

Desenvolvimento em comunidade

A história técnica e política de um plugin do WordPress

Vinicius Massuchetto

<http://vmassuchetto.github.io>

Download

Código fonte da apresentação:

<https://github.com/vmassuchetto/wp-bfpui-history>

PDF compilado:

<http://tinyurl.com/wcsp2013>

Sobre o que falaremos

- 1 Introdução
- 2 Problema
- 3 Solução diplomática
- 4 Solução prática
- 5 Considerações finais

O que é o WordPress

O que é o WordPress

- O CMS mais utilizado no mundo

O que é o WordPress

- O CMS mais utilizado no mundo
- 10 anos de desenvolvimento

O que é o WordPress

- O CMS mais utilizado no mundo
- 10 anos de desenvolvimento
- 18,9% dos sites do mundo usam WordPress

O que é o WordPress

- O CMS mais utilizado no mundo
- 10 anos de desenvolvimento
- 18,9% dos sites do mundo usam WordPress
- 29,3% das pessoas nos EUA sabem o que é WordPress

O que é o WordPress

- O CMS mais utilizado no mundo
- 10 anos de desenvolvimento
- 18,9% dos sites do mundo usam WordPress
- 29,3% das pessoas nos EUA sabem o que é WordPress
- 2.000 temas

O que é o WordPress

- O CMS mais utilizado no mundo
- 10 anos de desenvolvimento
- 18,9% dos sites do mundo usam WordPress
- 29,3% das pessoas nos EUA sabem o que é WordPress
- 2.000 temas
- 27.000 plugins

O que é o WordPress

- O CMS mais utilizado no mundo
- 10 anos de desenvolvimento
- 18,9% dos sites do mundo usam WordPress
- 29,3% das pessoas nos EUA sabem o que é WordPress
- 2.000 temas
- 27.000 plugins
- 20.000 desenvolvedores pelo mundo



O que são plugins do WordPress?

O que são plugins do WordPress?

- Rotinas que mudam o comportamento padrão do WordPress, adicionando, removendo ou modificando suas funcionalidades

O que são plugins do WordPress?

- Rotinas que mudam o comportamento padrão do WordPress, adicionando, removendo ou modificando suas funcionalidades
- Qualquer coisa que está no diretório `wp-content/plugins` e tem este cabeçalho:

```
<?php
/*
 * Plugin Name: Nome do plugin
 * Plugin URI: http://url.do/plugin
 * Description: Descricao do plugin
 * Author: Nome de quem fez o plugin
 * Version: 0.1
 * Author URI: http://blog.do/autor-do-plugin
 */
```


Better Front Page UI

Lets you specify the URL for your posts home, so there's no need for dummy pages as placeholders of home.php and front-page.php.

[Download Version 0.03](#) Favorite[Description](#) [Installation](#) [Other Notes](#) [Changelog](#) [Stats](#) [Support](#) [Reviews](#) [Developers](#)

The default way of defining a front page in WordPress is:

1. Create a `front-page.php` for the site home and (optionally) a `home.php` for the posts home.
2. Go to the site admin and create two dummy pages.
3. Select these dummy pages in the reading settings for the front page and the

Requires: 3.0 or higher
Compatible up to: 3.5.2
Last Updated: 2013-7-25
Downloads: 1,922

Ratings



Por que esse plugin?

Por que esse plugin?

- O propósito é reduzido, porém amplo

Por que esse plugin?

- O propósito é reduzido, porém amplo
- A implementação é simples, porém complexa

Falando em números: aspecto técnico

Falando em números: aspecto técnico

- 163 linhas

Falando em números: aspecto técnico

- 163 linhas
- 9 *commits* realizados

Falando em números: aspecto técnico

- 163 linhas
- 9 *commits* realizados
- 3 versões

Falando em números: aspecto técnico

- 163 linhas
- 9 *commits* realizados
- 3 versões
- 2 desenvolvedores

Falando em números: aspecto técnico

- 163 linhas
- 9 *commits* realizados
- 3 versões
- 2 desenvolvedores
- 1 plugin

Falando em números: aspecto político

Falando em números: aspecto político

- 189 mensagens trocadas no Trac e na lista wp-hackers

Falando em números: aspecto político

- 189 mensagens trocadas no Trac e na lista wp-hackers
- 20 *core patches* propostos

Falando em números: aspecto político

- 189 mensagens trocadas no Trac e na lista wp-hackers
- 20 *core patches* propostos
- 10 propostas de usabilidade

Falando em números: aspecto político

- 189 mensagens trocadas no Trac e na lista wp-hackers
- 20 *core patches* propostos
- 10 propostas de usabilidade
- 1 novo campo no admin

