

**GUIA BÁSICO
DE
DESIGN
PARA
MAGENTO**



Baseado no Magento design Guide

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779> 

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

Guia básico de design para Magento

Baseado no Magento design Guide

Versão 1.0

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779>



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

Índice

1 Visão geral	5
Escopo deste manual	5
Pré-requisitos	5
Sobre este Guia	6
Sobre a tradução	6
2 Design Magento - Conceitos e terminologia	7
Sites, Lojas e visualizações de loja	7
Pacotes de temas (design package) e temas	9
Temas	10
Tema default	11
Variações ou tema personalizado (não default)	12
O modelo FALLBACK do Magento	13
Blocks e layouts	14
Blocks	14
Widgets	16
Terminologia widget	16
Exemplos de widget	16
3 Estrutura de diretórios dos Pacotes de temas e dos temas no Magento	18
O pacote BASE	18
O pacote default	20
Pacotes de tema personalizado	20
Temas	21
4 Aplicando temas em Magento	22
Passo a passo 1: Criando e aplicando um tema	22
Atribuindo um pacote e um tema para a loja	22
Passo a passo 2 – Aplicando múltiplos temas	25
Hierarquia dos temas	30
Importantes regras de hierarquia a se lembrar	32
Exceções de design	33

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779>



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

5 Customizando temas Magento	34
Criando os diretórios básicos para seu pacote/tema	34
Criando um novo pacote e tema(s)	34
Aplicando seu novo pacote e tema para seu site	35
Desabilite o sistema de cache	35
Aplique seu pacote/tema personalizado em sua loja	36
Customizando os arquivos skin	36
Exercícios rápidos para começar	36
Exercício 1: Modifique o CSS	36
Exercício 2: Alterar a LOGO	36
Customizando com arquivos de layout	37
Introdução ao layout	37
Como o layout do Magento funciona	40
A anatomia de um arquivo de layout	42
Controladores	42
<block>	42
<reference>	43
Regras do XML	44
Como encontrar o arquivo de layout para modificar	45
Exercícios rápidos para começar	46
Exercício 1	46
Exercício 2	47
Customizando com o arquivo de layout local (local.xml)	47
Customizando com arquivos templates	49
Exercício 1	50
6 Guia rápido para criação de um tema do zero	52
Primeiro: desabilite o cache do sistema	52
Segundo: defina todos os tipos de estruturas possíveis para sua loja	52
Terceiro: corte seu xHTML de acordo para funcionalidade	53
Quarto: altere o layout para refletir seu design	55
7 Outros Recursos	56

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779> 

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

1 Visão geral

Guia básico de design para Magento é uma introdução ao conhecimento sobre o funcionamento estrutural do Magento e os métodos para desenvolvimento de temas. Explica como criar um tema totalmente personalizado com todas suas preferências, descreve o método mais rápido e descomplica o desenvolvimento de um novo tema. Se você está começando já deve ter percebido que o Magento é um sistema extremamente flexível e complexo, não cabendo aqui explicar todos os diferentes métodos de personalização, no entanto, este guia trará o básico indispensável para criação de temas que garantirá a você um ponto de partida para sua curva de aprendizagem. Para se manter atualizado, visite sempre o fórum oficial da comunidade, o wiki para adquirir mais detalhes e meu blog www.ronaldodiniz.com.br onde sempre incluo dicas importantes de design em geral com foco principal no Magento. Além disso, lembre-se que o Magento está em constante atualização, portanto, esta documentação pode não refletir fielmente a sua versão do Magento, mas ainda sim serão úteis todos os conceitos apresentados a você.

Com este guia você entenderá a estrutura de um tema para Magento, se familiarizará com termos mais comuns usados pela comunidade e principalmente entenderá que este sistema não é um bicho de sete cabeças (risos).

Escopo deste manual

O foco deste guia voltado para o Magento Community Edition e suporta a partir da versão 1.4, não vou aqui falar sobre o Magento Go ou Enterprise pois não atendem ao público brasileiro em geral, mas fique tranquilo pois a diferença fica apenas na nomenclatura de algumas pastas, para mais detalhes consulte o Magento Design Guide.

Pré-requisitos

Você precisará de uma versão Magento CE instalada para praticar os exercícios mostrados neste manual. Eu recomendo, mas não é obrigatório o domínio de XHTML, CSS, e noções básicas de PHP e XML

Além disso, para facilitar o entendimento do Magento aconselho a instalação da tradução feita pelo Mário SAM, que na minha opinião é a tradução mais fiel e atualizada, confira o pacote de tradução no www.mariosam.com.br e baixe a tradução mais recente.

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779> 

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

Sobre este Guia

Este guia foi baseado no Magento design Guide, distribuído gratuitamente pela Magento inc (original em inglês).

Este guia contém capítulos que podem ser ignorados a fim de acessar informações mais importantes para você no momento. Cada capítulo, no entanto, funciona como um prelúdio para o próximo, então aconselho a estudar este manual na ordem descrita e no futuro use como fonte de consulta.

Sobre a tradução

Antes de ler este manual entenda que eu não sou especializado em traduções, meu intuito aqui é trazer a você, caro leitor, um conteúdo que me ajudou muito no início do meu aprendizado com o Magento e juntamente melhorar meu conhecimento em inglês.

Por isso perdoe alguns possíveis erros que este manual possa conter e ficarei muito grato que me informe dos mesmos pois assim oderei melhorar cada vez mais.

Esta é a primeira versão da tradução do guia, portanto periodicamente visite meu site para saber sobre versões mais atuais deste guia, ou uma versão melhorada do mesmo. use o link no topo da pagina ou acesse: <http://www.ronaldodiniz.com.br>.

Bons estudos.

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779>



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

2 Design Magento - Conceitos e terminologia

Para acompanhar este guia é importante que você entenda a terminologia e os conceitos fundamentais dos aspectos frontend do Magento. É provável que estes termos sejam novidade para você, não desanime caso não entenda tudo de imediato. Aqui darei uma visão geral e no decorrer dos próximos capítulos entraremos com mais detalhes, desta forma você entenderá estes termos com mais fluidez.

Sites, Lojas e visualizações de loja

O Magento suporta a criação e manutenção de múltiplas lojas em uma única instalação, por isso, estabeleceu uma hierarquia que define várias lojas individuais em uma única instalação.

No Magento um site é uma coleção de lojas e estas são subdividas em visualizações de lojas. Estas camadas, embora pareça confuso inicialmente, garante uma poderosa flexibilidade quando se projeta soluções para negócios com Magento.

Um **site** é composto de uma ou mais lojas que compartilham as mesmas informações, formas de pagamento e carrinho de compras

Lojas são coleções de visualizações de loja e podem ser configuradas de diferentes maneiras. Sua principal função é fornecer uma plataforma que permite agrupar as **visualizações de loja** dentro de uma lógica para se relacionarem entre si junto ao site. Visualizações ou visões de lojas (como preferir) são as instâncias de lojas propriamente ditas. A maioria das lojas terá uma única visão de loja associada, Mas a loja também pode ter múltiplas visões de loja, este recurso é normalmente usado para diferenciar idiomas. Portanto, caso queira ter uma loja em Português e Espanhol, por exemplo, com o mesmo banco de dados podemos criar duas visualizações de lojas diferentes para cada idioma.



Figura 1 – Hierarquia sobre Sites, Lojas e Visualizações de lojas no Magento

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779>



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

Estes são conceitos muito gerais e podem ser adaptados para atender às necessidades exclusivas de cada loja, veja a seguir alguns cenários para ilustrar os diferentes usos de sites, lojas e visualizações de lojas.

Cenário 1 – loja única

Um a empresa chamada “Bongo Ferramentas” quer expandir seus negócios para internet. A Bongo tem um único catálogo e não precisa de suporte para idiomas. Este é o cenário mais simples onde a estrutura será: Site, Loja, Visualização de loja.

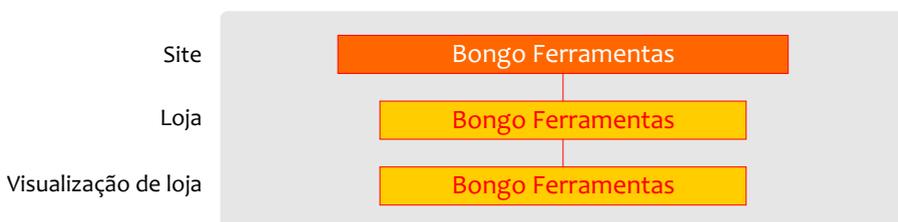


Figura 2 – Simples Site, Loja e Visualização de Loja

Cenário 2 – Múltiplas lojas relacionadas

Uma Companhia chamada “Dubloo” cria uma empresa on-line, com três lojas de roupas com diferentes níveis de preço para públicos específicos. Dubloo quer que todas as lojas compartilhem seus clientes, de forma que ao se cadastrar em uma das lojas ele estará cadastrado em todas. Neste cenário Dubloo terá um Site com três lojas online. Pelo fato de cada loja suportar um único idioma, cada loja terá apenas uma visualização de loja.

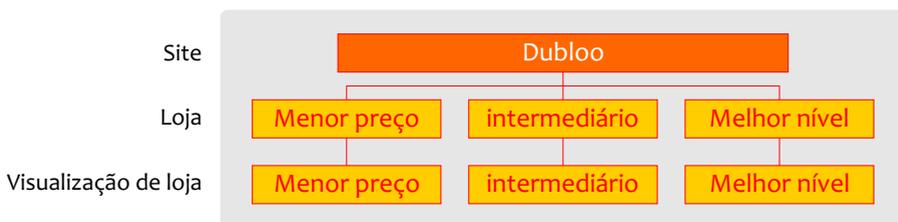


Figura 3 – Único Site com múltiplas lojas

Cenário 3 – Múltiplas lojas independentes

A companhia “Home Laptops” quer uma presença online com duas lojas distintas em que ambas vendem notebooks, mas com diferentes preços e diferentes seleções de produtos em algumas categorias. Eles também querem oferecer opções de idioma inglês e português em cada loja. O cliente precisa se cadastrar individualmente em cada loja, mas as

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779>



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

lojas não precisam compartilhar estas informações entre si. Neste cenário Home Laptops teria dois sites (com clientes separados compartilhados em um único backend) cada uma com uma loja e duas visualizações de loja (uma para inglês e outra para português).

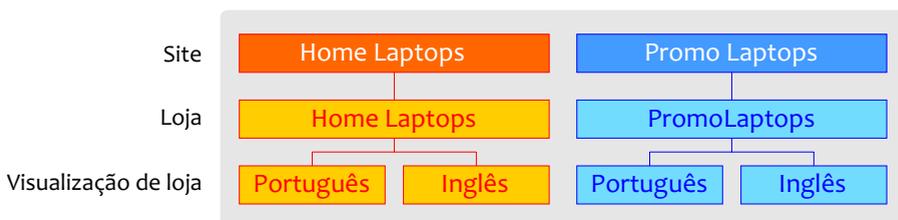


Figura 4 – Múltiplos sites com múltiplas lojas e visualizações de lojas

Pacotes de temas (design package) e temas

Ao observar a hierarquia de sites, lojas e visualizações de lojas na seção anterior, você pode ver que uma única instalação Magento pode executar várias lojas que podem, ou não, compartilhar o visual e algumas informações.

Para controlar exatamente a aparência de cada loja na sua instalação, o Magento permite que você crie temas. Temas relacionados são agrupados em pacotes de temas (original: design packages, algumas vezes chamado apenas de pacote o package). Cada loja Magento pode usar seu próprio tema, ou todos eles podem compartilhar um único tema, ou uma combinação de ambos.

Um pacote de temas é uma coleção de temas relacionados. Obrigatoriamente terá ao menos um pacote instalado, mas você pode ter quantos pacotes quiser.

Quando instalado, Magento tem um pacote de tema especial chamado “base” e outro pacote de exemplo chamado “default”. Você pode instalar quantos pacotes de tema desejar.

Um Tema dentro de um pacote de temas contém os arquivos reais que determinam a aparência e a funcionalidade do frontend de sua loja. Os temas são abordados com mais detalhes na próxima seção, mas resumidamente um tema contém arquivos de estrutura (layout, template e tradução específicos) e informações de estilo (arquivos css, imagens e arquivos javascripts específicos do tema). Cada tema pode pertencer a apenas um pacote. Por convenção cada pacote de temas deve conter um tema padrão chamado “default”. Além disso, um pacote de temas pode ter um número ilimitado de variações desse tema padrão e podem ter o nome que desejar.

