Observe que ao selecionar uma ferramenta da caixa acima, abaixo aparecem suas propriedades.

### **Camadas**

Observe que as camadas são adicionadas no menu Camadas e são monitoradas no painel da direita.

### **Efeitos Especiais**

Os efeitos no Gimp ficam no menu Filtros.

Antes crie alguma imagem ou entre algum texto, então selecione e aplique o filtro.

### **Criando um Simples Gif Animado**

- Crie todas as imagens do mesmo tamanho e do tipo PNG ou GIF e as coloque em um diretório qualquer

- Abra o Gimp e clique em Arquivo – Abrir como camada...

- Selecione todas as imagens, uma-a-uma até a última abrindo todas como camada

- Clique em salvar como

- Escolha salvar como GIF
  - Salvar como animação – Exportar
- Veja a tela abaixo e reproduza algo similar:



Com isso podemos criar um banner animado para o site e qualquer outra arte que a criatividade permitir.

Caso tenha ficado alguma camada em branco, apenas clique sobre ela com o botão direito e a remova salvando novamente a imagem.

### Usando o Gimp para criar um Logo Simples para o site

Iremos criar um logo que contém uma frase com Filtro Cromo

- Abrir o Gimp (Aplicativos - Graficos - Editor de Imagens Gimp)
- Arquivo - Nova
- Entre com a largura e a altura do logo (Ex.: 800x100) e OK
- Clique na ferramenta Texto (A maiúsculo na caixa de ferramentas)
- Clique no botão Fonte e selecione a fonte
- Clique na caixa Tamanho e digite ou selecione o tamanho
- Clique no botão Cor e selecione a cor da fonte
- Clique na área do logo e digite na caixa do editor de texto a frase do Logo

- Clique no menu Filtros - Alfa para logo e Cromar...
- Clique no botão cor de fundo e escolha uma cor que combine com a cor de frente e tecla OK
- Veja se está do seu gosto e repita os passos até gostar.  
Então salve.

**Bons tutoriais sobre o Gimp podem ser encontrados na Internet.**

<http://www.gimp.org/tutorials/>

[http://video.google.com.br/videosearch?hl=pt-BR&source=hp&q=gimp+tutorial&um=1&ie=UTF-8&ei=DjVKS5DGOtCLuAebufyIAg&sa=X&oi=video\\_result\\_group&ct=title&resnum=4&ved=0CCIQqwQwAw#](http://video.google.com.br/videosearch?hl=pt-BR&source=hp&q=gimp+tutorial&um=1&ie=UTF-8&ei=DjVKS5DGOtCLuAebufyIAg&sa=X&oi=video_result_group&ct=title&resnum=4&ved=0CCIQqwQwAw#)

<http://gimp-tutorials.net/>

## Kolourpaint

O Kolourpaint é um bom editor gráfico simples para Linux.

Como o nome sugere é semelhante ao MSPaint do Windows, mas conta com bons recursos que o paint não tem.

### Transparência

Geralmente imagens a serem colocadas sobre outras precisam ter fundo transparente. Para realizar este efeito em alguns softwares dá algum trabalho mas no Kolourpaint é simples.

Apenas selecione a área a ser transparente, clique na ferramenta transparente (pirâmide azul abaixo e à esquerda) então clique no balde de preenchimento e clique na área.

### Salvar Seleção como Arquivo

Um ótimo recurso existente no Paint.

Selecione a região que deseja transformar em arquivo  
Clique em Edit – Copy to file

### Capturando uma cor em Hexadecimal

Efetue um duplo clique em uma das cores abaixo para abrir a paleta.  
Veja que aí podemos entrar com cores usando HSV, RGB ou Hexa.

### Alguns Efeitos

Para acessar alguns poucos efeitos existentes  
Selecione uma região  
Clique no menu Selection – More Effects

Existem vários outros bons recursos.

### Tutoriais na Internet

<http://brmaker.com.br/forum/index.php?showtopic=4477>

<http://www.google.com.br/url?sa=t&source=web&ct=res&cd=4&ved=0CByQFjAD&url=http%3A%2F%2Fwebeduc.mec.gov.br%2Flinuxeducacional%2Fmanuais%2Fkolourpaint.pdf&rct=j&q=kolourpaint+tutorial&ei=0TVKS4fllYaOuAfyqb2MAg&usg=AFQjCNEAJm6la6bPXmd6EfOr-JVhpPWeRQ&sig2=Q-moRbXT-5Ogk24YhBJBwQ>

## PhotoScape

Uma boa opção para editar imagens no Windows.

Download - <http://www.photoscape.org/ps/main/download.php>

### Criando um Gif Animado

- Insert : Add a photo
- Ctrl-V : Add a clipboard photo
- Del : Delete the selected photo at the list
- Ctrl-Del : Delete all photos at the list
- Ctrl-Left : Move forward selected photos
- Ctrl-Right : Move backward selected photos
- Home,End,Left,Right : Change selection
- Ctrl-A : Select all frames
- Ctrl-C : Copy selected photo to clipboard
- Ctrl-S : Save your animated GIF file

Tutoriais:

<http://www.photoscape.org/ps/main/help.php>

Veja aqui os 11 melhores Editores Gráficos Free:

<http://graphicssoft.about.com/od/pixelbasedwin/tp/freephotoedw.htm>

## MS Paint

Este é muito simples de usar mas tem alguns recursos poucos pouco conhecidos:

Criar arquivo com seleção – veja no Kolourpaint.

Mover o cursor com as teclas.

Criar quadrados e círculos segurando a tecla SHIFT.

Tutoriais:

<http://www.lkwdpl.org/classes/MSPaint/paint.html>

<http://www.celsojunior.net/blog/2006/10/29/tutorial-desenhando-um-carro-no-ms-paint/>  
Espetacular

## 7.2 – Framework

Framework é a biblioteca do Joomla, que fica no diretório `libraries/joomla`. Acontece que um framework é bem mais que apenas uma biblioteca. Existem vários acordos definidos no framework, que sendo usados pelo programador que cria extensões para o Joomla acarreta várias vantagens em termos de segurança e de estabilidade, além de estar bem mais adequado para atualizar para as novas versões do Joomla que apareçam.

Todas as bibliotecas de terceiros usadas pelo Joomla encontram-se na pasta `libraries`. As específicas do Joomla estão na pasta `libraries/joomla`.

### 7.2.1 - Boa Prática

1 - Adicionar um arquivo `index.html` com apenas esta linha:

`index.html`

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

Isso evita o acesso direto ao diretório. Caso alguém acesse o diretório de um componente com este arquivo, apenas verá a tela em branco.

No caso este arquivo deve ser copiado para todas as pastas da extensão.

2 - Criar arquivo `leiam.txt` para cada extensão, contendo a função e como usar.

3 – Cada script deve conter uma seção de comentário inicial, contendo data, autor, função, nome, classe, etc do tipo do phpdoc para gerar documentação automática.

4 – Separar posições de formatação no CSS, sendo um arquivo para formatação (`template.css`) e outro para as posições (`posicoes.css` ou `positions.css`).

5 – Ao criar uma extensão, sempre tenha as pastas `site`, `admin` e `sql` quando for o caso. Na pasta `site` devem ficar os arquivos que serão destinados ao site, na `admin` para o administrador, na `sql` os scripts de instalação e desinstalação de tabelas e registros.

## 7.2.2. - Convenções para Nomes

**Nome de Form** – O nome do form do script “admin.nomex.html.php” deve ser **frmAdmin**. Isso para tirar o melhor proveito do framework do Joomla, como funções em JavaScript.

**Classes** – Iniciar com maiúsculas e usar o padrão CamelCase

**Métodos** – Iniciar com minúsculas e usar camelCase

Todos os elementos nomeados devem usar camelCase

Elementos não públicos devem iniciar com um sublinhado (privates e protected).

### Tabelas

#### Devem usar o prefixo

Iniciar com o nome da extensão a que pertence. Ex.: #\_\_minhaExtensao\_itens

Todos os nomes de campos devem ser em minúsculas e palavras compostas separadas por sublinhado. Exs.: email e email\_secundario.

Para uma maior eficiência da sua extensão no Joomla use os campos primary key nomeados com "id" e integer, auto\_increment e not null.

Nós devemos usar alguns campos comuns em nossas tabelas, para tirar vantagens do Framework do Joomla. Esses campos são usados na JTable: published, hits, checked\_out, checked\_out\_time, ordering,

Idealmente, ao criar a tabela, criar com suporte a UTF-8.

Execute essa consulta SQL para ver se a sua versão do MySQL tem suporte a UTF-8:

```
SHOW CHARACTER SET;
```

Caso apareça:

```
utf8          UTF-8 Unicode          utf8_general_ci
```

poderá criar as tabelas usando o collation utf8\_general\_ci.

Caso queira usar uma consulta SQL para criar seu banco com suporte a UTF-8:

```
CREATE DATABASE db_name CHARACTER SET = utf8 COLLATE = utf8_general_ci
```

**Padrão Factory** – é usado para construir e reoprnar objetos.

**Padrão Singleton** – é usado para permitir a criação de um simples objeto de uma classe específica.



**Indentação e comprimento de linhas**

Usar tabulação para indentar ao invés de espaços. A parada de tabulação deve conter 4 espaços.

Não deve existir limite para o comprimento das linhas.

**Comentários**

Os comentários devem seguir o padrão do phpdoc.

Todos os arquivos fontes devem vir com um bloco de comentário inicial contendo informações sobre o mesmo.

Assim como classes, métodos, constantes, propriedades de classes e métodos.

**Includes**

Ao incluir arquivo com condição (if) use `include_once`.

Quando incluir incondicionalmente use `require_once`.

**Tags do PHP**

Use as tags:

```
<?php
```

```
?>
```

**SQL**

Use as palavras chaves do SQL em maiúsculas e todos os identificadores em minúsculas.

**Consultas delimitadas com apóstrofes (aspas )**

Todos os campos tipo string das consultas precisam usar o método `quote()` para facilitar futura compatibilidade com outros SGBDs.

Todos os nomes de tabelas devem iniciar com `#_` ao contrário de `jos_` para acessar o conteúdo do Joomla e permitir que o usuário possa definir outros prefixos para serem aplicados.

Todas as variáveis tipo inteiro ou de ponto flutuante precisam sofrer cast com `(int)`, `(float)` ou `(double)` como for adequado.

**Nomes de classes**

Devem ter nomes bem descritivos de suas funções.

Devem iniciar com maiúsculas e usar CamelCase. Algumas podem usar tudo maiúsculo

como HTML e XML.

Todas as classes do framework do Joomla precisam iniciar com J maiúsculo com a próxima letra também maiúscula, como:

JHtmlHelper, JXmlParser e JModel

### **Nomes de Métodos e Funções**

Precisam seguir o método camel caps. A inicial deve ser minúscula e cada palavra seguinte com inicial maiúscula:

connect(), buildSomer(), JDoAlgo()

### **Membros privados de classes**

Iniciados com um simples sublinhado:

```
_membropriv = 3;
```

### **Constantes**

Todas em maiúsculas, separando as palavras por sublinhado.

Nomes de constantes usam como prefixo o nome da classe a que pertence:

As constantes da classe "JError" devem iniciar com JERROR\_

### **Variáveis Globais**

Nomes devem iniciar com sublinhado seguido por maiúsculas do nome da classe/package e outro sublinhado. Exemplo:

```
#_JERROR_LEVELS
```

### **Variáveis de Classe e variáveis regulares**

Seguem as mesmas convenções das funções.

### **Referência**

Deve haver um espaço entre o sinal de igual e o de referência e nenhum espaço entre este e a função ou variável.

```
$ref1 = &$this->_sql;
```

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

### **Controllers**

[Nome]Controller

```
class NomeController extends JController
```

```
class PrimeiroNomeController extends JController
```

Para múltiplos controllers:

```
[Component]Controller[Nome]
```

```
class BannerControllerClient extends JController
```

### **Models**

Usar [Component]Model[Nome]

```
class BannerModelClient extends JModel
```

```
com_banner
    /models/
    /banner.php
    /client.php
```

### **Views**

Usar [Component]View[Nome]

```
class ContactViewCategory extends JView
```

```
com_contact
    /views/
        /view1/
            /view.html.php
        /view2/
            /view.html.php
```

```
class ContentView extends JView
class ContentViewArchive extends JView
```

### **Plugins**

Usar [Diretório]Plugin[Elemento]

```
class ContentPluginPagebreak extends JPlugin
```

### **Layouts**

Componentes devem suportar diferentes layouts para renderizar os dados supridos pelas views e seus models.

Um arquivo de layout usualmente contem HTML e algum código PHP, somente a lógica

para exibir. Nenhuma função e nenhuma classe.

O layout consiste pelo menos um arquivo php e um arquivo de manifesto com mesmo nome e extensão XML localizado na pasta /tmpl da views, ambos refletem o nome interno do layout. O layout padrão chama-se default.

```
com_content
  /views
    /article/
      /tmpl/
        /default.php
        /form.php
        /form.xml
        /pagebreak.php
        /pagebreak.xml
```

O layout usa arquivos php suplementares para oferecer mais controle granular para renderizar partes individuais ou repetitivos itens de dados.

Usuários podem personalizar a saída do layout através do Template Layout Override

Referência: [http://docs.joomla.org/Coding\\_style\\_and\\_standards](http://docs.joomla.org/Coding_style_and_standards)

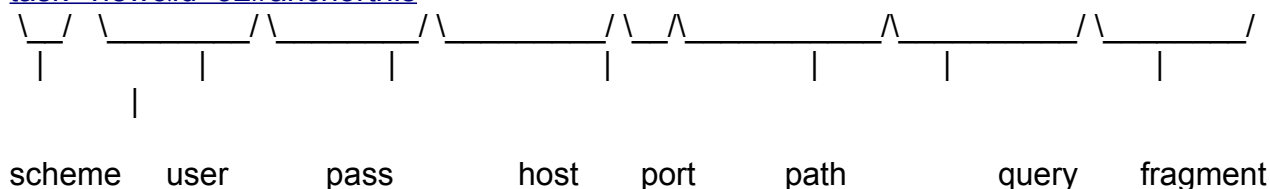
### 7.2.3 - Estrutura da URI do Joomla

[http://localhost/portal/index.php?option=com\\_exemplo](http://localhost/portal/index.php?option=com_exemplo)

Chamando o item1 do menu:

[http://localhost/portal/index.php?option=com\\_exemplo&Itemid=1](http://localhost/portal/index.php?option=com_exemplo&Itemid=1)

<http://fredbloggs:itsasecret@www.example.com:8080/path/to/Joomla/index.php?task=view&id=32#anchorthis>



Get method	Set method	Description	Example
<a href="#">getFragme nt</a>	<a href="#">setFragme nt</a>	Fragment (everything after the '#'). This is often referred to as an anchor.	anchorthis
<a href="#">getHost</a>	<a href="#">setHost</a>	Hostname or IP address. For example, 'www.joomla.org' or '192.168.2.45'.	www.example.com
<a href="#">getPass</a>	<a href="#">setPass</a>	Password part of the authority.	itsasecret
<a href="#">getPath</a>	<a href="#">setPath</a>	Path string. Note that the path always includes the leading "/" character.	/path/to/Joomla/index.php
<a href="#">getPort</a>	<a href="#">setPort</a>	Port number. Specific schemes (protocols) have their own defaults (for example, 'http' is 8080 port 80, 'ftp' is port 21).	
<a href="#">getQuery</a>	<a href="#">setQuery</a>	Query in string format. For example, "foo=bar&x=y". See also <a href="#">buildQuery</a> .	task=view&id=32
<a href="#">getScheme</a>	<a href="#">setScheme</a>	Scheme (protocol). For example, 'http', 'https', 'ftp'.	http
<a href="#">getUser</a>	<a href="#">setUser</a>	Username part of the authority.	fredbloggs
<a href="#">getVar</a>	<a href="#">setVar</a>	An individual query item value from within the query part. See also <a href="#">delVar</a> .	32

### Outros Métodos

Met. name	Description
<a href="#">base</a>	Returns the base URI of the request, regardless of the URI object.
<a href="#">buildQuery</a>	Build the query part of a URI from an array of query items.
<a href="#">current</a>	Returns the current URI of the request without the query part, regardless of the URI object.
<a href="#">delVar</a>	Removes an individual query item. See also <a href="#">getVar</a> and <a href="#">setVar</a> .

<a href="#">getInstance</a>	Returns a reference to the global JURI object, only creating it if it doesn't already exist. The global URI object represents the URI that caused the current page to be rendered.
<a href="#">isInternal</a>	Checks whether the URI is internal; that is, on the same host as Joomla.
<a href="#">isSSL</a>	Checks whether the URI is using the 'https' scheme (protocol).
<a href="#">parse</a>	Parse the URI and populate the class fields.
<a href="#">root</a>	Returns the root URI of the request, regardless of the URI object.
<a href="#">toString</a>	Returns the full URI represented by the URI object in string format.

**Exibindo conteúdo de páginas em formatos diferentes:**

[http://localhost/portal/index.php?option=com\\_exemplo&format=pdf](http://localhost/portal/index.php?option=com_exemplo&format=pdf)

Outro valor comun de consulta é o "task". Para mostrar a task view:

[http://localhost/portal/index.php?option=com\\_exemplo&task=view](http://localhost/portal/index.php?option=com_exemplo&task=view)

### 7.2.4 - Application Message Queue

- message,
- notice e
- error

```
$mainframe->enqueueMessage('Uma mensagem do tipo message');  
$mainframe->enqueueMessage('Uma mensagem do tipo notice', 'notice');  
$mainframe->enqueueMessage('Uma mensagem do tipo error', 'error');
```

Para notice ou error os mais usados são:

```
JError::raiseNotice()  
JError::raiseWarning()
```

```
$css = '/* Bespoke Error Messages */  
#system-message dt.bespoke  
{  
display: none;  
}  
dl#system-message dd.bespoke ul  
{  
color: #30A427;  
border-top: 3px solid #94CA8D;  
border-bottom: 3px solid #94CA8D;  
background: #C8DEC7 url(notice-bespoke.png) 4px 4px no-repeat;  
}';  
$doc =& JFactory::getDocument();  
$doc->addStyleDeclaration($css);
```

## Message

### Message

```
/** PHP 4 */  
$app = JFactory::getApplication()  
$app->enqueueMessage( 'Message' );  
  
/** Alternatively, in PHP 5 */  
JFactory::getApplication()->enqueueMessage( 'Message' );
```

## Notice

### Notice

```
JError::raiseNotice( 100, 'Notice' );
```

## Warning

---

### Warning

---

```
JError::raiseWarning( 100, 'Warning' );
```

## Error

```
JError::raiseError( 4711, 'A severe error occurred' );
```

Retrieved from "[http://docs.joomla.org/Display\\_error\\_messages\\_and\\_notices](http://docs.joomla.org/Display_error_messages_and_notices)"



### **7.2.5 - Redirecionamento (exemplos)**

```
$mainframe->redirect('index.php?option=com_example');
```

```
$mainframe->redirect('index.php?option=com_example', 'Alguma Mensagem', 'aviso');
```

```
$this->setRedirect('index.php?option=com_example');
```

```
$this->setRedirect('index.php?option=com_example', 'Some Message', 'notice');
```

```
$controller = new ExampleController();  
$controller->execute(JRequest::getCmd('task'));  
$controller->redirect();
```

## 7.2.6 - Metadatas

```
$keywords = 'macaco, gorila, chipansé, orangotango';  
$document->setMetaData('keywords', $keywords);  
  
$document->setMetaData('MSTHEMECOMPATIBLE', 'no', true);  
  
$keywords = explode(',', $document->getMetaData('keywords'));  
$keywords[] = 'append me';  
$keywords[] = 'and me';  
$document->setMetaData('keywords', implode(',', $keywords));
```

### 7.2.7 - Texto de Dicas - Tooltips

```
echo JHTML::_('tooltip', $stooltip);

echo JHTML::_('tooltip', $stooltip, $title);

echo JHTML::_('tooltip', $stooltip, $title, 'warning.png');

echo JHTML::_('tooltip', $stooltip, $title, null, $text);

// prepare o CSS
$css = /* Tooltips */
.tool-tip
{
min-width: 100px;
opacity: 0.8;
filter: alpha(opacity=80);
-moz-opacity: 0.8;
}
.tool-title
{
text-align: center;
}
.tool-text {
font-style: italic;
};
// add the CSS to the document
$doc =& JFactory::getDocument();
$doc->addStyleDeclaration($css);
```

## 7.2.8 - Constantes Pré-definidas

Name	Description
DS	Directory Separator (/)
JPATH_ADMINISTRATOR (/joomla/administrator)	Administrator path
JPATH_BASE	Path to the entry directory (/joomla)
JPATH_CACHE	Cache path (/joomla/cache)
JPATH_COMPONENT  (/joomla/components/com_example)	Component path
JPATH_COMPONENT_ADMINISTRATOR (/joomla/administrator/components/com_example)	Component backend path
JPATH_COMPONENT_SITE	Component frontend path (/joomla/components/com_example)
JPATH_CONFIGURATION	Configuration path (/joomla)
JPATH_INSTALLATION	Installation path (/joomla/installation)
JPATH_LIBRARIES	Libraries path (/joomla/libraries)
JPATH_PLUGINS	Plugins path (/joomla/plugins)
JPATH_ROOT (/joomla)	Path to the frontend entry directory
JPATH_SITE	Path to the public directory (/joomla)
JPATH_THEMES	Templates path (/joomla/templates)

### Datas

Name	Default Format	Example
DATE_FORMAT_LC	%A, %d %B %Y	Sunday, 23 June 1912
DATE_FORMAT_LC2	%A, %d %B %Y %H:%M	Sunday, 23 June 1912 00:00
DATE_FORMAT_LC3	%d %B %Y	23 June 1912
DATE_FORMAT_LC4	%d.%m.%y	23.06.12

### Tradução

```
echo JText::_('MONTH');
$value = JText::sprintf('SAVED_ITEMS', 3);
```

### Acesso negado:

```
JError::raiseError( 403, 'Access Denied' );
```

## 7.2.9 - Trabalhando com Arquivos, Diretórios e Permissões

Se quisermos trabalhar com arquivos no Joomla podemos usar diretamente as funções do PHP para isso mas existem boas vantagens usar as funções das classes estáticas JFile, JFolder e FPath, que fazem parte do Framework do Joomla.

Para trabalhar com arquivos e diretórios devemos adicionar um ou mais dos imports:

```
jimport('joomla.filesystem.file');  
  
jimport('joomla.filesystem.folder');  
  
jimport('joomla.filesystem.path');
```

### Aumentando a Segurança

Caso tenha necessidade de aumentar a segurança em relação a arquivos pode armazená-los no banco de dados ao invés do sistema de arquivos. Isso previne acesso direto ao sistema de arquivos pelo servidor web. Lembre de armazenar o nome original do arquivo e também o MIME type.

### Testar de Arquivo ou Pasta Existe

```
$arquivoExiste = JFile::exists($arquivoPath);
```

```
$pastaExiste = JFolder::exists($pastaPath);
```

O retorno será um booleano.

### Lendo o conteúdo de um Arquivo

```
$arquivoPath = JPATH_COMPONENT.DS.'files'.DS.'meuarquivo.csv';  
$conteudoArquivo = Jfile::read($arquivooPath);
```

### Excluindo Arquivo ou Pasta

```
$arquivvoPath = JPATH_COMPONENT . DS . 'excluame.txt';  
$arquivoExcluido = JFile::delete($arquivoPath);
```

```
$pastaPath = JPATH_COMPONENT . DS . 'excluame';  
$pastaExcluida = JFolder::delete($pastaPath);
```

### Copiando um arquivo ou pasta

```
// preparar paths  
$arquivoPath = JPATH_COMPONENT . DS . 'copieme.txt';  
$arquivoCopiaPath = JPATH_COMPONENT . DS . 'novacopia.txt';  
  
// tentando copiar arquivo  
$arquivoCopiado = JFile::copy($arquivoPath, $arquivoCoiaPath);
```

O método estático `JFile::copy()` sobrescreve arquivos existentes.

```
$pastaPath = JPATH_COMPONENT . DS . 'copieme';  
$pastaCopiadaPath = JPATH_COMPONENT . DS . 'novacopia';  
  
$pastaCopiada = JFolder::copy($pastaPath, $pastaCopiadaPath);
```

Os métodos estáticos `JFile::copy()` e `JFolder::copy()` não sobrescrevem pastas existentes.

### **Mover e Renomear Arquivos e Pastas**

```
$arquivoPath = JPATH_COMPONENT . DS . 'movame.txt';  
$arquivoMovidoPath = JPATH_COMPONENT . DS . 'aqui.txt';  
  
$arquivoMovido = JFile::move($arquivoPath, $arquivoMovidoPath);  
  
$pastaPath = JPATH_COMPONENT . DS . 'movame';  
$pastaMovidaPath = JPATH_COMPONENT . DS . 'movameaqui';  
  
$pastaMovida = JFolder::move($pastaPath, $pastaMovidaPath);
```

### **Criando uma Pasta**

```
$pastaPath = JPATH_COMPONENT . DS . 'minhapasta';  
$pastaCriada = JFolder::create($pastaPath);
```

### **Criando uma sub Pasta**

```
$pastaPath = JPATH_COMPONENT . DS . 'pastaprincipal' . DS . 'subpasta';  
$pastasCriadas = JFolder::create($pastaPath);
```

### **Lendo uma Estrutura de Diretórios**

```
$pastas = JFolder::listFolderTree($path, '.');
```

É uma boa prática sempre verificar se o path para o qual estamos apontando existe.

## Alterando Permissões de Arquivos e Pastas

```
jimport('joomla.filesystem.path');
```

### **Recebendo as permissões atuais de um arquivo:**

```
$permissoes = Jpath::getPermissions($path);
```

### **Checar se arquivos têm permissão de escrita:**

```
if(JPath::canChmod($path)) {
    // aqui efetuar as alterações
}
```

### **Alterando permissões:**

```
JPath::setPermissions($path, '0644', '0755');
```

### **Alterando somente as permissões de uma pasta:**

```
JPath::setPermissions($path, null, '0755');
```

### **Alterando somente permissões de arquivos:**

```
JPath::setPermissions($path, '0644', null);
```

## Efetuando Upload de arquivos

```
JFile::upload($src, $dest);
```

Este método adiciona às funções originais do PHP uma checagem para ver se o arquivo existe e as permissões de origem e destino.