Comentários

Aparência

Plugins 6

Usuários

Ferramentas

Configurações

Geral

Escrita

Leitura

Discussão

Mídia

Links

Permanentes

Recolher Menu

Configurações Comuns

<input type="radio"/> Padrão	<code>http://10.11.12.13/?p=123</code>
<input type="radio"/> Dia e nome	<code>http://10.11.12.13/2013/10/17/post-exemplo/</code>
<input type="radio"/> Mês e nome	<code>http://10.11.12.13/2013/10/post-exemplo/</code>
<input type="radio"/> Numérica	<code>http://10.11.12.13/arquivos/123</code>
<input checked="" type="radio"/> Nome do Post	<code>http://10.11.12.13/post-exemplo/</code>
<input type="radio"/> Estrutura Personalizada	<code>http://10.11.12.13 /%postname%/</code>

Opcional

Se preferir, você pode digitar estruturas personalizadas para os URLs das suas categorias e tags. Por exemplo, formato `http://example.org/topicos/uncategorized/`. Se você deixar em branco, serão usados os valores padrão.

Base das categorias	<input type="text"/>
Base das tags	<input type="text"/>

Post Home Page

`http://10.11.12.13/`

This will be the home for your posts

Salvar alterações

Causas dos problemas

Causas dos problemas

- Regra confusa no controlador para a escolha dos templates

Causas dos problemas

- Regra confusa no controlador para a escolha dos templates
- Terminologia conflituosa entre *home* e *front page*

Causas dos problemas

- Regra confusa no controlador para a escolha dos templates
- Terminologia conflituosa entre *home* e *front page*
- Alta sensibilidade às ações do usuário

Consequências dos problemas

Consequências dos problemas

- Temas construídos invariavelmente de forma errada

Consequências dos problemas

- Temas construídos invariavelmente de forma errada
- Confusão e verborragia evitável na documentação

Consequências dos problemas

- Temas construídos invariavelmente de forma errada
- Confusão e verborragia evitável na documentação
- Perda de visões de conteúdo

Consequências dos problemas

- Temas construídos invariavelmente de forma errada
- Confusão e verborragia evitável na documentação
- Perda de visões de conteúdo
- Visões de conteúdo não desejadas

Consequências dos problemas

- Temas construídos invariavelmente de forma errada
- Confusão e verborragia evitável na documentação
- Perda de visões de conteúdo
- Visões de conteúdo não desejadas
- Escolha de templates não desejados

Consequências dos problemas

- Temas construídos invariavelmente de forma errada
- Confusão e verborragia evitável na documentação
- Perda de visões de conteúdo
- Visões de conteúdo não desejadas
- Escolha de templates não desejados
- Sites ‘fora do ar’

Áreas envolvidas no processo decisório

Áreas envolvidas no processo decisório

- Análise de requisitos – implementações compatíveis e ajustáveis com a necessidade do projeto

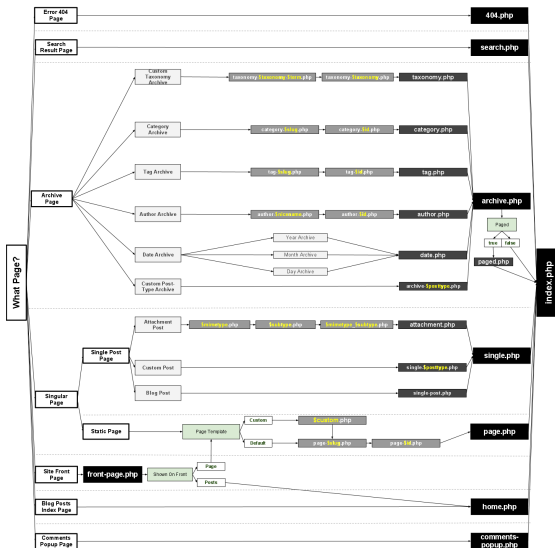
Áreas envolvidas no processo decisório

- Análise de requisitos – implementações compatíveis e ajustáveis com a necessidade do projeto
- Usabilidade na geração de conteúdo – facilidade de treinamento e manutenção do projeto em produção

Áreas envolvidas no processo decisório

- Análise de requisitos – implementações compatíveis e ajustáveis com a necessidade do projeto
- Usabilidade na geração de conteúdo – facilidade de treinamento e manutenção do projeto em produção
- Agilidade de desenvolvimento – tornar a vida do desenvolvedor mais fácil

Problema



Exemplos de hierarquia de template: datas

Exemplos de hierarquia de template: datas

- `http://site/2010`

Exemplos de hierarquia de template: datas

- `http://site/2010`
`http://site/2010/09`

Exemplos de hierarquia de template: datas

- `http://site/2010`
`http://site/2010/09`
`http://site/2010/09/22`

Exemplos de hierarquia de template: datas

- `http://site/2010`
`http://site/2010/09`
`http://site/2010/09/22`
- `→ date.php`