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779>



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

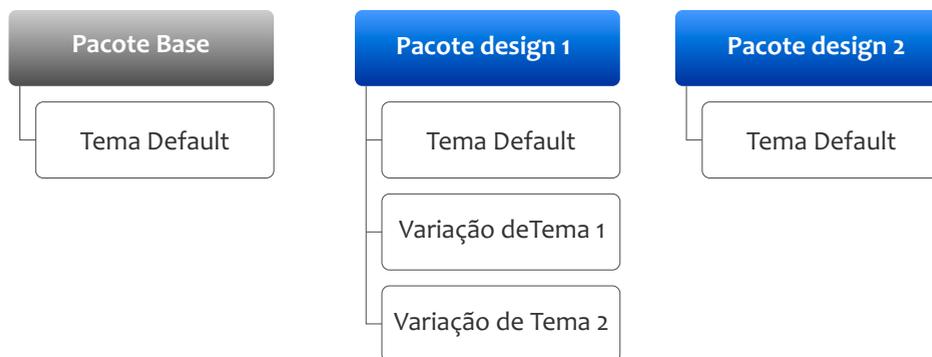


Figura 5 – Relação entre pacote de temas e temas no Magento

Tanto o pacote de temas quanto o tema pode ser atribuído ao site e/ou a visualização de loja através do painel administrativo do Magento. Se você atribuir um pacote no nível de site, todas as visualizações de loja herdarão esse pacote. Você pode atribuir mais de um pacote na loja e/ou visualização de loja de sobrepondo a configuração do site de forma simples e eficaz.

Temas

Um tema Magento é composto de arquivos estruturais (layout, template e locale) e arquivos de skin (css, imagens e JavaScripts específicos do tema) que são responsáveis pela experiência visual da sua loja. Estes arquivos residem em duas principais pastas dentro do seu sistema:

- Diretório “app/design” são arquivos que controlam a forma como os modelos de página são montados.
- Diretório “skin” arquivos que controlam os aspectos visuais do site, arquivos css, imagens, etc.

O Magento quebra seus arquivos de tema em diretórios separados para um maior nível de segurança de cada pasta em seu servidor. Os arquivos do diretório skin precisam ser acessados pelo browser e tem uma configuração de permissão muito aberto. Os arquivos no diretório app/design só precisam ser acessíveis ao aplicativo e tem um nível mais fechado, bloqueando o acesso via browser.

Dentro de cada diretório, os arquivos são divididos em mais subdiretórios de acordo com o tipo:

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779>



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

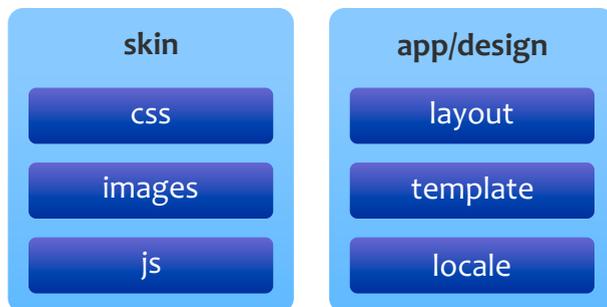


Figura 6 – Tipos de arquivos associados aos temas Magento e suas localizações

Abaixo uma pequena explicação sobre cada diretório e os arquivos contidos.

Arquivos de template na pasta `app/design/frontend/< pacote de temas>/<tema>/` são organizados nas seguintes pastas:

- **Layout** - contém arquivos XML básicos que definem a estrutura de diferentes blocos para todas as páginas bem como os metadados e codificação.
- **Template** - contém arquivos PHTML neles temos marcações (x)HTML e PHP necessário para apresentação lógica. Alguns arquivos são páginas do tema e outras são blocos do mesmo.
- **Locale** - simples arquivos de texto CSV organizados em uma base por conteúdo de tradução (como pares de nome e valor) para todo texto produzido pelo Magento (por exemplo: para elementos de interface e mensagens e não para produtos e categorias).

Arquivos de visual são organizados em `skin/frontend/< pacote de temas>/<tema>/` nas seguintes subcategorias:

- **css** - contém os arquivos CSS usadas no visual do site.
- **Images** - contém todas as imagens usadas pelo tema.
- **Js** - contém JavaScript específicos do tema como rotinas e funções. Bibliotecas javaScript e demais arquivos que possam ser compartilhados entre os temas são colocados na pasta `/js` no diretório raiz do Magento.

Tema default

Cada pacote de temas do Magento deve conter um tema chamado “default”, o tema principal do pacote.

Ao atribuir um pacote de tema em sua loja, a menos que descrema um tema específico, o Magento automaticamente procurará pelo tema “default” no diretório do pacote e irá usá-lo para montar o frontend.



Figura 7 – Os pacotes de tema devem ter um tema “default”.

O tema default deve conter todos os requeridos layouts, templates e skins necessários para montagem da loja. Para qualquer arquivo que não for encontrado no tema personalizado o Magento irá olhar o tema default do pacote.

Variações ou tema personalizado (não default)

Uma variação do tema não precisa ter todos os arquivos como no default, ele deve conter os arquivos necessários para a personalização. Variações de tema permitem que você faça facilmente modificações do tema padrão e que podem ser aplicadas a toda sua loja, subseções de loja, páginas específicas ou a uma loja separada dentro da mesma instalação Magento. Você pode alterar o tema default para o tema personalizado, além de carregá-lo junto com o padrão.

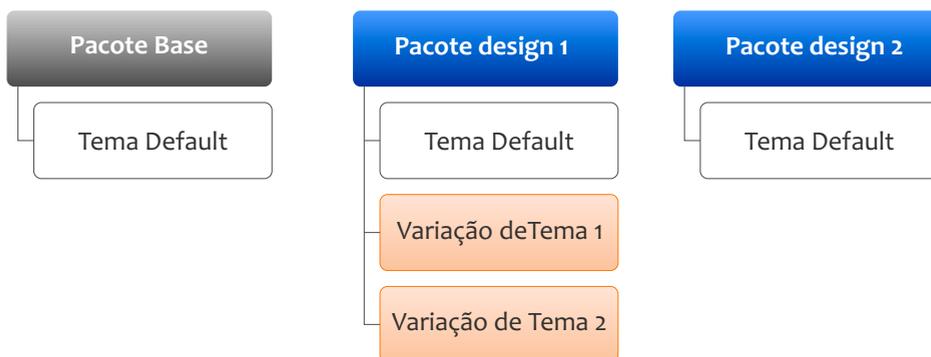


Figura 8 – Variações de tema em um pacote de temas.

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779>



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

Para exemplificar, nós podemos:

- Sobrescrever uma parte do seu site que precise de menos detalhe na página de produto.
- Criar uma seção de sua loja que antecipa a próxima coleção de produtos, mas sem oferecê-los a venda.
- Mudar de visual para uma data comemorativa, mas apenas por um período.
- Criar um hotsite de outra divisão da sua empresa, que precisa da mesma marca global como a do site principal, mas com modificações para adequar o catálogo e o público.

O modelo FALLBACK do Magento

A arquitetura dos temas para Magento mudaram da versão 1.3 para 1.4 de forma a garantir maior integridade perante tantas atualizações. Nessa mudança os arquivos do tema foram reformulados consideravelmente para melhorar o desempenho e acessibilidade, mas a maior mudança foi na estrutura e na hierarquia usada pelo Magento.

Magento sempre usou a lógica fallback para renderização da loja, mas na versão 1.4 isso foi ampliado para incluir o pacote de temas base como ponto de fallback final. Em versões anteriores o fallback era apenas dentro do tema e uma função personalizada não poderia ser compartilhada entre os temas sem replicar todos os arquivos.

Veja este exemplo para entender o quê é fallback e como ele funciona dentro do Magento, caso seu tema personalizado exija um arquivo CSS chamado “styles.css”, mas o Magento é incapaz de encontrar este arquivo dentro do seu tema personalizado, o Magento irá então pesquisar na pasta do tema seguinte da hierarquia e continuará a procurar seguindo a hierarquia até encontrar o arquivo “styles.css”. Este método de construção é chamado de “fallback” porque a aplicação “cai para traz” para a próxima fonte retroativa possível de arquivos necessários para recuperar e carregar um arquivo perdido.

A hierarquia fallback do Magento funciona como mostrada no diagrama a seguir.

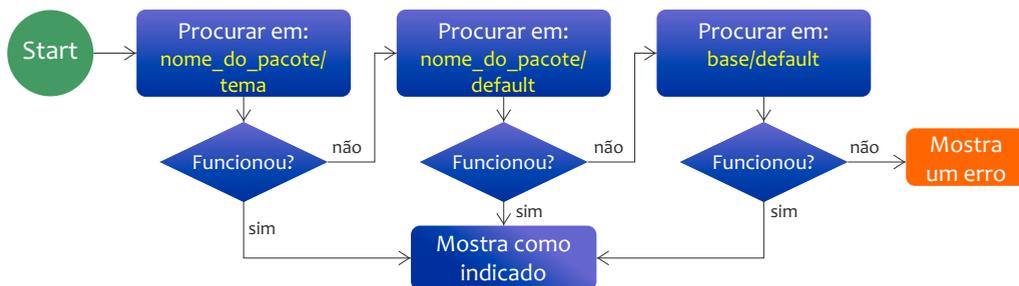


Figura 9 – lógica de fallback seguida pelos temas do Magento.

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779> 

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

Quando usada de forma correta, este mecanismo permite que você edite e mantenha apenas os arquivos que precisa personalizar para seu tema personalizado e todas as outras funcionalidades são fornecidas pelo tema default ou pelo pacote de temas base.

Por exemplo, o Magento mantém 4 temas de demonstração – default, blank, modern e iphone. Antes da introdução do pacote base, todos os arquivos do tema tinham que ser copiados para cada pacote e mantido, testado e debugado em cada um. Com a introdução do pacote base mantemos três vezes menos código. Para entender veja os temas default/default e default/blank que são implementados somente com mudanças de estilo CSS e obtém todos os arquivos de template do pacote base.

E por último, faz que seus temas sejam mais “upgrade proof” (literalmente a prova de upgrades) e permite que você precise administrar bem menos código. A maioria dos temas personalizados só altera um subconjunto das funcionalidades Magento, por isso, todas as principais funcionalidades estarão disponíveis no pacote base e pode ser sobrescrito por um arquivo específico do tema personalizado.

Blocks e layouts

Existem alguns conceitos e ferramentas que você precisa entender para ser um designer bem sucedido no Magento:

- Blocos estruturais (Structural Blocks)
- Blocos de conteúdo (Content Blocks)
- Layout

Blocks

Blocos é uma maneira pela qual o Magento distingue o conjunto de funcionalidades no sistema e forma modular para gerenciá-lo tanto do ponto de vista visual como funcional. Existem dois tipos de blocos que trabalham juntos para criar uma saída visual.

Blocos estruturais - criados com o único propósito de atribuir uma estrutura visual a uma página da loja como head, coluna da esquerda, coluna principal e footer.

Blocos de conteúdo - produzem o conteúdo real dentro de cada bloco estrutural. Estes representam cada funcionalidade de cada recurso dentro de uma página e utilizam os arquivos de template para gerar o xHTML a ser inserido no seu bloco estrutural pai. Alguns blocos de conteúdo são: Category list, mini cart, product tags, product listing, etc. Ao invés de incluir conteúdo após conteúdo a fim de reunir toda a produção do xHTML, algo típico em outros sistemas de loja, o Magento organiza e agrupa todo o conteúdo em blocos.

Layout - essencialmente são arquivos que mapeiam os blocos de conteúdo para seus blocos estruturais. Layouts em Magento possuem duas funções: Definir os blocos estruturais e de conteúdo e informar ao Magento como e onde eles se conectam.

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779>



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

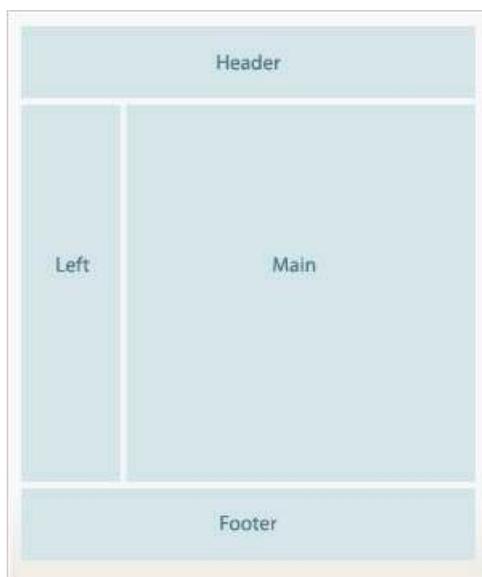


Figura 10 – Um exemplo de blocos estruturados.



Figura 11 – Um exemplo de blocos de conteúdos mapeado para blocos estruturados específicos.