Exemplo prático:

```
<form name="upload" method="post" enctype="multipart/form-data">
<input type="file" name="file_upload" />
<input type="submit" />
</form>
```

Upload code looks like this:

```
<?php
*****
*           *
* File upload example *
*           *
*****
//Retrieve file details from uploaded file, sent from upload form
$file = JRequest::getVar('file_upload', null, 'files', 'array');
```

```
//Import filesystem libraries. Perhaps not necessary, but does not hurt
jimport('joomla.filesystem.file');

//Clean up filename to get rid of strange characters like spaces etc
$filename = JFile::makeSafe($file['name']);

//Set up the source and destination of the file
$src = $file['tmp_name'];
$dest = JPATH_COMPONENT . DS . "uploads" . DS . $filename;

//First check if the file has the right extension, we need jpg only
if ( strtolower(JFile::getExt($filename) ) == 'jpg' ) {
    if ( JFile::upload($src, $dest) ) {
        //Redirect to a page of your choice
    } else {
        //Redirect and throw an error message
    }
} else {
    //Redirect and notify user file is not right extension
}
?>
```

Adicionando opções de upload a componentes:

[http://docs.joomla.org/Creating\\_a\\_file\\_uploader\\_in\\_your\\_component](http://docs.joomla.org/Creating_a_file_uploader_in_your_component)

### **Criando uma árvore como lista de uma estrutura de pastas**

```
JFolder::listFolderTree($path, $filter, $maxLevel = 3, $level = 0, $parent = 0);
```

### **Recebendo a extensão de um arquivo:**

```
$ext = Jfile::getExt($filename);
```

### **Recebendo nome de arquivo sem a extensão:**

```
$name = JFile::stripExt($filename);
```

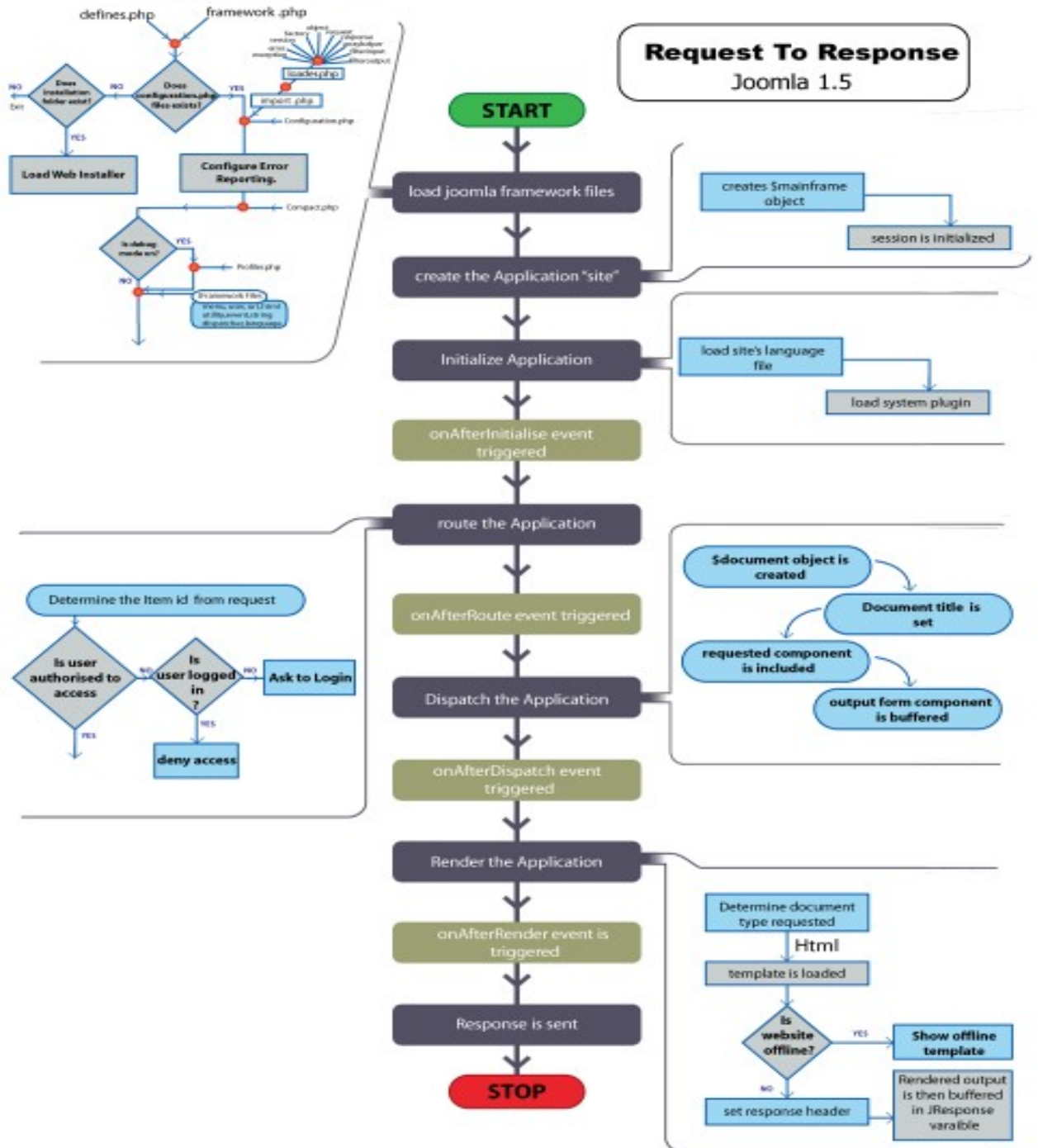
Original em: "[http://docs.joomla.org/How\\_to\\_use\\_the\\_filesystem\\_package](http://docs.joomla.org/How_to_use_the_filesystem_package)"

Referência: capítulo 12 do livro Joomla! 1.5 Development Cookbook do James Kennard, editora Packt Publishing.



### 7.2.10 - Fluxo Request/Response

Pune IT Labs



## 7.2.11 - Segurança na Programação

### Mantendo Extensões Seguras

#### Tornando o nome de um arquivo seguro

```
jimport('joomla.filesystem.file');  
$safeFilename = JFile::makeSafe($unsafeFilename);
```

#### Tornando seguro um diretório

```
jimport('joomla.filesystem.folder');  
$safeDirPath = JFolder::makeSafe($unsafeDirPath);
```

#### Tornando seguro um path

```
$myPath = JRequest::getVar('myPath', 'default', 'REQUEST', 'PATH');  
jimport('joomla.filesystem.path');  
$myPath = JPath::clean($myPath);  
JPath::check($myPath);
```

### Servidores de Hospedagem

Caso queira uma maior segurança para o site não use servidores compartilhados mas ao contrário use servidores dedicados, onde terá maior controle.

### Configurando o PHP (php.ini)

```
register_globals = off  
magic_quotes_gpc = on  
allow_url_fopen = off  
expose_php = off  
safe_mode = off  
ServerSignature = Off
```

```
/**
```

```
* Use 1 to emulate register_globals = on
```

```
*
```

```

* Use 0 to emulate register_globals = off [sic]
*/
define( 'RG_EMULATION', 1 );
/**
* Adds an array to the GLOBALS array and checks that the GLOBALS
variable is
* not being attacked
* @param array
* @param boolean True if the array is to be added to the GLOBALS
*/

```

## **.htaccess**

### **Recurso importante para segurança**

```

##### Begin - Rewrite rules to block out some common exploits
## If you experience problems on your site block out the operations listed
below
## This attempts to block the most common type of exploit `attempts` to
Joomla!
#
#IF the URI contains a "http:" or "ftp:" or "https"
RewriteCond %{QUERY_STRING} http\: [OR]
RewriteCond %{QUERY_STRING} ftp\: [OR]
RewriteCond %{QUERY_STRING} https\: [OR]
#OR if the URI contains a "["
RewriteCond %{QUERY_STRING} \[ [OR]
#OR if the URI contains a "]"
RewriteCond %{QUERY_STRING} \] [OR]
# Block out any script trying to set a mosConfig value through the URL
RewriteCond %{QUERY_STRING} mosConfig_[a-zA-Z]{1,21}(=|\%3D) [OR]
# Block out any script trying to base64_encode crap to send via URL
RewriteCond %{QUERY_STRING} base64_encode.*\(.*\) [OR]
# Block out any script that includes a <script> tag in URL
RewriteCond %{QUERY_STRING} (\<|\%3C).*script.*(\>|\%3E) [NC,OR]
# Block out any script trying to set a PHP GLOBALS variable via URL
RewriteCond %{QUERY_STRING} GLOBALS(=|\[|\%[0-9A-Z]{0,2}) [OR]
# Block out any script trying to modify a _REQUEST variable via URL
RewriteCond %{QUERY_STRING} _REQUEST(=|\[|\%[0-9A-Z]{0,2})
# Send all blocked request to homepage with 403 Forbidden error!

```

```
RewriteRule ^(.*)$ index.php [F,L]
#
##### End - Rewrite rules to block out some common exploits
```

### **Permissões de arquivos:**

```
.htaccess.....644
configuration.php... 644
Diretórios .....755
Arquivos..... 644
```

### **Organize-se em relação ao backup**

Uma ótima alternativa é o uso do JoomlaBack

### **Proteja seu computador local:**

- Use um bom antivírus
- Antispy
- Firewall
- Proteja suas senhas
- Evite spam
- Atente para uma boa manutenção do hardware e evite tomar café próximo ao teclado ou outros líquidos e sujantes
- Use uma senha forte para sua internet Wireless

### **Referência:**

Livro Joomla! Web Security do Tom Canavan - Editora Packt Publishing

### **Cuidado com a instalação de Extensões de Terceiros**

Sempre verifique a lista de suspeitos e perigosos:

[http://docs.joomla.org/Vulnerable\\_Extensions\\_List](http://docs.joomla.org/Vulnerable_Extensions_List)

**Ao concluir o site faça testes para ver como está a segurança.**

**Fique de olho nas atualizações** e as execute sempre que aparecerem. Uma boa idéia é assinar o RSS do Joomla ou cadastrar seu e-mail na página de downloads (<http://www.joomla.org/download.html>) para maior agilidade.

**Use senhas fortes** para a seção de administração do Joomla. Use um sistema operacional estável como o Linux e esteja sempre atualizado. Tome cuidado redobrado com entradas de usuários em formulários ou outras.

## Injeção SQL

Para receber ID de usuários do MySQL, veja o mysql.user:

```
http://[host]/activate.php?  
userName='/**/union/**/select/**/1,2,3,4,concat(user,0x203a3a20,password),  
6,7,8,9,9,9,9,9/**/from/**/mysql.user/*
```

O exploit acima tira vantagem da seguinte vulnerabilidade:

```
$userName = $_GET["userName"];  
$code = $_GET["activate"];  
$sql = "SELECT activated FROM users WHERE username = '$userName'  
AND activated = '$code'";
```

Ver: <http://developer.joomla.org/tutorials/181-preventing-sql-injections.html>

Caso magic\_quotes esteja como OFF este exploit derrubará seu sistema .

Cuidado com as configurações do PHP e com a elaboração das consultas SQL.

Oculte as mensagens de erros para evitar que informações caiam nas mãos dos curiosos:

```
# suppress php errors  
php_flag display_startup_errors off
```

```
php_flag display_errors off
php_flag html_errors off
php_value docref_root 0
php_value docref_ext 0
```

### **Prevenindo visualização de conteúdo de diretórios**

```
# Disable directory browsing
Options All -Indexes
```

### **Ocultando algumas extensões**

```
# prevent display of select file types
IndexIgnore *.wmv *.mp4 *.avi *.etc
```

### **Sites com ferramentas para mostrar vulnerabilidades, exploits e notícias sobre segurança:**

[www.secunia.org](http://www.secunia.org)  
[www.us-cert.gov](http://www.us-cert.gov)  
[www.milw0rm.com](http://www.milw0rm.com)  
[www.nist.gov](http://www.nist.gov)  
[www.sans.org](http://www.sans.org)  
[www.frsirt.com](http://www.frsirt.com)  
[www.joomla.org](http://www.joomla.org)  
[www.redhat.org/apps/support](http://www.redhat.org/apps/support)  
[www.freebsd.org/security](http://www.freebsd.org/security)  
[www.microsoft.com/technet/security/notify.asp](http://www.microsoft.com/technet/security/notify.asp)  
[www.openbsd.org/security](http://www.openbsd.org/security)  
[www.debian.org/security](http://www.debian.org/security)  
<http://sunsolve.sun.com/pub-cgi/secBulletin.pl>  
<http://osvdb.org/>

Elabora uma lista de itens sobre segurança que deverá averiguar com frequência.

Portas usadas por Backdoor Tools

(Source: [www.garykessler.net/library/bad\\_ports.html](http://www.garykessler.net/library/bad_ports.html))

## Códigos de Status do Apache

### 1xx Info / Informational

100 Continue  
HTTP\_CONTINUE  
101 Switching Protocols  
HTTP\_SWITCHING\_PROTOCOLS  
102 Processing  
HTTP\_PROCESSING

### 2xx Success / OK

200 OK  
HTTP\_OK  
201 Created  
HTTP\_CREATED  
202 Accepted  
HTTP\_ACCEPTED  
203 Non-Authoritative Information  
HTTP\_NON\_AUTHORITATIVE  
204 No Content  
HTTP\_NO\_CONTENT  
205 Reset Content  
HTTP\_RESET\_CONTENT  
206 Partial Content  
HTTP\_PARTIAL\_CONTENT  
207 Multi-Status  
HTTP\_MULTI\_STATUS

### 3xx Redirect

300 Multiple Choices  
HTTP\_MULTIPLE\_CHOICES  
301 Moved Permanently  
HTTP\_MOVED\_PERMANENTLY  
302 Found  
HTTP\_MOVED\_TEMPORARILY  
303 See Other  
HTTP\_SEE\_OTHER  
304 Not Modified

HTTP\_NOT\_MODIFIED

305 Use Proxy

HTTP\_USE\_PROXY

307 Temporary Redirect

HTTP\_TEMPORARY\_REDIRECT

4xx Client Error

400 Bad Request

HTTP\_BAD\_REQUEST

401 Authorization Required

HTTP\_UNAUTHORIZED

402 Payment Required

HTTP\_PAYMENT\_REQUIRED

403 Forbidden

HTTP\_FORBIDDEN

404 Not Found

HTTP\_NOT\_FOUND

405 Method Not Allowed

HTTP\_METHOD\_NOT\_ALLOWED

406 Not Acceptable

HTTP\_NOT\_ACCEPTABLE

407 Proxy Authentication Required

HTTP\_PROXY\_AUTHENTICATION\_REQUIRED

408 Request Time-out

HTTP\_REQUEST\_TIME\_OUT

409 Conflict

HTTP\_CONFLICT

410 Gone

HTTP\_GONE

411 Length Required

HTTP\_LENGTH\_REQUIRED

412 Precondition Failed

HTTP\_PRECONDITION\_FAILED

413 Request Entity Too Large

HTTP\_REQUEST\_ENTITY\_TOO\_LARGE

414 Request-URI Too Large

HTTP\_REQUEST\_URI\_TOO\_LARGE

415 Unsupported Media Type

HTTP\_UNSUPPORTED\_MEDIA\_TYPE

416 Requested Range Not Satisfiable



HTTP\_RANGE\_NOT\_SATISFIABLE

417 Expectation Failed

HTTP\_EXPECTATION\_FAILED

422 Unprocessable Entity

HTTP\_UNPROCESSABLE\_ENTITY

423 Locked

HTTP\_LOCKED

424 Failed Dependency

HTTP\_FAILED\_DEPENDENCY

425 No code

HTTP\_NO\_CODE

426 Upgrade Required

HTTP\_UPGRADE\_REQUIRED

5xx Server Error

500 Internal Server Error

HTTP\_INTERNAL\_SERVER\_ERROR

501 Method Not Implemented

HTTP\_NOT\_IMPLEMENTED

502 Bad Gateway

HTTP\_BAD\_GATEWAY

503 Service Temporarily Unavailable

HTTP\_SERVICE\_UNAVAILABLE

504 Gateway Time-out

HTTP\_GATEWAY\_TIME\_OUT

505 HTTP Version Not Supported

HTTP\_VERSION\_NOT\_SUPPORTED

506 Variant Also Negotiates

HTTP\_VARIANT\_ALSO\_NEGOTIATES

507 Insufficient Storage

HTTP\_INSUFFICIENT\_STORAGE

510 Not Extended

HTTP\_NOT\_EXTENDED

Boa relação de portas

<http://www.iana.org/assignments/port-numbers>

## 7.2.12 - Trabalhando com Bancos de Dados

Vale observar que geralmente não precisamos efetuar a conexão com o banco de dados ao trabalhar com o Joomla. Isso porque ao abrir o Joomla a conexão já foi efetuada pelo mesmo.

Apenas precisamos usar o objeto que contém a conexão.

O acesso ao banco de dados no Joomla é efetuado com o objeto global Jdatabase. Veja em seguida um exemplo de acesso ao Jdatabase usando Jfactory.

Conexão com o Banco de Dados

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

```
$query = 'SELECT nome FROM #__alo WHERE id = ' . $nome_id;  
$db->setQuery( $query );  
$db->query($query);
```

quote() e nameQuote() são usados para filtrar os valores e os nomes de campos e tabelas respectivamente, como no exemplo:

```
$query = "  
SELECT *  
FROM ".$db->nameQuote('#__example_table')."  
WHERE ".$db->nameQuote('id')." = ".$db->quote('999999').";  
";
```

O prefixo #\_\_ é substituído pelo método \$db->setQuery() pelo correspondente prefixo da tabela.

O método query() é o método básico para a execução de consultas SQL.

Executar uma consulta SQL :

```
query()
```

**Receber número de registros**

```
getNumRows( $cur=null )
```

Exemplo:

```
$db = JFactory::getDBO();  
$query = "SELECT * FROM #__example2";  
$db->setQuery($query);  
$db->query(); // Obrigatório para o getNumRows(). Somente após query() recebe valor  
$rows = $db->getNumRows();
```

**Retornar primeiro campo do primeiro registro retornado pela query**

loadResult()

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

$nome_id = JRequest::getVar( 'id' );
$query = 'SELECT nome FROM #__alo WHERE id = ' . $nome_id;
$db->setQuery( $query );
$nome = $db->loadResult();
```

**Retorna um array indexado de um único campo da tabela:**

loadResultArray(\$numinarray = 0)

**Retorna um array indexado com o primeiro registro retornado pela query**

loadRow()

```
$query = "
SELECT ".$db->nameQuote('field_name')."
FROM ".$db->nameQuote('#__my_table')."
WHERE ".$db->nameQuote('some_name')." = ".$db->quote($some_value).";
";
$db->setQuery($query);
$row = $db->loadRow();
print_r($row);
```

**Retorna o primeiro registro em um array associado da tabela.**

loadAssoc()

```
$query = "
SELECT ".$db->nameQuote('field_name')."
FROM ".$db->nameQuote('#__my_table')."
WHERE ".$db->nameQuote('some_name')." = ".$db->quote($some_value).";
";
$db->setQuery($query);
$row = $db->loadAssoc();
print_r($row);
```

**Retorna o um objeto em PHP de um simples registro da tabela:**

loadObject()

```
$query = "
SELECT ".$db->nameQuote('field_name')."
FROM ".$db->nameQuote('#__my_table')."
WHERE ".$db->nameQuote('some_name')." = ".$db->quote($some_value).";
";
$db->setQuery($query);
$row = $db->loadObject();
print_r($row);
```

```
$row->field_name;
```

**Retorna um array indexado de objetos dos registros retornados da tabela pela query:**

```
loadObjectList()
```

e

```
loadObjectList($key)
```

```
$query = "
```

```
  SELECT name, email, username
  FROM . . . ";
```

```
$db->setQuery($query);
```

```
$result = $db->loadObjectList();
```

```
print_r($result);
```

```
$query = "
```

```
  SELECT name, email, username
  FROM . . . ";
```

```
$db->setQuery($query);
```

```
$row = $db->loadObjectList('username');
```

```
print_r($row);
```

```
    $query = "SELECT * FROM #__alo";
```

```
    $db->setQuery( $query, $limitstart, $limit );
```

```
    $rows = $db->loadObjectList();
```

**Retorna um array indexado de um único campo da tabela:**

```
loadResultArray()
```

```
$query = "
```

```
  SELECT name, email, username
  FROM . . . ";
```

```
...
```

```
$db->setQuery($query);
```

```
$column= $db->loadResultArray();
```

```
print_r($column);
```

```
$column['email'];
```

```
$column[1];
```

**loadResultArray(\$index)**

```
$query = "
```

```
  SELECT name, email, username
  FROM . . . ";
```

```
$db->setQuery($query);
```

```
$column= $db->loadResultArray(1);
```

```
print_r($column);
```

Permite interagir através das colunas do retorno.

```

$query = "
  SELECT name, email, username
  FROM . . . ";
$db->setQuery($query);
for ( $i = 0; $i <= 2; $i++ ) {
  $column= $db->loadResultArray($i);
  print_r($column);
}

```

### Carrega um array de arrays ou um array associativo de arrays (indexando a coluna numérica)

```

loadRowList( $key=" )
$query = "
  SELECT name, email, username
  FROM . . . ";
$db->setQuery($query);
$row = $db->loadRowList();
print_r($row);

```

### loadRowList()

```

$db = & JFactory::getDBO();

$query = 'SELECT id, nome FROM #__alo';
$db->setQuery( $query );
$lista = $db->loadRowList();

```

### Retornar um array indexado de array associado dos registros retornados tabela via query:

```

loadAssocList()
e
loadAssocList($key)

```

```

$query = "
  SELECT name, email, username
  FROM . . . ";
$db->setQuery($query);
$row = $db->loadAssocList();
print_r($row);

```

```

$query = "
  SELECT name, email, username
  FROM . . . ";
$db->setQuery($query);
$row = $db->loadAssocList('username');
print_r($row);

```

**Lista tabelas do banco**

```
getTableList()
```

**Recebe informações sobre uma tabela**

```
getTableFields( $tables, $typeonly = true )
```

**Get a quoted database escaped string**

```
Quote( $text, $escaped = true )
```

require\_once(JApplicationHelper::getPath('front\_html','com\_alo')) - usado para receber o arquivo html do componente com\_alo, no caso recebe o arquivo alo.html.php

Mais detalhes e exemplos:

[http://docs.joomla.org/How\\_to\\_use\\_the\\_database\\_classes\\_in\\_your\\_script](http://docs.joomla.org/How_to_use_the_database_classes_in_your_script)

**Conexão com o Banco de Dados**

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

```
// Obtem da URL o id do nome escolhido
```

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

**CRUD****SELECT**

```
//monta a consulta (observe a cláusula where)
```

```
$query = 'SELECT nome FROM #__alo WHERE id = ' . $nome_id;
```

```
$db->setQuery( $query );
```

```
$db->query($query);
```

```
//Executa a consulta
```

```
$nome = $db->loadResult();
```

```
$db->setQuery($query);
```

**INSERT**

```
// Obtem da URL o id do nome escolhido
```

```
$nome = JRequest::getVar( 'nome' );
```

```
//monta a consulta tipo Insert
```

```
$query = "INSERT INTO #__alo (id, nome) VALUES ('$nome_id' , '$nome)";
```

```
$db->setQuery($query);
```

**UPDATE**

```
//monta a consulta tipo Update
```

```
$query = "UPDATE #__alo SET nome = '$nome' WHERE id='id$'";
```

```
$query="UPDATE $tbl SET published = 0 WHERE module = '$modulo';
$db->setQuery($query);
```

Outro:

```
$db = & JFactory::getDBO();
$nome = JRequest::getVar('nome');
$id = JRequest::getVar('id');
if ( $id != 0 ) {
    $sql = "update #__alo set nome='". $nome.'" where id='". $id;
}
else {
    $sql = "insert #__alo (nome) values ('". $nome."')";
}
$db->setQuery($sql, 0);
$db->query();
```

## DELETE

//monta a consulta tipo Delete

```
$query = "DELETE nome FROM #__alo WHERE id='$id'";
$db->setQuery($query);
```

Veja também:

[http://www.joomlarj.com.br/site/index?option=com\\_content&view=article&id=83:entenda-o-objeto-database-do-joomla-15-&catid=26:entenda-o-objeto-database-do-joomla-15&Itemid=15](http://www.joomlarj.com.br/site/index?option=com_content&view=article&id=83:entenda-o-objeto-database-do-joomla-15-&catid=26:entenda-o-objeto-database-do-joomla-15&Itemid=15)

## Classe database e seus métodos

```
function getCategories() {
    global $database;

    $sql = 'SELECT * FROM #__categories';
    $database->setQuery( $sql );
    $rows = $database->loadObjectList();
    foreach ( $rows as $row ) {
        echo "$row->title: $row->description\n";
    }
}

getCategories();
```

**database->getTableCreate**

Retorna um array com a declaração em SQL 'CREATE TABLE'.

```
function generateSQL() {
    global $database;

    $prefix = $database->_table_prefix;
    $tables = array( $prefix.'content_frontpage', $prefix.'content_rating' );
    $result = $database->getTableCreate( $tables );
    print_r( $result );
}

generateSQL();
```

**database->getTableFields**

Retorna uma lista de campos e tipos de campos de uma lista de tabelas.

```
function generateSQL() {
    global $database;

    $prefix = $database->_table_prefix;
    $tables = array( $prefix.'content_frontpage', $prefix.'content_rating' );
    $result = $database->getTableFields( $tables );
    print_r( $result );
}

generateSQL();
```

**database->loadResult()**

Retorna o primeiro campo do primeiro registro retornado da mais recente consulta ao banco ou false se a consulta falhar.

Exemplo: Esta função retorna o title da primeira categoria da tabela de categorias do Joomla.

```
function getCategoryTitle() {
    global $database;

    $sql = 'SELECT title FROM #__categories';
    $database->setQuery( $sql );
    echo $database->loadResult();
}

getCategoryTitle();
```



**database->loadResultArray()**

Retorna um array contendo um simples campo de todos os registros retornados da consulta.

Exemplo: Esta função mostra um array de titles da tabela categorias do Joomla.

```
function getCategoryTitles() {
    global $database;
    $sql = 'SELECT id, title, description FROM #__categories';
    $database->setQuery( $sql );
    $titles = $database->loadResultArray( 1 );
    print_r( $titles );
}
getCategoryTitles();
```

**database->loadRow**

Retorna o primeiro registro da consulta atual como um array. Se a consulta falhar então retorna null

Exemplo: Esta função mostra um array contendo todos os campos de um dado registro da tabela categoria do Joomla.

```
function getCategory( $title ) {
    global $database;

    $title = $database->Quote( $title );
    $sql = "SELECT * FROM #__categories WHERE title=$title";
    $database->setQuery( $sql );
    $category = $database->loadRow();
    print_r( $category );
}
getCategory( 'Latest' );
```

**database->loadRowList**

Retorna um array de registros do banco com colunas numéricas indexadas. Retornará null se a consulta falhar.

Exemplo: Esta função mostra um array contendo registros da tabela categorias do Joomla. Note que neste exemplo o array é indexado pelo campo title de category.

```
function getCategories() {
    global $database;

    $sql = "SELECT * FROM #__categories";
    $database->setQuery( $sql );
    $categories = $database->loadRowList( 'title' );
```

```
print_r( $categories );
}
```

```
getCategories();
```

### **database->NameQuote**

Adiciona aspas para um identificador como o nome de uma tabela, de um campo, etc., usando a específica marca de quote. Este método foi introduzido no Joomla 1.1.

Example:

```
global $database;
```

```
echo $database->NameQuote( 'myTable' );
```

### **database->query**

Executa a string da consulta SQL atual.

```
function countCategories() {
    global $database;

    $sql = "SELECT * FROM #__categories";
    $database->setQuery( $sql );
    if (!$result = $database->query()) {
        echo $database->stderr();
        return false;
    }

    return $database->getNumRows( $result );
}
```

```
echo 'Existem ' . countCategories() . ' categorias';
```

### **database->query\_batch**

Executa a string SQL da consulta atual como uma transação simples. Esta função pode aceitar um ponto e vírgula (;) como separador de lista da string SQL e executa então em uma chamada simples. Se o parâmetro \$transaction\_safe for true então toda a string precisa ser inteiramente bem sucedida caso contrário toda ela será rolled back (não será executada).

### **database->Quote**

Retorna uma string com aspas em todos os caracteres que têm função especial no banco (serão escaped). Esta função é similar à função database->getEscaped exceto que a string retornada será delimitada por uma aspa simples.

```
function doQuery( $text ) {
```

```

global $database;

$text = $database->Quote( $text );
$sql = "SELECT * FROM #__content WHERE introtext=$text";
$database->setQuery( $sql );
if (!$database->query()) {
    echo $database->stderr();
    return false;
}

echo $database->getQuery();

}

```

```
doQuery( "Joomla 1.5 é 'Poder em Simplicidade!'" );
```

deve produzir:

```
SELECT * FROM mos_content WHERE introtext='Joomla 1.5 é 'Poder em Simplicidade\''
```

#### **database->stderr**

Retorna a última mensagem de erro do banco em um formato padrão. O número do erro e a mensagem de erro são incluídos. A declaração SQL que causou o erro pode opcionalmente ser incluída.

