

# Plugins no WordPress

## Fazendo o Negócio Direito

Vinicius Massuchetto

WordCamp São Paulo 2012

Agosto de 2012

# Download

Esta apresentação está disponível em:  
<http://vinicius.soylocoporti.org.br/?p=2191>

# Sobre a Palestra

# Sobre a Palestra

- Motivação, dificuldades e vantagens dos métodos

# Sobre a Palestra

- Motivação, dificuldades e vantagens dos métodos
- Padrões de desenvolvimento no WordPress

# Sobre a Palestra

- Motivação, dificuldades e vantagens dos métodos
- Padrões de desenvolvimento no WordPress
- Estrutura de código

# Sobre a Palestra

- Motivação, dificuldades e vantagens dos métodos
- Padrões de desenvolvimento no WordPress
- Estrutura de código
- Interfaces com o WordPress

# Sobre a Palestra

- Motivação, dificuldades e vantagens dos métodos
- Padrões de desenvolvimento no WordPress
- Estrutura de código
- Interfaces com o WordPress
- Ferramentas úteis já presentes no WordPress

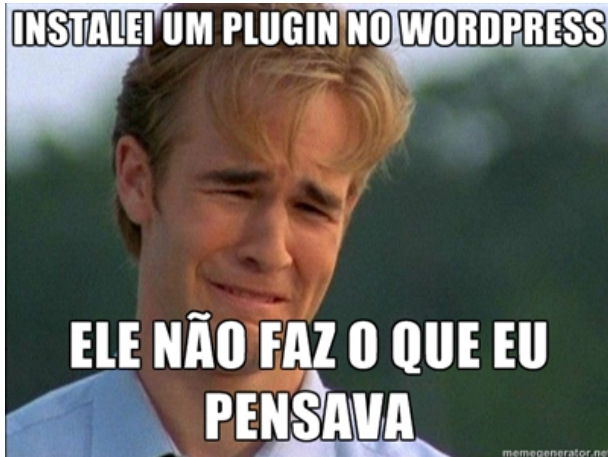


# Motivação



[quidmeme.com](http://quidmeme.com)

# Motivação



# Motivos para se criar um plugin

# Motivos para se criar um plugin

- Funcionalidade inexistente

# Motivos para se criar um plugin

- Funcionalidade inexistente
- Diferente implementação de uma funcionalidade existente

# Motivos para se criar um plugin

- Funcionalidade inexistente
- Diferente implementação de uma funcionalidade existente
- Códigos de tema portáveis

# Motivos para se criar um plugin

- Funcionalidade inexistente
- Diferente implementação de uma funcionalidade existente
- Códigos de tema portáteis
- Implementações modulares para clientes

# Motivos para se criar um plugin

- Funcionalidade inexistente
- Diferente implementação de uma funcionalidade existente
- Códigos de tema portáteis
- Implementações modulares para clientes
- Forks para ajustes e extensões de plugins existentes



# Perguntas

# Perguntas

- Demonstração, apoio ou funcionalidade crítica?

# Perguntas

- Demonstração, apoio ou funcionalidade crítica?
- Comunidade, visibilidade ou emprego?

# Perguntas

- Demonstração, apoio ou funcionalidade crítica?
- Comunidade, visibilidade ou emprego?
- Tempo para suporte?

# Dificuldades em se escrever um plugin

# Dificuldades em se escrever um plugin

- PHP X WordPress

# Dificuldades em se escrever um plugin

- PHP X WordPress
- Cultura de leitura de documentação e inspeção de código

# Dificuldades em se escrever um plugin

- PHP X WordPress
- Cultura de leitura de documentação e inspeção de código
- Barreira de idioma



# Dificuldades em se escrever um plugin

- PHP X WordPress
- Cultura de leitura de documentação e inspeção de código
- Barreira de idioma
- Entrega de produtos deficitários

# Vantagens de se seguir algumas boas práticas

# Vantagens de se seguir algumas boas práticas

- Código legível

# Vantagens de se seguir algumas boas práticas

- Código legível
- Padronização de estruturas

# Vantagens de se seguir algumas boas práticas

- Código legível
- Padronização de estruturas
- Melhor aprendizado de outros desenvolvedores

# Vantagens de se seguir algumas boas práticas

- Código legível
- Padronização de estruturas
- Melhor aprendizado de outros desenvolvedores
- Melhor manutenção do código

# Vantagens de se seguir algumas boas práticas

- Código legível
- Padronização de estruturas
- Melhor aprendizado de outros desenvolvedores
- Melhor manutenção do código
- Extensibilidade

# Vantagens de se seguir algumas boas práticas

- Código legível
- Padronização de estruturas
- Melhor aprendizado de outros desenvolvedores
- Melhor manutenção do código
- Extensibilidade
- Distributividade na comunidade do software livre