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779> 

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

Widgets

Você deve estar familiarizado com o termo “widgets” de outras plataformas online. No Magento widgets são blocos do frontend com um conjunto de opções configurações pré-definidas. Estas opções de configurações são exibidas em uma área específica no backend da loja onde um widget é criado ou editado pelo administrador da loja. A facilidade de configurar opções do widget permite o total controle de uma página para os administradores. Um aspecto importante dos widgets é que a sua utilização elimina alguns níveis de personalização do tema que era exigida para a configuração de uma loja. Essencialmente é um método que permite aos usuários inserir códigos dinâmicos sem nenhum conhecimento de programação (incluindo os dados do produto, por exemplo), em áreas da Loja Magento pré-definidas pelo designer / programador. Isso garante maior controle e flexibilidade na criação de informação e de conteúdo de marketing por meio de ferramentas do administrador, permitindo um controle intuitivo e eficiente do conteúdo, tais como:

- Dados de produtos dinâmicos em uma página de marketing.
- Informações dinâmicas como produtos vistos recentemente em uma página.
- Imagens promocionais posicionadas em diferentes blocos, coluna do site e em outros locais em todo frontend.
- Elementos interativos e blocos de ação (sistemas externos, Chats, vídeos, votação, etc).
- Elementos flash dinâmico e interativo facilmente editado e incorporado dentro de uma página aprimorando a experiência interativa para o usuário.

Terminologia widget

- Frontend Block – elemento que cria uma saída estrutura visual atribuindo uma estrutura visual ou produção de conteúdo real
- Magento Widget - um bloco frontend que implementa um widget especial que permite diferentes opções de configuração por cada instância do bloco, bem como a capacidade de ter múltiplos blocos independentes na página.
- Magento widget instance – um bloco em uma ou várias páginas que receba estas opções de configuração definidas pelo administrador da loja. O mesmo widget pode ser adicionado várias vezes em lugares diferentes, gerando várias instâncias do mesmo widget.

Note que a mesma instância pode ser adicionada em várias páginas (com as mesmas opções de configuração) e editada em um único lugar.

Exemplos de widget

Por padrão o Magento inclui os seguintes widgets configuráveis, novos widgets podem ser criados por desenvolvedores livremente.

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779> 

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

- **CMS Page Link** - mostra um link de uma página selecionada e permite especificar o texto e título. Dois temas estão disponíveis neste widget, links em linha ou em bloco.
- **CMS Static Block** - exibe o conteúdo de um bloco estático.
- **Catalog Category Link** - mostra um link de uma categoria selecionada e permite especificar o texto e o título, dois temas (em linha ou bloco) estão disponíveis.
- **Catalog Product Link** - mostra um link de um produto selecionado e permite especificar o texto e o título, dois temas (em linha ou bloco) estão disponíveis.
- **Recently Compared Products** - exibe um bloco que contém produtos que foram comparados recentemente. Este widget permite especificar o número de produtos a serem exibidos e tem dois modelos disponíveis (lista ou grade).
- **Recently Viewed Products** - exibe os produtos recentemente visualizados. Permite especificar o número de produtos a serem exibidos e tem dois modelos disponíveis: lista e grade.

3 Estrutura de diretórios dos Pacotes de temas e dos temas no Magento

Temas são agrupados em um pacote de temas, mas, como descrito no capítulo anterior, os arquivos de tema são separados em dois diretórios. Isto soa muito estranho quando se está iniciando na tematização do Magento. Por isso este capítulo mostra a estrutura de diretório associada a cada um dos principais pacotes e temas usados no Magento. À medida que passar por esta seção observe que os nomes de diretórios para os pacotes e temas devem ser os mesmos em ambos app/design e skin. Observe também que abaixo desse nível em cada diretório, o Magento tem um conjunto de diretórios necessários em que seus arquivos de estrutura e skin residem.

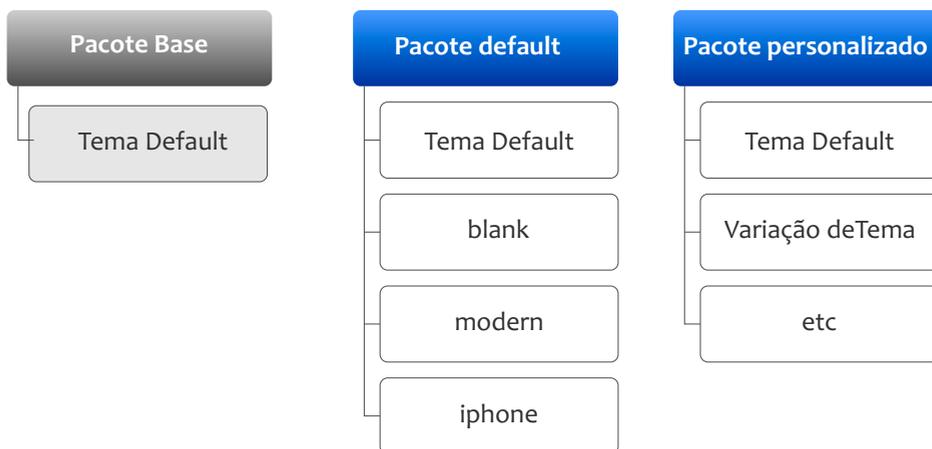


Figura 12 – Tipos de pacotes e temas no Magento

O pacote BASE

O pacote base foi introduzido no CE v1.4. Sua função é garantir uma base para todas as principais funcionalidades do Magento, de modo que seus temas personalizados possam incluir somente as alterações específicas do projeto ou negócio que se destinam. O Magento faz isso usando o modelo “fallback”, no qual ele procura os arquivos necessários para processar em uma página primeiramente dentro de seu pacote de temas e caso não encontre ele seguirá para o pacote base.

Nota: O pacote base é o ponto final de fallback para todos os seus pacotes de temas, nunca deve ser alterado quando você criar seus próprios pacotes de tema.

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779> 

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

Isto promove uma base de código limpa e um melhor meio de atualização para seus temas porque o pacote base conterá todos os arquivos que controlam o comportamento padrão do Magento (e existe um monte deles), mas seu tema personalizado apenas precisa conter os arquivos que você tenha alterado ou usado para substituir o comportamento padrão. O pacote base possui apenas o tema default e nenhuma variação pacote/tema deve ser incluída dentro do pacote base.

Na estrutura de diretório, o tema base tem tanto templates quanto skin que estão nos diretórios app/design e skin, respectivamente. O pacote base possui apenas um tema default associado a ele, embora não seja realmente um tema completo, pois lhe falta a maioria dos arquivos de skin.

Estrutura de diretório do pacote base

app/design/frontend/base/default

Contem todos os arquivos de layout e template necessários para as funcionalidades do core do Magento

<diretório_magento>

```
- app
+ core
- design
  + adminhtml
  - frontend
    - base
      + default
    + default
  + install
+ etc
+ locale
Mage.php
```

skin/frontend/base/default

Contém alguns arquivos CSS e JavaScript que suportam as funcionalidades do core. Não se trata, no entanto, de todos os CSS e de imagem necessários ao visual de um site, porque estes são específicos do design e não do core.

<diretório_magento>

```
- skin
  - adminhtml
  - frontend
    - base
      + default
  + instal
```

Base é realmente um repositório que torna as funcionalidades do core do Magento válidas para o frontend. Portanto não faça alterações no pacote base – arquivos no pacote base nunca devem ser editados. Temas personalizados devem ser feitos dentro das pastas de seus pacotes. Qualquer arquivo que precise alterar deve ser copiado do pacote base para eu tema personalizado ou ainda pode criar um conjunto de arquivos para manter suas alterações.

Regras para trabalhar com o pacote base:

- NUNCA editar os arquivos do pacote base.
- NUNCA criar um tema personalizado dentro do pacote base.

Nota: versões do Magento anteriores ao CE v1.4 não tem um pacote base, estas

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779>



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

informações não são válidas para eles, é importante que você saiba disso para não se basear em tutoriais de tema que não use o tema base.

O pacote default

Você já notou que em cada instalação Magento não possui apenas o pacote de temas base, mas também pelo menos outro pacote de tema disponível por padrão, este pacote é chamado de default.

Na estrutura de diretórios este tema tem tanto arquivos de tema quanto de layout, que estão nos diretórios **app/design** e **skin**, respectivamente. Diferente do pacote base, os temas neste pacote de demonstração são temas completos, terão todos os arquivos CSS e imagens necessários. Estes temas podem ser usados caso queira usar o tema default do Magento para edição de sua plataforma (eles se parecem com as lojas que você vê de demonstração) como um ponto de partida. Sua melhor finalidade, no entanto, é ser como uma referência ou exemplo para a criação do seu próprio pacote de temas.

O pacote default possui vários temas e sua estrutura é semelhante a esta:

Estrutura do pacote default

app/design/frontend/default/

<MAGENTO_BASE_DIR>

```
- app
  - design
    - frontend
      - base
      + default
    - default
      + default
      + blank
      + default
      + iphone
      + modern
```

skin/frontend/default/

<MAGENTO_BASE_DIR>

```
- skin
  - frontend
    + base
    - default
  - default
    + default
    + blank
    + blue
    + default
    + iphone
    + modern
```

Pacotes de tema personalizado

Como criar um pacote de temas? A maior parte do restante deste manual abrange os passos para fazer isso, mas você já deve entender que, com base no que vimos até agora, você começará criando as pastas do seu pacote nos diretórios **app/design/** e **skin/**.

Por exemplo, imagine que você queira criar um novo tema chamado “sublime”, você deve criar uma pasta de pacote de temas chamado sublime dentro de suas pastas de frontend e criar uma pasta default dentro de cada uma delas.

Diretório do tema personalizado Sublime:

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779>



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

Estrutura do pacote personalizado Sublime

app/design/frontend/sublime/default/

<MAGENTO_BASE_DIR>

```
- app
  + code
  - design
    - frontend
      + base
      + default
    - sublime
      + default
```

skin/frontend/sublime/default/

<MAGENTO_BASE_DIR>

```
- skin
  - frontend
    + base
    + default
  - sublime
    + default
```

Temas

O que vai dentro de cada pasta do tema sempre dependerá do nível de customização utilizado. Tenha em mente que com o fallback do Magento, que sempre vai achar os arquivos no pacote base, cada tema precisa apenas conter os arquivos que diferem daqueles que estão no pacote base.

Como você já sabe, cada pacote precisa ter um tema default, variações de tema você terá quantos quiser. Dentro de cada tema, a estrutura de arquivos deve refletir a estrutura que o Magento espera. Continuando com nosso exemplo do pacote sublime, o tema default deve ter ao menos as pastas principais do tema como um esqueleto para que tenha onde alocar seus arquivos enquanto trabalha. Todos os temas precisam dessa estrutura, sendo que algumas destas pastas podem acabar vazias e poderão ser excluídas posteriormente.

Estrutura para o tema default do pacote sublime

app/design/frontend/sublime/default/

<MAGENTO_BASE_DIR>

```
- app
  - design
    - frontend
      + base
      + default
    - sublime
      - default
        + layout
        + template
```

skin/frontend/sublime/default/

<MAGENTO_BASE_DIR>

```
- skin
  - frontend
    + base
    + default
  - sublime
    - default
      + css
      + images
      + js
```

Outra alternativa é criar seu tema dentro do pacote default como uma variação dele, muitos temas prontos vendidos na internet trabalham desta forma, mas aqui sugiro que crie seu próprio pacote para que não mova ou apague algum arquivo acidentalmente da pasta default.

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779> 

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

4 Aplicando temas em Magento

Neste capítulo veremos em dois exemplos como o Magento processa os arquivos do pacote e seus temas

Passo a passo 1: Criando e aplicando um tema

Neste passo, mostrarei a você como criar um pacote e um tema e aplicar em uma loja Magento. O próximo capítulo vai aprofundar em como customizar um tema. Neste ponto assumo que já tenha criado um novo pacote com pequenas alterações, seja por download a partir do Magento connect ou de terceiros.

Atribuindo um pacote e um tema para a loja

Agora que você criou seu próprio pacote e um tema (seja default ou personalizado), você precisa atribuí-lo à sua loja/site para que este faça efeito. Navegue em Magento admin (ex.: www.seudominio.com/admin), na aba visual (sistema > Configuração > guia Design ou visual, dependendo da sua tradução). Caso tenha mais de um site, loja ou visualização de loja definida na sua instalação Magento, no canto superior da coluna da esquerda, você verá uma caixa com o título de “Aplicar em:” (“Current Configuration Scope” em inglês) onde você pode selecionar a loja desejada para aplicar suas alterações de configuração.



Figura 13 – Setando o escopo no painel admin do Magento.

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779>



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

Para gerenciar seu design no nível de site, selecione o nome do seu site a partir da lista suspensa, em seguida, siga os seguintes passos. O que queremos dizer por nível de site? Lembra-se exemplo da empresa Dubloo?



Figura 14 – Exemplo Dubloo onde todas as lojas compartilham o mesmo tema.