Exemplos de hierarquia de template: datas

- `http://site/2010`
`http://site/2010/09`
`http://site/2010/09/22`
- `→ date.php`
`→ archive.php`

Exemplos de hierarquia de template: datas

- `http://site/2010`
`http://site/2010/09`
`http://site/2010/09/22`
- `→ date.php`
`→ archive.php`
`→ index.php ★`

Exemplos de hierarquia de template: post do tipo livro

Exemplos de hierarquia de template: post do tipo livro

- `http://site/qualquer-post-do-tipo-livro/`

Exemplos de hierarquia de template: post do tipo livro

- `http://site/qualquer-post-do-tipo-livro/`
- `→ single-livros.php`

Exemplos de hierarquia de template: post do tipo livro

- `http://site/qualquer-post-do-tipo-livro/`
- `→ single-livros.php`
`→ single.php`

Exemplos de hierarquia de template: post do tipo livro

- `http://site/qualquer-post-do-tipo-livro/`
- `→ single-livros.php`
 - `→ single.php`
 - `→ index.php ★`

Exemplos de hierarquia de template: categoria

Exemplos de hierarquia de template: categoria

- `http://site/category/qualquer-categoria`

Exemplos de hierarquia de template: categoria

- `http://site/category/qualquer-categoria`
- `→ category-qualquer-categoria.php`

Exemplos de hierarquia de template: categoria

- `http://site/category/qualquer-categoria`
- `→ category-qualquer-categoria.php`
`→ category-22.php`

Exemplos de hierarquia de template: categoria

- `http://site/category/qualquer-categoria`
- `→ category-qualquer-categoria.php`
`→ category-22.php`
`→ category.php`

Exemplos de hierarquia de template: categoria

- `http://site/category/qualquer-categoria`
- `→ category-qualquer-categoria.php`
 - `→ category-22.php`
 - `→ category.php`
 - `→ archive.php`

Exemplos de hierarquia de template: categoria

- `http://site/category/qualquer-categoria`
- `→ category-qualquer-categoria.php`
 - `→ category-22.php`
 - `→ category.php`
 - `→ archive.php`
 - `→ index.php ★`

Exemplos de hierarquia de template: lista dos livros de história

Exemplos de hierarquia de template: lista dos livros de história

- `http://site/livros/historia`

Exemplos de hierarquia de template: lista dos livros de história

- `http://site/livros/historia`
- `→ taxonomy-livros-historia.php`

Exemplos de hierarquia de template: lista dos livros de história

- `http://site/livros/historia`
- `→ taxonomy-livros-historia.php`
`→ taxonomy-livros.php`

Exemplos de hierarquia de template: lista dos livros de história

- `http://site/livros/historia`
- `→ taxonomy-livros-historia.php`
 - `→ taxonomy-livros.php`
 - `→ taxonomy.php`

Exemplos de hierarquia de template: lista dos livros de história

- `http://site/livros/historia`
- `→ taxonomy-livros-historia.php`
 - `→ taxonomy-livros.php`
 - `→ taxonomy.php`
 - `→ archive.php`

Exemplos de hierarquia de template: lista dos livros de história

- `http://site/livros/historia`
- `→ taxonomy-livros-historia.php`
 - `→ taxonomy-livros.php`
 - `→ taxonomy.php`
 - `→ archive.php`
 - `→ index.php ★`

Exemplos de hierarquia de template: página inicial

Exemplos de hierarquia de template: página inicial

- `http://site/`

Exemplos de hierarquia de template: página inicial

- `http://site/`
- `→ p*-problema.php ★`

Configurações de Leitura

A página inicial mostra

☒ Seus posts recentes

☐ Uma [página estática](#) (selecionar abaixo)

Página inicial:

Página de posts:

As páginas do blog mostram no máximo posts

Os feeds RSS mostram os últimos itens

Para cada post em um feed, mostrar

☒ Texto completo

☐ Resumo

Visibilidade nos mecanismos de busca

☐ Evitar que mecanismos de busca indexem este site

Cabe aos mecanismos de busca atender esta solicitação.