# Pensando Em Um Plugin

# Pensando Em Um Plugin

- Definição de escopo e pesquisa de funcionalidades

# Pensando Em Um Plugin

- Definição de escopo e pesquisa de funcionalidades
- Se parecer redundante, perguntar e descrever a ideia em listas e fóruns

# Pensando Em Um Plugin

- Definição de escopo e pesquisa de funcionalidades
- Se parecer redundante, perguntar e descrever a ideia em listas e fóruns
- Escolha de nome único e relevante

# Pensando Em Um Plugin

- Definição de escopo e pesquisa de funcionalidades
- Se parecer redundante, perguntar e descrever a ideia em listas e fóruns
- Escolha de nome único e relevante
- Avaliação do uso de outras tecnologias e frameworks



# Primeiro Padrão



# Arquivos



# Arquivos

- Nomear o diretório e os arquivos coerentemente

# Arquivos

- Nomear o diretório e os arquivos coerentemente
- Incluir somente arquivos necessários e sob demanda no código

# Arquivos

- Nomear o diretório e os arquivos coerentemente
- Incluir somente arquivos necessários e sob demanda no código
- Permitir que o diretório do plugin mude usando funções como:

# Arquivos

- Nomear o diretório e os arquivos coerentemente
- Incluir somente arquivos necessários e sob demanda no código
- Permitir que o diretório do plugin mude usando funções como:
  - `plugins_url`

# Arquivos

- Nomear o diretório e os arquivos coerentemente
- Incluir somente arquivos necessários e sob demanda no código
- Permitir que o diretório do plugin mude usando funções como:
  - `plugins_url`
  - `plugin_dir_url`

# Arquivos

- Nomear o diretório e os arquivos coerentemente
- Incluir somente arquivos necessários e sob demanda no código
- Permitir que o diretório do plugin mude usando funções como:
  - `plugins_url`
  - `plugin_dir_url`
  - `plugin_dir_path`

# Nomeação

# Nomeação

```
./some-plugin
./img
./inc
./js
+ sp.js
+ somelib.js
+ sp.php          <-- arquivo principal
+ sp-functions.php
+ sp-admin.php
+ sp-cache.php
```



# Inclusão Condicional

# Inclusão Condicional

```
if ( is_admin() ){  
  
    /* interface administrativa */  
  
    add_action ...  
    add_filter ...  
  
} else {  
  
    /* frontend */  
  
    add_action ...  
    add_filter ...  
  
}
```

# Padrões de Código

# Padrões de Código

- Ater-se aos padrões recomendados para código e documentação

# Padrões de Código

- Ater-se aos padrões recomendados para código e documentação
- Nomear as estruturas e funções com um identificador único

# Padrões de Código

- Ater-se aos padrões recomendados para código e documentação
- Nomear as estruturas e funções com um identificador único
- Clareza é melhor do que praticidade

# Tag PHP

# Tag PHP

## Errado

```
<? /* code */ ?>
```

```
<?=$var?>
```

```
<% /* code */ %>
```



# Tag PHP

Certo

```
<?php /* code */ ?>
```

```
<?php echo $var; ?>
```

# Chaves

## Errado

```
if(condition)
{
    action1();
    action2();
}
else
{
    action3();
    action4();
}
```

# Chaves

## Certo

```
if ( condition ) {  
    action1();  
    action2();  
} else {  
    action3();  
    action4();  
}
```

# Funções

## Errado

```
function some_function($param1='foo', $param2='bar') {  
    /* code */  
}  
  
class SomeClass {  
    /* code */  
}
```

# Funções

## Certo

```
function some_function( $param1 = 'foo', $param2 = 'bar' ) {  
    /* code */  
}  
  
class Some_Class {  
    /* code */  
}
```

# Vetores

## Errado

```
$x = $foo[$bar];
```

```
$x = $foo[ 'bar' ];
```

# Vetores

Certo

```
$x = $foo[ $bar ];
```

```
$x = $foo['bar'];
```

# Padrões de SQL



# Padrões de SQL

- Evitar escrever consultas

# Padrões de SQL

- Evitar escrever consultas
- Escrever as palavras SQL em caixa alta

# Padrões de SQL

- Evitar escrever consultas
- Escrever as palavras SQL em caixa alta
- Validar os tipos de dados antes de utilizá-los

# Padrões de SQL

- Evitar escrever consultas
- Escrever as palavras SQL em caixa alta
- Validar os tipos de dados antes de utilizá-los
- Selecionar só o que for necessário

# Padrões de SQL

- Evitar escrever consultas
- Escrever as palavras SQL em caixa alta
- Validar os tipos de dados antes de utilizá-los
- Selecionar só o que for necessário
- Utilizar a wpdb