Neste caso, Dubloo quer personalizar cada uma das visualizações de loja (estão todas em laranja), de modo que precisamos apontar seu tema por site.

Para gerenciar o tema no nível de visualização de loja, selecione o nome da visualização de loja no menu dropdown, em seguida aplicando os seguintes passos, vejamos como ficou a Dubloo.

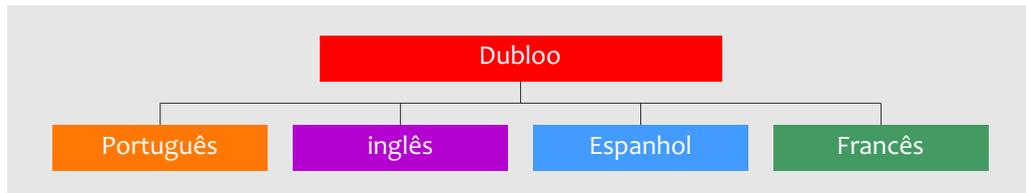


Figura 15 – Exemplo Dubloo onde cada visualização de loja possui um tema diferente.

Vamos dizer que Dubloo queira que seus usuários vejam um design diferente dependendo da visão de loja estão usando, para conseguir isso eles precisam gerenciar o design no nível de visualização de loja.

Passo 1:

Na guia Visual, em “Nome do Template Atual”, coloque o nome da pasta em que seu novo pacote de tema reside. Magento automaticamente usa o pacote default caso este campo esteja em branco.

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779>



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



Figura 16 – definindo o pacote de tema no painel administrativo do Magento.

Passo 2:

No campo default (no painel do Themes), entre com o nome do novo tema que gostaria de aplicar em seu site, loja ou visualização de loja. Caso deixe este campo em branco, o Magento aplicará o tema default do pacote de tema definido no passo 1. Lembre-se: de acordo com o esquema de fallback do Magento, se você especificar um nome de tema, o Magento irá olhar primeiro os arquivos deste tema, caso não encontre ele pesquisará na pasta default do seu pacote de tema e finalmente só então irá procurar em “base/default”. Observe que você também pode substituir o tema a ser aplicado para tipos de arquivos específicos, como Layouts, templates, translations e skins, se você especificar um nome diferente em cada um desses campos, o Magento irá procurar os arquivos nestas pastas e depois executa o fallback para o default do seu pacote de temas.

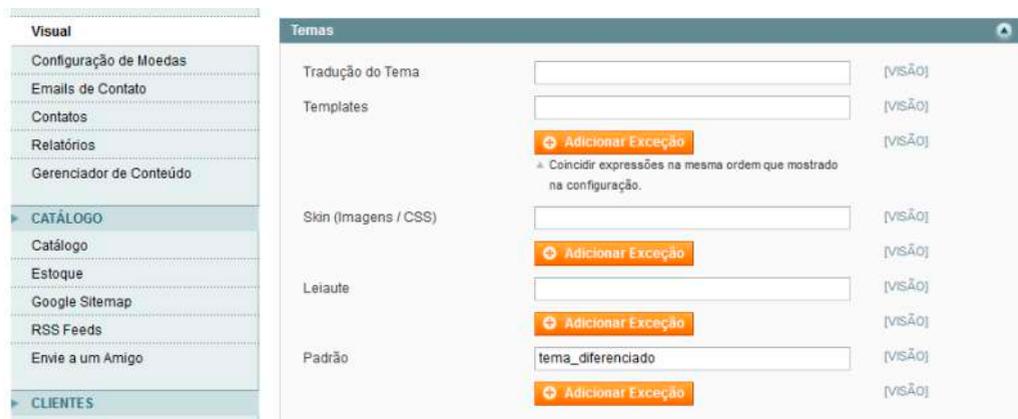


Figura 17 – definindo o tema no painel administrativo do Magento.

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779> 

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

Passo 3

Quando estiver pronto clique em “salvar configurações” (“Save config”) e atualize sua loja, Voila! Agora você vê seu novo tema aplicado no frontend.

Agora que vimos como criamos e configuramos um tema no Magento, veremos como o Magento lida com os temas.

Passo a passo 2 – Aplicando múltiplos temas.

O Magento permite a flexibilidade de usar múltiplos temas mesmo dentro do mesmo layout e modelo. Vamos dizer que você quer ter a mesma estrutura e modelo de layout para todas as visualizações de loja, mas que varie as imagens e a paleta de cores. No Magento isso é realmente simples.

Lembra como as pastas de layout e template estão localizadas em `app/design/frontend/[pacote]/[tema]` e nosso atual visual é localizado em `skin/frontend/[pacote]/[tema]`? Os dois não precisam ter o mesmo nome!

Si você olhar para a instalação padrão do Magento, você vai ver a pasta do tema default em `app/code/design/frontend/default/default`. Correspondente a isso são varias pastas de skin. Dê uma olhada em `skin/frontend/default` e você verá “blank,” “blue,” “default,” “French” e “german.” Todas as cinco são diferentes skins que podem ser usadas com o tema default. Os temas “blank,” “iPhone” e “modern” são listados também, mas interagem separadamente a partir do tema default. Existem mais temas definidos no diretório skin do que no diretório app/design, isso porque os temas “blue,” “French” e “german” diferem do tema “default” apenas nos arquivos “CSS”.

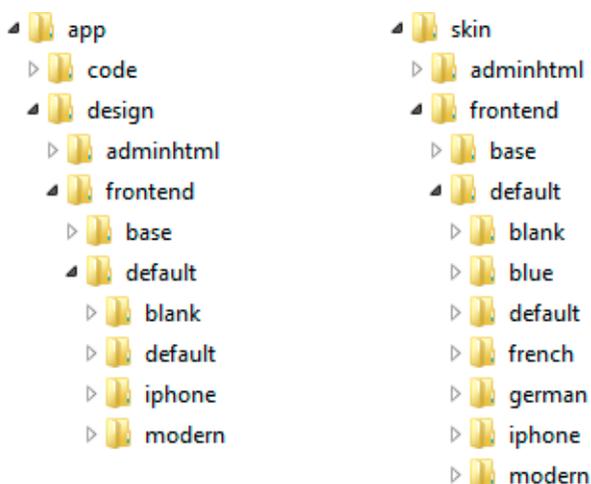


Figura 18 – ScreenShot dos diretórios (temas) em pacote default.

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779> 

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

Voltemos ao terceiro cenário da nossa sessão sobre terminologias. Lembra da loja Home Laptops?

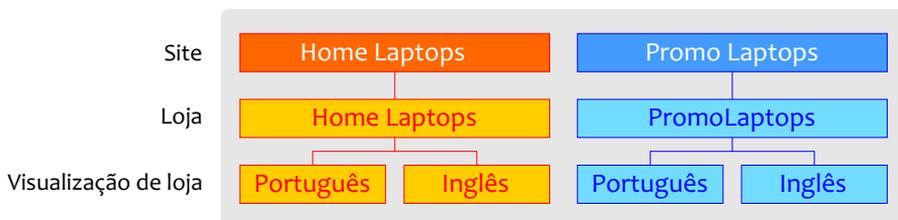


Figura 19 – Home Laptops com múltiplos sites, lojas e visualizações de loja.

Neste caso você poderia escolher “default” para seu layout e templates. Vamos supor que você queira deixar a visualização em inglês usando a skin default. Como definimos esta como o padrão para a configuração geral, não será necessário fazer qualquer alteração para esta visualização.

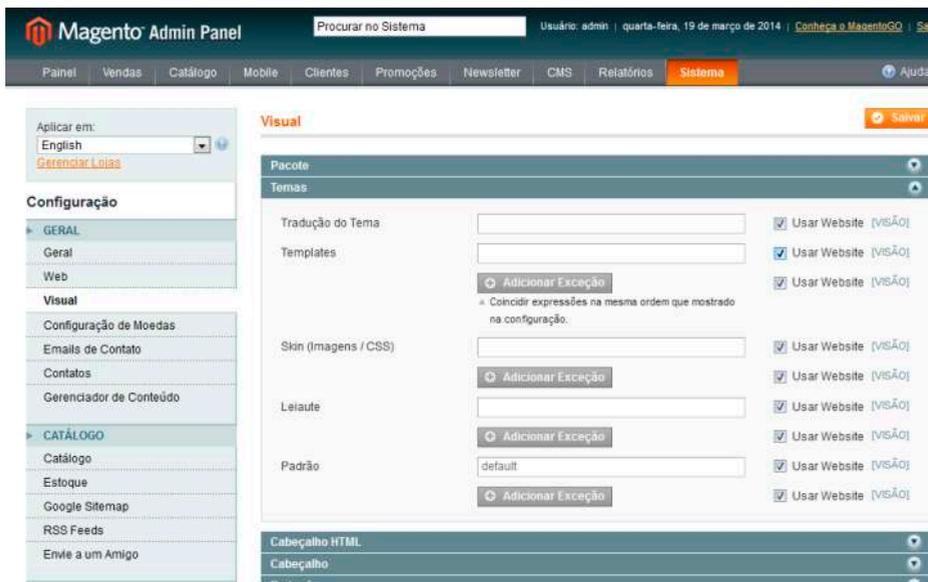


Figura 20 – Configuração a visão de inglês para o tema default/default.

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779>



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



Figura 21 – frontend da loja em inglês com tema default/default.

Agora, digamos que você queira que seus usuários que português vejam uma variação, além da linguagem, quando eles mudarem a visão da loja. Faça isso marcando a variação de tema “blue” para sua visualização de loja em português.

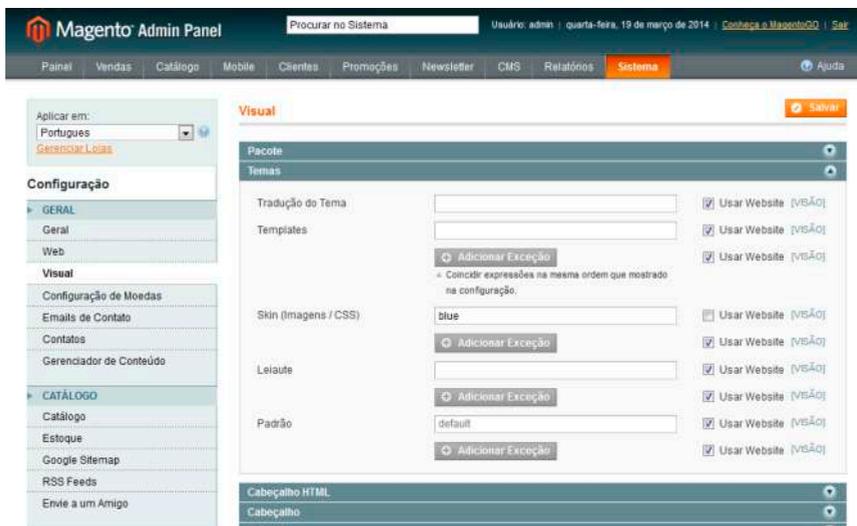


Figura 22 – Configurando a visão em português para tema default/blue.

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779> 

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

Deixamos o padrão como “default” assim qualquer lugar que não tenha alteração em nossa variação de tema “blue” dará um fallback para o “default”. Isso permitirá que você tenha um segundo conjunto de imagens e paleta de cores baseado no seu público alvo. Veja o que acontece quando mudamos a visão da loja para o português:



Figura 23 – Configurando a visão em espanhol para tema default/blue .

Vejamos outro exemplo: Datas comemorativas, de longe, oferecem a melhor oportunidade de vendas para qualquer loja de e-commerce. Clientes fazem fila para comprar presentes para sua família e amigos. E mais filas para comprar fantasias para carnaval e dia das bruxas, por exemplo. A fim de se adaptar para estes clientes sazonais, sua loja deve refletir fielmente estas ocasiões, de forma a inspirar seus clientes a navegar mais por sua loja. Uma loja com o visual do dia a dia mostrada abaixo apenas faz uma mudança durante o Natal. O que precisa apenas de alguns vermelhos, flocos de neve e de um Papai Noel – Assim como é mostrado abaixo no visual de Natal.

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779>



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



Figura 24 – Visuais sazonais são baseados em algum tema mas com algumas variações.

O Magento oferece a capacidade de lidar com vários temas de sua escolha exatamente para aqueles momentos em que um toque extra é necessário. Neste caso, você não precisa criar uma nova visão de loja, como fizemos com o Inglês e Português. Tudo que você precisa fazer é enviar uma nova skin de Natal para a sua pasta de skins e, em seguida, fazer referência a ela em sua configuração onde foi atribuído "default" e "português". Ao fazer isso, você ainda está preservando o seu projeto e estrutura da loja não sazonal normal, reforçando-a com o tema natalino. Quando a época de Natal termina, simplesmente mude o nome da skin de volta para sua escolha típica e voila! Sua loja voltou ao normal. Você perceberá a semelhanças de conceito nos dois projetos acima. O visual passou para Natal no diagrama 4, mas por traz disso é possível ver a estrutura do design da loja normal. As únicas diferenças reais são em alguns CSS, imagens e alterações de palavras nos arquivos de template. Porque as mudanças são de fato mínimas, você não precisa de um tema padrão para alocar seu tema de natal. O que você precisa está em poucas alterações de arquivos e você chegará a uma loja muito mais alegre. A função do multi-temas do Magento é exatamente esta, dando o poder de ligar e desligar os temas sazonais sem alterar seu tema padrão.