```

$query = "SELECT * FROM 'bad_table'";
$database->setQuery( $query );
$rows = $database->loadObjectList();
if ($database->getErrorNum()) {
    echo $database->stderr();
}

```

#### **Escapando e quotando dados do banco de dados:**

```

$db =& JFactory::getDBO();
$query = false;
if( $id = JRequest::getVar('id', 0, 'GET', 'INT') )
{
    $data = JRequest::getVar('content', 0, 'GET', 'STRING',
        JREQUEST_ALLOWRAW);
    $query = " UPDATE ".$db->nameQuote('#__test')." SET ".$db->
    >nameQuote('content')."=" . $db->Quote($data). " WHERE ". $db->nameQuote('id')."=" . $id;
}

```

Assumindo \$id=123 e \$data="Foo's bar", o valor de \$query deve ser:

```
UPDATE `#__test` SET `content`='Foo\'s bar' WHERE `id`=123;
```

## database->setQuery

Configura a string SQL para execução posterior. Esta função também substitui todas as ocorrências da string SQL contendo “#\_\_” pelo respectivo prefixo da tabela. Vale observar que esta função não executa a consulta SQL. Quem executa é a função query().

Syntax (anterior ao Joomla 1.0)

```
void setQuery ( string $query [, string $prefix] )
```

\$query

é a string SQL.

\$prefix

é a string de identificação de prefixo que será substituída pelo atual prefixo das tabelas. Este parâmetro é opcional e se omitido valerá o default '#\_\_'.

Syntax (igual ou superior a Joomla 1.0)

```
void setQuery ( sting $query [, integer $offset [, integer $limit [, string $prefix ]]] )
```

\$query

é a string SQL

\$offset

é o offset da tabela para ser usado com paginação de registros. Este parâmetro é opcional e se omitido valerá null.

\$limit

é o maior número de registros que será retornado pela consulta SQL. É opcional e se omitido valerá null.

\$prefix

é a string identificadora do prefixo das tabelas que deverá ser substituída pelo valor atual do prefixo das tabelas do banco. Este é opcional e se omitido terá o valor '#\_\_'.

Eemplos

```
function getCategories() {
    global $database;

    $sql = 'SELECT * FROM #__categories';
    $database->setQuery( $sql );
    $rows = $database->loadObjectList();
    foreach ( $rows as $row ) {
        echo "$row->title: $row->description\n";
    }
}

getCategories();
```

deve resultar em:

Latest: The latest news from the Joomla Team

Joomla: A selection of links that are all related to the Joomla project.

Newsflash:

Contacts: Contact Details for this web site

### **database->insertObject**

Insere um novo registro na tabela usando dados de um objeto. Retorna true se a inclusão foi bem sucedida; false em caso contrário.

Sintaxe:

```
boolean insertObject ( string $table, object &$object [, string $keyname [, boolean $verbose ] ] )
```

**\$table**

é o nome da tabela.

**\$object**

é um objeto contendo os campos a serem inseridos. Os atributos são arrays ou objetos, terão o valor null ou terão o nome iniciando com um underscore (sublinhado), são ignorados quando configurando os campos do banco.

**\$keyname**

é o nome da chave primária da tabela. Se este não for null então o atributo \$keyname do \$object deve ser atualizado com o ID único retornado pela inserção no banco. Este parâmetro é opcional e se omitido valerá null.

**\$verbose**

é um flag. Se true então a consulta SQL e o ID único retornado pela consulta será mostrado. Se false então nenhuma saída será produzida. Este parâmetro é opcional e se omitido valerá false.

### Exemplos

Exemplo: Este exemplo insere um novo registro na tabela users. Observe que o campo 'id' é setado para null antes de chamar insertObject e deve ser atualizado com o ID único alocado para a tabela.

```
// Cria um novo registro na tabela users e retorna o id único alocado se bem sucedido  
// ou false se falhar.
```

```
function newUser( $name, $username) {  
    global $database;  
  
    $user = new stdClass;  
    $user->id = NULL;  
    $user->name = $name;
```

```
$user->username = $username;

if (!$database->insertObject( '#__users', $user, 'id' )) {
    echo $database->stderr();
    return false;
}

return $user->id;
}

echo newUser( 'Linus Torvalds', 'linus' );
```

### **database->updateObject**

Atualizar um registro de uma tabela usando dados contidos em um objeto. Retornará false se a atualização falhar, caso contrário retornará o registro atualizado.

Sintaxe:

```
object updateObject ( string $table, object &$amp;object, string $keyname [, boolean $updateNulls ] )
```

**\$table**

é o nome da tabela.

**\$object**

é um objeto contendo os campos a serem atualizados.

**\$keyname**

é o nome da chave primária da tabela.

**\$updateNulls**

é um flag. Se true então atribui null ao objeto que deve sobrescrever os correspondente campos. Se false então a atribuição de null ao objeto será ignorada.

### Exemplos

Exemplo: Neste exemplo a função newUser é chamada para criar um registro na tabela users do Joomla, então a função updateUser é chamada para atualizar um desses campos.

```
// Criar novo registro na tabela users e retornar o ID único alocado se bem sucedido
// ou false em caso contrário.
function newUser( $name ) {
    global $database;

    $user = new stdClass;
    $user->id = NULL;
    $user->name = $name;
```

```
if (!$database->insertObject( '#__users', $user, 'id' )) {
    echo $database->stderr();
    return false;
}

return $user->id;
}

// Atualizar um registro existente.
function updateUser( $id, $username ) {
    global $database;

    $user = new stdClass;
    $user->id = $id;
    $user->username = $username;

    if (!$database->updateObject( '#__users', $user, 'id' )) {
        echo $database->stderr();
        return false;
    }
}

$id = newUser( 'Linus Torvalds' );
updateUser( $id, 'linus' );
```

### **database->getPrefix**

Retorna o prefixo atual das tabelas do banco de dados. Por padrão o prefixo é 'jos\_' mas isso pode ser alterado durante a instalação do Joomla. Este método foi introduzido no Joomla 1.1.

Sintaxe:

```
string getPrefix ( )
```

Exemplos

Exemplo:

```
echo $database->getPrefix();
```

deve produzir a seguinte saída:

```
jos_
```

**database->insertid**

Retorna o número exclusivo do último registro mais um para ser inserido na tabela.

Sintaxe:

```
int insertid ( )
```

Exemplos

Exemplo: Este exemplo cria um novo registro default na tabela users e retorna o ID único alocado para o usuário.

```
// Retorna o ID único do novo registro do usuário ou false
// se a inclusão falhar.
function newUser() {
    global $database;

    $sql = 'INSERT INTO #__users () VALUES()';
    $database->setQuery( $sql );
    if (!$database->query()) {
        echo $database->stderr();
        return false;
    }

    return $database->insertid();
}
```

**database->loadAssocList**

Carrega uma lista associativa de registros.

Sintaxe:

```
array loadAssocList ( string $key )
```

\$key

é o nome do campo da chave primária. Este parâmetro é opcional e se for omitido terá o valor empty.

Exemplos

Exemplo: A função deste exemplo mostra um array associativo de campos da tabela categorias do Joomla.

```
function getCategories() {
    global $database;
```



```
$sql = 'SELECT * FROM #__categories';
$database->setQuery( $sql );
$rows = $database->loadAssocList();
print_r( $rows );

}

getCategories();

deve produzir:
Array
(
    [0] => Array
        (
            [id] => 1
            [parent_id] => 0
            [title] => Latest
            [name] => Latest News
            [image] => taking_notes.jpg
            [section] => 1
            [image_position] => left
            [description] => The latest news from the Joomla Team
            [published] => 1
            [checked_out] => 0
            [checked_out_time] => 2000-00-00 00:00:00
            [editor] =>
            [ordering] => 0
            [access] => 0
            [count] => 1
            [params] =>
        )
    [1] => Array
        (
            [id] => 2
            [parent_id] => 0
            [title] => Joomla
            [name] => Joomla
            [image] => clock.jpg
            [section] => com_weblinks
            [image_position] => left
            [description] => A selection of links that are all related to the Joomla project.
            [published] => 1
            [checked_out] => 0
            [checked_out_time] => 0000-00-00 00:00:00
            [editor] =>
        )
)
```

```
    [ordering] => 0
    [access] => 0
    [count] => 0
    [params] =>
  )
}
```

### **database->loadObject**

Carrega um objeto com cammpos do primeiro registro retornados da consulta atual. Se um objeto for passado como um parâmetro os campos do registro os campos do registro estarão contidos nele e o objeto atualizado será retornado.

Sintaxe:

```
boolean loadObject ( object &$bind_object )
```

`$bind_object`

é o objeto que deve ser atualizado do primeiro registro retornado pela consulta. Se `$bind_object` for null então `loadObject` deverá criar um objeto contendo todos os campos do primeiro registro da consulta.

Exemplos

Exemplo: Neste exemplo nós permitimos que `loadObject` crie um objeto para nós.

```
function getUser() {
    global $database;

    $sql = 'SELECT * FROM #__users';
    $database->setQuery( $sql );
    $user = NULL;
    $database->loadObject( $user );
    echo "Name: $user->name\n";
    echo "Username: $user->username\n";
}
```

```
getUser();
```

deve produzir:

```
Name: Administrator
Username: admin
```

**database->loadObjectList**

Retorna um array de objetos do banco usando a consulta atual. Retorna false se a consulta falhar (antes do Joomla 1.1 retorna null).

Sintaxe:

```
array loadObjectList ( [ string $key ] )
```

**\$key**

é o nome do campo da chave primária. Se \$key for empty ou null então loadObjectList retornará uma lista sequencial dos registros retornados pela consulta. Se \$key não for empty ou null então o array retornado deve ser indexado pelo valor da chave da tabela. Este parâmetro é opcional e se omitido terá o valor null (empty para versões anteriores ao Joomla 1.1).

Exemplos

Exemplo:

```
function getCategories() {  
    global $database;  
  
    $sql = 'SELECT * FROM #__categories';  
    $database->setQuery( $sql );  
    $rows = $database->loadObjectList();  
    foreach ( $rows as $row ) {  
        echo "$row->title: $row->description\n";  
    }  
}
```

```
getCategories();
```

deverá produzir:

Latest: The latest news from the Joomla Team

Joomla: A selection of links that are all related to the Joomla project.

Newsflash:

Contacts: Contact Details for this web site

**database->schemaUpdate**

Atualiza o esquema AXMLS. Retorna true se bem sucedido; false em falha. Este método foi introduzido na versão 1.1 do Joomla.

Sintaxe:

```
boolean updateSchema ( string $filename )
```

**\$filename**

é o nome do arquivo XML que contém o schema.

## 7.2.13 - Manipulando Campos comuns

Existem alguns nomes de campos reservados pelo Joomla, que são usados em várias tabelas e inclusive devemos adicionar nas tabelas que criamos: published, hits, ordering, etc.

Ao usarmos esses campos nas nossas tabelas estaremos tirando vantagem do Framework do Joomla.

### Publishing

Para publicar e despublicar dados nós podemos usar o método publish(). Ele publica e despublica vários registros ao mesmo tempo. Se a tabela incluir um campo chamado "checked\_", nós podemos ficar seguros de que o registro não está "checado" ou está "checado" para o usuário atual. Este exemplo publica um registro.

```
$publishIds = array($id);
$user =& JFactory::getUser();
if (!$table->publish($publishIds, 1, $user->get('id')))
{
    // erro ao publicar, então
    // use $table->getError() para uma explicação das causas do erro
}
```

### Hits

Para incrementar o campo hits nós podemos usar o método hit(). Neste exemplo nós setamos o ID do registro de buffer e usamos o método hit().

```
$table->set('id', $id);
$table->hit();
```

Alternativamente nós podemos especificar o ID quando nós usamos o método hit(). Se nós optarmos por fazer isso, nós precisaremos lembrar que o ID do buffer deve ser atualizado para valor semelhante ao ID do hit.

```
$table->hit($id);
```

### Checking Out

Antes de iniciar a checagem dos registros nós precisamos primeiro checar se um registro está realmente checado. Lembrar que quando um registro está checado nós não devemos permitir que ninguém modifique esse registro. Nós podemos usar o método isCheckedOut() para fazer essa verificação. Neste exemplo nós testamos para ver se algum usuário que não seja o atual checou o registro:

```
$table->load($id);
$user =& JFactory::getUser();
if ($table->isCheckedOut($user->get('id')))
{
    // handle record is already checked-out
}
```

Logo qye nós determinamos que um registro não está checado, então nós podemos usar o método checkout() para checar o registro. Neste exemplo nós checamos o registro para o usuário atual; isto seta o campo checked\_out para o ID do usuário e o campo checked\_out\_time para o tempo atual.

```
$table->load($id);
$user =& JFactory::getUser();
if (!$table->checkout($user->get('id')))
{
    // handle failed to checkout record
}
```

```
$table->load($id);
$user =& JFactory::getUser();
if (!$table->checkin($user->get('id')))
{
    // handle failed to checkin record
}
```

### Ordering

Quando nós desejamos ordenar itens JTable nos oferece alguns métodos úteis. O primeiro é o reorder(). Este método verifica cada registro e move então a ordem para cima enquanto qualquer gaps em ordem deve ser removido. Neste exemplo nós reordenamos os registros da nossa tabela:

```
$table->reorder();
```

exemplo, nós reordenamos os registros no grupo 1:

```
$db =& $table::getDBO();
$where = $db->nameQuote('group').' = 1';
$table->reorder($where);
```

neste exemplo, nós recebemos a próxima posição entre os registros do grupo 1:

```
$db =& $table::getDBO();
$where = $db->nameQuote('group').' = 1';
$nextPosition = $table->getNextOrder($where);
```

Por último nós podemos usar o método move() para mover um registro para uma posição acima ou abaixo.

Neste exemplo, nós movemos um registro para uma posição acima na ordem:

```
$table->load($id);
$table->move(-1);
```

Novamente temos a opção para especificar agrupamentos. Imagine que nossa tabela também tem um campo chamado group; neste exemplo, nós movemos um registro para baixo na ordem no group 1:

```
$db =& $table::getDBO();  
$where = $db->nameQuote('group').' = 1';  
$table->load($id);  
$table->move(1, $where);
```

## 7.2.14 - JTable

A JTable é uma poderosa classe, onde usamos esses nomes de campos. Ela permite usar com simplicidade várias funções com os registros das tabelas. Para cada tabela que usamos devemos estender JTable.

### Convenções para JTable

-Ao criar um componente a sub classe de JTable deve ficar em:  
portal/administrator/com\_exemplo/tables

-Nome da classe:  
TableExemploTabela

-Primeiro definir as propriedades públicas que representam cada campo da tabela, cada uma com o mesmo nome do campo

-Definir o construtor, que tem um único parâmetro (o objeto do banco)

-Sobrescrever o método check(). Caso falhe devemos usar setError() para mostrar a razão da falha.

Instanciar a classe criada:

```
JTable::addIncludePath(JPATH_COMPONENT_ADMINISTRATOR.DS.'tables');
```

```
$table = Jtable::getInstance('alguma', 'Table');
```

### Criando uma classe JTable

A classe JTable deve ser criada no diretório:

administrator/components/com\_meucomponente/tables

Exemplo de classe JTable para a tabela #\_\_simples:

```
/**
 * Manipular a tabela #__simples
 */
class TableSimples extends JTable
{
    /** @var int */
    var $id = null;
    /** @var string */
    var $sim = "";
    /** @var string */
    var $ples = "";
}
```



```

/**
 * Criando um novo TableSimples
 */
function __construct(&$db) {
    parent::__construct('#__meucomponente', 'id', $db);
}
/**
 * Is the data valid?
 */
function check() {
    // check for valid id (int or null)
    if (!preg_match('~^\d+$~', $this->id) || $this->id !== null) {
        $this->setError(JText::_('ID IS INVALID'));
        return false;
    }
    // check for valid sim
    if(JString::trim($this->sim) == "") {
        $this->setError(JText::_('Simples requer sim'));
        return false;
    }
    // all OK, data is valid!
    return true;
}
}

```

### Criando um novo registro usando JTable

```

// Valores a usar para criar um novo registro
$post = JRequest::get('POST');

// Não entrar com ID
$post['id'] = false;

if (!$table->save($post)) {
    // uh oh failed to save
}

// values to use to create new record
$post = JRequest::get('POST');
// do not provide an ID
$post['id'] = false;
// allow raw untrimmed value for foo
$post['foo'] = JRequest::getString('foo', '', 'POST', JREQUEST_
ALLOWRAW | JREQUEST_NOTRIM);
// bind $post with $table

```

```
if (!$table->bind($post)) {
    // failed
}
// check the data is valid
if (!$table->check()) {
    // failed
}
// store the data in the database table and update nulls
if (!$table->store(true)) {
    // failed
}
// check the record in
if (!$table->checkin()) {
    // failed
}
// reorder the entire table (not grouped in any way)
if (!$table->reorder()) {
    // failed
}
}
```

### **Atualizar um registro usando JTable**

```
// values to use to update an existing record
// $post contains the ID of the record we are updating
$post = JRequest::get('POST');
if (!$table->save($post)) {
    // uh oh failed to save
}
}
```

### **Lendo um registro existente usando JTable**

```
if ($table->load(JRequest::getInt('id'))) {
    // success!
}

$someField = $table->get('someField');

if ($table->delete(JRequest::getInt('id'))) {
    // successfully deleted
}
}
```

### **Checando se registro está bloqueado usando JTable**

```
// get current user
```

```
$user =& JFactory::getUser();
// load the record
$table->load($id);
// check if record is already checked out
if ($table->isCheckedOut($user->get('id'))) {
    // someone beat us to it!
}

// check out the current record
$table->checkout($user->get('id'));

// check the current record in
$table->checkin();

// get current user
$user =& JFactory::getUser();
// check if is checked out
if (JTable::isCheckedOut($user->get('id'), $checkedOut)) {
    // record is checked out by another user
}

// check out a record identified by $oid
$table->checkout($user->get('id'), $oid);

// check in a record identified by $oid
$table->checkin($oid);
```

### Alterando a ordem dos registros com JTable

```
// prepare grouping
$db =& JFactory::getDBO();
$group = $db->nameQuote('catid') . ' = ' . intval($catid);
// get next space
$next = $table->getNextOrder($group);

// prepare grouping
$db =& JFactory::getDBO();
$group = $db->nameQuote('catid') . ' = ' . intval($catid);
// get next space
$next = $table->reorder($group);

// prepare grouping
$db =& JFactory::getDBO();
$group = $db->nameQuote('catid') . ' = ' . intval($catid);
// move current record up one
$table->move(-1, $group);
```

**Publicando e despublicando registros com JTable**

```
// get array of records to publish
$cids = JRequest::getVar('cid', array(), 'REQUEST', 'ARRAY');
// publish the records!
$table->publish($cids);
```

```
// get array of records to publish
$cids = JRequest::getVar('cid', array(), 'REQUEST', 'ARRAY');
// get the current user
$user =& JFactory::getUser();
// unpublish the records!
$table->publish($cids, 0, $user->get('id'));
```

**Incrementando a contagem de um registro (hits) usando JTable**

```
// increment number of hits for current record
$table->hit();
```

```
// increment number of hits
$table->hit($id);
```

## 7.2.15 - Incrementando Contador de Hits

```
/**
 * Increments the hit counter
 *
 */
function hit()
{
    //
    $db =& JFactory::getDBO();
    $db->setQuery('UPDATE '.$db->nameQuote('#__minhaextensao').'
        .SET '.$db->nameQuote('hits').' = '.$db->nameQuote('hits').' + 1 .'WHERE id = '
    $this->_id);
    $db->query();
}

// build COUNT query
$sql = "SELECT COUNT(*) FROM $tableName";
$db->setQuery($sql);
$total = $db->loadResult();
```

## 7.2.16 - Manipulando Erros com DBO

```
JDatabase::getErrorNum()

if ($db->getErrorNum() == 0) {
    // no errors occurred
} else {
    // errors occurred
}

// on fail
$error = $db->getErrorMsg();

// display error
JError::raiseWarning(500, $error);

// on fail
$error = $db->stderr();

// display error
JError::raiseError(500, $error);
```

## 7.2.17 - Conectar a um Banco de Dados Externo

Para conectar a um banco de dados diferente do banco do nosso site em Joomla. Que esteja no mesmo servidor ou mesmo em um servidor remoto e até mesmo em outro tipo de driver de banco de dados. Para isso podemos usar o método [JDatabase->getInstance](#).

```
<?php
$options = array(); //prevent problems

$options['driver'] = 'mysql'; // Database driver name
$options['host'] = 'db.myhost.com'; // Database host name
$options['user'] = 'fredbloggs'; // User for database authentication
$options['password'] = 's9(39s£h[%dkFd'; // Password for database authentication
$options['database'] = 'bigdatabase'; // Database name
$options['prefix'] = 'abc_'; // Database prefix (may be empty)

$db = & JDatabase::getInstance( $options );
?>
```

\$db é agora um objeto do tipo [JDatabase](#) e podemos executar operações nele os métodos usuais.

Note que se o banco usa uma porta com número não padrão então ela pode ser especificada adicionando ao final do hostname. Exemplo: 'db.myhost.com:3307'.

## 7.2.18 - Trabalhando com Usuários no Joomla

### Trabalhando com Session

Session na web é uma forma de oferecer um armazenamento temporário de facilidades enquanto durar a visita do usuário.

Em PHP os dados são acessados com a variável global `$_SESSION`, mas no Joomla devemos fazer diferente, usando os objetos e métodos do Framework, através do objeto `Jsession`, que estão globalmente acessíveis através da interface `static JFactory`. Como `JFactory` retorna um objeto precisamos usar `=&` ao atribuir o objeto à variável.

```
$session =& JFactory::getSession();  
$valor = $session->get('exemplo');
```

```
$ValorCampo = JRequest::getString('meuCampo', '', 'POST', JREQUEST_ALLOWRAW);
```

```
$session->set('meuCampo', $ValorCampo, 'meuForm');
```

### Remover Session

```
$session =& JFactory::getSession();
```

```
$session->clear('meuCampo', 'meuForm');
```

```
$session->set('valorOculto', null);
```

### Garantindo que uma variável seja setada:

```
JRequest::setVar('nome', Jrequest::getVar('nome', 'unknown'));
```

### Setar o novo valor e receber o antigo:

```
$valorAntigo = $session->set('avisoOculto', $valor);
```

### Namespace

```
$session->set('algo', $value, 'meunamespace');
```

### Application

O método `Japplication::setUserState()` era para retornar o valor antigo, mas um bug o leva a retornar o valor novo.

```
// receber a aplicação
```



```
$app =& JFactory::getApplication();
```

```
// setar algum dado  
$app->setUserState('com_minhaextensao', $valor);
```

**Recebendo o valor de um diferente Namespace:**

```
$value = $session->get('hideAnnouncements', $defaultValue, 'mynamespace');
```

**Acessando o estado dos dados do usuário:**

```
// receber a aplicação (semelhante a $mainframe)  
$app =& JFactory::getApplication();
```

```
// receber o estado de dados do usuário  
$value = $app->getUserState('com_minhaextensao');
```

**Atualizar estado dos dados do usuário com****Japplication::getUserStateFromRequest():**

```
// receber estado de dados do user e atualizar pelo request  
$value = $app->getUserStateFromRequest(  
    'com_minhaextensao',  
    'inputNome',  
    $defaultValue,  
    'INTEGER'  
);
```

**Método muito usado para manipular estado de exibição de dados, como em paginações:**

```
// receber o limit default da paginação global  
$defaultListLimit = $app->getCfg('list_limit');
```

```
// receber o limit baseado no estado dos dados requisitados pelo usuário  
$limit = $app->getUserStateFromRequest(  
    'global.list.limit',  
    'limit',  
    $defaultListLimit,  
    'INTEGER'  
);
```

**Token**

```
$token = JSession::getToken();
```

Recebendo o objeto do JUser:

```
$user =& JFactory::getUser();
```

Recebendo usuário pelo ID e pelo username:

```
// recebendo pelo ID
$aUser =& JFactory::getUser(100);

// recebendo pelo username
$anotherUser =& JFactory::getUser('anotherusername');
```

**Recebenndo user pelo estado (guest/registered):**

```
if ($user->get('guest') {
    // user é guest
} else {
    // user é registrado
}

// receber username e name do user
$username = $user->get('username');
$name     = $user->get('name');

// mostrar informações sobre o user
echo "Login é $username nome é $name";

if ($user->get('guest') {
    // user é guest
    $username = JText::_('ANONYMOUS');
    $name     = JText::_('ANONYMOUS');
} else {
    // user é registrado
    $username = $user->get('username');
    $name     = $user->get('name');
}
```

**ID dos grupos de usuários:**

Frontend:  
registrado - 18  
autor - 19  
editor - 20  
publisher – 21

Backend:  
manager – 23  
administrator – 24  
super administrator – 25

**Recebendo o ID do grupo (gid):**

```
$groupID = $user->get('gid');
```

**Tipo do grupo:**

```
$usertype = $user->get('usertype');
```

**Níveis de Acesso**

```
public – 0  
registered 1  
special - 2
```

**Recebendo o ID do nível de acesso (aid):**

```
$aid = $user->get('aid');
```

```
// recebendo o DBO  
$db =& JFactory::getDBO();
```

```
// preparando o nome do campo de acesso  
$access = $db->nameQuote('access');
```

```
// restringindo a query pelo nível de acesso  
$where = "WHERE $access <= " . intval($aid);
```

```
// make sure the user has the necessary access rights  
if ($table->get('access') > $aid) {  
    JError::raiseError(403, JText::_('ALERTNOTAUTH'));  
    jexit();  
}
```

**Timezones**

```
$user =& JFactory::getUser();  
$timezone = $user->getParam('timezone');
```

```
$timezone = $user->getParam('timezone', '0');
```

```
// receber parameters com arquivo de definição em XML carregado  
$params =& $user->getParameters(true);
```

```
// receber timezone (hours UTC offset)  
$timezone = $params->get('timezone', '0');
```

**Setando os Parâmetros do Usuário**

```
$user =& JFactory::getUser();
```

```
// setando valor de parâmetro
```

```
$user->setParam('editor', 'none');
```

Ou

```
$params->set('editor', 'none');
```

### Extendendo parâmetros dos usuários

Arquivo user.xml definido na pasta models de um componente.

```
<?xml version="1.0" encoding="utf-8"?>
<user>
  <params group="com_mycomponent">
    <param name="myparameter"
      type="text"
      size="20"
      default=""
      label="MY PARAMETER"
      description="MY PARAMETER DESCRIPTION" />
  </params>
</user>
```

```
// define path to the XML file
```

```
$path = JPATH_COMPONENT_ADMINISTRATOR . DS . 'models';
```

```
// load the XML file and get the JParameter object
```

```
$params =& $user->getParameters(true, $path);
```

```
// output parameters
```

```
echo $params->render('bespokeparams', 'com_mycomponent');
```

```
echo JHTML::_('form.token');
```

```
// make sure the token is valid
```

```
JRequest::checkToken() or jexit('Invalid Token');
```

```
// load the raw array of params
```

```
$rawParams = JRequest::getVar(
```

```
  'bespokeparams',
```

```
  array(),
```

```
  'POST',
```

```
  'ARRAY'
```

```
);
```

```
// build an array from the raw array
```

```
$besokeParams['myparameter'] = $rawParams['myparameter'];
```

```
// bind with the user's parameters
```

```
$params->bind($bespokeParams);
```

```
// save changes to the user
```

```
if ($user->save()) {  
    JError::raiseNotice("200", JText::_("Saved User"));  
}
```

## 7.2.19 - Envio de e-mail

```
$mailer =& JFactory::getMailer();
$mailer->setSender('example@example.org');

$sender = array('example@example.org', 'example')
$mailer =& JFactory::getMailer();
$mailer->setSender($sender);

$reply = array('example@example.org', 'Example');
$mailer->addReplyTo($reply);
$reply0 = array('example@example.org', 'Example');
$reply1 = array('example@example.org', 'Example');
$replies = array($reply0, $reply1);
$mailer->addReplyTo($replies);

    $mailer->addRecipient('foo@example.org');
    $recipients = array('bar@example.org', ' baz@example.org ');
    $mailer->addRecipient($recipients);

    $mailer->setSubject('Some Email');
    $mailer->setBody('Lorem ipsum dolor sit amet.');
```

\$mailer->IsHTML(true);

```
if ($mailer->Send() !== true)
{
    // an error has occurred
    // a notice will have been raised by $mailer
}
```

Ou

```
jimport('phpmailer.phpmailer');
$mailer = new PHPMailer();

if (!JMailHelper::isEmailAddress($someEmailAddress))
{
    $this->setError(JText::_('INVALID_EMAIL_ADDRESS'));
    return false;
}
```

```
jimport('joomla.utilities.mail');
```

**Enviando e-mail para o usuário**

```
// get the Joomla! configuration
$config =& JFactory::getConfig();

// get the email 'from' details
$from = $config->getValue('mailfrom');
$fromname = $config->getValue('fromname');

$recipient = $user->get('email');

// get the users name
$name = $user->get('name');
$subject = 'My First Joomla! Email';
$body = "Hi $name \n\nDo you like my email?";

if (JUtility::sendMail($from, $fromname, $recipient, $subject, $body)
    !== true) {
    JError::raiseNotice(
        '500',
        JText::_('ERROR SENDING EMAIL')
    );
}
```

## 7.2.20 - Lista de Controle de Acesso - ACL

```
$acl =& JFactory::getACL();
$acl->_mos_add_acl('com_myExtension', 'manage', 'users', 'super administrator');
```

```
$user =& JFactory->getUser();
if( ! $user-> authorize('com_myExtension', 'manage') )
{
    JError::raiseError(403, JText::_('Access Forbidden'));
}
```

```
$task = JRequest->getVar('task', 'view', 'GET', 'WORD');
$controller = new myExtensionController();
$controller->setAccessControl('com_myExtension');
$controller->execute($task);
```

```
$task = JRequest->getVar('task', 'view', 'GET', 'WORD');
$controller = new myExtensionController();
$controller->setAccessControl('com_myExtension', 'manage');
$controller->execute($task);
```

```
$acl =& JFactory::getACL();
$acl->_mos_add_acl('com_myExtension', 'manage', 'users', 'super administrator',
'category', 'some category');
```

```
$user =& JFactory->getUser();
if( ! $user-> authorize('com_myExtension', 'manage', 'category', 'some category') )
{
    JError::raiseError('403', JText::_('Access Forbidden'));
}
```

### **JFactory oferece:**

- \* getApplication( \$id = null, \$config = array() )
- \* getConfig( \$file = null, \$type = 'PHP' )
- \* getSession( \$options = array() )
- \* getLanguage( )
- \* getDocument( )
- getUser( \$id = null )
- \* getCache( \$group = "", \$handler = 'callback', \$storage = null )
- \* getACL( )
- \* getTemplate( )
- \* getDBO( )
- \* getMailer( )
- \* getXMLParser( \$type = 'DOM', \$options = array() )
- \* getEditor( \$editor = null )
- \* getURI( \$uri = 'SERVER' )



```
$user =& JFactory::getUser();

$user =& JFactory::getUser();
$language = $user->getParam('language', 'the default');

$user =& JFactory::getUser();

if ($user->guest) {
    echo "<p>You must login to see the content. I want your email address.</p>";
} else {

}

$user =& JFactory::getUser();

if ($user->authorize('com_content', 'edit', 'content', 'all')) {
    echo "<p>You may edit all content.</p>";
} else {
    echo "<p>You may not edit all content.</p>";
}

if ($user->authorize('com_content', 'publish', 'content', 'own')) {
    echo "<p>You may publish your own content.</p>";
} else {
    echo "<p>You may not publish your own content.</p>";
}
}
```

## ACL

Podemos criar nossas próprias permissões.

Adicionar o código a um local que deva ser executado, como o início do componente que estamos criando.

Primeiro receber o objeto authorization usando a função getACL() do JFactory.