Salvar alterações

O problema das *dummy pages*

O problema das *dummy pages*

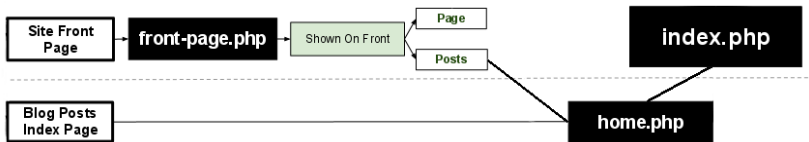
- `get_option('show_on_front')`

O problema das *dummy pages*

- `get_option('show_on_front')`
== 'posts' → lista de posts na página inicial, como usual

O problema das *dummy pages*

- `get_option('show_on_front')`
 - == 'posts' → lista de posts na página inicial, como usual
 - == 'page' → condiciona a seguinte escolha de templates:



O problema das *dummy pages*

O problema das *dummy pages*

- `int(get_option('page_on_front'))`

O problema das *dummy pages*

- `int(get_option('page_on_front'))`
→ `front-page.php`

O problema das *dummy pages*

- `int(get_option('page_on_front'))`
→ `front-page.php`
→ `page-minha-pagina.php`

O problema das *dummy pages*

- `int(get_option('page_on_front'))`
 - `front-page.php`
 - `page-minha-pagina.php`
 - `page-22.php`

O problema das *dummy pages*

- `int(get_option('page_on_front'))`
 - `front-page.php`
 - `page-minha-pagina.php`
 - `page-22.php`
 - `page.php`

O problema das *dummy pages*

- `int(get_option('page_on_front'))`
 - `front-page.php`
 - `page-minha-pagina.php`
 - `page-22.php`
 - `page.php`
 - `index.php` ★

O problema das *dummy pages*

- `int(get_option('page_on_front'))`
 - `front-page.php`
 - `page-minha-pagina.php`
 - `page-22.php`
 - `page.php`
 - `index.php` ★
- `int(get_option('page_for_posts'))`

O problema das *dummy pages*

- `int(get_option('page_on_front'))`
 - `front-page.php`
 - `page-minha-pagina.php`
 - `page-22.php`
 - `page.php`
 - `index.php` ★
- `int(get_option('page_for_posts'))`
 - `home.php`

O problema das *dummy pages*

- `int(get_option('page_on_front'))`
 - `front-page.php`
 - `page-minha-pagina.php`
 - `page-22.php`
 - `page.php`
 - `index.php` ★
- `int(get_option('page_for_posts'))`
 - `home.php`
 - `index.php` ★

Exemplos de hierarquia de template: página inicial

Exemplos de hierarquia de template: página inicial

- `http://site/`

Exemplos de hierarquia de template: página inicial

- `http://site/`
- `→ (p*-problema.php)`

Exemplos de hierarquia de template: página inicial

- `http://site/`
- `→ (p*-problema.php)`
- `→ front-page.php`

Exemplos de hierarquia de template: página inicial

- `http://site/`
- `→ (p*-problema.php)`
- `→ front-page.php`
- `→ home.php`

Exemplos de hierarquia de template: página inicial

- `http://site/`
- `→ (p*-problema.php)`
- `→ front-page.php`
- `→ home.php`
- `→ index.php ★ (!!!)`

Exemplos de hierarquia de template: página inicial

- `http://site/`
- `→ (p*-problema.php)`
- `→ front-page.php`
- `→ home.php`
- `→ index.php ★ (!!!)`
- ... e vai ficar sem a listagem de posts principal

Problema

Problema

- Criação obrigatória das *dummy pages* para ter a completude de funcionalidades para os templates

Problema

- Criação obrigatória das *dummy pages* para ter a completude de funcionalidades para os templates
- Solução sensível à interferência do usuário

Problema

- Criação obrigatória das *dummy pages* para ter a completude de funcionalidades para os templates
- Solução sensível à interferência do usuário
- Recurso indesejado para boa parte dos sites com perfil comercial e que se distanciem da dinâmica de blogs

Nosso ponto de vista

Nosso ponto de vista

- O usuário deve ter acesso a algumas funcionalidades administrativas ao mesmo tempo que não deve ser capaz de interferir na estrutura lógica do site

Nosso ponto de vista

- O usuário deve ter acesso a algumas funcionalidades administrativas ao mesmo tempo que não deve ser capaz de interferir na estrutura lógica do site
- A decisão sobre a capacidade do usuário em ter acesso a determinadas funcionalidades possivelmente desnecessárias deve estar sob responsabilidade do desenvolvedor

Nosso ponto de vista

- O usuário deve ter acesso a algumas funcionalidades administrativas ao mesmo tempo que não deve ser capaz de interferir na estrutura lógica do site
- A decisão sobre a capacidade do usuário em ter acesso a determinadas funcionalidades possivelmente desnecessárias deve estar sob responsabilidade do desenvolvedor
- Quanto mais simples para o usuário, melhor