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779> 

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

Hierarquia dos temas

Quando você atribuir vários temas para sua loja, você estará mais próximo de entender que, embora o seu principal objetivo na construção de um tema seja criar a interface gráfica mais útil e visualmente agradável, a aplicação é capaz de localizar e carregar todos os arquivos do tema necessário para manter o sistema em execução livre de erros. Por exemplo, se sua categoria listing Page chama o template “view.phtml” (neste caso o template se torna um arquivo requerido), mas o sistema não consegue achar o arquivo do tema no ponto mais alto da hierarquia. Se isso acontece o sistema vai continuar descendo na hierarquia até ser capaz de encontrar o arquivo “view.phtml”, então vai carregá-lo e encerrar a pesquisa. Este método de trabalho chamamos de fallback porque o sistema vai retrocedendo para o próximo local possível onde o arquivo possa estar e carregá-lo. Veja o modelo fallback do Magento para entendê-lo.

A hierarquia do fallback do Magento funciona assim:

- 1 - Procura o arquivo em:
 - app/design/frontend/pacote/tema_personalizado
 - skin/frontend/pacote/tema_personalizado
 - Procura por substituições específicas no arquivo de layout local.xml
- 2 - Se não encontra, procura em:
 - app/design/frontend/pacote/default
 - skin/frontend/pacote/default
- 3 - Se não encontra, procura em:
 - app/design/frontend/base/default
 - skin/frontend/base/default
- 4 - Se não encontra, ocorrerá um erro de renderização

Por exemplo, digamos que você tenha três temas instalados dentro da sua loja e cada um desses temas tem os seguintes arquivos:

Tabela 1

base	default	tema personalizado
Todos os arquivos	templates/3-col-layout.phtml	templates/3-col-layout.phtml
	templates/header.phtml	css/base.css
	images/logo.gif	
	css/base.css	
	css/boxes.css	

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779> 

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

Vamos supor também que esses três temas são dispostos nesta hierarquia:

Tabela 2

Mais alto	Tema personalizado
	default
Mais baixo	base

Você vai ver que existem alguns arquivos que são redundantes como o `templates/3-col-layout.phtml` e `css/base.css` na tabela 1. Agora vamos arrumar a tabela de forma que os arquivos redundantes fiquem na mesma linha.

Tabela 3

base	default	tema personalizado
Todos os arquivos		
	<code>templates/3-col-layout.phtml</code>	<code>templates/3-col-layout.phtml</code>
	<code>templates/header.phtml</code>	
	<code>images/logo.gif</code>	
	<code>css/base.css</code>	<code>css/base.css</code>
	<code>css/boxes.css</code>	

Você Talvez pergunte “Ok, legal. Mas o que isso quer dizer?”

Bem, a tabela 3 é como você vê os arquivos em cada pasta do tema e não como o Magento vê.

Vejamos como o Magento vê a mesma estrutura de arquivos na tabela 4

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779>



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

Tabela 4

base	default	tema personalizado
Todos os arquivos		
		templates/3-col-layout.phtml
	templates/header.phtml	
	images/logo.gif	
		css/base.css
	css/boxes.css	

Você pode perceber como o Magento elimina a redundância, ignorando os arquivos no nível mais baixo e reconhecendo apenas os arquivos mais altos da hierarquia. Isso acontece porque o arquivo já foi encontrado e não pesquisou mais, portanto, encerrou a pesquisa desses arquivos encontrados e continuou apenas dos arquivos faltantes.

Quando usado corretamente, o mecanismo de fallback permite a você editar e manter somente os arquivos que precisa alterar para um tema personalizado e todos os outros arquivos serão fornecidos pelo tema default do pacote personalizado ou pelo pacote base. Por exemplo, O Magento mantém quatro temas de demonstração – default, black, iPone e Modern. Antes da introdução do pacote base, todos os arquivos de tema tinham que ser copiados para cada tema, testados e debugados um a um. O código no pacote base é mais eficiente.

Por exemplo, o tema default/default e default/blank são implementados apenas com alterações de CSS herdadas todos os arquivos de template e de layout do pacote base. Para finalizar, esse método torna seu tema mais “upgrade proof” ou mais compatível com atualizações do sistema e permite a você manter muito menos código. Muitos temas apenas personalizam um subconjunto de funcionalidades do tema default do Magento. Portanto, toda a funcionalidade do core do Magento vem do tema base e podem ser substituídos seletivamente por um pacote/tema personalizado.

Importantes regras de hierarquia a se lembrar

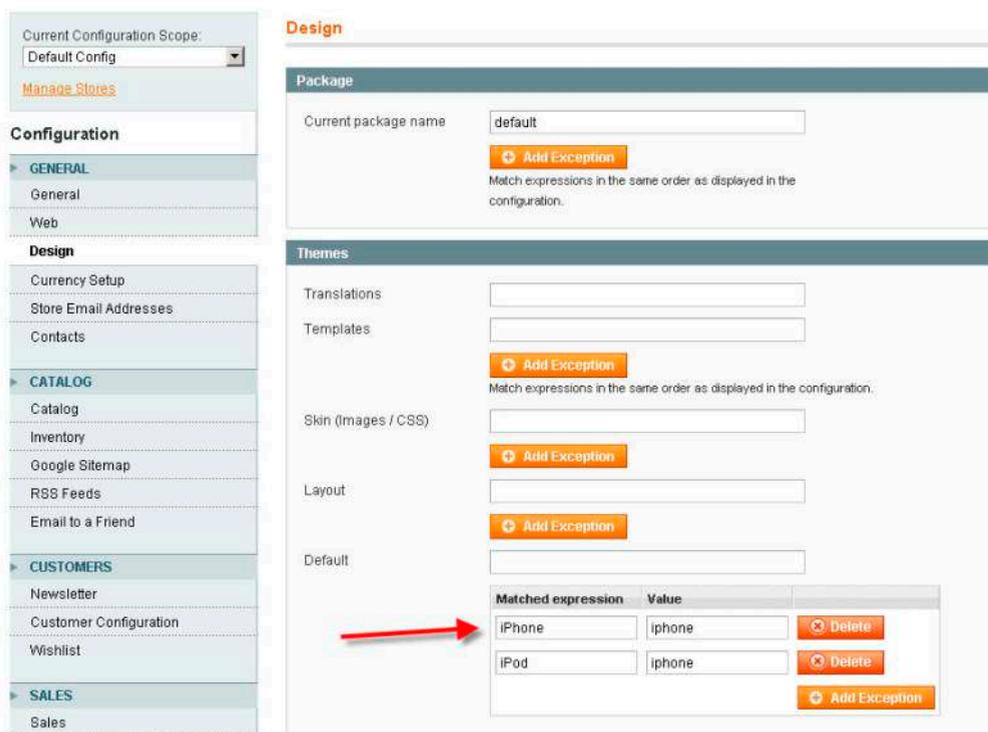
- Crie seu tema customizado dentro de seu próprio pacote. Faça diretórios como app/design/frontend/seu_pacote/default e skin/frontend/seu_pacote/default e construa seu tema personalizado lá.
- Nunca copie todos os arquivos de base/default no seu pacote personalizado. Copie apenas os arquivos que você precisa modificar. Isso ajudará você a saber o que procurar, quando os arquivos personalizados necessitarem de atualização sem ter que procurar no meio de arquivos não modificados desnecessariamente.

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779> 

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

Exceções de design

Exceções de design permitem aos gerenciadores escolher um tema alternativo para um sistema de usuário específico ao invés de criar uma visão de loja separada para um agente que visualiza a loja (ex.: iPhone), exceções de design permitem ao administrador sobrepor configurações de uma visão de loja, limitando a quantidade de visões de loja que precisará administrar.



The screenshot shows the Magento administration interface for Design configuration. On the left is a navigation menu with sections: Configuration (General, Web), Design (Currency Setup, Store Email Addresses, Contacts), CATALOG (Catalog, Inventory, Google Sitemap, RSS Feeds, Email to a Friend), CUSTOMERS (Newsletter, Customer Configuration, Wishlist), and SALES (Sales). The main content area is titled 'Design' and includes a 'Package' section with 'Current package name' set to 'default' and an 'Add Exception' button. Below is the 'Themes' section with input fields for Translations, Templates, Skin (Images / CSS), Layout, and Default, each with an 'Add Exception' button. A table at the bottom lists 'Matched expression' and 'Value' for 'iPhone' and 'iPod', both set to 'iphone', with 'Delete' buttons for each. A red arrow points to the 'Matched expression' column.

Matched expression	Value	
iPhone	iphone	Delete
iPod	iphone	Delete

Figura 25 – Aplicando exceção de design permitindo que o Magento use um tema diferente para diferentes clientes / browsers.

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779> 

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

5 Customizando temas Magento

O Magento foi construído seguindo um modelo totalmente modular, que oferece flexibilidade quase ilimitada. Naturalmente isto é aproveitado para programação e desenvolvimento de temas para sua loja. Neste capítulo Entenderemos o que isso significa e veremos exatamente como desenvolver um tema para sua loja com Magento.

Este capítulo ajudará a entender como modificar um tema existente. A partir disso você ganhará confiança e conhecimento para desenvolver você mesmo seu próprio tema. Lembre-se que construiremos sobre o pacote base ao invés de totalmente do zero. Nosso objetivo final é customizar a loja visualmente e com funcionalidade que deseja, mas que seu tema tenha o mínimo de arquivos para editar e deixe que o tema base cuide de todo o resto.

Criando os diretórios básicos para seu pacote/tema

Com a inclusão do pacote base, criar um novo pacote e o tema default nunca foi tão fácil. Você pode criar um novo tema incluindo apenas os arquivos que precisam ser diferente do pacote base. Uma regra que nunca pode deixar de seguir é sempre manter a convenção dos subdiretórios do Magento. Um bom ponto para começar a criação de um pacote personalizado e um tema é criar um “esqueleto” que reflita todo o template principal e diretórios de temas usado pelo Magento, então salvar os arquivos em seus locais corretos. Quando estiver pronto, se algum desses diretórios permanecer vazios e sejam desnecessários, você pode deletá-los.

Criando um novo pacote e tema(s)

Por favor, ignore as instruções legais do Magento e tutoriais que ensinam a criar seu tema personalizado dentro do pacote default, ou editar arquivos diretamente no diretório default/default. Ao invés disso, o método que proporciona melhor resultado para atualizações e mais proteção de alterações acidentas é criar um novo pacote e criar seu tema personalizado dentro dele.

Dentro de app/design e skin, crie um novo diretório de pacote e os diretórios de templates e de skins requeridos pelo Magento como mostrado abaixo.

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779> 

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

Estrutura de diretórios para temas personalizados no pacote sublime

app/design/frontend/sublime/default/

<MAGENTO_BASE_DIR>

- + app
 - + design
 - + frontend
 - + base
 - + default
 - + seu_pacote
 - + default
 - + layout
 - + template

skin/frontend/dubloo/default/

<MAGENTO_BASE_DIR>

- + skin
 - + frontend
 - + base
 - + default
 - + seu_pacote
 - + default
 - + css
 - + images
 - + js

Isto é tudo que precisa fazer para começar, o Magento irá procurar os arquivos no tema padrão para seu pacote personalizado e em seguida retorna ao pacote base para todos os arquivos que não encontrou. Como o diretório base não tem o CSS completo, é uma boa idéia copiar os arquivos de skin de um dos temas do pacote default como um ponto de partida. Os CSS e imagens do tema default/blanck são mais clean.

Aplicando seu novo pacote e tema para seu site

Agora que criamos o esqueleto de diretórios para acomodar seu novo pacote e tema, precisamos dizer ao Magento que use este pacote e assim acompanhar as alterações em tempo real.

Desabilite o sistema de cache

Prepare o Magento para produção do tema, precisaremos desabilitar o cache indo no painel administrativo ([HTTP://seu-dominio.com.br/admin](http://seu-dominio.com.br/admin) ou localhost/magento/admin) e navegando em Sistema > Gerenciar Cache (system > cache management). Selecione as check boxes referentes a **Layouts**, **Blocks HTML output** e **Translations** e dentro de **Ações (Actions)** selecione **Desabilitar (Disable)** e clique em **Enviar (Submit)**. Em cada um desses itens, na área Status, aparecerá como “DESATIVADO” (“DISABLED”) com fundo vermelho. Isso garante que as alterações sejam aplicadas na loja conforme você altera seu tema.