```
$auth =& JFactory::getACL();
```

```
$auth->addACL('com_userinfo15', 'persuade', 'users', 'super administrator');
$auth->addACL('com_userinfo15', 'persuade', 'users', 'administrator');
$auth->addACL('com_userinfo15', 'persuade', 'users', 'manager');
```

```
$user =& JFactory::getUser();
```

```
if ($user->authorize('com_userinfo15', 'persuade')) {
    echo "<p>You may persuade the system to do what you wish.</p>";
} else {
```

```
    echo "<p>You are not very persuasive.</p>";  
}
```

```
$user =& JFactory::getUser();
```

```
echo "<p>Seu nome é $user->name, seu email é $user->email  
e seu nome de usuário é $user->username</p>";  
echo "<p>Seu tipo de usuário é $user->usertype, o qual tem a identificação  
de grupo $user->gid.</p>";
```

```
$usuario =& JFactory::getUser();
```

```
if ($usuario->authorize('com_content', 'edit', 'content', 'all')) {  
    echo "<p>Você pode editar todos os artigos.</p>";  
} else {  
    echo "<p>Você não pode editar todos os artigos.</p>";  
}
```

```
if ($usuario->authorize('com_content', 'publish', 'content', 'own')) {  
    echo "<p>Você pode publicar seus próprios artigos.</p>";  
} else {  
    echo "<p>Você não pode publicar seus próprios artigos.</p>";  
}
```

```
$autoriza =& JFactory::getACL();
```

```
$autoriza->addACL('com_seucomponente', 'convencer', 'users', 'super administrator');  
$autoriza->addACL('com_seucomponente', 'convencer', 'users', 'administrator');  
$autoriza->addACL('com_seucomponente', 'convencer', 'users', 'manager');
```

```
$usuario =& JFactory::getUser();
```

```
if ($usuario->authorize('com_seucomponente', 'convencer')) {  
    echo "<p>Você pode convencer o sistema para fazer o que quiser.</p>";  
} else {  
    echo "<p>Você não é muito convincente.</p>";  
}
```

Veja sobre o objeto JUser em Numa Boa - [http://www.joomlarj.com.br/site/index.php?option=com\\_content&view=article&id=82:o-objeto-juser-&catid=16:como-criar-componentes--modulos&Itemid=15](http://www.joomlarj.com.br/site/index.php?option=com_content&view=article&id=82:o-objeto-juser-&catid=16:como-criar-componentes--modulos&Itemid=15)

## JUser

De Joomla! Documentation

JUser é uma classe abstrata que provém alguns métodos para manipulação de usuários.

Disponível para Joomla 1.5 e Joomla 1.6

Definido em - /libraries/joomla/user/user.php

Extends

\* JObject

Métodos Get e Set

Get method	Set method	Description	Property
	setLastVisit	Seta a última data de visita	
getParam	setParam	Parâmetro User	
getParameters	setParameters	Parâmetros User	
getTable		Recebe o objeto user da tabela	

### Outros métodos

Nome do método	Descrição
defParam	Seta um parâmetro default, caso não exista.
authorize	Checa a autorização do objeto JUser novamente para acessar objeto de controle e opcionalmente um objeto de extensão acesso.
bind	Liga um array associativo de dados para o objeto user.
save	Salva o objeto JUser no banco de dados.
getInstance	Retorna uma referência para o objeto global JUser de um certo tipo, criando caso este ainda não exista. Em muitos casos podemos
somente	JFactory::getUser.
delete	Deleta o objeto JUser do banco.

### Import

```
jimport( 'joomla.user.user' );
```

Exemplos

### Create um novo user

```
/*
```

```
First set up some variables/objects
```

```
*/
```

```
// get the ACL
$acl =& JFactory::getACL();

/* get the com_user params */

jimport('joomla.application.component.helper'); // include
libraries/application/component/helper.php
$usersParams = &JComponentHelper::getParams( 'com_users' ); // load the Params

// "generate" a new JUser Object
$user = JFactory::getUser(0); // it's important to set the "0" otherwise your admin user
information will be loaded

$data = array(); // array for all user settings

// get the default usertype
$userType = $usersParams->get( 'new_usertype' );
if (!$userType) {
    $userType = 'Registered';
}

// set up the "main" user information

$data['name'] = $firstname.' '.$lastname; // add first- and lastname
$data['username'] = $username; // add username
$data['email'] = $email; // add email
$data['gid'] = $acl->get_group_id( "", $userType, 'ARO' ); // generate the gid from the
usertype

/* no need to add the usertype, it will be generated automatically from the gid */

$data['password'] = $password; // set the password
$data['password2'] = $password; // confirm the password
$data['sendEmail'] = 1; // should the user receive system mails?

/* Now we can decide, if the user will need an activation */

$useractivation = $usersConfig->get( 'useractivation' ); // in this example, we load the
config-setting
if ($useractivation == 1) { // yeah we want an activation

    jimport('joomla.user.helper'); // include libraries/user/helper.php
    $data['block'] = 1; // block the User
    $data['activation'] = JUtility::getHash( JUserHelper::genRandomPassword() ); // set
activation hash (don't forget to send an activation email)

}
```

```
else { // no we need no activation

    $data['block'] = 0; // don't block the user

}

if (!$user->bind($data)) { // now bind the data to the JUser Object, if it not works....

    JError::raiseWarning(" JText::_( $user->getError()); // ...raise an Warning
    return false; // if you're in a method/function return false

}

if (!$user->save()) { // if the user is NOT saved...

    JError::raiseWarning(" JText::_( $user->getError()); // ...raise an Warning
    return false; // if you're in a method/function return false

}

return $user; // else return the new JUser object
```

Para toda requisição em Joomla existe um usuário.

```
$user =& JFactory::getUser();
```

O código abaixo exibe nome, e-mail, user name, user type e id do grupo para o usuário atual:

```
echo "<p>Your name is {$user->name}, your email is {$user->email}, and your username
is {$user->username}</p>";
echo "<p>Your usertype is {$user->usertype} which has a group id of {$user->gid}.</p>";
```

Outras variáveis membros geradas automaticamente ao chamar getUser():

\* id - The unique, numerical user id. Use this when referencing the user record in other database tables.

\* name - The name of the user. (e.g. Vint Cerf)

\* username - The login/screen name of the user. (e.g. shmuffin1979)

\* email - The email address of the user. (e.g. crashoverride@hackers.com)

\* password - The encrypted version of the user's password

\* password\_clear - Set to the user's password only when it is being changed.

Otherwise, remains blank.

\* usertype - The role of the user within Joomla!. (Super Administrator, Editor, etc...)

- \* gid - Set to the user's group id, which corresponds to the usertype.
- \* block - Set to '1' when the user is set to 'blocked' in Joomla!.
- \* registerDate - Set to the date when the user was first registered.
- \* lastvisitDate - Set to the date the user last visited the site.
- \* guest - If the user is not logged in, this variable will be set to '1'. The other variables will be unset or default values.

Para receber um dos parâmetros acima:

```
$user =& JFactory::getUser();  
$language = $user->getParam('language', 'the default');
```

```
echo "<p>Your language is set to {$language}.</p>";
```

Garanta que o usuário está logado antes de continuar. A função membro guest deve estar setada para '1' quando o usuário atual não está logado. Quando o usuário está autenticado guest vale '0'.

```
$user =& JFactory::getUser();  
  
if ($user->guest) {  
    echo "<p>You must login to see the content. I want your email address.</p>";  
} else {
```

## Permissão para executar Tarefas

Sabemos que o super administrador está habilitado a editar qualquer conteúdo enquanto que o publisher poderá editar apenas os seus próprios.

A função authorize() pode ser usada para determinar se o usuário atual tem permissão para executar uma determinada tarefa.

```
$user =& JFactory::getUser();  
  
if ($user->authorize('com_content', 'edit', 'content', 'all')) {  
    echo "<p>You may edit all content.</p>";  
} else {  
    echo "<p>You may not edit all content.</p>";  
}  
  
if ($user->authorize('com_content', 'publish', 'content', 'own')) {  
    echo "<p>You may publish your own content.</p>";  
} else {  
    echo "<p>You may not publish your own content.</p>";  
}
```

A função `authorize('com_content', 'edit', 'content', 'all')` recebe 4 parâmetros:

`com_content` - componente ou função  
`edit` - tarefa  
`content` - permissão ou tipo de requisição  
`all` - permissão do dono

### **Podemos criar nossas próprias permissões.**

Adicionar o código a um local que deva ser executado, como o início do componente que estamos criando.

Primeiro receber o objeto `authorizathion` usando a função `getACL()` do `JFactory`.

```
$auth =& JFactory::getACL();

$auth->addACL('com_userinfo15', 'persuade', 'users', 'super administrator');
$auth->addACL('com_userinfo15', 'persuade', 'users', 'administrator');
$auth->addACL('com_userinfo15', 'persuade', 'users', 'manager');

$user =& JFactory::getUser();

if ($user->authorize('com_userinfo15', 'persuade')) {
    echo "<p>You may persuade the system to do what you wish.</p>";
} else {
    echo "<p>You are not very persuasive.</p>";
}
```

Original em inglês: [http://www.jlleblanc.com/joomla/Tutorials/JUser\\_tutorial\\_for\\_Joomla!\\_1%115/](http://www.jlleblanc.com/joomla/Tutorials/JUser_tutorial_for_Joomla!_1%115/)

### **Customizar Backend**

Ocultar mainmenu:

```
JRequest::setVar('hidemainmenu', 1);
```

ou via URI

```
index.php?option=com_mycomponent&task=sometask&hidemainmenu=1
```

Setando título e ícone da toolbar:

```
JToolBarHelper::title('My Title');
```

Via CSS

```
JToolBarHelper::title('My Title', 'myicon');
```

```
// create the CSS
```

```
$iconStyle = '.icon-48-myicon { background-image: url(components/  
    com_mycomponent/assets/icon-48-myicon.png) }';  
// add the CSS to the document  
$document =& JFactory::getDocument();  
$document->addStyleDeclaration($iconStyle);
```



## 7.2.21 - Funções para trabalhar com Navegadores

```
$browser =& JBrowser::getInstance();
if ($browser->hasFeature('javascript'))
{
    // the browser has JavaScript capabilities
}
```

Lista das features que podem ser checadas com hasFeature()

- accesskey
- cite
- dom
- frames
- hdml
- homepage
- html
- iframes
- images
- java
- javascript
- optgroup
- rte
- tables
- utf
- wml
- xmlhttpreq

```
$browser =& JBrowser::getInstance();
if ($browser->hasQuirk('avoid_popup_windows'))
{
    // the browser does not like popups
}
```

```
$browser =& JBrowser::getInstance();
$string = ucfirst($browser->getBrowser()).' ';
$string .= $browser->getVersion().' (';
$string .= $browser->getPlatform().)';
```

```
$browser =& JBrowser::getInstance();
if ($browser->isRobot())
{
    JError::raiseError('403', JText::_('Robots are disallowed'));
}
```

## **Extensões suportadas nos pacotes dos instaladores**

zip, tar, gz, tgz, gzip, bz2, tbz2 e bzip2

## 7.2.22 - Tratamento de Erros

Error Level Error Type Class Method

```
1 (E_ERROR) Error JError::raiseError()
2 (E_WARNING) Warning JError::raiseWarning()
8 (E_NOTICE) Notice JError::raiseNotice()
```

```
JError::raiseError('403', JText::_('Access Forbidden'));
```

```
JError::raiseError('500', JText::_('An error has occurred.'));
```

```
JError::raiseWarning('ERROR_CODE', JText::_('Look out! There is a giant boxing kangaroo behind you!'));
```

```
$result = $SomeController->execute('someTask');
if(JError::isError($result))
{
    // handle invalid task
}
```

Mode Description

Ignore Error is ignored

Echo Prints the JException message to screen

Verbose Prints the JException message and back-trace information to screen

Die Terminates the application and prints the JException message to screen

Message Adds a message to the application queue

Log Adds a log entry to the application error log

Trigger Triggers a PHP error

Callback Calls a static method in another class

```
JError::setErrorHandling(E_NOTICE, 'Ignore');
```

```
define('MY_ERROR', 666);
if( !JError::registerErrorLevel(MY_ERROR, 'My Extension Error',
                               'Message') )
{
    JError::raiseError('SOME_ERROR', JText::_('Error level already defined').'
[.MY_ERROR.]);
}
```

```
JError::raise(MYEXT_ERROR, 'SOME_ERROR', JText::_('Look out!
It's those boxing kangaroos again!'));
```

**Recebendo valores do configuration.php**

Receber o número de registros por página da paginação:  
`$limit = JRequest::getVar('limit', $mainframe->getCfg('list_limit'));`

**CAST (Conversões de unidades)**

`JarrayHelper::toInteger($cid); // Converte $cid para inteiro`

### 7.3 – Criando Extensões

Por mais rico que seja o Joomla e por mais extensões de terceiros que exista ainda assim de vez em quando precisamos criar uma extensão para adicionar uma nova funcionalidade não existente.

Neste momento, o Joomla também facilita a nossa vida. Aqui há necessidade de conhecer a programação em PHP e quanto mais melhor, mas não obrigatoriamente requer um expert. O mais importante é a disposição para aprender.

O Framework do Joomla já vem com funções e classes que facilitam muito comparando com programar diretamente com as funções do PHP, portanto veja o capítulo sobre o Framework e estude ainda outros assuntos do mesmo que considerar necessários. Em especial visite este site para buscar informações:

<http://docs.joomla.org/Developers>

### 7.3.1 – Criando Módulos para o Joomla

Os módulos são as extensões mais usadas no Joomla. Para começo de conversa apenas um componente é permitido para exibição por vez, enquanto que é ilimitada a quantidade de módulos que podemos exibir de cada vez num site em Joomla.

Os módulos também são as extensões mais simples e seu código pode ser extremamente simples como veremos. Claro que existem módulos sofisticados mas geralmente são bem simples.

O site do Joomla tem seu layout controlado por uma outra extensão, o template. No layout do Joomla, geralmente reserva-se a região central para se exibir o conteúdo. Quando solicitamos um artigo ele aparece nessa região. Essa região central é onde são exibidos os componentes, que formam outro tipo de componente. As regiões do layout são controladas pelo template. Os módulos não tem como objetivo ser o conteúdo principal do site, mas sim secundário.

Os módulos são extensões que são exibidas nessas regiões controladas pelo template. Essas regiões (posições) ficam ao redor da região central.

Muitas vezes os módulos trabalham em conjunto com os componentes, mas também podem trabalhar independentes.

Uma diferença entre um módulo e um componente é que o módulo é criado para aparecer no site, por conta disso ao ser criado ou instalado ele não está habilitado e precisamos habilitá-lo para que apareça no site. Logo que habilitamos ele automaticamente já aparece no site. Já o componente ao ser criado ou instalado ele já está habilitado mas não tem recurso para aparecer no site (a não ser que o componente tenha mudado o comportamento padrão do Joomla, como é o caso do Remository). Para o componente aparecer no site precisamos adicionar um item de menu que o chame.

Exemplos de Módulos: Formulário de login, Busca, Menus, Logo, Banner, Popular, Novidades, etc.

Módulos que trabalham com componentes: banner, contato, pesquisa, etc.

Mais detalhes em: <http://docs.joomla.org/Developers>

Então agora iremos criar extensões que aparecerão nestas pequenas regiões ao redor do centro. É bom lembrar isso, pois se nosso módulo tem um formulário ele deverá atentar para que a área é pequena (vide o form de login, onde os rótulos ficam acima dos campos para economizar espaço).

## Estrutura de Arquivos e Diretórios de um Módulo após a Instalação

Supondo o Joomla instalado no diretório "portal" do diretório web do Apache, então um módulo simples chamado alomundo ficaria assim:

### *Diretório*

portal/mod\_alomundo

### *Arquivos*

portal/mod\_alomundo/alomundo.php

portal/mod\_alomundo/alomundo.xml

No mínimo um módulo precisa ter dois scripts, o script em PHP e o script XML.

## Função dos Scripts

O script PHP é o que tem o código que será executado.

O script XML é utilizado para instalar o módulo, levando informações sobre ele, autor, data, versão, posições, arquivos, etc. Também é utilizado pelo backend do Joomla para gerenciar a extensão.

## Ponto de Acesso Único ao Joomla

Por conta da segurança. para executar qualquer coisa no Joomla sempre somos obrigados a passar pela index.php. Isso é conseguido criando-se uma constante na index.php e checando seu valor em todos os demais scripts.

```
index.php  
define( '_JEXEC', 1 );
```

Em todos os demais scripts a primeira linha testa a constante \_JEXEC.

```
defined('_JEXEC') or die('Restricted access');
```

Assim também no backend tudo passa pelo administrator/index.php

## Ponto de Entrada no Módulo

Sempre o ponto de entrada ao módulo, o primeiro arquivo do módulo que será executado, sempre será o arquivo que tem o mesmo nome do módulo com extensão .php.

## Projeto do Módulo Simples

Criaremos um pequeno módulo cuja função será apenas a de mostrar a frase “Olá Mundo do Joomla” na tela do site.

*Criar o diretório:*

mod\_ola

*Dentro do diretório acima criar os arquivos:*

mod\_ola.php

mod\_ola.xml

Neste caso, mesmo que este módulo tivesse diversos outros scripts, o script que será executado em primeiro lugar será o alomundo.php.

### Conteúdo do mod\_ola.xml

```
<?xml version="1.0" encoding="utf-8"?>
<install type="module" version="1.5.0">
<name>Olá Joomla!</name>
<author>Seu Nome</author>
<creationDate>08 de Novembro de 2009</creationDate>
<copyright>(C) 2005 Open Source Matters. All rights reserved.</copyright>
<license>http://www.gnu.org/copyleft/gpl.html GNU/GPL</license>
<authorEmail>seuemail@email.com</authorEmail>
<authorUrl>http://suauri.ext</authorUrl>
<version>1.0.0</version>
<description>Pequeno módulo com a finalidade de mostrar como funciona um
módulo no Joomla 1.5.
</description>
<files>
<filename module="mod_ola">mod_ola.php</filename>
<filename module="mod_ola">index.html</filename>
</files>
</install>
```

Veja que é um XML que contém informações para que o Joomla instale o módulo.

A primeira linha diz o tipo de arquivo (XML) e sua versão e também a codificação (utf-8).

A tag install diz o tipo de extensão e a versão do Joomla.

As demais tags mostram informações sobre o módulo e sobre os arquivos do módulo.

Aqui se for citado um arquivo que não existe no módulo este dispsrará um erro e não poderá ser instalado. Deveremos corrigir isso e empacotar novamente para que possa ser instalado.



### Conteúdo do mod\_alo.php

```
<?php
defined('_JEXEC') or die('Acesso Restrito!');
?>
<b>Olá Mundo do Joomla!</b>
```

Agora devemos compactar a pasta com os scripts, gerando o pacote mod\_olamundo.zip, por exemplo.

Então temos um módulo para o Joomla 1.5 pronto para ser instalado e distribuído. Instale, habilite o módulo e visualize o site. Depois mude sua posição para right, por exemplo e visualize. Veja que é um módulo simples mas com todas as principais funcionalidades de um módulo Joomla.

### Projeto de Módulo que exibe Link

Vamos criar um módulo que exibe um link do Google e ao ser clicado abre a página do Google em outra seção do navegador.

Nós poderíamos apenas criar um item de menu que abrisse o site do Google, mas podemos desejar criar um módulo, pois assim realça mais o link, visto que um módulo tem seu título maior e posso mudar sua posição para uma que considero mais conveniente.

### O nome do módulo será mod\_google.

*Criar o diretório:*  
mod\_google

*Dentro do diretório acima criar os arquivos:*  
mod\_google.php  
mod\_google.xml

### Conteúdo do mod\_google.xml

```
<?xml version="1.0" encoding="utf-8"?>
<install type="module" version="1.5.0">
<name>Linka Google</name>
<author>Seu Nome</author>
<creationDate>08 de Novembro de 2009</creationDate>
<copyright>(C) 2005 Open Source Matters. All rights reserved.</copyright>
<license>http://www.gnu.org/copyleft/gpl.html GNU/GPL</license>
<authorEmail>seuemail@email.com</authorEmail>
<authorUrl>http://suauri.ext</authorUrl>
<version>1.0.0</version>
```

```
<description>Pequeno módulo com a finalidade de abrir o site do Google noutra seção do navegador.
</description>
<files>
<filename module="mod_google">mod_google.php</filename>
</files>
</install>
```

### Conteúdo do mod\_google.php

```
<?php
defined('_JEXEC') or die('Acesso Restrito!');
?>
<a href="http://www.google.com.br" target="_blank">Google</a>
```

### Gerar o Instalador

Empacote e instale.

Habilite e teste no site.

Este pequeno módulo pode ser utilizado para exibir o link de um outro site que deseje.

### Projeto de Módulo para Exibir Conteúdo Estático

Agora vamos criar um módulo que recebe conteúdo web (HTML ou JavaScript) e exibe no site. Este é um módulo que tem utilidade prática.

Para receber o conteúdo usaremos um parâmetro no XML.

Como este módulo pode dar margem a dúvidas sobre seu funcionamento criaremos um pequeno arquivo texto chamado "leiam.txt" que ficará na raiz do módulo explicando a função e como usar o módulo. Isso é uma boa prática, que torna mais eficiente o uso de extensões. Este arquivo não precisa aparecer no XML.

#### *Diretório:*

mod\_conteudo

#### *Arquivos:*

mod\_conteudo.php

mod\_conteudo.xml

leiam.txt

index.html

### Conteúdo do mod\_conteudo.xml

```
<?xml version="1.0" encoding="utf-8"?>
<install type="module" version="1.5.0">
<name>Conteúdo Estático</name>
```

```
<author>Seu Nome</author>
<creationDate>18 de outubro de 2009</creationDate>
<copyright>(C) 2005 Open Source Matters. All rights reserved.</copyright>
<license>http://www.gnu.org/copyleft/gpl.html GNU/GPL</license>
<authorEmail>seu e-mail</authorEmail>
<authorUrl>seu site</authorUrl>
<version>1.0.0</version>
<description>Use o parâmetro na administração do módulo para gerenciar conteúdo na
textarea.
Seu Nome - seu site.
</description>
<files>
<filename module="mod_conteudo">mod_conteudo.php</filename>
</files>
<params>
<param name="conteudo" type="textarea" label="Conteúdo"
description="Conteúdo em HTML ou JavaScript" default="" rows="5" cols="50" />
</params>
</install>
```

### Conteúdo do mod\_conteudo.php

```
<?php
defined('_JEXEC') or die('Restricted access');
$conteudo = $params->get( 'conteudo' ); // receber o valor do parâmetro.
print $conteudo;
?>
```

Agora empacote, instale e entre com conteúdo na textarea do parâmetro.

### Sugestões para o conteúdo.

*Links com cores personalizadas:*

```
<a href="http://cursos.ribafs.org"><font color="red"><b>Cursos Gratuitos
Online</b></font></a><br>
<a href="http://www.google.com.br"><b>Oráculo (sabe tuuuudo...)</b></a><br>
```

*Banner em JavaScript:*

```
<script type="text/javascript"
src="http://www.ubuntu.com/files/countdown/display.js"></script>
```

### Melhorando o Módulo com Parâmetro

Veja que um módulo com os links é mostrado.

Mas será que poderíamos deixar este módulo com a mesma aparência dos módulos de menus?

Aqueles com os cantos arredondados e bordas azuis?

Caso esteja usando o template padrão do Joomla 1.5, então basta adicionar um parâmetro no XML para mostrar o Sufixo de Classe do Módulo.

Bem, vale lembrar que sempre que precisarmos de uma funcionalidade e não soubermos como implementar, podemos ver como outras extensões o fazem. Tanto as extensões do core do Joomla quanto outras extensões de terceiros. Inclusive é bom lembrar de visitar o <http://extensions.joomla.org> sempre antes de criar uma nova extensão, para ver se ela já existe e evitemos reinventar a roda. Mesmo que não exista, pode ser que encontre uma parecida e possa aproveitar algo.

Como já sabemos que o parâmetro fica no script XML do módulo, então vamos dar uma olhada no código de um módulo do Joomla que implementa o sufixo. O módulo mod\_mainmenu tem esse parâmetro. Então vamos até seu xml para dar uma olhada. Na pasta modules encontramos o mod\_mainmenu. Então vamos editar o script mod\_mainmenu.xml:

Como é um script relativamente grande eu procuro por "suffix". Encontro duas ocorrências:

```
<param name="class_sfx" type="text" default="" label="Menu Class Suffix"
description="PARAMMENUCLASSSSUFFIX" />
```

```
<param name="moduleclass_sfx" type="text" default="" label="Module Class Suffix"
description="PARAMMODULECLASSSSUFFIX" />
```

A que precisamos é a Module Class Suffix, então adicionaremos este parâmetro para nosso mod\_conteudo.xml:

```
<param name="moduleclass_sfx" type="text" default="" label="Module Class Suffix"
description="PARAMMODULECLASSSSUFFIX" />
```

Sem nenhuma alteração. Empacotamos novamente, desinstalamos, reinstalamos e entramos para o novo parâmetro com o valor "\_menu" sem as aspas e mantemos sua posição em left. Após salvar e visualizar o site verá que nosso módulo agora se parece com um menu.

Poderá ver um módulo com essas características (inspiração deste exemplo) aqui: <http://cursos.ribafs.org> É o primeiro módulo: "Cursos Disponíveis".

## Projeto de Módulo para Ocultar e Exibir Outros Módulos

Este módulo tem como objetivo exibir um outro módulo mas somente para usuários registrados, ou seja, quando o usuário efetuar login.

Quando o usuário efetuar logoff o outro módulo será ocultado novamente. Nosso módulo

ocultará um segundo módulo para os visitantes.

Veja que nos dois módulos anteriores praticamente não utilizamos código do framework\* do Joomla.

\* Framework - o conjunto de classes das bibliotecas do Joomla que trabalham em harmonia compartilhando informações.

Agora neste módulo começaremos a usar.

#### *Diretório*

mod\_ocultamodulov

#### *Arquivos*

mod\_ocultamodulov.xml

mod\_ocultamodulov.php

### **Conteúdo do mod\_ocultamodulov.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<install type="module" version="1.5.0">
<name>Oculta MóduloV</name>
<author>seu Nome</author>
<creationDate>18 de outubro de 2009</creationDate>
<copyright>(C) 2005 Open Source Matters. All rights reserved.</copyright>
<license>http://www.gnu.org/copyleft/gpl.html GNU/GPL</license>
<authorEmail>e-mail</authorEmail>
<authorUrl>site</authorUrl>
<version>1.0.0</version>
<description>Altere o parâmetro para entrar com o módulo que deseja ocultar.
Este módulo é um exercício do Curso de Joomla 1.5 do Sigma para o Governo do Estado
do Ceará em 11/2009.
Nome - Site.
</description>
<files>
<filename
module="mod_ocultamodulov">mod_ocultamodulov.php</filename>
</files>
<params>
<param name="modulo" type="text" label="Modulo" description="Modulo"
default="mod_nomemodulo" />
</params>
</install>
```

## Conteúdo do mod\_ocultamodulov.php

```
<?php
// no direct access
defined('_JEXEC') or die('Restricted access');
$modulo = $params->get( 'modulo' );
// Aqui acima ao usar $module deu problema. O nome $module deve ser reservado
$user =& JFactory::getUser();
$db =& JFactory::getDBO();
$prefix = $db->getPrefix();
$tbl = $prefix.'modules';
$query = "SELECT published FROM $tbl WHERE module='$modulo'";
$db->setQuery( $query );
$rows = $db->loadObjectList();
foreach($rows as $pub){
    $p = $pub->published;
}
// To guest published. To login unpublished
if($user->guest && $p == 1) { // guest and unpublished
    $query="UPDATE $tbl SET published = 0 WHERE module = '$modulo'";
    $db->setQuery($query);
    $db->query();
    print "<script>window.location.reload();</script>";
}elseif(!$user->guest && $p == 0){ // login and published
    $query="UPDATE $tbl SET published = 1 WHERE module = '$modulo'";
    $db->setQuery($query);
    $db->query();
    print "<script>window.location.reload();</script>";
}
?>
```

Após empacotar, instalar e publicar devemos ocultar o título deste módulo, que não tem função de exibição e entrar o nome do módulo a ser ocultado para o visitante e exibido para o registrado no parâmetro.