Exemplos de funcionalidade suprimíveis nos temas

Função `add_theme_support()`

Exemplos de funcionalidade suprimíveis nos temas

Função `add_theme_support()`

- `automatic-feed-links`: Referências para feeds na `<head>`

Exemplos de funcionalidade suprimíveis nos temas

Função `add_theme_support()`

- `automatic-feed-links`: Referências para feeds na `<head>`
- `post-thumbnails`: Campos de imagens destacadas nos posts

Exemplos de funcionalidades suprimíveis nos temas

Exemplos de funcionalidades suprimíveis nos temas

- `add_editor_style()`: Aparência e funcionalidade do TinyMCE – editor de conteúdo textual

Exemplos de funcionalidades suprimíveis nos temas

- `add_editor_style()`: Aparência e funcionalidade do TinyMCE – editor de conteúdo textual
- `add_custom_background()`: Controle do fundo do site

Exemplos de funcionalidades suprimíveis nos temas

- `add_editor_style()`: Aparência e funcionalidade do TinyMCE – editor de conteúdo textual
- `add_custom_background()`: Controle do fundo do site
- `add_custom_image_header()`: Controle da imagem do cabeçalho

Exemplos de funcionalidades suprimíveis nos temas

- `add_editor_style()`: Aparência e funcionalidade do TinyMCE – editor de conteúdo textual
- `add_custom_background()`: Controle do fundo do site
- `add_custom_image_header()`: Controle da imagem do cabeçalho
- `register_nav_menus()`: Distribuição de links ao longo de menus

Exemplos de funcionalidades suprimíveis nos temas

- `add_editor_style()`: Aparência e funcionalidade do TinyMCE – editor de conteúdo textual
- `add_custom_background()`: Controle do fundo do site
- `add_custom_image_header()`: Controle da imagem do cabeçalho
- `register_nav_menus()`: Distribuição de links ao longo de menus
- `register_sidebar()`: Distribuição de widgets sobre certas áreas do site

Nosso ponto de vista

Nosso ponto de vista

- Conteúdo e estrutura lógica devem ser coisas diferentes

Nosso ponto de vista

- Conteúdo e estrutura lógica devem ser coisas diferentes
- Funcionalidades que possibilitam a extensibilidade moderadamente complexa do WordPress o popularizaram para a utilização em grandes serviços

Nosso ponto de vista

- Conteúdo e estrutura lógica devem ser coisas diferentes
- Funcionalidades que possibilitam a extensibilidade moderadamente complexa do WordPress o popularizaram para a utilização em grandes serviços
- O WordPress só é o que é hoje porque pôde flexibilizar-se o suficiente para ser aplicável às mais diversas situações, desde um simples blog a mega portais.

Duas frentes de discussão

Duas frentes de discussão

- Discussões de funcionalidade na lista
`wp-hackers@lists.automattic.com`

Duas frentes de discussão

- Discussões de funcionalidade na lista `wp-hackers@lists.automattic.com`
- Implementação de patches e propostas de usabilidade no Trac do core do WordPress

“Eu acho que vocês estão se confundindo um pouco com a terminologia home e front page.

home.php é o seu template para posts do blog (o seu archive para o post do tipo post). É sempre usado quando is_home() é verdade, mesmo que esta não seja a página inicial do seu site. Você nunca deveria “colocar qualquer coisa que quisesse ali” neste template. Ele serve para mostrar os posts do seu blog.

front-page.php é para sobrescrever qualquer coisa que seja mostrada na página inicial do site, independente de qualquer outra configuração.”

– Justin Tadlock – *theme reviewer*

...problema entendido na discussão ...

“Sou um ferrenho defensor de que os temas permaneçam em seu escopo funcional: que é apresentar o conteúdo, e não definir ou gerenciar o conteúdo e as opções do site.”

– Chip Bennet– *theme reviewer*

“Isto faz sentido quando você está fazendo um blog geral, mas não é verdade quando você está fazendo um tema para um site especificadamente. Não existe situação em que o administrador do site vai querer que a página inicial mude.

Não é bom ter que explicar toda vez para o cliente: ‘Vê esta página qui? Ela não tem conteúdo nenhum e qualquer coisa que você escrever nela não vai aparecer em lugar nenhum, mas deixe ela aí porque se você removê-la vai quebrar o seu site’.”

– Leo Germani

“Em um desenvolvimento encomendado, o desenvolvedor nada mais é do que um agente do usuário, que é o cliente. Não existe razão para o desenvolvedor, que desenvolve, instala, e configura o Tema, não possa também configurar as opções de leitura para que o site exiba o conteúdo adequadamente.

De qualquer forma, o caso de uso que você descreve não é aquele compartilhado pela grande maioria dos usuários do WordPress, e a habilidade do tema em editar o que vai aparecer na capa do site seria danoso para esta grande maioria.”