Nota: Dependendo do que você está alterando, pode ser necessário desativar itens adicionais nesta área ou até mesmo deletar os arquivos de cache em sua estrutura de diretórios. Não exclua as pastas manualmente a menos que saiba o que está fazendo.

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779> 

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

Aplique seu pacote/tema personalizado em sua loja

Para ver seu novo tema enquanto o desenvolve, tenha certeza de aplicar para loja em desenvolvimento usando o painel administrativo. Vá em ADMIN: sistema > Configuração na aba Visual (Design). No painel Pacote coloque o nome do seu pacote de temas <meu_pacote> em Nome do Template Atual. No painel Temas, ponha como “default” para usar seu tema padrão ou o nome do seu tema personalizado conforme for.

Volte ao frontend e atualize o browser para recarregar a página. Ela deve conter todas as funcionalidades padrão validada pelo pacote base e verá algum estilo CSS caso tenha copiado algum para dentro de seu tema. Caso não tenha movido nenhum arquivo de estilo e imagens no seu tema, o site mostrará apenas texto, sem nenhuma formatação.

Customizando os arquivos skin

Manipular imagens e arquivos CSS vai personalizar a loja Magento de forma eficiente. Se buscar nos exemplos passados, o tema blue do pacote default usa apenas CSS. Não use CSS para mostrar ou esconder blocos ou conteúdos do Magento, use-o para trabalhar todo o estilo do seu site.

Como o Magento tem muitas funcionalidades escritas envolto de divs pré-definidas, até que você tenha mais experiência em criação de temas Magento, será melhor começarmos com folhas de estilo já existentes e modificá-los para se adequar à sua identidade e visual. Quando tiver mais experiência com Magento, você poderá escrever do zero ou a partir do framework de sua preferência. Se você vai iniciar com CSS e imagens já existentes, tenha certeza de instalar uma ferramenta como o plugin firebug do Firefox, que vai ajudar você para achar qual regra CSS está controlando um elemento em uma página.

Exercícios rápidos para começar

Exercício 1: Modifique o CSS

1. Use o firebug para explorar o css usado na página.
2. Faça alterações de cor, padding, etc. no arquivo CSS principal do seu tema. Lembre-se de sempre fazer as alterações dentro do seu pacote e tema personalizado. Se o arquivo CSS principal é o “style.css”, você o encontrará neste local: skin/frontend/<seu_pacote>/<seu_tema>/css/style.css
3. Volte ao frontend do seu site e recarregue a página para ver suas alterações.

Exercício 2: Alterar a LOGO

Certamente uma das primeiras coisas que todos querem fazer quando customizam uma loja Magento é alterar a logo. A logo da loja está associada ao tema que vai ser usada para esta loja e fica armazenada em skin/diretório de imagens desta loja. Por convenção, o Magento usa três arquivos de logo chamados:

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779> 

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

- **logo_gif** – usado no cabeçalho (header) do próprio site
- **logo_email.gif** – usado em emails de saída do site
- **logo_print.gif** – usado quando uma página do site é impressa.

Se olhar para o diretório de imagens da pasta skin de um tema existente, como o default/blank, no diretório skin/frontend/default/blank/images, você verá estes três arquivos.

Você pode alterar a logo do site usando seu tema simplesmente colocando seu próprio logo em seu tema personalizado da pasta skin usando o mesmo nome, se desejar.

- 1 - Coloque seu logo com o mesmo nome no diretório
skin/frontend/<seu_pacote>/<seu_tema>/images
Você pode trocar o nome e o tipo de arquivo do logo alterando o nome do arquivo no painel administrativo.
- 2 - Vá ao ADMIN e no menu Sistema > Configurações (System > Configuration) na aba visual (Design), no painel header, você vai setar a pasta e o nome do arquivo principal da logo. Isto altera apenas o arquivo usado no header do site, não nas versões_email e_print.
- 3 - Insira seu logo principal com novo nome na pasta
skin/frontend/<seu_pacote>/<seu_tema>/images

Customizando com arquivos de layout

Layouts são uma das características mais poderosas e mais originais do Magento para criação de temas. Mesmo que esteja familiarizado com outros sistemas de templates, trabalhar com layouts provavelmente será uma novidade para você. Se você dominar seu uso, você poderá rapidamente ligar, desligar, mover quase todos os componentes e funções do Magento.

Introdução ao layout

Layout é a ferramenta com que você pode atribuir informação para cada bloco que criou. Os arquivos de layout do Magento estão no formato XML e modificando o Layout você pode mover os blocos de uma página e atribuir templates para blocos de conteúdo produzindo marcações de blocos de estrutura. Na verdade com alguns arquivos de layout apenas você é capaz de modificar o visual de cada página de sua loja.

Como por exemplo, veja esta screenShot da categoria landing Page no Magento:

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779>



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



Figura 26 – Exemplo da página de categoria da loja de exemplo do Magento.

Agora veja a quebra do exemplo acima em blocos estruturais do Magento e em blocos de conteúdo atribuídos a cada um desses blocos.

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779> 

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



Figura 27 – Representação do uso dos blocos estruturais e blocos de conteúdo do Magento.

Os quadrados na imagem a esquerda acima são os **blocos estruturais**. Eles são os pais dos blocos de conteúdo, servem para posicionar seus blocos de conteúdo dentro do contexto da página (como na imagem da direita acima). Tais blocos estruturais existem na forma de Header, coluna left, coluna right, etc. que criam a estrutura visual de uma página de loja. Neste capítulo, trabalharemos com a estrutura padrão de blocos do Magento. Blocos estruturais são manipulados pela página de módulos do Magento e são definidos no arquivo `app/design/frontend/base/default/layout/page.xml` e nas páginas de template padrão em `app/design/frontend/base/default/template/page/`.

Um **bloco de conteúdo** é cada bloco colorido que compõem um bloco estrutural. Em síntese, o verdadeiro conteúdo da loja. São representações de cada funcionalidade em uma página (como uma lista de categorias, tags do produto, etc.), implementam os arquivos de template para gerar o (x)HTML a ser inserido no bloco estrutural pai. Blocos de conteúdo combinam a maior parte dos arquivos de template da pasta

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779> 

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

app/design/frontend/base/default/template/.

Com Magento, você não terá mais um arquivo de template chamado “left_colun.ext” com intermináveis marcações que devam ser gerenciadas manualmente, dependendo de cada funcionalidade necessária para cada página de loja. Em vez disso, suas funções são gerenciadas em um modelo por funcionalidade que permite carregar e descarregar recursos da sua página de loja com apenas alguns comandos de layout.

Como o layout do Magento funciona

Layout é um componente virtual da aplicação do Magento. Modificando os componentes do layout, você pode trabalhar as páginas de loja com melhor compatibilidade com atualizações do sistema. Os arquivos de layout podem parecer confusos no começo, mas na realidade, são construídos usando um pequeno conjunto de tags XML que são fáceis de aprender. Ao aprender alguns conceitos de chaves e comandos do layout, em pouco tempo, você terá mais confiança e conhecimento para facilmente modificar o design da loja com suas especificações desejadas.

Os arquivos base de layout ficam no diretório app/design/frontend/base/default/layout.

Arquivos de layout podem ser copiados para o diretório do seu tema

app/design/frontend/<seu_pacote>/<seu_tema>/layout.

Um layout é composto por layout padrão e aytout de atualização que são feitas de tags XML simples. Com estes comandos de layout, você pode atribuir / modificar blocos estruturais, blocos de conteúdo e qualquer controle de funções da loja como carregar ou remover um específico bloco javaScript para uma página.

Cada módulo do Magento tem seu arquivo de layout próprio, por exemplo o “catalog.xml” é o layout do módulo de catálogo e “customer.xml” é o módulo de clientes. Cada arquivo é ainda dividido em controladores. Os controladores que são variáveis que estão pré-definidas pelo core do Magento e em todas as extensões ativas na sua loja. Para a maior parte, cada controlador corresponde a um tipo de página em sua loja Magento. Como o <catalog_category_default> e <catalog_product_view>, mas existem alguns controladores que se aplicam a todas as páginas, tais como o <default> e também controladores aplicados em páginas com status específicos, como o <customer_logged_in> e <customer_logged_out>.

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779>



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

```

<default> → Handle
<!-- Mage_Catalog -->
<reference name="top.menu">
    <block type="catalog/navigation"
</reference>
<reference name="left">
    <block type="core/template" name
        <action method="setImgSrc"><
        <action method="setImgAlt"><
        <action method="setLinkUrl"><
        <action method="setLinkUrl">
    </block>
</reference>
<reference name="right">
    <block type="core/template" name
    <block type="core/template" name
</reference>
</default>

<!--
Category.default.layout
-->

<catalog_category_default> → Handle
<reference name="left">
    <block type="catalog/navigation"
</reference>
<reference name="content">
    <block type="catalog/category_vi
        <block type="catalog/product
    </block>
</reference>
</catalog_category_default>
    
```

Figura 28 – Exemplo de código de arquivo de layout do Magento mostrando o uso dos controladores

Um controlador pode aparecer em vários arquivos de layout e os blocos podem ser atribuídos a ele ou sobrescritos em cada arquivo de layout. Por exemplo, vários arquivos de layout podem ter o controlador <default>. Analisando os arquivos de layout, o Magento primeiro pega as atualizações de layout dentro de <default> em cada um dos arquivos de layout, lendo na ordem que foram atribuídas em app/etc/modules/Mage_All.xml. Em seguida, analisa os controladores específicos, novamente olhando todos os arquivos de layout e finaliza a construção da página da loja.

O sistema de renderização do Magento é construído dessa forma para permitir adição perfeita e remoção de módulos sem afetar outros módulos do sistema. Isto significa que você pode adicionar, remover e mover muitas das funções do Magento, basta adicionar, remover ou mover as declarações de blocos nos arquivos de layout.

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779>



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

A anatomia de um arquivo de layout

Arquivos de layout contêm um pequeno conjunto de tags XML que atuam como instruções detalhadas para aplicação em como trabalhar uma página, o que fazer e o comportamento de cada construção de bloco. Aqui estão algumas propriedades comportamentais de cada tag XML do layout.

Controladores

Um controlador (diagrama 1) é um identificador pelo qual a aplicação determina o que fazer com as atualizações aninhadas por ele.

Se o nome do controlador é <default>, a aplicação sabe que suas atualizações deverão ser carregadas em quase todas as páginas da loja antes de carregar o layout da página específica ("quase todas", porque algumas páginas excepcionais, como a imagem de pop-up do produto não carregam o <default>).

Se o magento encontra um controlador diferente do <default>, ele atribuirá as atualizações listadas no controlador de acordo com a página específica desse controlador. Por exemplo, <catalog_product_view> contém as atualizações de layout da página de visualização do produto. Controladores são pedaços de algo maior que um designer, sem um conhecimento profundo da programação Magento, nunca deve precisar modificar.

<block>

Magento determina o comportamento e a representação visual de cada bloco trabalhado pra uma página através da tag <block>. Nós já falamos sobre os dois tipos de bloco que o Magento aplica, bloco de estrutura e bloco de conteúdo. A melhor maneira de distinguir os dois é olhando o comportamento atribuído a cada um pelas tags atribuídas. Um bloco estrutural normalmente tem o atributo "as" através do qual o aplicativo é capaz de comunicar-se com a área designada (usando o método getChildHtml) em um template. Você perceberá várias ocorrências deste atributo "as" no layout default, porque naturalmente o layout default é um dos que constroem as bases sobre as quais os layouts específicos podem começar para adicionar. Por exemplo, no layout default temos blocos estruturais como "left," "right," "content" e "footer" sendo introduzidos. Embora estes blocos possam existir em atualizações de layout normais, é melhor setar os blocos estruturais recorrentes no layout padrão primeiro, e então, adicionar os blocos de conteúdo por página. Os atributos válidos para o <block> são:

- **type** – este é o identificador de módulo de classe que define o funcionamento do bloco. Este atributo não pode ser modificado.
- **name** – é o nome pelo qual outros blocos podem fazer referência para o bloco em que este atributo foi dado (diagrama 3)
- **before** e **after** – são dois métodos de posicionar um bloco de conteúdo dentro de um

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779> 

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

bloco estrutural. `before=""` e `after=""` são comandos usados para posicionar o bloco em primeiro ou em último lugar dentro de um bloco estrutural

- **template** – determina o template que representa a funcionalidade do bloco em que este atributo é dado. Por exemplo, se neste atributo é escrito `'catalog/category/view.phtml'`, a aplicação vai carregar o arquivo de template `'app/design/frontend/template/catalog/category/view.phtml'`
- **action** - `<action>` é usado para controlar funcionalidades da frente da loja, como carregar ou remover um JavaScript. Algum dia teremos uma lista completa de todos os métodos do action, mas no momento a melhor maneira de estudar sobre os diferentes métodos do action é brincar com eles nas atualizações de layout disponíveis
- **as** – este é o nome que um template chama um bloco em que este atributo é designado. Quando você vê um método PHP chamada `getChildHtml('block_name')`, ele está fazendo referência ao bloco em que o atributo `“as”` foi definido como `“block_name”` (ex.: `<?=$this->getChildHtml('header')?>` em um arquivo de template faz referência ao `<block as=“header”>`).