Caso ainda não tenha o módulo Formulário de Login, crie agora e publique para testar nosso módulo mod\_ocultamodulov.

Lembre que este módulo tem apenas ação e até parece um plugin, portanto oculte seu título para que não apareça.

Veja outros exemplos de módulos simples e úteis no DVD distribuído no curso, pasta 1material/modulos ou no site da lista:

<http://groups.google.com/group/cursosdejoomla>

Seção de arquivos. Arquivo Modulos.zip e Componentes.zip.

Veja por exemplo, os módulos para consulta da bíblia que precisam ter os respectivos componentes instalados para funcionar (aqui o componente apenas instala o banco).

O capítulo 5 do livro Joomla Avançado é sobre a criação de módulos e ali encontrará também os fundamentos teóricos com exemplos práticos. Vide DVD fornecido em sala, diretório 1material.

O Livro Learning Joomla 1.5 também no capítulo 5 aborda a criação de módulos.

Código Fonte disponível no site da editora:

<http://www.packtpub.com/support/>

Procure pelo Título do Livro " Livro Learning Joomla 1.5 Extension Development" e entre com seu e-mail para receber o código fonte por e-mail.

O Livro Mastering Joomla! 1.5 aborda a criação de módulos também no capítulo 5. Código fonte disponível também no site acima.

Na s site do Joomla também encontramos exemplos de criação de módulos:

<http://docs.joomla.org/Developers#Modules>

## 7.3.2 - Criando um Módulo para adicionar o Logo do Site no Joomla 1.5

Caso o Template ainda não tenha um ID que atenda ao desejado

### 1) Edite o CSS do Template e adicione:

```
#logo {  
position:absolute;  
top:40px;  
left:150px;  
}
```

E comente o ID `div#logo` { existente com `/* ... */`

Ajuste top e left de acordo com seu template e o Logo ou deixe para ajustar após visualizar.

Obs.: Claro que se no CSS já existir um ID logo use outro.

### 2) Adicionar mais uma posição ao `templateDetails.xml`, ao final da relação:

```
<position>logo</position>
```

### 3) Criar um Módulo HTML Personalizado

- Extensões - Administrar Módulo - Novo - HTML Personalizado
- No Título entre Logo
- Em Exibir Título marque Não
- Em Posição selecione logo
- Role a tela, adicione a imagem do logo e clique em Salvar
- Uma boa idéia é adicionar link para o próprio site ou se for um banner, o link para o respectivo site.

### 4) Adicionar ao final do `index.php` do template:

```
<?php if($this->countModules('logo')) : ?>  
<div id="logo">  
<jdoc:include type="modules" name="logo" style="xhtml" />  
</div>  
<?php else : ?>  
<?php endif; ?>
```

Remova ou comente o `id="logo"` existente no `index.php`

Agora para trocar a logomarca basta editar o módulo Logo e trocar a imagem. Lembre de adicionar um link à imagem para a index do site.



### 7.3.3 – Criando Componentes para o Joomla

Componentes são mini-aplicativos que rodam em um site Joomla ou na seção de administração. Geralmente têm uma área sofisticada na seção de administração.

Componentes são extensões maiores, geralmente, que os módulos e quando exibidos no site ocupam a região central do site. Temos um componente que exibe o conteúdo dos artigos nessa região e que já vem por default na maioria dos templates.

Os módulos são muito utilizados para exibir conteúdo processado pelos componentes. Como ocupa a região central, somente um componente pode aparecer no site de cada vez, ao contrário do módulo, que pode aparecer em várias posições.

Exemplos de componentes: banner, contato, enquete, os que oferecem arquivos para download (Phoca, Remository), como as galerias de imagens (Phoca, MorpheuShow), como é o caso dos gerenciadores de backup, sistemas de comentários, como o JoomlaComments, entre outros. Todos estes, ao ser chamados, ocupam a região central do site.

Uma outra diferença entre os módulos e os componentes é que os módulos, no backend, tem apenas alguns poucos recursos de personalização, enquanto que os componentes algumas vezes chegam a ter até mais recursos em sua área administrativa até que o Joomla, como é o caso do Remository e do JoomlaPack.

Geralmente a interface dos componentes na administração oferecem recursos completos para gerenciar o banco de dados para o componente (CRUD), enquanto que a interface dos módulos oferece apenas alguns poucos recursos para exibição.

Alguns componentes tem apenas funções administrativas e não aparecem no site, como é o caso do NinjaXplorer e do JoomlaPack (este tem acesso pelo site mas opcional).

Componentes ocupam a região central da página enquanto que módulos ocupam as regiões ao redor da região central. Módulos podem aparecer em qualquer página e componentes tem sua página para aparecer.

Tanto módulos quanto componentes podem ser criados apenas para efetuar algumas operações que não seja mostrar conteúdo.

Mais detalhes em: <http://docs.joomla.org/Developers>

Aqui, quando vamos criar o componente já devemos ter em mente que ele ocupará a região central do site e em comparação com o módulo, terá uma região maior à sua disposição.

**Alerta:** lembre que um componente precisa ter nome único, sob pena de nem instalar. Ao informar a tag <name> no XML devemos informar um nome único.

Os módulos, os templates e os plugins têm sua programação simplificada pelo framework do Joomla, usando parâmetros e outras facilidades, mas os componentes precisam de um maior esforço por parte do programador. O que num módulo é um parâmetro no XML, num componente é um formulário na parte administrativa. Os componentes tem como função inicial exibir os artigos, seções e categorias.

### Fluxo das Informações

solicita-se portal/index.php  
carga do framework  
solicita portal/templates/templateX/index.php  
ativa jdoc:component (que chama geralmente o primeiro link do menu)  
ao mesmo tempo em que carrega os módulos disponíveis

### Fases de Criação do Componente para o Site

Existem componentes que serão usados no site (frontend) e outros que serão usados no admin (backend) do Joomla.

1 – Componente para mostrar um simples "Olá mundo" (sem e com \_JEXEC)  
Chamar pela URL, já que o componente não está registrado mas sim será criado dentro da própria estrutura do Joomla.

Estrutura de arquivos e diretórios de um componente para o site do Joomla.  
Supondo que o Joomla tenha sido instalado no diretório portal.

portal/components/com\_simples

Então iniciaremos criando o diretório com\_simples em portal/components.  
Agora vamos criar o arquivo simples.php no diretório com\_simples.

Edite o script simples.php e digite as três linhas abaixo:

```
<?php  
print "Meu primeiro componente para Joomla!";  
?>
```

Agora vamos executar nosso componente. Isso já é um componente, embora ainda não tenha praticamente utilidade alguma. Como ele ainda não está registrado, vamos chamá-lo pela URL:

[http://localhost/portal/index.php?option=com\\_simples](http://localhost/portal/index.php?option=com_simples)

Agora renomeie o arquivo simples.php para qualquer outro nome com extensão PHP. Veja que existe um acordo entre o programador e o Joomla. O arquivo não pode ter um nome qualquer, mas deve ter o mesmo nome do componente apenas sem o prefixo com e o

sublinhado.

com\_simples → simples.php

com\_alo → alo.php

Outro detalhe importante é que no Joomla tudo passa pelo script index.php, pois existe uma constante definida neste script que é verificada em todos os demais. A constante é `_JEXEC` e é definida no index.php como:

```
define( '_JEXEC', 1 );
```

Em cada outro script a ser executado ela é testada. No nosso caso, ainda não adicionamos essa constante, mas em termos de segurança é importante adicionar em todos os scripts. Vamos tentar chamar o arquivo passando seu endereço completo:

[http://localhost/portal/components/com\\_simples/simples.php](http://localhost/portal/components/com_simples/simples.php)

Experimente e veja que será executado sem problemas.

Agora edite novamente o script simples.php e adicione a linha da constante para que fique assim:

```
<?php
defined('_JEXEC') or die( 'Acesso Restrito!' );
print "Meu primeiro componente para Joomla!";
?>
```

Agora execute novamente com a URL anterior e verá que não será executado mas receberemos uma mensagem da constante:

[http://localhost/portal/components/com\\_simples/simples.php](http://localhost/portal/components/com_simples/simples.php)

Mas se chamar passando pela porta de entrada, ou seja, pela index.php:

[http://localhost/portal/index.php?option=com\\_simples](http://localhost/portal/index.php?option=com_simples)

Nesse caso funcionará sem problema.

## 2 – Usar jDump para melhorar debug

Agora vamos testar uma pequena ferramenta que ajuda no debug do Joomla. Antes vamos ativar o debug existente.

Alerta: não ative em site em produção, somente em site em testes localmente.

Configuração Global – Sistema

Configurações de Depuração – Ativar Depuração de Sistema – Sim – Salvar

Agora cada erro ecoará exibindo na tela uma grande quantidade de informações.  
Instalando a Ferramenta jDump

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

Faça o download do componente e do plugin.

Instale ambos e habilite o plugin.

### Usando o jDump

Funções disponíveis:

`dump('variavel', 'observação')`

`dumpMessage()`

`dumpSysinfo()`

`dumpTrace()`

`dumpTemplate()`

Ver Administrator - jDump - Preferências e informações

`dump($nome, 'Nome do usuário');`

Vamos testar na próxima etapa, onde nosso componente receberá algum código.

Já que vimos como funciona um componente no Joomla agora veremos um componente com algo mais interessante.

### Projeto Componente Cumprimentos

Vamos criar um novo componente chamado `com_cumprimentos`. Este componente será separado em duas camadas, a camada de apresentação e a camada de lógica. Ele terá como objetivo cumprimentar os visitantes de acordo com os feriados cadastrados. Caso seja natal desejará "Feliz Natal".

*Criar o diretório*

`portal/components/com_cumprimentos`

*Crie dentro do diretório `com_cumprimentos` o script*

`cumprimentos.php`.

Agora edite o script `cumprimentos.php` (camada de lógica) e adicione o seguinte conteúdo:

```
<?php
defined('_JEXEC') or die('Acesso restrito!');
jimport('joomla.application.helper');
require_once(JApplicationHelper::getPath('html')); // No admin usar admin_html
```

```
//require_once('cumprimentos.html.php'); // Como se fosse isso
$arefa = JRequest::getVar('tarefa',' ');
$nome = JRequest::getVar('nome',' ');

switch ($arefa) {
case 'mostrar':
ApresentacaoCumprimentos::apresentaNome($nome);
break;

default:
ApresentacaoCumprimentos::apresentaDefault();
break;
}
?>
```

**Dica:**

O import: jimport('joomla.application.helper'); corresponde a:  
include('libraries/joomla/application/helper.php');

Agora o script cumprimentos.html.php, que conterà a camada de apresentação:

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

function apresentaNome($nome) {
print 'Como vai ' . $nome . '? Bem-vindo ao Curso de Joomla!';
}

function apresentaDefault() {
print 'Como vai visitante? Bem-vindo ao Curso de Joomla!';
}
}
?>
```

Agora vamos chamar o navegador com uma URL adequada para testar nosso componente:

[http://localhost/portal/index.php?option=com\\_cumprimentos&tarefa=mostrar&nome=Ribamar](http://localhost/portal/index.php?option=com_cumprimentos&tarefa=mostrar&nome=Ribamar)

Agora testemos quando não for entrado um nome:

[http://localhost/portal/index.php?option=com\\_cumprimentos](http://localhost/portal/index.php?option=com_cumprimentos)

**Como as informações caminham aqui?**

Acessa-se o arquivo cumprimentos.php, que chama um dos métodos da classe ApresentacaoCumprimentos no script cumprimentos.html.php.

Veja agora como é o mesmo componente mas com apenas uma camada:

```
<?php
defined('_JEXEC') or die('Acesso restrito!');
jimport('joomla.application.helper');

$arefa = JRequest::getVar('arefa', ' ');
$nome = JRequest::getVar('nome', ' ');

switch ($arefa) {
case 'mostrar':
print "Olá ".$nome. ", tudo bem?";
break;
default:
print "Como vai visitante. Sinta-se em casa!";
break;
}
?>
```

Veja que para separar em duas camadas, removemos para outro arquivo a parte correspondente à apresentação, apenas as linhas com os comando print.

Caso alguém questione que em duas camadas tem o mesmo efeito e dá mais trabalho, realmente é verdade. Mas isso somente é verdade em um aplicativo deste tamanho, pois quando começa a crescer e a ficar complexo, então percebe-se o benefício da separação em camadas.

Programar assim, em uma única camada, para programas simples não tem problema, mas quando o programa começa a crescer e a ficar complexo devemos dividir pelo menos em duas camadas e idealmente em três usando MVC.

### 7.3.4 - Desenvolvendo Componentes usando MVC

Um software do tipo framework é a base de uma aplicação que pode ser usada por um desenvolvedor.

O framework do Joomla 1.5 desencadeou um grande acordo de poder.

MVC – Model-View-Controller – é um padrão de projeto de software que pode ser usado para organizar código, a maneira como as regras de negócio e a apresentação dos dados é separada.

MVC foi originalmente desenvolvido para mapear a tradicional regra de entrada, processamento e saída em uma arquitetura lógica de GUI.

**Model** – encapsula os dados da aplicação. No Joomla o model contém métodos para adicionar, remover, receber e atualizar informações do banco de dados. São ativas representações das tabelas do banco de dados: podem conectar para o banco de dados, efetuar consultas nele (se instruído para fazer isso por um controller) e salvam dados no banco de dados. Na arquitetura MVC não deve haver interação entre modelos e views.

Toda a lógica é manipulada pelos controllers.

Se uma aplicação está sendo migrada de um sistema que utiliza arquivos texto para uma aplicação que utiliza bancos de dados. Neste caso o modelo será a única camada que precisará ser alterada, nada na view nem no controller.

**Controller** – é responsável por responder às ações dos usuários. No caso da web, uma ação de um usuário geralmente é uma requisição de página (link, submit de form, etc). O controller deve determinar que requisição foi feita pelo usuário e responder apropriadamente disparando o modelo para manipular os dados apropriadamente e passar os dados para a view. contém a lógica da aplicação. Cada controller pode oferecer diferentes funcionalidades. Controles recebem e modificam os dados acessando as tabelas do banco de dados através do modelos. Registram variáveis e objetos que podem ser usados nas views.

**View** – usada para renderizar os dados do model, recebidos através do controller, de uma maneira adequada para a interação com o usuário. O Joomla mostra na view os dados em forma de HTML, PDF ou para impressão. podem ser descritos como arquivos de templates que apresentam seu conteúdo para o usuário: variáveis, arrays e objetos que são usados em views são registrados através de um controller.

Views não deveriam conter complexas lógicas de negócios, somente a estrutura de controle elementar necessária para executar operações particulares como com a iteração de coleções de dados através de uma construção foreach, deve ser continua na view.

View e model não devem ter nenhuma integração. A view não deve modificar dados, apenas exibir os dados recebidos.

O uso do padrão de projeto MVC é indicado para projetos de grande e médio porte, onde dará ganho de produtividade evitando a complexidade de manutenções por separar sem as camadas. A separação de camadas aumenta a flexibilidade e a reutilização.

O Joomla oferece em seu Framework uma API para desenvolvimento usando o padrão

## **MVC:**

**Models** - `jimport( 'joomla.application.component.model' );`

**Controllers** - `jimport( 'joomla.application.component.controller' );`

**Views** - `jimport( 'joomla.application.component.view' );`

## **Estrutura de arquivos e diretórios de um componente MVC no Joomla 1.5**

Temos componentes tanto para o site quanto para o admin e a estrutura de arquivos é a mesma, com a diferença de um ficar no site (portal) e o outro no admin (portal/administrator):

### **/site**

Este diretório guarda os arquivos da parte do frontend do componente.

`site/{nomedocomponente}.php`

Este é o arquivo principal do componente e o ponto de entrada para a parte frontend.

`/site/controller.php`

Este arquivo manipula o controller default do frontend, que contém uma classe chamada "`{NomeComponente}Controller`". Esta classe precisa estender a classe base "`JController`".

`/site/views`

Este diretório manipula as diferentes views para o componente.

`/site/views/{nomedaview}`

Este diretório manipula os arquivos para a view `{NomeViewName}`.

`/site/views/{viewname}/view.html.php`

Este é o ponto de entrada para a view `{ViewNome}`. Este deve declarar a classe `{ComponentNome}View{ViewNome}`. Esta classe precisa estender a classe base "`JView`".

`/site/views/{viewname}/tmpl`

Este diretório manipula os arquivos do template para a view `{ViewNome}`.

`/site/views/{viewname}/tmpl/default.php`

Este é o template default para a view `{ViewNome}`.

`/site/models`



---

Este diretório manipula modelos adicionais, caso seja necessário para a aplicação.

/site/models/{modelnome}.php

Este arquivo manipula a classe model {ComponentNome}Model{ModelNome}. Esta classe precisa estender a classe base "JModel". Note que a view nomeada {ViewNome} deve por default carregar um model chamado {ViewNome} se essa existir. Os models precisam ser nomeados após a sua view ter a intenção de ser usada com ele.

/site/controllers

Este diretório manipula controllers adicionais, caso a aplicação necessite.

/site/controllers/{controllername}.php

Este manipula a classe controller {ComponentNome}Controller{ControllerNome}. Esta classe precisa estender a classe base "JController".

## **/admin**

Para o admin a estrutura é idêntica.

Adaptado de [http://docs.joomla.org/File\\_Structure\\_and\\_Naming\\_Conventions](http://docs.joomla.org/File_Structure_and_Naming_Conventions) onde encontrará a versão em inglês. Caso encontre alguma divergência favor me avisar para que eu atualiza (ribafs@gmail.com).

Exemplo simples para um componente do site

/site

site/ola.php

/site/controller.php

/site/views

/site/views/Saudacao

/site/views/Saudacao/view.html.php

/site/views/Saudacao/tmpl

/site/views/Saudacao/tmpl/default.php

/site/models

/site/models/AloModelSaudacao.php

/site/controllers

/site/controllers/OlaControllerSaudacao.php

Obs.: podemos ter várias views, cada uma em seu sub-diretório dentro de views e com estrutura semelhante.

Um bom exemplo de extensão usando o padrão MVC pode ser encontrado nos fontes do livro do Joseph LeBlanc (vide referência) em seu capítulo 9. O livro Joomla Avançado também aborda no seu capítulo 4.

A classe JModel já encapsula o banco de dados, por isso dispensa que usemos: JFactory::getDBO()

para ter acesso ao banco. Basta usar  
`$this->_db`

As views basicamente implementam o método display de JView.

Algumas vezes o contrller apenas precisa ser definido sem implementação, pois herdará de JController.

A implementação se faz necessária apenas quando precisamos determinar qual view deve ser instanciada, indicar o layout ou tipo de documento a ser aplicado e então sobrescrever apenas o método display().

O uso Messages Queue torna as mensagens de erro e de alerta bem mais atraentes. O livro do James Kennard as aborda no capítulo 9.

### **Adicionando CSS na View**

Para adicionar o nosso CSS na view e assim nossa extensão não usar o do template mas o nosso CSS devemos criar o diretório:

`com_alo/assets`

E nele adicionar o arquivo `saudacao.css`

Depois devemos editar:

`com_alo/views/saudacao/tmpl/default.php`

E adicionar o `include` para o nosso CSS.

Lembrar que no CSS o comando `!important` força a precedência deste artigo sobre os demais.

### 7.3.5 – Criando Plugins para o Joomla

Os plugins são extensões destinadas a executar código quando for disparado algum dos eventos do Joomla. O Joomla também oferece possibilidade de o programador criar seus próprios eventos para que o plugin funcione de acordo com eles.

São utilizados quando uma peça de código é necessária em todo o site. São geralmente utilizados para formatar a saída de um componente ou módulo quando imediatamente antes da página ser construída.

Diferente dos componentes e módulos, os plugins não tem uma pasta para cada um, eles são agrupados por tipo. Todos do mesmo tipo ficam numa única pasta.

Se os módulos são feitos para aparecer no site, assim como os componentes aparecem o site ou na administração, geralmente os plugins não aparecem, somente executam alguma verificação (autenticação do usuário), alteração no conteúdo antes que seja exibido.

Com um plugin podemos interceptar um artigo antes que seja exibido, alterá-lo e exibir com as alterações.

Plugins usam o padrão de projeto Observer.

Neste padrão existe um objeto central que vigia alterações em seu próprio estado. Este objeto é observado por todos os Observers. Cada observer se registra e então informa ao objeto central a sua existência.

Quando o objeto central muda de estado então notifica a todos os Observers. Então os

Observers executam as ações como consequência da notificação.

O objeto JeventDispatch notifica a todos os plugins registrados para responder a um certo evento.

Os plugins afetam o conteúdo de diferentes maneiras.

#### Existem plugins de alguns tipos já definidos no core do Joomla:

- authentication – autenticação de usuários no processo de login
- content – processa itens de conteúdo antes de serem exibidos
- editors – editores WYXIWYG que podem ser usados para editar conteúdo
- editors-xtD – extensões para editores (normalmente botões adicionais ao final do editor)
- search – extender as funcionalidades do componente search
- system – escutadores (listeners) dos eventos do sistema
- user - processam usuários quando ações são executadas
- xmlrpc – criam respostas XML-RPC

Segundo a Wikipédia ([http://pt.wikipedia.org/wiki/Chamada\\_de\\_procedimento\\_remoto](http://pt.wikipedia.org/wiki/Chamada_de_procedimento_remoto)):  
RPC - Chamada Remota de Procedimento (ou RPC, acrônimo de Remote Procedure Call) é uma

tecnologia de comunicação entre processos que permite a um programa de computador chamar um procedimento em outro espaço de endereçamento (geralmente em outro computador, conectado por uma rede). O programador não se preocupa com detalhes de implementação dessa interação remota: do ponto de vista do código, a chamada se assemelha a chamadas de procedimentos locais.

RPC é uma tecnologia popular para a implementação do modelo cliente-servidor de computação distribuída. Uma chamada de procedimento remoto é iniciada pelo cliente enviando uma mensagem para um servidor remoto para executar um procedimento específico. Uma resposta é retornada ao cliente. Uma diferença importante entre chamadas de procedimento remotas e chamadas de procedimento locais é que, no primeiro caso, a chamada pode falhar por problemas da rede. Nesse caso, não há nem mesmo garantia de que o procedimento foi invocado.

Adicionalmente nós podemos definir outros tipos de plugins.

Os plugins reagem a qualquer evento mas somente aos do seu grupo.

### **Lista de Eventos em Ordem Alfabética**

- onAfterContentSave (content)
- onAfterDeleteUser (user)
- onAfterDisplay (content)
- onAfterDisplayTitle (content)
- onAfterDisplayContent (content)
- onAfterInitialise (system)
- onAfterRoute (system)
- onAfterDispatch (system)
- onAfterRender (system)
- onAfterStoreUser (user)
- onAuthenticate (authentication)
- onAuthenticateFailure (authentication)
- onBeforeContentSave (content)
- onBeforeDeleteUser (user)
- onBeforeDisplay (content)
- onBeforeDisplayContent (content)
- onBeforeStoreUser (user)
- onCustomEditorButton (editors-xtd)
- onDisplay (editors)
- onGetContent (editors)
- onGetInsertMethod (editors)
- onGetWebServices (xmlrpc)
- onInit (editors)
- onLoginFailure (user)
- onLogoutFailure (user)
- onLoginUser (user)
- onLogoutUser (user)
- onPrepareContent (content)

onSave (editors)  
onSearch (search)  
onSearchAreas (search)  
onSetContent (editors)  
onSubmitContact (contact)  
onValidateContact (contact)

Fonte: <http://docs.joomla.org/Tutorial:Plugins>

### Código para Implementação Básica de Plugin

NomeDoPlugin.php

```
<?php
defined('_JEXEC') or die('Acesso Restrito!');

jimport('joomla.event.plugin');

class plgTipoDoPluginNomeDoPlugin extends JPlugin
{
    function plgTipoDoPluginNomeDoPlugin(&$subject)
    {
        // Chamar o construtor da super classe
        parent::__construct($subject);

        // Obtém os parâmetros relacionados ao plugin
        $this->_plugin => JPluginHelper::getPlugin('TipoDoPlugin', 'NomeDoPlugin');
        $this->_params = new JParameter($this->_plugin->params);
    }
    /**
     * Tratamento de eventos. São métodos desta classe com o mesmo nome
     * dos eventos.
     */
    function NomeDoEvento()
    {
        global $mainframe;
        // O código para o funcionamento do plugin
        return true;
    }
}
?>
```



**Outra forma de implementar plugin no Joomla:**

NomeDoPlugin.php

```
<?php
defined('_JEXEC') or die('Acesso Restrito!');

jimport('joomla.event.plugin');
$mainframe->registerEvent('NomeDoEvento', 'NomeDaFuncao');

function NomeDaFuncao
{
    // Código para o funcionamento desta função em relação ao evento NomeDoEvento
}
?>
```

**Eventos por Grupos****Grupo Content**

**onPrepareContent** – Este é o na preparação de conteúdo para exibir. É o evento mais comum dos plugins orientados a conteúdo. Desde que o artigo e parâmetros relacionados sejam passados por referência então os manipuladores de eventos podem modificar o conteúdo antes que seja exibido.

**Parâmetros deste evento:**

&\$article – uma referência ao conteúdo do artigo

&\$params – referência aos parâmetros que são carregados junto ao conteúdo

\$limitstart – número da página inicial

return – true se sucesso

**OnBeforeDisplayContent** – Esta é uma requisição de informação que deve ser adicionada imediatamente antes do conteúdo ser gerado. Para views que geral XHTML/HTML elas devem incluir o uso de estilo que estiver especificado como parte do conteúdo ou parâmetros relacionados. Este não é um evento para modificar conteúdo. Caso queira modificar use onPrepareContent.

**Parâmetros**

&\$article – referência ao conteúdo do artigo

&\$params – referência aos parâmetros que são carregados juntamente com o conteúdo.

\$limitstart – número da página inicial

return – retorna uma string em XHTML que será apresentada antes do conteúdo.

**Exemplo:** Criar um pequeno plugin que inserirá uma frase em XHTML entre o título e o texto de um artigo. Grave na pasta

portal/plugins/content  
exemplo.php:

```
<?php
defined( '_JEXEC' ) or die( 'Acesso Restrito!' );
class plgContentExemplo extends JPlugin {
function onBeforeDisplayContent( &$amp;article, &$amp;params, $limitstart )
{
return '<H1> Exemplo do tratamento do Evento <B>onBeforeDisplayContent</B></H1>';
}
}
?>
```

### Registrando

Registre no banco adicionando este registro à tabela jos\_plugins:

```
INSERT INTO jos_plugins (name,element,folder, published) values ('Exemplo Plugin
Conteúdo','exemplo','content',1);
```

### Testando

Crie o artigo com nome Exemplo Plugin Conteúdo e adicione algum texto. Abra no site para ver o resultado do plugin.

**onAfterDisplayContent** – Esta é também uma requisição de informação que deve ser colocada imediatamente após a geração do conteúdo. Para views que geram XHTML/HTML elas devem incluir o uso de estilo que estiver especificado como parte do conteúdo ou parâmetros relacionados. Este não é um evento para modificar conteúdo. Caso queira modificar use onPrepareContent.

### Parâmetros

`&$article` – referência ao conteúdo do artigo

`&$params` – referência aos parâmetros que são carregados juntamente com o conteúdo.

`$limitstart` – número da página inicial

`return` – retorna uma string em XHTML que será apresentada após o conteúdo.

**onAfterDisplayTitle** – Esta é mais uma requisição de informação que deve ser adicionada entre o título do conteúdo e o corpo do mesmo. Este não é um evento para modificar conteúdo. Caso queira modificar use onPrepareContent.

### Parâmetros

`&$article` – referência ao conteúdo do artigo

`&$params` – referência aos parâmetros que são carregados juntamente com o conteúdo.

`$limitstart` – número da página inicial



return – retorna uma string em XHTML que será apresentada após o título do artigo.

### **Grupo User**

**onBeforeStoreUser** – Este evento é disparado imediatamente antes de se inserir um novo usuário no banco ou antes de atualizar um usuário existente.

#### **Parâmetros**

\$user – array com informações sobre o usuário

\$id

\$isNew – verdadeiro para novos usuários

return – não existe retorno

**onAfterStoreUser** – Este evento é disparado imediatamente após se inserir um usuário no banco ou atualizar um existente.

#### **Parâmetros**

\$user – array com informações sobre o usuário

\$isNew – verdadeiro para novos usuários

\$success – verdadeiro se usuário inserido sem problema

\$msg – mensagem de erro caso a tentativa de inserir o usuário tenha falhado

return – não existe retorno

**onBeforeDeleteUser** – Este evento é disparado imediatamente antes da exclusão do usuário do sistema.

#### **Parâmetros**

\$user – array com informações sobre o usuário

return – não existe retorno

**onAfterDeleteUser** – Este evento é disparado imediatamente após a exclusão do usuário do sistema.

#### **Parâmetros**

\$user – array com informações sobre o usuário

\$success – retorna true caso a exclusão tenha sido bem sucedida

\$msg – mensagem de erro para exclusão mal sucedida

**OnLoginUser** – Este evento é disparado após o usuário ser autenticado pelo Joomla como um usuário do sistema. Se o retorno for false a autenticação falhou.

#### **Parâmetros:**

\$user – array com dados do usuário

\$option – array com opções extras

**onLogoutUser** – Este evento é disparado antes do usuário efetuar o logout do sistema. Se o retorno for false o logout falhou.

**Parâmetros:**

\$user – array com credenciais do usuário

return – true ou false (para sucesso ou falha da ação)

**onLoginFailure** – Este evento é disparado sempre que a autenticação de um usuário falha.

**Parâmetros:**

\$user – array com dados do usuário

JAuthenticaeResponse que causou a falha

### Grupo Authentication

**onAuthenticate** – Este evento é disparado sempre que um login for válido.

**Parâmetros:**

\$credentials – array com dados do usuário

### Grupo Editor

**onInit** – Ações executadas no momento em que o editor é instanciado.

**Parâmetros:**

Sem parâmetro

return – tag XHTML a ser adicionada ao cabeçalho do documento.

**onDisplay** – tratamento do conteúdo ao ser mostrado.

**Parâmetros:**

\$name – nome do campo da área de edição

\$content – conteúdo do editor

\$width – largura do editor em pixel

\$height – altura do editor em pixel

\$col – largura do editor em colunas

\$row – altura do editor em linhas

\$buttons – true ou false para mostrar/ocultar os botões que tratam os eventos do grupo editor-xtd

return – texto XHTML para usar no editor

**onGetContent** – tratamento do conteúdo do editor quando for solicitado

**Parâmetros:**

\$editor – nome do editor

\$return – conteúdo

**onSetContent** – tratamento do conteúdo antes que seja armazenado no editor.

**Parâmetros:**

\$editor – nome do editor

\$html – conteúdo do editor

return – JavaScript que, ao ser executado, atribui a \$html o conteúdo do editor.

**Grupo Editor-xtd**

**onDisplay** – executado no momento em que o editor é instanciado.

**Grupo System**

**onAfterInitialize** – acontece quando o usuário, através no navegador, faz uma rebusca ao Joomla (aplicativo).

**OnAfterRoute** – é disparado depois de o framework ter sido carregado, a aplicação ter sido iniciada, o evento onAfterInitialize ter sido disparado e roteador ter direcionado a requisição do cliente a um componente. Nesse contexto o roteador é um elemento do Joomla que examina uma requisição e determina qual componente deve tratar essa requisição.

**OnAfterDispatch** – acontece depois de o framework ter sido carregado e o conteúdo ter sido enviado para o buffer.

**OnAfterRender** – acontece depois de o framework acomodar o conteúdo no template e enviá-lo à área de resposta (conteúdo final que será enviado ao navegador). Ocorre antes de “response” ser enviado.

Detalhes: <http://docs.joomla.org/Tutorial:Plugins>

**Criando um Plugin Simples**

Para criar um plugin precisamos basicamente de dois arquivos, um PHP e o XML.

Veja agora o código de um plugin bem simples, que sempre exibirá a mensagem Olá

Mundo quando abrirmos qualquer artigo. Este será um plugin do grupo system.

```
plg_ola  
ola.xml  
ola.php  
index.html
```

```
ola.php
```

```
<?php
defined( '_JEXEC' ) or die( 'Restricted access' );
jimport( 'joomla.plugin.plugin' );
/**
 * Example system plugin
 */
class plgSystemOla extends JPlugin
{
function plgSystemOla( &$subject, $config )
{
parent::__construct( $subject, $config );
// Do some extra initialisation in this constructor if required
}
// Ao inicializar do Portal exibirá a mensagem Olá Mundo
function onAfterInitialise()
{
print '<font color="white"><h1>Olá Mundo</h1></font>';
}
}
?>
```

### **ola.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<install version="1.5.2" type="plugin" group="system" method="upgrade">
<name>System - Ola</name>
<author>Autor</author>
<creationDate>Mês Ano</creationDate>
<copyright>Copyright (C) 2008 Holder. All rights reserved.</copyright>
<license>GNU General Public License</license>
<authorEmail>email</authorEmail>
<authorUrl>url</authorUrl>
<version>1.0.1</version>
<description>Plugin Olá mundo para o Curso de Joomla 1.5</description>
<files>
<filename plugin="ola">ola.php</filename>
<filename plugin="ola">index.html</filename>
</files>
<params>
<param name="ola"
type="text"
default=""
label="Exemplo"
description="Um exemplo de parâmetro tipo texto" />
</params>
</install>
```

Fonte: Código adaptado de exemplo da documentação do Joomla:

<http://developer.joomla.org/tutorials/184-how-to-create-a-joomla-plugin.html>

Código do index.html é semelhante aos anteriores.

### **Gerar Instalador**

Empacotar tudo como plg\_ola

### **Testando**

Instalar e habilitar. Então apenas vá até o site e tecele F5.

### **Exemplo Plugin "Abreviações"**

Veja agora o código de um plugin simples que substitui algumas abreviações pelo seu respectivo valor por estendendo em todos os artigos criados a partir da publicação deste plugin.

plg\_abreviacoes  
abreviacoes.xml  
abreviacoes.php  
index.html

#### **abreviacoes.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
<install version="1.5" type="plugin" group="content">
<name>Content - Substituir Abreviações</name>
<author>Dan Rahmel. Adaptação de Ribamar FS</author>
<creationDate>July 2007</creationDate>
<authorEmail>admin@joomlajumpstart.com</authorEmail>
<authorUrl>www.joomlajumpstart.com</authorUrl>
<version>1.0</version>
<description>Substitui abreviações existentes em artigos por seu respectivo
valor.</description>
<files>
<filename plugin="abreviacoes">abreviacoes.php</filename>
</files>
<params>
<param name="replace_limit" type="text" size="5" default="50" label="Limite de
Busca" description="Quantos itens retornar"/>
</params>
</install>
```

**abreviacoes.php**

```
<?php
defined( '_JEXEC' ) or die( 'Restricted access' );
class plgContentAbreviacoes extends JPlugin {
// PHP4 compatible constructor
function plgContentAbreviacoes( & $subject ) {
parent::__construct( $subject );
}
// Name function same as event so it will be called.
function onPrepareContent(&$article, &$params, $limitstart=0) {
// Get the parameters in case we need them in a later plug-in
$plugin =& JPluginHelper::getPlugin('content', 'abreviacoes');
$pluginParams = new JParameter( $plugin->params );
$num =1;
if (!$pluginParams->get('replace_limit', 1)) {
$num = $pluginParams->get('replace_limit', 1);
}
if ( !JString::strpos( $article->text, '{abreviacoes=off}' ) === false ) {
$article->text = str_replace( '{abreviacoes=off}',
", $article->text );
return true;
}
$article->text =$this->abreviacoesStr($article->text, $num);
return true;
}
function abreviacoesStr($myStr, $num) {
// Create an array of text abbreviations
$abbr=array( "PHP", "CMS", "CE", "BR"
);
// Create a matching array of the phrases the words represent.
// These are coded two per line so they are more human readable
// making it easier to match them up to the above array.
$sactual=array("Personal Home Page", "Content Management System",
"Ceará", "Brasil");
// Add the regular expression notation to the abbreviations in the
// array so the strings will only be matched if they are found as
// lone words. Also make search case insensitive.
for($i=0;$i < sizeof($abbr);$i++) {
$abbr[$i] = "\\b" . $abbr[$i] . "\\b/i";
}
// Do a regular expression replace
$myStr = preg_replace ( $abbr, $sactual, $myStr, $num);
// Return altered string
return $myStr;
}
}
```

```
// $myPlugin =& new plgContentAbreviacoes( JEventDispatcher::getInstance() );  
// Agora JEventDispatcher mudou para JDispatcher  
$myPlugin =& new plgContentAbreviacoes( JDispatcher::getInstance() );  
?>
```

**Adicione também nosso index.html.**

Empacote os três juntos e instale.

**Testando o Plugin**

Após a instalação crie um artigo contendo as abreviações:

PHP, CMS, CE, BR e veja que o plugin fará a substituição.

Isso dá para imaginarmos outros usos para esta idéia, como censura de algumas palavras.

Como está o plugin somente substitue a primeira ocorrência da abreviação.

Um outro plugin do CD procura por algumas palavras que sejam título de resenhas dos restaurantes e substitui por links para as respectivas resenhas.

Mais detalhes em: <http://docs.joomla.org/Developers>

### 7.3.6 – Criando Templates Simples para o Joomla

Os templates são as extensões que cuidam da apresentação do conteúdo na tela. Tanto em termos de aparência: tipo de fonte, tamanho, cores, imagens, etc, como também em termos de posição: em que posição aparecerá cada um dos módulos, quantas colunas terá, qual a largura da página e de cada coluna, etc.

Os templates são formados de vários arquivos, mas apenas o `index.php` e o `templateDetails.xml` são obrigatórios.

#### Estrutura de arquivos de um template:

```
templateDetails.xml
index.php
template_thumbnails.ext
css\tempalte.css
images\logomarca.png
html\
component.php
```

Entre outros menos comuns.

**index.php** - oferece a lógica central do template, incluindo a exibição de qualquer módulo ou componente. Erroneamente muitos templates incluem código de layout no `index.php`.

**template.css** - oferece os estilos para as fontes, cores, posições, imagens e outros aspectos da apresentação do template.

**templateDetails.xml** - metainformações sobre o próprio template usadas pela interface de administração do Joomla para a instalação e manutenção do template.

**banner.ext** - não absolutamente necessário, muitos templates oferecem um central header gráfico que dá ao template uma marca.

#### Arquivos de Parâmetros

São arquivos chamados `params.ini` e que contém parâmetros que serão lidos na seção de administração, como por exemplo:

```
showComponent = 1
```

#### Arquivos de Imagens

Imagens de qualidade e adicionadas com competência fazem uma grande diferença nos templates.



## Arquivo Template Thumbnail

Ao mover o mouse sobre o nome dos templates na seção de administração aparecerá a imagem miniatura do template (caso ele a tenha). Isso ajuda a escolher o template sem precisar instalar.

Suas dimensões são de aproximadamente 206 x 150 pixels.

## Arquivo CSS

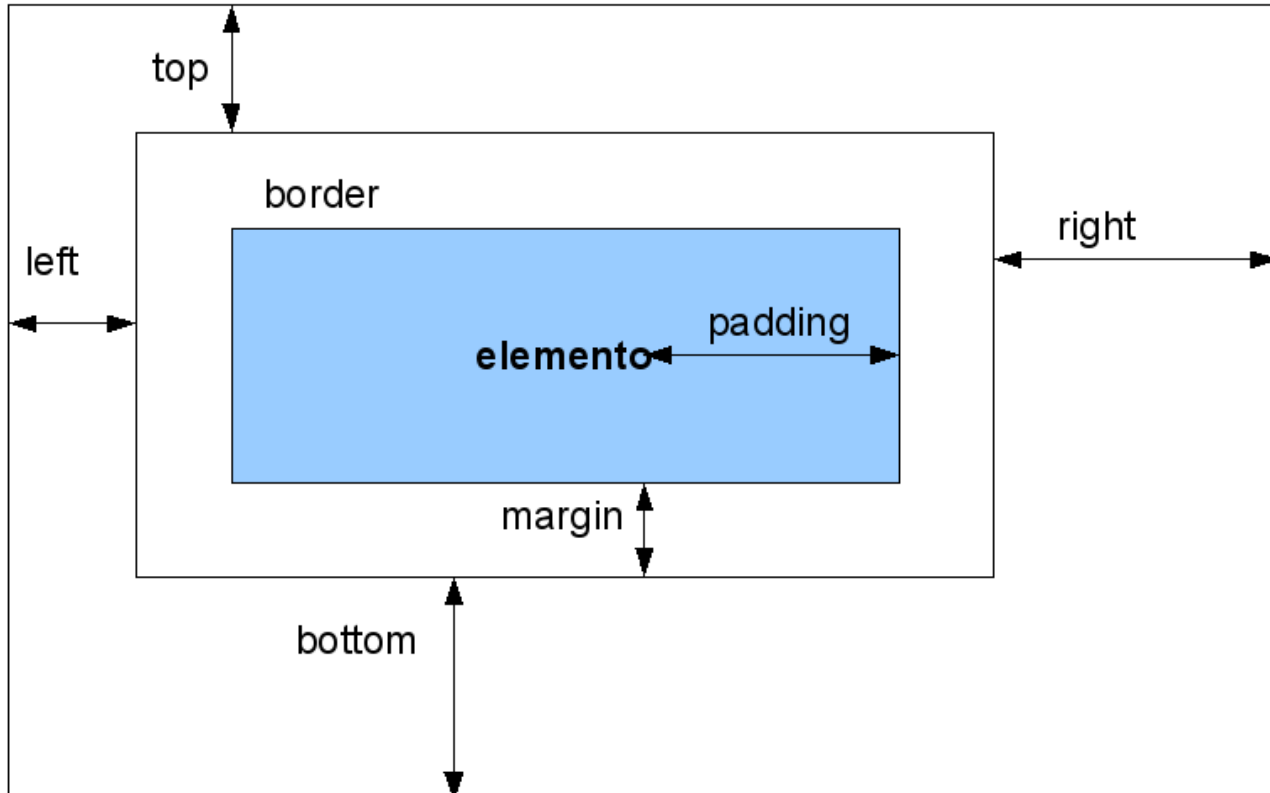
O segundo mais importante arquivo de um template é o arquivo CSS. Templates podem usar um ou mais arquivos CSS para controlar as fontes, cores, imagens e outros recursos da apresentação do template.

Caso um template seja perfeito para você, com exceção de que os links de texto aparecem vermelho e você gostaria que fossem verdes, então você apenas precisa alterar o CSS para ter seu template realmente perfeito.

Você precisa ser hábil para posicionar acuradamente cada item da página na devida posição.

Um bom conhecimento de CSS é exigido aqui.

## Como o CSS posiciona os elementos dentro da página:



## **Algumas Recomendações para padronizar Sites**

### **Sites de Conteúdo**

Conteúdo com fundo branco ou pastel claro e apenas os destaques são coloridos

### **Sites de Promoção (Lojas virtuais)**

Cores fortes e com muita animação

### **Lojas Virtuais**

Menus no topo com as categorias dos produtos

### **Quase todos os sites**

Logo no canto superior esquerdo com link para a index.

Os padrões se tornam importantes porque se os seguirmos o visitante já sabe onde encontrar o que procura, já que aprendeu isso em todos os sites que andou.

Complementando alguns padrões devemos criar nosso site sempre pensando em tornar a visita o mais fácil possível, sempre criando um site pensando no cliente.

**Exemplo:** mostre no rodapé um telefone e um e-mail para que ele possa tirar dúvidas se as tiver.

O menu de navegação deve ser fácil de encontrar, inclusive deve ser destacado e de fácil navegação. Não é mais importante que seja bonito e cheio de efeitos e chegue a ficar com parte escondida em algumas situações, mas importa que seja fácil e funcione.

Se ele puder efetuar uma operação com um único clique não o faça dar 2 ou mais cliques. Na combo lance um evento para que ele não precise clicar num botão para confirmar.

Num form jogue o foco no primeiro campo que ele deverá preencher.  
Use o TABINDEX para maior conforto.

Use fundo branco com fontes uniformes para facilitar a leitura.  
Evite largura de coluna muito grande em textos. Prefira dividir em duas colunas.

Um layout agradável e de fácil navegação chega a ser mais importante que muitas animações e truques visuais.

Use de bom senso ao criar o site. Atente para contrastes, combinação de cores, tons de uma cor, etc. Use bem os recursos do CSS para personalizar os elementos do site e destacar alguns. Assim também como deve ter boas imagens e fontes.

Coloque as coisas onde o visitante espera encontrar.  
É muito importante usar um template no site, para todas as seções, para que o visitante perceba que ainda está no site, pois identifica suas características.

*Adaptado do Guia Prático de Criação de Sites de Milton Chicoli*

### Testes

É importante que você tenha pelo menos dois navegadores instalados para seus testes: IE e FF.

Para adicionar a referência de um arquivo CSS no arquivo index.php, adicione as linhas abaixo entre as:

tags <head> e </head>.

```
<link rel="stylesheet" href="templates/_system/css/general.css" type="text/css" />
<link rel="stylesheet" href="templates/<?php echo $this->template ?>/css/template.css"
type="text/css" />
```

A primeira linha carrega o CSS de sistema provido pelo Joomla. Ela define alguns elementos do HTML como a tag <body>.

A segunda linha adiciona o path do nosso CSS.

### Arquivo de Metadados (XML)

Detalhes do Template – templateDetails.xml

Informações sobre o autor, tipo de extensão, versão do template, codificação de caracteres, arquivos, idiomas, posições, parâmetros, etc.

### Exemplo simples de templateDetails.xml

```
<?xml version="1.0" encoding="utf-8"?>
<install version="1.5" type="template">
  <name>Three Columns template</name>
  <description>
    Two CSS columns in the Joomla world.
  </description>
  <files>
    <filename>index.php</filename>
    <filename>templateDetails.xml</filename>
    <filename>images/LSlogo.jpg</filename>
    <filename>css/template.css</filename>
  </files>
  <positions>
    <position>left</position>
    <position>right</position>
  </positions>
  <param name="showComponent" type="radio" default="1"
    label="Show Component" description="Show component output">
    <option value="0">No</option>
    <option value="1">Yes</option>
  </param>
</install>
```

As posições relacionadas são meramente ilustrativas e não serão checadas pelo Joomla, mas são uma boa prática.

## Módulos e Posições

Alguns módulos e posições

```

<jdoc:include type="modules" name="icon" />
<jdoc:include type="modules" name="header" /> - Aparece somente entre as tags <head>
        para renderizar o conteúdo do estilo, scripts e meta elementos
        associados com a página atual.
<jdoc:include type="installation" /> - Usado somente pelo instalador do Joomla
<jdoc:include type="message" /> - Este elemento deve aparecer somente na tag <body>
        do Template para renderizar as mensagens de erro que ocorrerem na requisição.
<jdoc:include type="module" name="breadcrumbs" />
<jdoc:include type="module" name="menu" />
<jdoc:include type="module" name="submenu" style="rounded" id="submenu-box" />
<jdoc:include type="modules" name="left" style="rounded" />
<jdoc:include type="modules" name="left" style="xhtml" /> - menu principal
<jdoc:include type="modules" name="right" style="xhtml" /> - pool
<jdoc:include type="modules" name="status" />
<jdoc:include type="modules" name="syndicate" /> - RSS
<jdoc:include type="modules" name="title" />
<jdoc:include type="modules" name="status" />
<jdoc:include type="modules" name="toolbar" />
<jdoc:include type="modules" name="top" />
<jdoc:include type="modules" name="top" style="xhtml" /> - newsflash
<jdoc:include type="modules" name="user1" style="xhtml" /> - novidades
<jdoc:include type="modules" name="user2" style="xhtml" /> - populares
<jdoc:include type="modules" name="user3" /> - top menu
<jdoc:include type="modules" name="user4" /> - busca
<jdoc:include type="modules" name="bottom" /> - abaixo, logo acima do footer
<jdoc:include type="modules" name="footer" /> - Copyright do Joomla e Banner
<jdoc:include type="modules" name="debug" /> - abaixo do footer
<jdoc:include type="component" /> - área central, artigo de boas vindas.

```

Este elemento deve aparecer somente na tag <body> do Template para renderizar o conteúdo principal da página com respeito ao conteúdo da página atual sendo visualizada.

### module

Renderiza um módulo simples dado pelo nome do atributo. O módulo precisa ser publicado e estar acessível para o usuário atual para que seja visível. Adicionais atributos podem ser oferecidos para controlar o layout e aparência do módulo, se suportado.

### modules

Este elemento renderiza todos os módulos atribuídos para o template na posição dados pelo nome do atributo. Os módulos precisam ser publicados e acessíveis pelo usuário atual para ser visível.

Adicionais atributos podem ser oferecidos para controlar o layout e aparência do módulo, se suportado.

### **O atributo style**

O atributo opcional style está disponível para o módulo e tipos de módulos. O valor do atributo refere-se ao estilo "chrome" usado para empacotar a saída gerada pelo módulo. Se nenhum estilo for provido, um valor "none" será usado como default.

Nomes dos módulos padrão chrome providos pelo Core do Joomla! são declarados em

```
/templates/system/html/modules.php
```

- \* table
- \* horz
- \* xhtml
- \* rounded
- outline
- 

Designers de templates devem criar nomes chrome adicionais apenas adicionando uma versão customizada deste arquivo para o usuário

```
-template/html/modules.php.
```

### **Muitos módulos ficam nas regiões left e right**

Regiões user5, user6, user7, ... ficam na região inferior, abaixo do artigo principal (componente).

Também existem módulos para regiões advertisement, que podem ficar em right ou abaixo, próximo do footer.

A Joomla page consists of a few modules and a component. These modules have certain positions.

To know module position in a Joomla template, we only need to append a query string tp=1. For the example:

<http://localhost/portalc/index.php?tp=1>

## Lista completa de jdoc:include

- component
- head
- installation
- message
- module
- modules

## Template padrão do Joomla 1.5 e suas posições com os respectivos módulos

The image shows a screenshot of the Joomla 1.5 default template layout. The page is filled with various modules, each highlighted with a black box and a label. The labels include: 'Newsflash - top', 'Top menu - user3', 'Search - user4', 'breadcrumbs', 'Main menu - left', 'Latest News - user1', 'Popular - user2', 'Login form - left', 'Componente - frontpage', 'Pool - right', 'Banner - footer', 'Footer - footer', 'Syndication syndicate', and 'debug'. The layout is complex, with many modules in the top and right areas, and a large central content area. The Joomla logo is visible in the top left corner.

Este é um template bem rico em termos de posições.

**Importante**

Lembre que somente podemos adicionar um módulo na tela em uma determinada posição se ele for definido antes no CSS e aparecer no index.php e no XML.

**Template Bom dá força ao Site**

Simplesmente com a troca de um template um site que parece medíocre pode se transformar em um site com fantástica presença na web.

**Bom mercado**

Para Joomla é um mercado bastante promissor, com dezenas de empresas oferecendo templates para vender e muita gente procurando templates free.

**Editando um template**

O Joomla nos oferece recursos para alteração dos templates através da interface de administração.

Vá em Extensões – Administrar Tema e clique no link de um template.

Aí podemos alterar o CSS e o HTML (no caso o index.php).

Também podemos alterar os parâmetros quando existirem.

Se for o template padrão do Joomla 1.5 (RHUK Milkyway), neste caso ele tem parâmetros para as cores e para a largura.

Ao clicar em Editar CSS, caso exista mais de um aparecerá uma lista onde devemos escolher um.

Todos os templates do site Joomla ficam no diretório /templates e os do admin focam em administrator/templates.

**O objeto JDocumentHTML**

```
echo $this->language  
echo $this->direction // diretório  
echo $this->template  
echo $this->title  
echo $this->description  
echo $this->link  
echo $this->countModules('user1')
```

Páginas semânticas - São páginas (X)HTML que cuidam somente do conteúdo, ficando com o CSS a apresentação.

**Resolução do Monitor (<http://marketshare.hitslink.com/report.aspx?qprid=17>)**

Apenas 23,64% usa 800x600 ou inferior  
O restante usa resolução de 1024x768 ou superior.

## Criação de Template Simples (sem CSS)

O mais básico template para o Joomla deve ter seu diretório e os dois arquivos:

- index.php
- templateDetails.xml

*Crie a pasta:*

portal/templates/inicial

*Crie os scripts:*

portal/templates/inicial/index.php

portal/templates/inicial/templateDetails.xml

### index.php

```
<?php defined('_JEXEC') or die('No access available'); ?>
<html>
<head>
    <jdoc:include type="head" />
</head>
<body>
    <h1>Olá Mundo dos Templates</h1>
</body>
</html>
```

### templateDetails.xml

```
<?xml version="1.0" encoding="utf-8"?>
<install version="1.5" type="template">
    <name>Template Inicial</name>
    <version>1.0</version>
    <creationDate>01/18/2007</creationDate>
    <author>Seu nome</author>
    <authorEmail>seu e-mail</authorEmail>
    <authorUrl>Seu site</authorUrl>
    <copyright>2007</copyright>
    <license>GNU/GPL</license>
    <description>Template inicial. Exercício do curso de Joomla 1.5</description>
    <files>
        <filename>index.php</filename>
    </files>
</install>
```

Acesse a administração e torne o template inicial o template padrão.

### Testando

Chame no navegador

<http://localhost/portal>



Observe que não é um template funcional, mas apenas um tradicional "Olá mundo". A intenção deste é apenas começar a mostrar o funcionamento de um template no Joomla 1.5.

## Exemplo de um template já com duas colunas usando CSS

### Template simples com duas colunas

#### **Estrutura de Arquivos**

```
\css
    template.css
\images
    joomlacurso.png
index.php
templateDetails.xml
```

#### **index.php**

```
<?php echo '<?xml version="1.0" encoding="utf-8"? .>'; ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
    <jdoc:include type="head" />
    <link rel="stylesheet" href="templates/_system/css/general.css" type="text/css" />
    <link rel="stylesheet" href="templates/<?php echo $this->template ?
>/css/template.css"
type="text/css" />
</head>
<body id="page_bg">
    <jdoc:include type="message" />
    <div id="logo">&nbsp;</div>
    <div id="container">
        <div id="col2">
            <jdoc:include type="component" />
        </div>
        <div id="col1">
            <jdoc:include type="modules" name="left" style="xhtml" />
        </div>
        <div class="myclear">&nbsp;</div>
    </div>
    <jdoc:include type="modules" name="debug" />
</body>
</html>
```

**templateDetails.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<install version="1.5" type="template">
  <name>Template de duas colunas</name>
  <version>1.0</version>
  <creationDate>2009</creationDate>
  <author>Seu nome</author>
  <authorEmail>email</authorEmail>
  <authorUrl>site</authorUrl>
  <copyright>2009</copyright>
  <license>GNU/GPL</license>
  <description>Template para treinamento do curso de Joomla 1.5.</description>
  <files>
    <filename>index.php</filename>
    <filename>templateDetails.xml</filename>
    <filename>images/logo.png</filename>
    <filename>css/template.css</filename>
  </files>
  <positions>
    Ribamar FS – http://cursos.ribafs.org
    Programando para o Joomla 1.5 40/43
    <position>left</position>
  </positions>
</install>
```

**template.css**

```
div#logo {
  width: 110%; height: 100px;
  margin-left: -10px;
  margin-bottom: 10px;
  background: url(../images/logo.png) left no-repeat;
  border: 1px solid #244223 ;
  padding: 0px;
}
#col1 {
  float:left;width:20%;
  display:inline;
  background:#244223;
  padding: 10px;
}
#col2 {
  float:right;width:75%;
  display:inline;
  border:3px solid #244223;
  background:#58a155;
  padding: 10px;
}
```

```
#container {
    float:left;width:85%;
    display:inline;
}
#myclear {
    clear:both;
}
#page_bg {
    font-family: Verdana, Arial, Helvetica, sans-serif;
    height: 100%;
    background: black;
}
.moduletable_menu, .moduletable
{
    color: white;
    border-bottom: 1px solid #fff;
    margin-bottom: -1em;
}
.moduletable_menu h3 {
    border-bottom: 1px solid #FFFFFF;
    margin-bottom: 0px;
}

```

Ribamar FS – <http://cursos.ribafs.org>  
Programando para o Joomla 1.5 41/43

```
.moduletable_menu ul {
    margin-left: 10px;
    margin-top: 0px;
    padding: 10px;
    font-size: 80%;
    list-style-type: square;
}
.moduletable_menu a:link, .moduletable a:link {
    color:yellow;
}
.moduletable_menu a:visited, .moduletable a:visited {
    color:cornsilk;
}
.contentheading {
    border-bottom: 2px solid Black;
    border-right: 2px solid Black;
    border-left: 1px solid LightGreen;
    border-top: 1px solid LightGreen;
    background:#244223;
    color: white;
    padding: 10px;
    font-size:2em;
}
```

```
.contentpaneopen h1 {
    font-size:1.5em;
    border-bottom: 1px solid #244223;
    padding: 10px;
}
```

Crie a pasta images e para dentro dela copie para a mesma a imagem do logo do template rhuk\_milkyway, que é o padrão do Joomla renomeando para logo.png.