– Chip Bennett – *theme reviewer*

“Se você não quer o seu cliente mudando as opções do site, então não o cadastre como um admin.

Mas eu concordo com alguns comentários que pode ser confuso ver duas páginas que não servem pra nada no admin. Uma modificação útil seria um aviso no topo destas páginas avisando que elas servem para o que o Tema exiba as informações corretas.”

– Bill Erickson – *freelancer*

“Eu acho que vocês não estão entendendo a questão do Chip sobre como os temas não devem ditar a estrutura do site para a grande maioria dos usuários. É sobre quem controla o site, e não a finalidade do site.

– Justin Tadlock – *theme reviewer*

Solução diplomática


Front page displays

☐ A single, editable pageVisiting `http://example.com` will show a page, instead of your latest posts.

Front page displays

☒ A single, editable page: titled Visiting `http://example.com` will show a page, instead of your latest posts.☒ Show my latest posts at `http://mysiteurl.com/` /

Front page displays

☒ A single, editable page: Visiting `http://example.com` will show a page, instead of your latest posts.☒ Show my latest posts at `http://mysiteurl.com/` /


After creating new "home" page in step above and saving the change

– Dave Martin – *designer na Automattic*

1

Initial view when enabled

Enable a static front page

Show a [page](#) instead of your latest posts

— Select —



Show latest posts on a separate page

— Select —

3

Maybe we want new pages for both

Enable a static front page

Show a [page](#) instead of your latest posts

— Add new page —

titled: Home



Show latest posts on a separate page

— Add new page —

Page title: Blog

Located at <http://mutrunk.dev/blog/>

Edit

2

Maybe add new homepage, choose existing for latest posts

Enable a static front page

Show a [page](#) instead of your latest posts

— Add new page —

titled: Home



Show latest posts on a separate page

Blog

— Drew Jaynes — *freelancer*

Introdução
oooooooooooo

Problema
oooooooooooooooo

Solução diplomática
oooooooooooooooo●

Solução prática
oooooooo

Considerações finais
ooo

Solução diplomática

- Jan 2011: Questão aberta por Mark Jaquith sobre a usabilidade desta seção do admin

- Jan 2011: Questão aberta por Mark Jaquith sobre a usabilidade desta seção do admin
- Out 2012: Funcionalidade considerada crítica e de alta prioridade

- Jan 2011: Questão aberta por Mark Jaquith sobre a usabilidade desta seção do admin
- Out 2012: Funcionalidade considerada crítica e de alta prioridade
- Nov 2012: Complicações de usabilidade e urgência de liberação de versões fazem as propostas de *patches* serem agendadas para um *release* indefinido. “Faço isso com peso no coração.” – Andrew Nacin, ao retirar esta funcionalidade da *milestone*

- Jan 2011: Questão aberta por Mark Jaquith sobre a usabilidade desta seção do admin
- Out 2012: Funcionalidade considerada crítica e de alta prioridade
- Nov 2012: Complicações de usabilidade e urgência de liberação de versões fazem as propostas de *patches* serem agendadas para um *release* indefinido. “Faço isso com peso no coração.” – Andrew Nacin, ao retirar esta funcionalidade da *milestone*
- Jan 2013: Última resposta sobre esta questão

Funcionalidade esperada

Funcionalidade esperada

- Fazer com que o desenvolvedor use somente a `front-page.php`

Funcionalidade esperada

- Fazer com que o desenvolvedor use somente a `front-page.php`
- Se o `front-page.php` estiver criado, utiliza-o para a capa do site e habilita-se uma nova URL para o índice de posts – `/blog` por padrão

Etapas de implementação

Etapas de implementação

- Desenhar uma estrutura base para o plugin

Etapas de implementação

- Desenhar uma estrutura base para o plugin
- Adicionar um campo no admin para que seja escolhida uma URL que vai responder pela lista de posts

Etapas de implementação

- Desenhar uma estrutura base para o plugin
- Adicionar um campo no admin para que seja escolhida uma URL que vai responder pela lista de posts
- Implementar as *rewrite rules* para gerar as requisições corretas para esta URL

Etapas de implementação

- Desenhar uma estrutura base para o plugin
- Adicionar um campo no admin para que seja escolhida uma URL que vai responder pela lista de posts
- Implementar as *rewrite rules* para gerar as requisições corretas para esta URL
- Suprimir a funcionalidade da opção `show_on_front` para o caso de plugins e temas que a utilizem

Etapas de implementação

- Desenhar uma estrutura base para o plugin
- Adicionar um campo no admin para que seja escolhida uma URL que vai responder pela lista de posts
- Implementar as *rewrite rules* para gerar as requisições corretas para esta URL
- Suprimir a funcionalidade da opção `show_on_front` para o caso de plugins e temas que a utilizem
- Definir funções auxiliares para que as URLs sejam utilizadas pelo tema