<reference>

`<reference>` é usado para fazer referência a outro `<block>`. Fazendo uma referência para outro bloco, as atualizações dentro de `<reference>` serão aplicadas no `<block>` para que se correlacione (diagrama 3).

Para fazer uma referência, você precisa marcar a referência para um bloco pelo atributo `“name”`. Este atributo chama o `<block>` marcado pelo atributo `“name”`. Então, quando você faz uma referência como `<reference name=“right”>`, você está marcando o bloco chamado `<block name=“right”>`.

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779>



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

```

catalog.xml
    </block>
</reference>
<reference name="right">
    ...<block type="catalog/product_list_related" name="
</reference>

</catalog_product_view>

page.xml

<block type="core/text_list" name="right" as="right"
<block type="page/html_footer" name="footer" as="f
<block type="core/store_switcher" name="store_
<block type="cms/block" name="footer_links">
    <action method="setBlockId"><block_id>foot
</block>
    
```

Figura 29 – Exemplo de código de arquivo de layout mostrando o uso do <reference>.

Regras do XML

A única regra que é preciso lembrar em relação a XML é que, quando uma tag se abre, ou deve ser seguido por uma tag de fechamento (<xml_tag> </xml_tag>) ou deve fechar em si mesma (<xml_tag />) exatamente como nas tags de xHTML.

```

XML
</reference>
<reference name="content">
    <block type="catalog/product_view" name="product.int
    <block type="catalog/product_view_super_config"
    template="catalog/product/view/super/config.phtml
    <block type="catalog/product_view_super_group"
    template="catalog/product/view/super/group.phtml
    <block type="catalog/product_list_upsell" name="
    template="catalog/product/list/upsell.phtml />
    <block type="catalog/product_view_additional" name="
</block>
</reference>
<reference name="right">
    <block type="catalog/product_list_related" name="cat
    template="catalog/product/list/related.phtml" />
</reference>

(X)HTML
<div class="catalog-list">
    <ul class="pager">
        <li><a href="#">1</a></li>
        <li><a href="#" class="active">2</a></li>
    </ul>
    <p></p>
</div>
    
```

Figura 30 – Exemplo de código de arquivo de layout sobre a sintaxe XML.

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779>



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

Como encontrar o arquivo de layout para modificar

Para acessar os arquivos de layout, vá para `app/design/frontend/pacote/tema/layout/`. Assim como os templates, os layouts são salvos com base no módulo, então, você pode facilmente localizar o arquivo de layout para modificar com a ajuda da dica de template. Para habilitar a dica de template, vá em Sistema > configurações > Avançado > Desenvolvedor. Neste ponto você deve se certificar que selecionou uma loja no dropdown que fica no canto superior esquerdo “Aplicar em” (Current Configuration Scope) ao invés de “todas as lojas”. Uma vez que seu site foi selecionado, clique em debugar. Nele você pode mudar “Exibir Caminho das Telas” para **sim**, se quiser mais informações, pode marcar o “Exibir Nome dos Blocos” para **sim**.

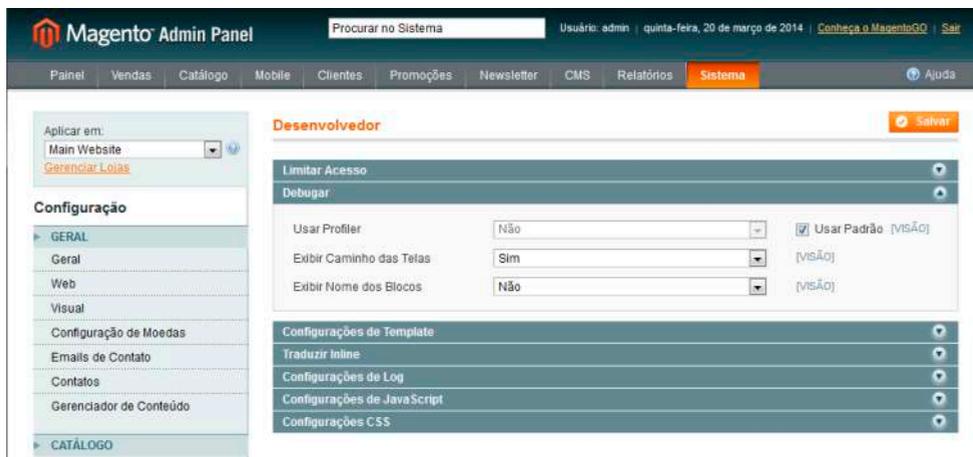


Figura 31, habilitando a dica para o caminho do template

Agora que tem a dica de template habilitado, atualize a a página que deseja modificar e veja o caminho do arquivo do template no qual a dica mostra, se você quer, por exemplo, mover o mini carrinho, referencie o caminho do template (ex: `app/design/frontend/base/default/template/checkout/cart/sidebar.phtml`) e use a primeira pasta do caminho dentro da pasta template (indicado em **negrito**, que também é o nome do módulo) para encontrar o arquivo de layout correspondente. Neste caso, portanto, você sabe que deve procurar em “**checkout.xml**” para modificar a posição do mini carrinho. Cada arquivo de layout é ainda dividido em controladores, que definem um layout de página específico. Cada área de layout por página é claramente marcado com comentários que indicam seu uso. Mas o próprio sistema reconhece essa divisão através dos controladores de cada arquivo de layout.

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779>



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

Exercícios rápidos para começar

Aqui temos dois exercícios para você sentir como é trabalhar com layout. Nestes exercícios você irá copiar os arquivos de layout do tema base e modifica-los dentro do seu tema personalizado. Exploraremos, depois, outros métodos que não exigem que copie o arquivo inteiro.

Exercício 1

Na página de categorias, mova o “Meu carrinho de compras” da coluna da direita para coluna da esquerda

1. Ligue a dica de template indo no admin e navegando para Sistema > Configuração. Quando a página carregar, selecione a loja que está trabalhando usando o menu dropdown no topo esquerdo da página. Espere a página recarregar e selecione a aba Desenvolvedor. Selecione “sim” para “Exibir caminho das telas”. Clique em salvar. Volte ao frontend da loja e recarregue a página.
2. Quando a página recarregar, veja o caminho do template do bloco “meu carrinho de compras” que vai mostrar algo como “frontend/base/default/template/checkout/cart/sidebar.phtml.” Olhando para a pasta, sabemos que o template é inicialmente introduzido pelo modulo “checkout”. Como sei disso? R= frontend/base/default/template/checkout/cart/sidebar.phtml. O caminho do template é relacionado ao seu módulo. O diretório imediatamente abaixo da pasta “template” é o nome do módulo pelo qual um template é introduzido.
3. Como agora sabemos que estamos lidando com o módulo “checkout”, copie o arquivo checkout.xml do pacote base para seu pacote personalizado. Copie app/design/frontend/base/default/layout/checkout.xml para app/design/frontend/<seu_pacote>/default/layout/checkout.xml
4. Abra o arquivo copiado app/design/frontend/<seu_pacote>/default/layout/checkout.xml, procure por “checkout/cart/sidebar.phtml” (nome do template “meu carrinho de compras”) nas atualizações de layout. Você vai encontrar uma área semelhante a esta:

```
<reference name=“right”>  
<block type=“checkout/cart_sidebar” name=“cart_sidebar” before=“.”  
template=“checkout/cart/sidebar.phtml”/”>  
</reference”>
```

Troque pelo seguinte código (note que estamos apenas trocando esta linha <reference name=“right”> para <reference name=“left”>).

```
<reference name=“left”>  
<block type=“checkout/cart_sidebar” name=“cart_sidebar” before=“.”
```

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779> 

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

```
template="checkout/cart/sidebar.phtml"/">  
</reference">
```

5. Salve e recarregue a página para ver a alteração

Exercício 2

Use o mesmo método para identificar e remover ou mover o bloco de conteúdo

1. Com a digitação de template, identifique qual módulo do Magento controla outro bloco que você gostaria de mover ou remover (ex.: include do cachorro, da pesquisa, no newsletter, etc.)
2. Copie o arquivo de layout que controla esse bloco do pacote base para seu pacote personalizado.
3. Modifique o arquivo de layout em seu pacote / tema personalizado da forma que você gostaria de fazer. (comente ou delete o bloco que você não quer ou mova para bloco left, right ou footer, etc.) salve suas alterações.
4. Recarregue a loja para ver suas alterações.

Customizando com o arquivo de layout local (local.xml)

Como vimos acima, podemos mudar dramaticamente qualquer bloco e conteúdo que o Magento mostra em um tema modificando alguns arquivos e incluindo apenas os arquivos modificados em seu tema personalizado. Isso funciona porque o Magento trabalha com o sistema de fallback que faz com que a aplicação procure por cada arquivo de tema ou declaração de bloco primeiramente em seu pacote de temas e só então, caso não encontre, pesquisará no pacote base. Seu tema precisa ter apenas os arquivos ou blocos que diferem do pacote base.

Você pode usar o método de alterar e salvar arquivos para todas as versões do Magento e é perfeitamente aceitável para customizar um tema. Uma abordagem mais elegante e sustentável para alterar, ligar ou desligar blocos, que permite fazer a maioria das alterações em um único arquivo é a criação de um arquivo layout em seu tema personalizado que substitui o comportamento de alguns blocos no tema base. Você não precisará copiar todos os arquivos do tema e deletar ou comentar cada bloco de código. Você pode contar com o tema base para todos os blocos padrão e em seu tema simplesmente remover a referência do bloco que não quer. Além disso, pode ainda fazer tudo em um único lugar, de modo que é mais fácil controlar. Na lógica do Magento funciona assim: depois de ter juntado todos os arquivos XML de layout e atualizações, ele irá puxar e processar o arquivo de layout local, chamado local.xml, finalizando, que pode ser usado como uma revisão final para todas as atualizações de tema anteriores. Faça isso dentro de seu pacote de temas, no diretório

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779>



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

app/design/frontend/<seu_pacote>/<seu_tema>/layout:

Crie um arquivo chamado local.xml. Dentro dele crie um bloco <default> que vai conter e consolidar suas alterações globais:

```
<?xml version="1.0" ?>
<layout>
  <default>
    <!-- suas atualizações aqui -->
  </default>
</layout>
```

Dependendo do que você deseja desligar, local.xml pode conter algumas das seguintes linhas:

```
<default>
  <remove name="left.permanent.callout" /> <!--the dog-->
  <remove name="right.permanent.callout" /> <!--back to school-->
  <remove name="catalog.compare.sidebar" /> <!--comparar produtos-->
  <remove name="paypal.partner.right.logo" /> <!--logo paypal-->
  <remove name="cart_sidebar" /> <!--cart sidebar-->
  <remove name="left.reports.product.viewed" /> <!--Produtos Visto recentemente-->
  <remove name="right.reports.product.viewed" /> <!--Produtos Visto recentemente-->
  <remove name="right.reports.product.compared" /> <!--Produtos comparados recentemente-->
</default>
```

O conteúdo vai variar dependendo do seu tema, aqui demos apenas uma orientação. Use esta lista como um guia, encontre o xml no diretório de layout do seu tema (ou no diretório do tema base) que contém o(s) bloco(s) que deseja desabilitar e encontre os nomes dos blocos nesse arquivo. Você poderá, então, inserir esses nomes no local.xml para remover os blocos. Além disso, o exemplo acima remove blocos do escopo padrão, talvez você prefira remove-los apenas de um bloco estrutural específico, como “left” ou “right”, aqui dou uma idéia de como funciona.

Também é possível usar o local.xml para atualizar controladores específicos. Por exemplo, se quisesse definir a página padrão de categorias como duas colunas com sidebar a esquerda (2column-left) você poderia usar os controladores responsáveis que renderizam as páginas de catálogo inserindo um código semelhante a este em seu local.xml.

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779>



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

```
<catalog_category_default>
  <reference name="root">
    <action method="setTemplate"> <template>page/2columns-left.phtml </template> </action>
  </reference>
</catalog_category_default>
<catalog_category_layered>
  <reference name="root">
    <action method="setTemplate"> <template>page/2columns-left.phtml </template> </action>
  </reference>
</catalog_category_layered>
```

Nós apenas arramnhamos a superfície ta usabilidade do arquivo local.xml para atualizar os layouts para os blocos que queira desativar, ao invés de copiar os arquivos e excluir o código. Usando este método, você também pode adicionar blocos, alterar suas posições, mudar alguns dos seus comportamentos, etc. Assim, substituindo o comportamento padrão em um único local sem nunca editar os arquivos de layout reais.