### Testando

Torne este template o padrão e visualize no site.

### Usando Objetos da classe JDocument

A JDocument é a classe mais usada em templates Joomla. São acessadas através do objeto jdoc.

O include dos templates do Joomla 1.5 nos arquivos index.php é da seguinte forma:

```
<jdoc:include type="head" />
```

Podemos também adicionar um include que exibirá um sistema de mensagens:

```
<jdoc:include type="message" />
```

Para adicionar ao template a saída do módulo 'top' usamos:

```
<jdoc:include type="module" name="top" style="xhtml" />
```

Para exibir o idioma selecionado pelo template atual:

```
<?php print $this->language; ?>
```

Adicionando este linha acima no nosso hello world veremos o idioma que o Joomla está usando.

### Outras propriedades interessantes do JDocumentHTML:

direction - ltr (left-to-right) ou rtl (right-to-left)

Exemplo:

```
<?php if($this->direction == 'rtl') : ?>
    <link href="<?php echo $this->baseurl ?
>/templates/rhuk_milkyway/css/template_rtl.css"
rel="stylesheet" type="text/css" />
<?php endif; ?>
```

template - diretório do template

title - contém o título do documento

description - descrição do documento (dos metadatas)  
link - contém a URL do documento

O objeto JDocumentHTML também nos oferece alguns métodos úteis, como o countModules().

Exemplo:

```
<div id="leftcolumn">
    <?php if($this->countModules('left')) : ?>
        <jdoc:include type="modules" name="left" style="rounded" />
    <?php endif; ?>
</div>
```

### Contando vários módulos:

```
$this->countModules('user1' + 'user2')
```

### Expressões booleanas:

```
$this->countModules('user1 and user2')
```

```
$this->countModules('user1 or user2')
```

```
<?php if($this->countModules('top')) : ?>
    <jdoc:include type="modules" name="top" />
<?php else : ?>
    <img src='placeholder.gif'>
<?php endif; ?>
```

### Acessando parâmetros de templates

Podemos personalizar um template apenas alterando seus parâmetros através da administração.

O template default do Joomla 1.5 (rhuk\_milkyway) é um ótimo exemplo disso. Tem parâmetros para alterar suas cores de fonte e de fundo (background).

### Código que lê o parâmetro e decide qual cor exibir:

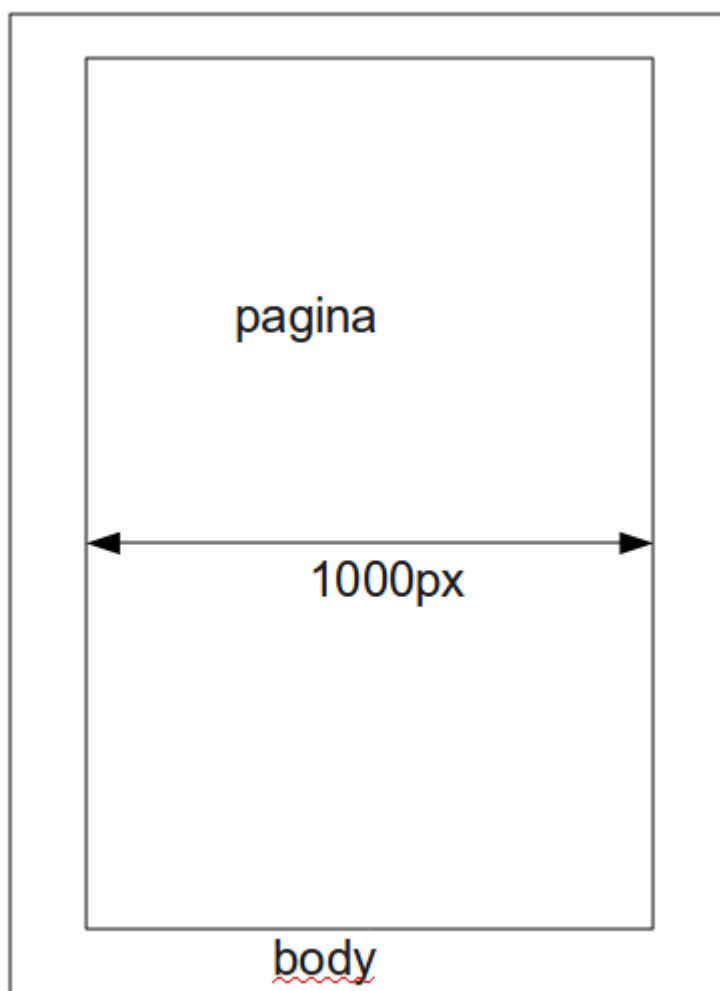
```
<?php if($this->params->get('showComponent')) : ?>
    <jdoc:include type="component" />
<?php endif; ?>
```

### Testes

Após a conclusão devemos efetuar vários testes, que devem ser, de preferência, em outro sistema operacional, em outro navegador e em outro tamanho de monitor.

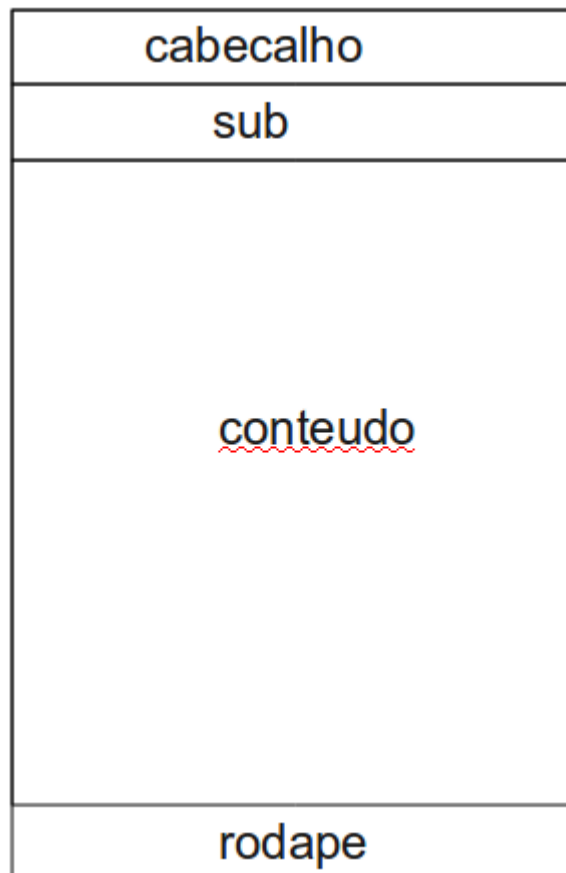
## Estrutura do Template Green

ID pagina, que representa toda a região útil e central do template:



Veja que nosso template é um template de largura fixa, com 1000px.

O ID pagina se subdivide em 4 grandes regiões:



As regiões sub e corpo se subdividem em outras regiões menores e as regiões cabeçalho e rodape não se subdividem.

Região cabeçalho:

cabeçalho

Região rodape:

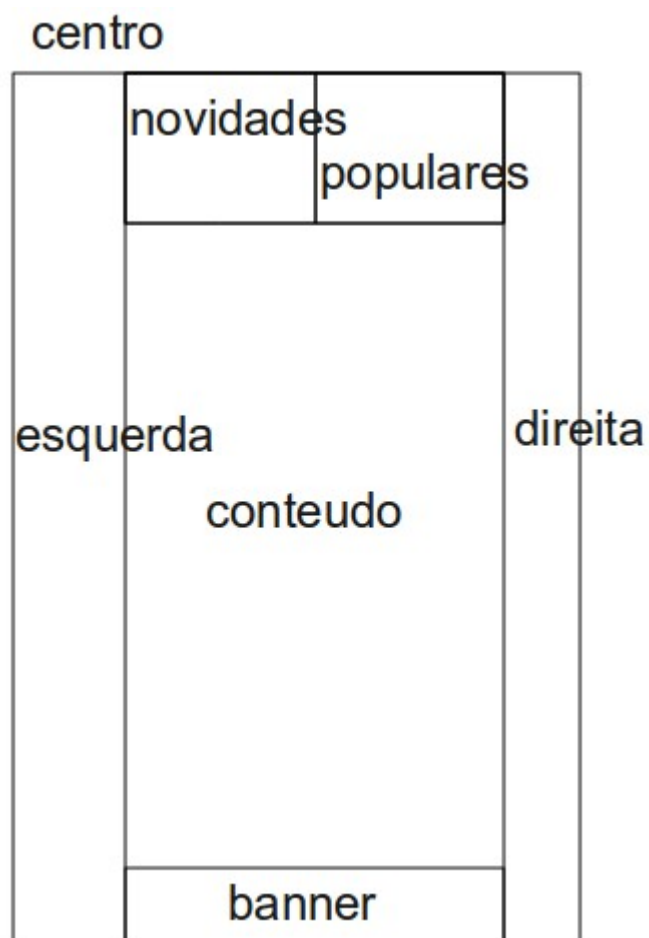
rodape

Região sub:

sub  
topmenu  
pathway search



Região centro:



Fortaleza, Ceará – Brasil, 28 de novembro de 2009.

### 7.3.7 - Parâmetros nos Templates

```
<params>
  <param name="templateColour" type="list" default="blue"
    label="Template Colour" description="Choose the template colour.">
    <option value="blue">Blue</option>
    <option value="red">Red</option>
    <option value="green">Green</option>
    <option value="black">Black</option>
  </param>
  <param name="authorCopyright" type="radio" default="1"
    label="Author Copyright" description="Show/Hide author copyright.">
    <option value="0">hide</option>
    <option value="1">show</option>
  </param>
</params>

<param name="mycalendar" type="calendar" default="5-10-2008" label="Select a date" description=""
format="%d-%m-%Y" />

<param name="mycategory" type="category" label="Select a category" description="" section="3" />

<param name="myeditor" type="editors" default="none" label="Select an editor" />

<param name="myfile" type="filelist" default="" label="Select a file" description="" directory="administrator"
filter="" exclude="" stripext="" />

<param name="myfolder" type="folderlist" default="" label="Select a folder" directory="administrator" filter=""
exclude="" stripext="" />

<param name="myhelpsite" type="helpsites" default="" label="Select a help site" description="" />

<param name="mysecretvariable" type="hidden" default="" />

<param name="myimage" type="imagelist" default="" label="Select an image" description="" directory=""
exclude="" stripext="" />

<param name="mylanguage" type="languages" client="site" default="en-GB" label="Select a language"
description="" />

<param name="mylistvalue" type="list" default="" label="Select an option" description="">
  <option value="0">Option 1</option>
  <option value="1">Option 2</option>
</param>

<param name="mymenu" type="menu" default="mainmenu" label="Select a menu" description="Select a
menu" />

<param name="mymenuitem" type="menuitem" default="45" label="Select a menu item" description="Select
a menu item" />

<param name="mypassword" type="password" default="secret" label="Enter a password" description=""
size="5" />
```

```
<param name="myradiovalue" type="radio" default="0" label="Select an option" description="">
  <option value="0">1</option>
  <option value="1">2</option>
</param>
```

```
<param name="mysection" type="section" default="" label="Select a section" description="" />
```

```
<param type="spacer" />
```

```
<param type="spacer" default="Advanced parameters" />
```

```
<param type="spacer" default="&lt;b&gt;Advanced parameters&lt;/b&gt;" />
```

```
<param name="title" type="sql" default="10" label="Select an article" query="SELECT id AS value, title
FROM #__content" />
```

```
<param name="title" type="sql" default="10" label="Select an article" query="SELECT id, title FROM
#__content" key_field="id" />
```

```
<param name="myfield" type="sql" default="10" label="Select an article" query="SELECT id AS value, title
AS myfield FROM #__content" />
```

Ou alternativamente:

```
<param name="myfield" type="sql" default="10" label="Select an article" query="SELECT id, title FROM
#__content" key_field="id" value_field="title" />
```

```
<param name="mytextvalue" type="text" default="Some text" label="Enter some text" description=""
size="10" />
```

```
<param name="mytextarea" type="textarea" default="default" label="Enter some text" description=""
rows="10" cols="5" />
```

```
<param name="mytimezone" type="timezones" default="-10" label="Select a timezone" description="" />
```

```
<param name="myusergroups" type="usergroup" default="" label="Select a user group" description="" />
```

Recebendo dados de parâmetros em arquivos de template (index.php):

```
<?php $myParam = $this->params->get( 'parameterName' ); ?>
```

```
<?php
$tplColour = $this->params->get( 'templateColour' );
$this->addStyleSheet( $this->baseurl . '/templates/
    . $this->template . '/css/ . $tplColour . '.css' );
?>
```

```
<?php if ( $this->params->get( 'authorCopyright' ) ) : ?>
  <div class="copyright">
    Copyright &copy; 2008 Fat Pigeon Templates
  </div>
<?php endif; ?>
```

## Contando os módulos de uma posição

```
<?php if ($this->countModules( 'user1' )) : ?>
  <div class="user1">
    <jdoc:include type="modules" name="user1" style="rounded" />
  </div>
<?php endif; ?>
```

```
$this->countModules( 'user1 + user2' );
```

```
$this->countModules( 'user1 and user2' );
```

```
$this->countModules( 'user1' and 'user2' );
```

```
<?php if ($this->countModules( 'user1 or user2' )) : ?>
  <div class="rightcolumn">
    <jdoc:include type="modules" name="user1" style="xhtml" />
    <jdoc:include type="modules" name="user2" style="xhtml" />
  </div>
<?php endif; ?>
```

```
<?php if ($this->countModules( 'user1 or user2' )) : ?>
  <div class="user1user2">

    <?php if ($this->countModules( 'user1' )) : ?>
      <jdoc:include type="modules" name="user1" style="xhtml" />
    <?php endif; ?>

    <?php if ($this->countModules( 'user1 and user2' )) : ?>
      <div class="greyline"></div>
    <?php endif; ?>

    <?php if ($this->countModules( 'user2' )) : ?>
      <jdoc:include type="modules" name="user2" style="xhtml" />
    <?php endif; ?>

  </div>
<?php endif; ?>
```

## Tipos de Media CSS

```
<link rel="stylesheet" media="screen" type="text/css" href="css/screen/default.css" />
<link rel="stylesheet" media="print" type="text/css" href="css/print/default.css" />
<link rel="stylesheet" media="handheld" type="text/css" href="css/handheld/default.css" />
```

### 7.3.8 - Sobre o Favicon

Dimensões: 16x16 até 64x64

Converta uma imagem PNG para ICO aqui:

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

Faça o download de um ícone aqui:

<http://www.freefavicon.com/freefavicons/objects/index.php>

<http://artscape.edwardtbabinski.us/favicon/dark-icons.html>

Ou crie o seu aqui:

<http://www.degraeve.com/favicon/>

<http://tools.dynamicdrive.com/favicon/>

Então copie o arquivo favicon.ico para o diretório do template em uso.

Antes renomeie o arquivo existente por segurança.

Para adicionar favicon em página que ainda não tem, edite o arquivo index.php e adicione as duas linhas

abaixo entre as tags <head> e </head>:

```
<link href="/diretoriosite/favicon.ico" rel="shortcut icon" type="image/x-icon" />
```

```
<link rel="shortcut icon" href="http://localhost/site/favicon.ico" />
```

Dica do site:

[http://kb.siteground.com/article/How\\_to\\_change\\_the\\_favicon\\_in\\_Joomla\\_15.html](http://kb.siteground.com/article/How_to_change_the_favicon_in_Joomla_15.html)

[http://alexandrecunha.com/artigo/internet/o\\_que\\_como\\_criar\\_um\\_favicon](http://alexandrecunha.com/artigo/internet/o_que_como_criar_um_favicon)

O que é um favicon?

O favicon é aquele ícone que aparece na barra de endereço do seu navegador e na lista de favoritos (ou marcadores) e permitem identificar o website com uma pequena imagem de 16x16 pontos. Digamos que funcionam como "ícone de marca" ou ícone de atalho dos websites. Na ausência de um favicon definido, os navegadores exibem seu próprio ícone ou um ícone padrão (normalmente parecido com uma folha em branco).

O termo favicon deriva dos termos favorite icon (ou ícone do favorito). A facilidade do favicon foi introduzida pela Microsoft no Internet Explorer e rapidamente se tornou amplamente utilizada por todos os navegadores.

Como produzir um favicon com o Gimp

Um favicon deverá estar no formato .ICO (ou Windows Icon) mas também poderá estar nos formatos .PNG e .GIF embora estes últimos sejam apenas suportados por navegadores que não o Microsoft Internet Explorer. O formato mais amplamente usado, é mesmo o favicon de 16x16 com 8 bit.

Para produzir um favicon a partir de uma qualquer imagem ou logótipo no Gimp, siga os seguinte passos:

- \* Abra a imagem ou logótipo que pretende utilizar
- \* Ajuste a imagem para a forma de quadrado
- \* Redimensione a imagem para as dimensões 16x16
- \* Grave a imagem no formato .ICO (basta colocar a extensão .ico ao gravar). Ao fazer isto, o Gimp irá apresentar uma janela "Save as Windows Icon" com várias opções. Escolha a opção de acordo com as definições de cores que pretende no resultado final. Poderá ser útil fazer várias experiências até chegar ao resultado que considerar melhor.

Mais sobre o favicon em: <http://en.wikipedia.org/wiki/Favicon>

## Customizando Templates para o Joomla 1.5

Simplesmente com a troca de um template um site que parece medíocre pode se transformar em um site com fantástica presença na web.

Embora seja mostrado como criar um template e seus componentes, aqui apenas veremos como personalizar/customizar templates para o Joomla 1.5. Tento em vista que existe uma grande quantidade de tempaltes free para o Joomla 1.5, nós iremos customizar um ou dois destes.

No Joomla é um mercado bastante promissor, com dezenas de empresas oferecendo templates para vender.

### Inserindo Flash em Código HTML

```
<object type="application/x-shockwave-flash" data="path/file.swf" width="WIDTH"
height="HEIGHT">
<param name="movie" value="file.swf" />
<p>Você precisa do flash player</p>
</object>
```

ou

```
<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0,4
0,0" height="60" width="455"><param name="width" value="455"><param name="height"
value="60">
<param name="src" value="logo.swf">
<embed type="application/x-shockwave-flash" src="logo.swf" height="60" width="455">
</object>
```

**Referência:**

<http://books.google.com.br/books?q=joomla&btnG=Pesquisar+livros>  
<http://www.joomla.com.br/tutoriais-mainmenu-47/239-template-joomla15.html>  
Livro Professional Joomla, de Dan Rahmel da Wrox editora

## Templates em Flash

<http://www.easytemplates.com/preview/300110158>  
<http://www.flashtemplatestore.com/preview/300110222>  
<http://www.easytemplates.com/preview/300110222>  
<http://www.templates247.com/previews/545/t545.html>  
<http://www.easytemplates.com/fla-templates.htm>  
<http://www.templates247.com/previews/557/t557.html>  
<http://www.flashtemplates.com/>

[http://www.flashmo.com/preview/flashmo\\_135\\_portfolio](http://www.flashmo.com/preview/flashmo_135_portfolio)  
[http://www.flashmo.com/preview/flashmo\\_127\\_curtain](http://www.flashmo.com/preview/flashmo_127_curtain)  
[http://www.flashmo.com/preview/flashmo\\_100\\_building](http://www.flashmo.com/preview/flashmo_100_building)  
<http://www.flashtemplatestore.com/>  
<http://www.templates247.com/previews/466/t466.html>  
<http://sunnyflash.com/view-furniture-design-xml-flash-template-2279.html>  
<http://www.flash-template.com/preview.php?tid=1087>

<http://www.flashtemplate.us/xmlswfdemo/cl4b/index.html>  
<http://www.myflashxml.com/>  
<http://www.flashtemplate.us/xmlswfdemo/cl3b/index.html>

## Aberturas

<http://www.templatesdream.com/flash-intros.html>

## Logotipos

<http://www.templatesdream.com/logos.html>

### 7.3.9 – Criação de Languages/Idiomas para o Joomla

Quanto a estas extensões, que não são bem extensões se comparadas com as demais, apenas são arquivos textos contendo variáveis com os respectivos valores num idioma.

Mas são importantes quando precisamos suportar mais de um idioma e dar mais visibilidade às nossas extensões.

Faça o download do pequeno módulo:

<http://extensions.joomla.org/extensions/core-enhancements/file-management/7364>

Descompacte e veja como é implementada a extensão language/idioma.

Observe que existem os arquivos do idioma e apenas sua referência no XML.  
Basta isso para tornar suas extensões com suporte a vários idiomas.



## 8 - Recursos Extras

Output Overrides  
Sufixo de Classes

### 8.1 - Overrides - Sobrescrevendo Saídas

O Joomla traz vários recursos que facilitam muito sua programação.

Temos os parâmetros passados para as views.

Além disso a equipe adicionou um recurso chamado Output Overrides, que permite controlar a saída (visualização) de praticamente tudo no Joomla 1.5.

Tudo no Joomla tem sua aparência controlada pelo template, mas usando Output Overrides podemos controlar isso e mudar a aparência.

Exemplo:

```
<?php
$document = &JFactory::getDocument();
$renderer = $document->loadRenderer('modules');
$options = array('style' => 'xhtml');
$position = 'article-banners';

echo $renderer->render($position, $options, null);
?>
```

Carrega o documento, depois recebe modules do \$document, configura o estilo do módulo e então exibe o módulo na posição desejada.

Carregar um único módulo dá um pouco mais de trabalho, veja:

```
<?php
$document = &JFactory::getDocument();
$renderer = $document->loadRenderer('module');
$options = array('style' => 'raw');

$module = JModuleHelper::getModule('mod_comments_comments');
$module->params = "heading=2\nlimit=10";
echo $renderer->render($module, $options);
?>
```

Desta vez carregamos apenas o "module" renderizado.

## 8.2 - Sufixos de Classes

### Para Módulos, Páginas e Menus do Joomla

Sufixo de Classe no Joomla 1.5 é uma forma fácil de alterar o CSS. Sufixo de classe é a parte final adicionada ao nome da classe.

Todo o estilo do site em Joomla está no script `template.css`. No caso do Joomla ser instalado na pasta `portal` temos:  
`portal/templates/rhuk_milkyway/css/template.css`

`_menu` é um sufixo que acrescenta-se aos módulos dos menus no menu default para fiquem com cantos arredondados e azuis.

A função dos sufixos de classes é a de personalizar um dos módulos, já que todos usam a mesma classe.

Quando queremos um deles diferente uma forma prática é adicionar um sufixo de classe ao CSS dos mesmos, na administração do módulo.

Alterar a classe altera todos os módulos, mas alterar o sufixo de classe de um deles altera somente ele.

Quando adicionamos um sufixo de classe para um módulo, por exemplo:

`_sufixo`

Então a classe existente fica assim:

```
class="classeexistente_sufixo"
```

### Adicionando uma Segunda Classe

Para adicionar uma segunda classe CSS para um módulo, inserimos um espaço antes do nome da segunda classe:

`segundaclasse`

Então deverá ficar assim:

```
class ="primeira segundaclasse"
```

O código CSS da `segundaclasse` deve ser criado para que funcione.

**Como exemplo vamos usar o módulo enquete (pools).****Adicione um Sufixo de Classe de Módulo**

Acesse o admin - Extensões - Administrar Módulos - Pools

Na caixa "Sufixo de Classe do Módulo" digite:

`_cssenquete`

Então clique em Salvar

**Agora edite o template.css**

Extensões - Administrar Tema

Clique no link do template **rhuk\_milkyway**

Clique em Editar CSS

Selecione template.css e clique em Editar

Role a tela e ao final do script digite:

```
/* Alteração para sufixo de classe do módulo pools */
div.moduletable_cssenquete td{
/*content */
color: #FF0000;
font-family: Arial, Helvetica, sans-serif;
text-align: left;
font-style: italic;
}
div.moduletable_cssenquete h3{
/*title */
color: #FF0000;
text-align: left;
font-style: italic;
}
```

Então clique em Salvar e Salvar novamente.

Acesse o site e veja que agora a enquete está em vermelho.

Observe a sutileza e que são diferentes:

`div.moduletable_cssenquete`

`div.moduletable cssenquete`

`div.moduletable _cssenquete`

Como o Joomla é uma aplicação complexa não se torna fácil personalizar o CSS.

## Exemplo de Customização de um Módulo usando Sufixo de Classe

Vamos customizar o módulo main menu.

Extensões - Administrar Módulo

Main menu - Parâmetros Avançados então digite `_estilo1` colado em `_menu`, para que fique assim:

`_menu_estilo1`

Então clique em Salvar

Agora faça semelhante com o anterior e adiciona ao final do `template.css` este código:

```
/* Alteração do main menu */
div.module_menu_estilo1 {
font-family: Courier;
font-size: 13px;
font-weight: bold;
color: #c393a2;
text-transform: capitalize;
text-align: left;
}
div.module_menu_estilo1 h3 {
font-family: Courier;
color: #c393a2;
}
```

## Sufixo de Classe de Página para o Joomla 1.5

Funciona quando queremos o mesmo estilo para o conteúdo.

Como Exemplo Vamos aplicar um Sufixo de Classe de Página no cabeçalho do artigo "Joomla! Overview".

Para visualizar antes do sufixo clique agora no link " Joomla! Overview".

No Admin clique na aba Menu - Main Menu

Clique no item Joomla! Overview

Clique em Parâmetros (Sistema) e na caixa Sufixo de Classe de Página digite:

`_sobrever`

Então clique em Salvar.