Introdução
oooooooooooo

Problema
oooooooooooooooo

Solução diplomática
oooooooooooooooo

Solução prática
oo●ooooo

Considerações finais
ooo

Solução prática

```
<?php // estrutura basica
```

```
<?php // estrutura basica

// plugins/better-front-page-ui/better-front-page-ui.php
// cabecalho
class Better_Front_Page_UI {
    function Better_Front_Page_UI() {
        /* acoes e filtros de inicializacao */
    }
    function nome_do_gancho() {
        /* codigo */
    }
}
```

```
<?php // estrutura basica

// plugins/better-front-page-ui/better-front-page-ui.php
// cabeçalho
class Better_Front_Page_UI {
    function Better_Front_Page_UI() {
        /* acoes e filtros de inicializacao */
    }
    function nome_do_gancho() {
        /* codigo */
    }
}

function better_front_page_ui_init() {
    new Better_Front_Page_UI();
}
add_action( 'plugins_loaded', 'better_front_page_ui_init' );
```

Introdução
oooooooooooo

Problema
oooooooooooooooo

Solução diplomática
oooooooooooooooo

Solução prática
ooo●oooo

Considerações finais
ooo

Solução prática


```
<?php // campo no admin
```

```
<?php // campo no admin

if ( is_admin() ) {
    add_action( 'admin_init', array( $this, 'admin_init' ) );
}
```

```
<?php // campo no admin

if ( is_admin() ) {
    add_action( 'admin_init', array( $this, 'admin_init' ) );
}

function admin_init() {
    add_settings_field( self::$option_name, __( 'Post Home
    Page', 'better_front_page_ui' ), array( $this,
    'output_setting_form' ), 'permalink', 'optional' );
    global $pagenow;
    if ( $pagenow == 'options-permalink.php' ) {
        if ( isset( $_POST[self::$option_name] ) ) {
            $value = $this->sanitize_option(
            $_POST[self::$option_name] );
            update_option( self::$option_name, $value );
        } else {
            delete_option( self::$option_name );
        }
    }
}
```

```
<?php // campo no admin

if ( is_admin() ) {
    add_action( 'admin_init', array( $this, 'admin_init' ) );
}

function admin_init() {
    add_settings_field( self::$option_name, __( 'Post Home
    Page', 'better_front_page_ui' ), array( $this,
    'output_setting_form' ), 'permalink', 'optional' );
    global $pagenow;
    if ( $pagenow == 'options-permalink.php' ) {
        if ( isset( $_POST[self::$option_name] ) ) {
            $value = $this->sanitize_option(
            $_POST[self::$option_name] );
            update_option( self::$option_name, $value );
        } else {
            delete_option( self::$option_name );
        }
    }
}
```

Introdução
oooooooooooo

Problema
oooooooooooooooooooo

Solução diplomática
oooooooooooooooooooo

Solução prática
oooo●ooo

Considerações finais
ooo

Solução prática

```
<?php //rewrite rule
```

```
<?php //rewrite rule
```

```
add_action( 'rewrite_rules_array', array( $this,  
    'rewrite_rules_array' ), 9999 );
```

```
<?php //rewrite rule

add_action( 'rewrite_rules_array', array( $this,
    'rewrite_rules_array' ), 9999 );

function rewrite_rules_array( $rules ) {
    $option = $this->get_option();
    $new_rules = array(
        $option . '/?$' => 'index.php?force_home=1',
        $option . '/page/?([0-9]{1,})/?$' =>
        'index.php?force_home=1&paged=$matches[1]',
    );
    return array_merge( $new_rules, $rules );
}
```



```
<?php //rewrite rule

add_action( 'rewrite_rules_array', array( $this,
    'rewrite_rules_array' ), 9999 );

function rewrite_rules_array( $rules ) {
    $option = $this->get_option();
    $new_rules = array(
        $option . '/?$' => 'index.php?force_home=1',
        $option . '/page/?([0-9]{1,})/?$' =>
        'index.php?force_home=1&paged=$matches[1]',
    );
    return array_merge( $new_rules, $rules );
}

add_filter( 'query_vars', array( $this, 'query_vars' ) );
```

```
<?php //rewrite rule

add_action( 'rewrite_rules_array', array( $this,
    'rewrite_rules_array' ), 9999 );

function rewrite_rules_array( $rules ) {
    $option = $this->get_option();
    $new_rules = array(
        $option . '/?$' => 'index.php?force_home=1',
        $option . '/page/?([0-9]{1,})/?$' =>
        'index.php?force_home=1&paged=$matches[1]',
    );
    return array_merge( $new_rules, $rules );
}

add_filter( 'query_vars', array( $this, 'query_vars' ) );

function query_vars( $vars ) {
    array_push( $vars, 'force_home' );
    return $vars;
}
```