Customizando com arquivos templates

Você viu que os arquivos de layout no Magento controlam a presença ou ausência de cada bloco de conteúdo em um tema. O que, especificamente, é renderizado dentro desses blocos é controlado pelos arquivos de template. Muitos dos arquivos de template não têm qualquer lógica do que vai ou não vai ser mostrado, lembre-se: os arquivos de layout é que fazem este controle. Uma vez que o arquivo de template é chamado, espera-se obter o que será processado e exibido.

Arquivos de template no Magento são os arquivos PHTML que contém código xHTML e PHP que serão passados e renderizados para o browser. Depois que você identificar qual arquivo de template é iniciado para gerar os conteúdos para um específico bloco, você pode modificar este arquivo de template designado. Ou pode-se modificar o arquivo de layout para associar um arquivo de template diferente com um bloco, que permite você a criar um arquivo de template completamente novo.

Como recomendação geral, ao invés de editar os arquivos de template do Magento, é uma boa prática copiar o arquivo de template que deseja alterar para seu tema com um nome ligeiramente diferente e editar esse novo arquivo renomeado. Então você altera o arquivo de layout para indicar que deve usar seu arquivo novo para este bloco, preservando o template original do Magento no caso de ser usado em outros lugares e assegura que quando o Magento for atualizado, qualquer alteração do arquivo original não terá impacto sobre sua alteração sem sua revisão explícita.

Tomando com base o exemplo anterior usando um arquivo local.xml para atualizar uma

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779>



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

declaração de layout, se quiser que suas páginas de categoria tenha uma página de template única, você pode criar uma página de template em `app/design/frontend/<seu_pacote>/default/template/page/` chamada “`2columns-custom.phtml`”. Neste novo arquivo talvez queira uma versão modificada de duas colunas do Magento com uma sidebar a esquerda para incluir também novos blocos estruturais para as promoções e mensagens de frete grátis que só deva aparecer nas páginas de categoria.

Neste caso, seu arquivo `local.xml` em seu diretório `app/design/frontend/<your_package>/default/layout` incluiria o seguinte código dizendo ao Magento para renderizar estas páginas usando seu novo arquivo de template.

```
<?xml version="1.0" ?>
<layout>
  <catalog_category_default>
    <reference name="root">
      <action method="setTemplate"> <template> page/2columns-custom.phtml </template>
    </action>
  </reference>
</catalog_category_default>
<catalog_category_layered>
  <reference name="root">
    <action method="setTemplate"> <template> page/2columns-custom.phtml </template>
  </action>
</reference>
</catalog_category_layered>
</layout>
```

Exercício 1

Separe os links de SEO na área de rodapé - ao invés de ter os links como uma lista, isole “Advanced serach” eo mova-o para o cabeçalho.

O módulo e arquivo de layout que atribui um conteúdo para uma página é, algumas vezes, um pouco difícil de encontrar no começo, especialmente se este bloco aparece em todas as páginas do cabeçalho ou rodapé. Mas vamos pensar um pouco, presumindo que os links SEO devam ser controlados pelo módulo `catalog` ou `catalogsearch` ou que seja parte do bloco estrutural do rodapé, abra o arquivo `layout/page.xml` e examine a lista de filhos sob o bloco `footer` para procurar um bloco que se chame `footer links` – encontraremos o código `<block name="footer_links">`, que é o que chama os links de SEO. Agora sabemos quem chama os links SEO. Então as atualizações de layout referente aos links SEO são através do

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779>



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

do name="footer_links", faça uma pesquisa em todos os arquivos xml para <reference name="foter_links">. Você vai encontrar referências de footer_links em catalog.xml (que chama o "Mapa do site" ou "site map"), catalogsearch.xml (que chama "Termos de Busca" e "Busca Avançada") e contact.xml (que chama "Contato").

1. Seja qual for o método utilizado, uma vez que você tenha localizado a área dos itens de link SEO individuais, você vai começar os passos para isolar a "Pesquisa Avançada" do grupo e torná-lo um item do cabeçalho.
2. Copie todos os arquivos que precise modificar para dentro de seu tema personalizado.
3. Primeiro vá em seu page.xml copiado e crie um novo block <block type="page/template_links" name="header_links" as="header_links" template="page/template/links.phtml"/> e aninhe dentro de <block name="header">. Isto faz com que o layout diga ao Magento que terá este link no header.phtml.
4. No caso de o arquivo header.phtml, o Magento é codificado para procurar esse nome, então vamos editar o arquivo de modelo diretamente no nosso tema personalizado para este exemplo. Abra o seu template/page/html/header.phtml, e digite <?= \$this->getChildHtml('header_links')?> onde você quer que o link apareça.
5. Agora vá para catalogsearch.xml e apague ou comente este código: <action method="addLink" translate="label title" module="catalogsearch"><label>Advanced Search</label><url helper="catalogsearch/getAdvancedSearchUrl" /><title>Advanced Search</title></action> que está em <reference name="footer_links">. Isto irá remover o link do rodapé.
6. Ainda em catalogsearch.xml, crie uma nova referência para o novo bloco header_links e aninhe o código dentro dele assim:
<reference name="header_links">
 <action method="addLink" translate="label title" module="catalogsearch"><label>Advanced Search</label><url helper="catalogsearch/getAdvancedSearchUrl" /><title>Advanced Search</title></action>
</reference>
7. Atualize a loja e veja as alterações. Temos agora Pesquisa avançada no cabeçalho em vez de no rodapé.

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779> 

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

6 Guia rápido para criação de um tema do zero

Muitos temas Magento, mesmo os disponíveis no mercado, são construídos na forma descrita nos capítulos anteriores, modificando arquivos ou modificando os blocos definidos no pacote base do Magento.

É possível, no entanto, criar um novo tema Magento a partir do zero e não usar a estrutura padrão do Magento e os blocos de conteúdo, você não está limitado a header, footer, left, right e containers como seus blocos estruturais, se você não quiser. Se você decidir construir um tema a partir do zero, parcial ou total, você assume várias complicações, mas dependendo do seu projeto ou seu fluxo de trabalho, pode ser o caminho mais coerente para você.

Este capítulo mostrará um pouco do que é necessário sobre como quebrar seu design e recriar um novo layout, template e skins do Magento.

Primeiro: desabilite o cache do sistema

Prepare seu Magento para produção, você precisa desabilitar o sistema de cache indo em seu painel administrativo (<http://seuhost.com/admin>) e navegue até sistema > gerenciar Cache. Selecione os checkboxes a seguir Layouts, Blocks HTML, Output e Translations e em ações selecione “desabilitar” e clique em executar. Cada um desses tipos de cache exibirão uma box vermelha na área de status escrito “DESABILITADO”. Fazendo isso, você assegura que sua frente de loja reflita fielmente todas as suas alterações.

Nota: dependendo do que você estiver alterando pode ser necessário desativar outros tipos de cache ou talvez apagar manualmente os arquivos de cache na sua estrutura de pastas. Nunca delete pastas se você não tiver certeza do que está fazendo.

Segundo: defina todos os tipos de estruturas possíveis para sua loja

Antes de iniciar as marcações para a loja, primeiramente você precisa se perguntar o tipo de estrutura de página que gostaria ter para cada uma das páginas de loja, faça uma lista semelhante a esta:

Página inicial usará três colunas estruturais

Página que lista as categorias usará duas colunas sendo uma barra lateral à direita

Páginas do cliente terão duas colunas sendo incluída a barra lateral na esquerda

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779>



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

O esqueleto do template

Quando sua lista estiver completa, crie as marcações xHTML para cada tipo de estrutura e salve-as como ESQUELETO DE TEMPLATES em `app/design/frontend/design_package/theme_variation/template/page/`. Um esqueleto de template é chamado assim devido sua finalidade, contém tudo (além dos elementos `<head>`), é a apresentação das marcações que servem para posicionar os BLOCOS ESTRUTURAIIS de acordo com suas marcações.

Exemplo de esqueleto de template:

```
<html>
<head></head>
<body>
<div class="header"><?=$this->getChildHtml('header')?></div>
<div class="middle">
  <div class="col-left"><?=$this->getChildHtml('left')?></div>
  <div class="col-main"><?=$this->getChildHtml('content')?></div>
</div>
<div class="footer"><?=$this->getChildHtml('footer')?></div>
</body>
</html>
```

Após a digitalização por meio do esqueleto de template de exemplo acima, você vai notar um método chamado PHP “`<?=$this->getChildHtml()?>`” dentro de cada marcação. É como o Magento carrega os blocos estruturais dentro de um esqueleto de template e nele serão posicionados todos os blocos de conteúdo dentro da página da loja.

Cada `getChildHtml` chama um nome como em `<div class="header"><?=$this->getChildHtml('header')?></div>` e estes nomes são métodos que cada bloco é identificado no layout, o conteúdo dos esqueletos de template são atribuídos à loja através do layout.

Terceiro: corte seu xHTML de acordo para funcionalidade

Depois de ter criado seus esqueletos de template, é hora de criar o template para cada bloco de conteúdo.

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779>



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

O Magento gosta de templates semânticos

Você terá que cortar a marcação xHTML construído para sua página e fornecer a marcação de acordo com todas as funcionalidades da página. Por exemplo, se você tem uma área de mini-carrinho em seu design, a marcação desta área se tornará um arquivo template sozinho. O mesmo vale para tags de produtos, assinar newsletter, etc. Todas estas funcionalidades já vêm com Magento, para que possa consultar os templates de modelos existentes para construir a sua lógica de marcação.

Onde salvar os templates

Diagrama 5



A marcação completa para qualquer página em sua loja é introduzida através de um array em que cada um representa a funcionalidade de um módulo. Para descobrir o caminho do template que está sendo usado que você gostaria de modificar você pode ativar as dicas do template, para habilitá-lo:

1. Vá ao admin da loja em sistema > configurações
2. Selecione sua loja no menu dropdown no topo esquerdo do admin
3. Depois que a página recarregar selecione a aba “Desenvolvedor” e selecione sim para mostrar caminho das telas.
4. Quando tiver terminado, volte para frente de loja e você verá a pasta para todos os templates listados de acordo com o bloco. Agora tudo que precisa fazer para modificar a marcação é seguir o caminho descrito pra modificar os templates.

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779>



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

Quarto: altere o layout para refletir seu design

Depois de distribuir as marcações, você pode querer mover os templates de uma página para ver se está funcionando corretamente.

Layout default VS atualizações de layout

Existem dois tipos de layouts, default e atualizações. Um layout default (page.xml) é o layout que, por padrão aplica-se a quase todas as páginas da loja. Todos os outros arquivos de layout são atualizações de layout que simplesmente atualiza o layout padrão em uma base por página.

Vejamos por exemplo o seu modelo de esqueleto:

No layout padrão, você tem que definir três colunas, o que significa que, por padrão, a maioria das páginas em sua loja terá a estrutura de três colunas, mas nas páginas de visualização de produto não é a estrutura de três colunas que você quer suar. Para isso, você quer uma estrutura de duas colunas que inclui uma coluna à direita. Então, você vai deixar o layout padrão de lado e com o catalog.xml aberto, você pode colocar alguns comandos de layout que informa o aplicativo para carregar a estrutura de duas colunas para a sua página do produto em vez do padrão de três. Isso é chamado de processo de atualização do layout.

Exemplo do método de atribuição do esqueleto de template:

```
<reference name="root">  
<action method="setTemplate"><template>page/2columns-  
right.phtml</template></action>  
</reference>
```

Vamos pegar outro exemplo:

Digamos que por default você quer uma assinatura de newsletter na coluna da direita, mas nas páginas do cliente você queira esconder isso. Deixe seu layout padrão e abra o customer.xml, coloque um comando que desativa o bloco do newsletter, excluindo a funcionalidade newsletter da página.

<https://www.facebook.com/pages/Ronaldo-Diniz/128411380596779> 

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

7 Outros recursos

O Magento oferece uma grande riqueza de ajuda para criação do design de sua loja, recomendamos que use e participe dos seguintes recursos (em inglês):

- Artigos de conhecimento básico (<http://www.magentocommerce.com/knowledge-base>)
- Webinars (<http://www.magentocommerce.com/media/webinars>)
- Forum (<http://www.magentocommerce.com/boards>)

Nós também convidamos você a compartilhar conosco seus projetos, discutir esta documentação e fazer muitas perguntas no fórum da comunidade “HTML, XHTML, CSS, Design Questions” (<http://www.magentocommerce.com/boards/viewforum/16/>).