Usando uma extensão para o Firefox como Web Developer ou Firebug, podemos ver:

```

<table class="contentpaneopen">
  <tbody>
    <tr>
      <td class="contentheading" width="100%">
        Joomla! Overview </td>
    </tr>
  </tbody>
</table>

```

A estrutura do CSS, que a classe contentheading está contida na contentpaneopen.

Então edite novamente o template.css e adicione ao final:

```

/* Alteração do item Joomla! Overview */
.contentheading_sobrever
{
background: #CCFF33;
font-family: "Agency FB";
}

```

Salve e veja o resultado. Agora o título está com fundo verde.

## Sufixo de Classe de Menu

Alterar o estilo de menus.

### Como exemplo vamos alterar o menu Key Concepts

Vá na administração do módulo e clique em Parâmetros Avançados e digite na caixa de texto Sufixo de Classe do Menu "-horizontal" sem as aspas e clique em Salvar.

Veja a estrutura do CSS com HTML visto com o Firebug:

```

<div class="module_menu">
  <div>
    <div>
      <div>
        <h3>Key Concepts</h3>
        <ul class="menu-horizontal">
          <li class="item40">
            <li class="item38">
              <a href="/portal/index.php?option=com_content&view=article&id=24&Itemid=38">
                <span>Content Layouts</span>
              </a>
            </li>
          <li class="item43">

```

Observe que o item de menu com classe item38 é uma outra classe dentro da classe menu-horizontal.

Como não pretendemos alterar o módulo vamos ignorar o `module_menu`.

Agora edite o `template.css` e adicione ao final:

```
/* Alteração do key concepts */  
.menu-horizontal .item38  
{  
border: thin #FF0000 groove;  
}
```

Salve e veja o resultado, que deve ser bordas ao redor do item 38, o do meio.

Observe que `".menu-horizontal .item38"` é muito diferente de `".menu-horizontal.item38"` sem o espaço central. Este último não funciona.

O sufixo de classe de menu é útil quando não desejamos alterar o módulo por completo.

Um recurso muito útil é o da Extensão Web Developer - `Ctrl+Shift+Y` então clique em uma região da página para a qual deseja conhecer o CSS.

Para desativar tecele novamente `Ctrl+Shift+Y`.

Tradução livre de <http://help.joomla.org/ghop/feb2008/task057/Joomla%20Class%20Suffix%20Guides.pdf>

## 9 - Extensões de Exemplo

Aqui pretendo mostrar o código de algumas extensões de exemplo. Quando iniciamos o aprendizado de uma linguagem ou conhecimento, ter por perto exemplos simples e funcionais que possamos estudar, alterar e personalizar ao nosso modo para ir aprendendo como funciona, essa é uma forma bem produtiva de aprender.

Para o aprendizado é importante que as extensões sejam funcionais mas que não sejam complexas, o que dificultaria o aprendizado.

As extensões que mostrarei agora tem complexidade superior às apresentadas no capítulo sobre extensões mas ainda com complexidade média para evitar dificultar o aprendizado.

Aque mostrarei um componente, um módulo e um plugin que tem como finalidade controlar a divulgação de hotéis num site Joomla. Onde os responsáveis cadastram seus hotéis pelo site. Após o login são autenticados e terão acesso ao cadastro. Somente têm acesso aos hotéis que cadastraram. Haverá um componente para a administração que permite cadastrar os usuários que terão acesso ao cadastro.

Um módulo mostrará os hotéis cadastrados aleatoriamente.

Um template com muitos bons recursos será criado seguindo vários passos, desde o início até sua conclusão.

### 9.1 - Código Fonte

Adicionei um pacote contendo 53 extensões, entre componentes, módulos, plugins e templates. A complexidade varia de inicial para média.

#### Download do código fonte

[http://cursodejoomla.ribafs.org/cursodejoomla\\_codigofonte.zip](http://cursodejoomla.ribafs.org/cursodejoomla_codigofonte.zip)

## 9.2 - Componente de Exemplo – com\_hoteiscrud2

Os componentes são as extensões que ocupam a região central da página e que por padrão tem um componente que mostra os artigos nesta região, juntamente com a paginação se for necessária (nas categorias).

Os componentes são pequenos ou grandes aplicativos completos e geralmente são as maiores extensões em termos de linhas de código como também as que têm mais recursos na administração.

Este componente conterà os arquivos:

```
\admin
  \tables
    contatos.php
    hoteis.php
    index.html
  admin.hoteis_crud.html.php
  admin.hoteis_crud.php
  controller.php
  index.html
  install.mysql.sql
  toolbar.hoteis_crud.html.php
  toolbar.hoteis_crud.php
  uninstall.mysql.sql
\site
  hoteis_crud.html.php
  hoteis_crud.php
  index.html
hoteis_crud.xml
index.html
leiametext
```

### ***hoteis\_crud.xml***

Instalador, que guarda várias informações importantes sobre o componente para o Joomla.

```
\admin – este diretório guarda os arquivos da parte administrativa do componente
  \tables – este diretório contém um arquivo para cada tabela do componente
    contatos.php – classe que representa a tabela contatos
    hoteis.php – classe que representa a tabela hoteis
    index.html – esconder listagem do diretório
  admin.hoteis_crud.html.php – apresentação do componente (html)
  admin.hoteis_crud.php – arquivo principal do componente admin
  controller.php – controller da parte admin do componente
```



index.html – esconder listagem do diretório  
install.mysql.sql – arquivo com o script SQL para criação da tabela  
toolbar.hoteis\_crud.html.php – apresentação da toolbar do componente  
toolbar.hoteis\_crud.php – arquivo da toolbar do componente do admin  
uninstall.mysql.sql – script SQL para desinstalação

\site

hoteis\_crud.html.php – apresentação do componente (parte do site)  
hoteis\_crud.php – arquivo principal do componente (parte do site)  
index.html – esconder listagem do diretório

Este componente utiliza de forma simples o padrão de projeto MVC.

Usa os principais recursos da administração do Joomla:

- JTable
- MVC
- Toolbar
- Paginação

Na parte do site encontram-se as principais operações com bancos de dados, o CRUD. Esta parte não usa MVC mas está dividida em duas camadas, o arquivo contendo .html no nome contém a parte da apresentação e o outro a parte da lógica.

Procurei separar bem o código, de forma que fique fácil de entender para o iniciante em Joomla.

### 9.3 - Módulo de Exemplo – mod\_hoteiscrud

Este pequeno módulo conterà os arquivos:

mod\_hoteiscrud.php  
mod\_hoteiscrud.xml  
index.html  
leiametext

#### *mod\_hoteiscrud.xml*

```
<?xml version="1.0" encoding="utf-8"?>
<install type="module" version="1.5.0">
<name>Módulo Hotéis</name>
<author>Ribamar FS</author>
<creationDate>janeiro de 2010</creationDate>
<copyright>(C) 2005 Open Source Matters. All rights reserved.</copyright>
<license>http://www.gnu.org/copyleft/gpl.html GNU/GPL</license>
<authorEmail>ribafs@gmail.com</authorEmail>
<authorUrl>http://ribafs.org</authorUrl>
<version>1.0.0</version>
<description>Exibição de um hotel aleatoriamente da tabela de hotéis. Parte do Livro
Curso de Joomla.
</description>
<files>
<filename module="mod_hoteiscrud">mod_hoteiscrud.php</filename>
<filename>index.html</filename>
</files>
</install>
```

#### *mod\_hoteiscrud.php*

```
<?php
defined('_JEXEC') or die('Restricted access');
//$modulo = $params->get( 'modulo' );

$db =& JFactory::getDBO();
$query = "SELECT * FROM #__hoteis ORDER BY rand( )";
$db->setQuery( $query, 0, 1 );
//$reg_rand = $db->loadObjectList();
$reg_rand = $db->loadRow();
// 0 - id
// 1 - nome
```

```
// 2 - login_contato
// 3 - published
if ($reg_rand[3]){
print '<table><tr><td>Hotel - </td><td>'.$reg_rand[0].'</td></tr>
      <tr><td>Nome - </td><td>'.$reg_rand[1].'</td></tr>
      <tr><td>Contato - </td><td>'.$reg_rand[2].'</td></tr>
      </table>';
}
?>
```

***index.html***

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

***leiam.txt***

Este módulo tem como objetivo exibir um hotel aleatoriamente vindo da tabela de hotéis.

janeiro de 2010

Ribamar FS - <http://ribafs.org>

Download do módulo mod\_hoteiscrud completo.

## 9.4 - Template de Exemplo – tmp\_green

Para criar este template eu tive como base o template padrão do Joomla 1.5 (rhuk\_milkyway), pelo fato de ser um template muito rico em posições e bem organizado. Acontece que ele ainda usa tabela para exibir algumas posições e resolvi aprender um pouco mais sobre CSS e ao final removi as tabelas e troquei tudo por CSS.

### Template Green0

Este template é um template didático, cujo objetivo é mostrar como se cria um template para o Joomla 1.5 de forma simples e com bons recursos.

Este template, em 15 etapas (15 templates funcionais, da 0 até a 14), teve como base um layout em CSS, que encontra-se aqui:

<http://cursos.ribafs.org/aplicativos-em-php/57-css-introducao/240-css-layouts>

Esse layout tem as principais posições definidas pelo CSS para parecer um template em Joomla. Observe que nesse layout e num template do Joomla quem controla as posições de cada um dos módulos, de cada uma das regiões da página, quem faz isso é o CSS. Então o XHTML (index.php) vem e controla que módulos e componentes aparecerão e em que ordem aparecerão.

Isso simula uma situação real bem exigente, onde o cliente te pede para desenvolver um template com um certo layout, ou simplesmente te passa um rascunho num pedaço de papel onde mostra como deseja o layout do seu site. Partindo desse "projeto" nós iremos desenvolver um template para o Joomla 1.5.

Para isso usaremos as ferramentas seguintes:

- Sistema operacional Linux Ubuntu ou Windows XP
- Navegador Firefox ou Internet Explorer (7 ou superior)
- gEdit para editar os script (Linux) ou NetBeans for PHP (Windows)
- Extensões para o Firefox:
  - Firebug
  - ColorZilla
  - Measuteit
- Gráficas:
  - Gimp (Linux e Windows)
  - Kolourpaint (Linux) ou Paint (Win)

A criação foi efetuada em várias etapas, cada uma tendo gerado um template funcional com os recursos da etapa.

Nesta primeira etapa, parti do layout CSS e no Joomla tive como base o template de exemplo distribuído com o livro Joomla Avançado, cujo código fonte está disponível em: [http://www.novateceditora.com.br/livros/joomlaavancado/joomla\\_avancado.zip](http://www.novateceditora.com.br/livros/joomlaavancado/joomla_avancado.zip)

Um recurso bastante didático abordado nesse exemplo e que adotei é a divisão do CSS em dois arquivos, um contendo a formatação e o outro as posições.

Observe que o arquivo formatacao.css tem duas seções comentadas:

```
/* Sobrescrição de algumas tags do HTML */
```

Onde se altera o comportamento padrão de algumas tags do HTML

```
/* Parte específica do CSS do Joomla */
```

Onde se altera o comportamento padrão de algumas classes e IDs do CSS de sistema do Joomla.

No arquivo posicoes.css eu deixei os nomes de cada posição como algo que lembre as posições dos módulos do Joomla.

Observe que temos um ID que abrange toda a página útil, que é o "pagina".

Dentro do página temos três regiões principais: cabeçalho, centro e rodape.

Dentro da cabeçalho temos logo.

Dentro do centro temos menu, conteúdo e módulos.

A rodape é única.

O comportamento padrão do HTML é que cada div seja mostrada na tela uma abaixo da outra.

Quem mudou esse comportamento aqui, permitindo que uma div seja mostrada ao lado de outra, foi o comando "float".

O comando clear (com o valor both - ambos) retira a influência do float e permite ao rodape ficar numa nova linha ao final.

Veja que nesta primeira fase temos um template simples, que não usa imagens, nem imagens arredondadas, apenas usa cores de frente e fundo com background-color.

Observe que utilizamos apenas 4 posições (templateDetails.xml):

logo - para a logomarca

left - para o menu principal e outros módulos

right - para outros módulos

footer - para o rodapé

O XHTML (index.php) é bem simples, apenas contemplando as 4 posições mais o componente central e, claro, incluindo os scripts CSS entre as tags <head> e </head>.

## Green1

Basicamente neste template foi utilizado como base o arquivo `template.css` do template padrão do Joomla 1.5 (`rhuk_milkyway`) no script `formatacao.css`. Os demais scripts basicamente ficaram como estava no Green0.

Veja que o **`formatacao.css`** agora tem duas seções:

```
/** Core html setup stuff **/
```

Que sobrescreve o comportamento padrão de algumas tags do HTML.

```
/** Joomla! specific content elements **/
```

Que sobrescreve o comportamento padrão de algumas classes e IDs do CSS de sistema do Joomla.

Como alteramos o script `formatacao.css`, mesmo não alterando os demais, veja que a aparência do site, com este template também foi bastante alterada. Como este script cuida apenas da formatação, nosso template foi alterado apenas nas cores, fontes, tamanhos de fontes, etc.

## Template Green2

Este adicionou as principais posições existentes em um bom template.

```
<position>logo</position>  
<position>breadcrumb</position>  
<position>left</position>  
<position>right</position>  
<position>user4</position>  
<position>footer</position>  
<position>syndicate</position>
```

Com isso o `index.php` também sobre alterações para contemplar as novas posições. Com isso podemos ver agora um site com mais informações, mais módulos, como o RSS, a busca e o breadcrumb (que mostra o caminho percorrido pelo usuário até chegar à seção atual).

O `posicoes.css` agora também recebeu duas novas posições, o ID `pathway`, que abriga o módulo breadcrumb e o ID `search`, que abriga a busca na posição `user4`. O módulo `syndicate` (RSS) ficou no ID `rodape`, que também recebeu um novo ID `power_by`, para mostrar os créditos para a equipe do Joomla.

### Template Green3

Com isso o index.php também sobre alterações para contemplar as novas posições. Com isso podemos ver agora um site com mais informações, mais módulos, como o RSS, a busca e o breadcrumb (que mostra o caminho percorrido pelo usuário até chegar à seção atual).

O posicoes.css agora também recebeu duas novas posições, o ID pathway, que abriga o módulo breadcrumb e o ID search, que abriga a busca.

O módulo syndicate (RSS) ficou no ID rodape, que também recebeu um novo ID power\_by, para mostrar os créditos para a equipe do Joomla.

Aqui temos uma posição para a logomarca e uma para o rodapé, portanto podemos criar um módulo do tipo HTML personalizado para exibir a logomarca na posição logo. Este módulo apenas contém a imagem da logomarca no editor ao final.

Um outro módulo com uma imagem ou apenas com os créditos ou como queira para o rodapé na posição rodape.

Também se pode adicionar logo em Flash ou em JavaScript ou outra alternativa do lado do cliente.

Para adicionar em Flash use algo como:

```
<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#versi
on=6,0,40,0" height="60" width="455"><param name="width" value="455"><param
name="height" value="60">
<param name="src" value="logo.swf">
<embed type="application/x-shockwave-flash" src="logo.swf" height="60" width="455">
</object>
```

### Template Green4

Este adiciona a posição user3, que no caso recebe o menu de top (topmenu). Mas isso foi feito mas ainda não de forma completa, pois falta detalhes no CSS. Não tem outras diferenças em relação ao anterior.

Esta versão também recebeu uma miniatura do template (template\_thumbnail.png).

### Template Green5

Este adiciona as posições user1 e user2, para abrigar os módulos novidades e populares.

Estes módulos foram adaptados do template padrão do Joomla 1.5, que utilizam tabela. Nesta versão utilizaremos também com tabela mas mais a frente substituiremos por CSS. Vale também observar que eles não ficaram centralizados, mas que o faremos nas próximas versões.

Na pasta images existe uma imagem que pode ser utilizada para a criação do módulo da logomarca.

### Template Green6

A diferença deste é o uso de imagens como background de algumas regiões e do body do formatacao.css.

Nesta versão o arquivo formatacao.css foi renomeado para template.css, que é um nome usado como padrão para atender à convenção do Joomla 1.5

Observa-se que mesmo com uaspecto razoável percebe-se como remendos, ou seja, não existe um tema bonito e coerente em termos de cores. Aqui sente-se a falta de domínio com ferramentas gráficas para criar bonitas imagens.

Foi então que procurei aprender pelo menos o básico sobre o Gimp. Nas próximas versões veremos o que acontecerá usando imagens criadas especificamente para o template.

Paralelamente o script template.css foi sendo melhorado e melhor adaptado para nossos templates.

### Template Green7

Aqui já aparecem imagens criadas no Gimp, em degradê dando um melhor aspecto e todas combinando umas com as outras.

Para visualizar melhor crie os dois módulos do tipo HTML personalizado e utilize as imagens logo e rodape existentes no diretório images.

As imagens ainda não têm cantos arredondados, o que somente usaremos nos próximos.

Percebe-se que o posicoes.css pode ser bem melhorado, removendo várias classes e IDs para o template.css, para que fiquem no local mais adequado, já que não são específicos



para posicionamento.

É bom observar que a esta altura o arquivo `posicoes.css` tem seus IDs e classes todos encadeados.

Como todas as regiões principais estão dentro da grande região ID `pagina`, então essas regiões são criadas subordinadas à página, como no exemplo do cabeçalho:

```
div.pagina div#cabecalho{
```

Aqui neste template temos um problema com a exibição do menu de topo, o que será corrigido no próximo.

As imagens degradê deste template foram criadas com o Gimp da seguinte forma:

- Abra o Gimp
- Crie uma nova imagem com 1000x90pixels
- Apenas tecele em "L" ou então clique na ferramenta "Ferramenta de Mistura: preenche uma área com um degradê de cores".

Ela fica à esquerda do lápis e à direita da Ferramenta de preenchimento.

Ao ser clicada abrirá o painel abaixo.

- Clique na imagem retangular à direita de Degradê (Selecionei Caribbean Blues)
- Em Forma selecione a forma desejada (Selecionei Linear)
- Veja que existe uma grande quantidade de parâmetros que podem alterar o resultado final
- Agora clique em alguma posição da figura e arraste para outra posição. Experimente arrastar na horizontal. Tecele CTRL+Z e experimente na vertical. Tecele CTRL+Z e altere outros parâmetros e refaça até ficar satisfeito.
- Então salve (sugestão `logo.png`) Para alterar a extensão clique em Tipo de arquivo (por extensão).

No `posicoes.css` deve haver um ID assim:

```
div#cabecalho div#logo {  
    height: 90px;  
    background: url(../images/logo.png) 0 0 no-repeat;  
}
```

## Template Green8

Uma boa novidade deste é o topmenu numa região/posição adequada e com imagem de fundo.

Adicionamos a posição debug.

Vale ressaltar que para uma posição ser útil precisamos seguir a sequência:

- Adicionar no templateDetail.xml
- Criar sua formatação e posição no posicao.css
- Adicionar o respectivo jdoc no index.php (<jdoc:include type="modules" name="debug" />)

Observe que debug não requer adição ao CSS, pois no caso já tem no CSS de sistema do Joomla, mas obrigatoriamente precisa aparecer no index.php para que fique disponível na administração para aparecer no site.

Aqui as posições user1 e user2 (novidades e populares) ainda estão usando tabela mas na versão 10 será substituída por CSS.

## Template Green9

Este template já começa a ter seu aspecto melhorado.  
As posições user1 e user2 já estão centralizadas.

## Template Green10

Uma boa organização facilita bem as coisas.  
Veja o que fiz no script templateDetail.xml em relação as posições:

```
<positions>
  <position>logo</position>

  <position>user3</position> <!-- topmenu -->
  <position>breadcrumb</position>
  <position>user4</position> <!-- search -->

  <position>left</position>

  <position>user1</position> <!-- novidades -->
  <position>user2</position> <!-- populares -->

  <position>banner</position>

  <position>right</position>

  <position>footer</position>
```

```
<position>syndicate</position>
<position>debug</position>
</positions>
```

Coloquei em ordem e agrupadas quando é o caso e completando com um comentário explicativo de algumas posições.

Também usei o mesmo princípio no script posicoes.css, indentando de forma adequada e adicionando um bloco de comentário no padrão do phpdoc no início:

```
/** arquivo: posicoes.css
 *
 * Criado por Ribamar FS para demonstrar a criação de um template para o Joomla 1.5
 *
 * http://ribafs.org - novembro de 2009 - Fortaleza - Ceará - Brasil
 */

/* div principal */
div.pagina{
    background: #DFEDED;
    width: 1000px; /*Largura da página total, toda a área útil*/
    margin: 20px auto; /*Centralizar o container na página*/
}

div.pagina div#cabecalho{
    height: 90px; /*Altura da seção do cabeçalho*/
    background: url(..images/bg.png) 0 0 repeat; /* Servir de fundo para o
cabeçalho */
}

div#cabecalho div#logo {
    height: 90px; /*Altura da seção do cabeçalho. auto para qq altura de
logomarca*/
    background: url(..images/logo.png) 0 0 no-repeat; /* Servir de fundo
para o cabeçalho */
}

div.pagina div#sub{
    clear: both;
    height: 75px;
    background: #99C5A1 url(..images/bg_sub.png) no-repeat top center;
}

div#sub div#topmenu{
    margin-top: 0px;
    margin-left: 285px;
```

```
margin-right: 299px;
white-space: nowrap;
height: 25px;
background: url(../images/modulo_bg.png) 0 0 repeat;
}
```

Uma margem de uns 10pixels ao redor da página e ao redor dos módulos ajuda.

Neste template iniciamos o uso de imagens com cantos arredondados criadas com o Gimp.

Vale ressaltar que essas imagens tem a largura da página, ainda não são apenas os cantos unidos com CSS.

Foram criadas assim:

- Abrir o Gimp
- Crie uma nova imagem com 1000x90 pixels
- À direita clique na ferramenta "Cores de frente e fundo" para selecionar uma outra cor (diferente de preto)

Eu selecionei um verde com esse código (087908) e clique em OK.

- Clique na ferramenta da preenchimento (à direita do A) e clique na sua nova imagem para que tenha fundo verde.

- Agora vamos inserir uma camada transparente sobre a atual: Camada - Transparência - Cor para alfa

- Tecle CTRL+A para selecionar tudo
- Clique no menu Seleção - Retângulo arredondado ...
- Em Raio arraste para 100% e clique em Ok.
- Clique no menu Seleção novamente - Inverter
- Tecle CTRL+X para remover os cantos
- Tecle CTRL+S para salvar. Escolha a extensão PNG e salve como logo\_redondo.png

Caso queira transformar uma das imagens degradê para cantos arredondados, apenas abra uma no Gimp e use os passos acima para cortar os cantos de forma arredondados.

Observe a sincronização do CSS com as imagens de cantos arredondados.

## Template Green11

Neste foi adicionado o favicon.ico:

- Selecione uma imagem que deseje transformar no símbolo do seu site. Pode ser ico, png, gif ou jpg.
- Redimensione para 16x16 pixels (outros tamanhos podem funcionar)

- Renomeie para favicon.ico
- Copie para o raiz do template

Neste resolvi inserir uma imagem diretamente na região logo, já no arquivo posicoes.css Também poderá criar o módulo para o logo e usar, caso queira.

Resolvi remover da parte inferior do logo e da superior do sub os cantos arredondados, de forma que fiquem arredondados apenas os cantos superiores do logo.

Nesta versão removi a última tabela que estava usando no template (topmenu). Agora o template é 100% CSS.

## Template Green12

Neste resolvi inserir uma imagem diretamente na região logo, já no arquivo posicoes.css Também poderá criar o módulo para o logo e usar, caso queira.

Resolvi remover da parte inferior do logo e da superior do sub os cantos arredondados, de forma que fiquem arredondados apenas os cantos superiores do logo.

Também usei os cantos arredondados no rodape.

## Banner

Alterei a posição do módulo banner para "banner" para ficar mais adequado. Agora fica abaixo do conteúdo (como no template rhuk\_milkyway).

É interessante observar a sintaxe do background com cor de fundo, em especial para cantos arredondados:

```
background: #99C5A1 url(..images/bg_sub_deg.png) no-repeat;
```

## Template Green13

Neste adicionei o supote a parâmetros, no caso, com variação de cor e de cor de fundo, adaptados do template rhuk\_milkyway.

Experimente alterar na seção de administração do template, clicando no link do template Green13.

Observe também o arquivo params.ini, que guarda os valores selecionados no template.

Agora o arquivo posicoes.css foi primeiramente duplicado formando:

verde.css e verde\_bg.css

O verde.css é uma cópia fiel do posicoes.css.

O verde\_bg.css contém apenas:

```
body {  
    background: url(../images/verde/bg_verde.png) 0 0 repeat;  
}
```

Mas deve ser melhorado.

Depois copieie estes dois arquivos:

O verde.css copieie para azul.css

O verde\_bg.css para azul\_bg.css

Altereie as imagens de fundo para imagens azuis nos scripts azul e azul\_bg e também as cores dos background.

## Template Green14

Neste adicioneie o supote a parâmetros, no caso além das cores também para a largura da página (fixa e fluida), adaptados do template rhuk\_milkyway.

Experimente alterar na seção de administração do template, clicando no link do template Green14.

Observe que não troqueie adequadamente as imagens nem cores para um final adequado, mas o projeto está completo e com exemplos funcionais.

Agora temos um tempalte de exemplo explicado com detalhes que pode servir de base para que crie templates bem melhores e com muitos recursos.

Obs.: a largura foi implementada corretamente para as duas opções mas somente para a cor azul.

Mas ainda falta trabalhar as imagens e também centralizar a região conteúdo quando usar largura máxima.

## Template Green15

Neste o diferencial é o logo que é grande e também tem ativado o news flash.

Como também teve a cor dos links alterada para verde.

## Template Green16

Este conta com recursos muito bonitos:

- Ícones à esquerda do título de cada artigo (classe contentheading)
- Itens de menu com imagem de fundo e item de menu ativo diferenciado (classes moduletable\_menu e menu)

## 10 – Referências

### Livros Iniciantes

- Beginning Joomla! do Dan Rahmel - Editora Apress - <http://www.apress.com>
- Joomla! with Flash do Suhreed Sarkar - Editora Packt Publishing - <http://www.packtpub.com>
- Up and Running with Joomla de R. Allen Wyke and Skip Matheny - Editora O'Reilly
- Building Websites with Joomla! 1.5 do Hagen Graf - Editora Packt Publishing
- Joomla! 1.5x Customization do Daniel Chapman - Editora Packt Publishing
- Joomla! Cash de Brandon Dawson e Tom Canavan - Editora Packt Publishing
- Joomla! For Dummies de by Steven Holzner, PhD e Nancy Conner, PhD - Editora Wiley Publishing, Inc. - <http://www.wiley.com>

### Avançados

- 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

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

### Links

Busca no site oficial do PHP - <http://php.net/>  
Manual - [http://www.php.net/manual/pt\\_BR/](http://www.php.net/manual/pt_BR/)  
[http://pt.wikibooks.org/wiki/Aplicativos\\_em\\_PHP](http://pt.wikibooks.org/wiki/Aplicativos_em_PHP)

### CSS

CSS1 - <http://www.w3.org/TR/REC-CSS1/>  
CSS2 - <http://www.w3.org/TR/CSS2/>

### HTML

HTML 4.01 - <http://www.w3.org/TR/html4/>  
XHTML1  
Online <http://www.w3.org/TR/xhtml1/>  
PDF - <http://www.w3.org/TR/xhtml1/xhtml1.pdf>



### JavaScript

<http://mediasrv.ns.ac.yu/extra/html/jsref/index.html>  
[https://developer.mozilla.org/en/JavaScript\\_Language\\_Resources](https://developer.mozilla.org/en/JavaScript_Language_Resources)  
<http://www.w3schools.com/jsref/default.asp>  
<http://docs.sun.com/source/816-6408-10/contents.htm>  
<http://javascript-reference.info/>

### 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://joomlancode.org/gf/project/portuguesebr/frs/>

### Extensões diversas

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

### Jumi

<http://jumi.vedeme.cz/>

Jumi is a set of universal multipurpose tools - modules and plugins - for Joomla!  
They natively include into Joomla! articles and/or at module positions any readable file from your server.

### Template Builder Online

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

### Live Chat

<http://www.dayanahost.com/phponline.cfm>

Download - <http://phponline.dayanahost.com/download.php?v=2.1.3>