Introdução
oooooooooooo

Problema
oooooooooooooooo

Solução diplomática
oooooooooooooooo

Solução prática
oooo●oo

Considerações finais
ooo

Solução prática

```
<?php // suprime a funcionalidade do show_on_front
```

```
<?php // suprime a funcionalidade do show_on_front  
  
add_filter( 'option_show_on_front', array( $this,  
    'filter_show_on_front' ) );
```

```
<?php // suprime a funcionalidade do show_on_front

add_filter( 'option_show_on_front', array( $this,
    'filter_show_on_front' ) );

function filter_show_on_front( $value ) {
    return ( is_home() && get_query_var( 'force_home' ) == 1 ) ?
        'force' : $value;
}
```

Introdução
oooooooooooo

Problema
oooooooooooooooo

Solução diplomática
oooooooooooooooo

Solução prática
oooooo●o

Considerações finais
ooo

Solução prática

```
<?php // fazendo as coisas com asseio
```



```
<?php // fazendo as coisas com asseio
```

```
register_activation_hook( __FILE__, array(  
    'Better_Front_Page_UI', 'activate' ) );  
function activate() { flush_rewrite_rules(); }
```

```
<?php // fazendo as coisas com asseio
```

```
register_activation_hook( __FILE__, array(  
    'Better_Front_Page_UI', 'activate' ) );  
function activate() { flush_rewrite_rules(); }
```

```
register_deactivation_hook( __FILE__, array(  
    'Better_Front_Page_UI', 'deactivate' ) );  
function deactivate() { flush_rewrite_rules(); }
```

```
<?php // fazendo as coisas com asseio
```

```
register_activation_hook( __FILE__, array(  
    'Better_Front_Page_UI', 'activate' ) );  
function activate() { flush_rewrite_rules(); }
```

```
register_deactivation_hook( __FILE__, array(  
    'Better_Front_Page_UI', 'deactivate' ) );  
function deactivate() { flush_rewrite_rules(); }
```

```
register_uninstall_hook( __FILE__, array(  
    'Better_Front_Page_UI', 'uninstall' ) );  
function uninstall() {  
    delete_option( self::$option_name );  
    flush_rewrite_rules();  
}
```

Introdução
oooooooooooo

Problema
oooooooooooooooo

Solução diplomática
oooooooooooooooo

Solução prática
ooooooo●

Considerações finais
ooo

Solução prática

```
<?php // funcoes auxiliares para a URL escolhida
```

```
<?php // funcoes auxiliares para a URL escolhida

function get_the_posts_home_url() {
    if ( class_exists( 'Better_Front_Page_UI' ) )
        return home_url( get_option(
            Better_Front_Page_UI::$option_name ) );
    return false;
}
```

```
<?php // funcoes auxiliares para a URL escolhida
```

```
function get_the_posts_home_url() {  
    if ( class_exists( 'Better_Front_Page_UI' ) )  
        return home_url( get_option(  
            Better_Front_Page_UI::$option_name ) );  
    return false;  
}
```

```
function the_posts_home_url() {  
    if ( $output = get_the_posts_home_url() )  
        echo $output;  
}
```

Quebrando o sono

Quebrando o sono

- Por que eu preciso reiniciar as regras de URL ao ativar e desativar o plugin, mas não ao salvar a opção no formulário?

Quebrando o sono

- Por que eu preciso reiniciar as regras de URL ao ativar e desativar o plugin, mas não ao salvar a opção no formulário?
- Se a opção tivesse que ser colocada fora do `options-permalink.php`, precisaríamos salvá-la?

Considerações finais

Considerações finais

- Antes de fazer um plugin que interfira no funcionamento estrutural do WordPress, verifique as discussões na comunidade e no Trac.

Considerações finais

- Antes de fazer um plugin que interfira no funcionamento estrutural do WordPress, verifique as discussões na comunidade e no Trac.
- Melhor do que esperar que a comunidade desenvolva, se o propósito está claro e é útil para você, simplesmente faça o plugin ou proponha o *patch*

Considerações finais

- Antes de fazer um plugin que interfira no funcionamento estrutural do WordPress, verifique as discussões na comunidade e no Trac.
- Melhor do que esperar que a comunidade desenvolva, se o propósito está claro e é útil para você, simplesmente faça o plugin ou proponha o *patch*
- Uma funcionalidade simples pode vir a ser elaborada do ponto de vista técnico, e inviável do ponto de vista político



Plugin Better Front Page UI

<http://wordpress.org/plugins/better-front-page-ui/>