# Padrões de SQL

- Evitar escrever consultas
- Escrever as palavras SQL em caixa alta
- Validar os tipos de dados antes de utilizá-los
- Selecionar só o que for necessário
- Utilizar a wpdb
- Se precisar criar tabelas no banco, use `$wpdb->prefix`

# Exemplo de Consulta

## Errado

```
$r = $wpdb->get_results("select * from wp_posts where post_date  
> '2012-08-25 00:00:00'");
```

# Exemplo de Consulta

Certo

```
$r = $wpdb->get_results( $wpdb->prepare(
    "
    SELECT ID, post_title, post_content
    FROM $wpdb->posts
    WHERE 1=1
        AND post_date > '%s'
    "
    , $some_date ) );
```





# Debug



# Constantes de debug no wp-config.php

# Constantes de debug no wp-config.php

- WP\_DEBUG

# Constantes de debug no wp-config.php

- WP\_DEBUG
- WP\_DEBUG\_LOG

# Constantes de debug no wp-config.php

- WP\_DEBUG
- WP\_DEBUG\_LOG
- WP\_DEBUG\_DISPLAY

# Cabeçalho

# Cabeçalho

Todo plugin começa pelo começo..



# Cabeçalho

Todo plugin começa pelo começo..

```
/*  
 * Plugin Name: Nome do Plugin  
 * Version: 0.01  
 * Description: Texto longo  
 * Author: Nome do Autor  
 * Contributors: Logins no WordPress.org  
 * Plugin URI: http://uma-url.com  
 */
```

# Estrutura Procedural

# Estrutura Procedural

```
function callback1 () { /* code */ }  
add_action( 'hook1', 'callback1' );  
  
function callback2 () { /* code */ }  
add_action( 'hook2', 'callback2' );  
  
function callback3 ($param) { /* code */ return $param; }  
add_filter( 'filter', 'callback3' );
```

# Estrutura Orientada a Objetos

# Estrutura Orientada a Objetos

```
class Some_Plugin {  
  
    function Some_Plugin() {  
        add_action( 'hook1', array($this, 'hook1' ));  
        add_action( 'hook1', array($this, 'hook2' ));  
    }  
  
    function hook1 { /* code */ }  
    function hook2 { /* code */ }  
  
}
```

# Vantagens da Orientação a Objetos em Plugins

# Vantagens da Orientação a Objetos em Plugins

- Organiza o código

# Vantagens da Orientação a Objetos em Plugins

- Organiza o código
- Melhora a extensibilidade



# Vantagens da Orientação a Objetos em Plugins

- Organiza o código
- Melhora a extensibilidade
- Reduz o impacto no escopo global do PHP

# Vantagens da Orientação a Objetos em Plugins

- Organiza o código
- Melhora a extensibilidade
- Reduz o impacto no escopo global do PHP
- Ajuda a não introduzir variáveis globais

# Ativação

# Ativação

- `register_activation_hook`

# Ativação

- `register_activation_hook`
- Criação de opções padrão

# Ativação

- `register_activation_hook`
- Criação de opções padrão
- Criação de tabelas

# Ativação

- `register_activation_hook`
- Criação de opções padrão
- Criação de tabelas
- Exibição de avisos para o usuário configurar o plugin

# Desativação



# Desativação

- `register_deactivation_hook`

# Desativação

- `register_deactivation_hook`
- Em geral não deve causar nenhuma perda de dados

# Desativação

- `register_deactivation_hook`
- Em geral não deve causar nenhuma perda de dados
- Desabilitar plugins dependentes

# Desinstalação

# Desinstalação

- `register_uninstall_hook`

# Desinstalação

- `register_uninstall_hook`
- Não deve deixar nenhum dado residual no WordPress

# Desinstalação

- `register_uninstall_hook`
- Não deve deixar nenhum dado residual no WordPress
- Remove opções do usuário

# Desinstalação

- `register_uninstall_hook`
- Não deve deixar nenhum dado residual no WordPress
- Remove opções do usuário
- Remove tabelas



# Desinstalação

- `register_uninstall_hook`
- Não deve deixar nenhum dado residual no WordPress
- Remove opções do usuário
- Remove tabelas
- Avisa o usuário antes de remover qualquer dado (`admin_notices`)

# Inicialização

# Inicialização

- `*_init()`

# Inicialização

- `*_init()`
- Geralmente através de um procedimento inicializador

# Inicialização

```
function someplugin_init () {  
    add_action( 'hook1', 'callback1' );  
    add_action( 'hook2', 'callback2' );  
    add_filter( 'filter', 'callback3' );  
}  
someplugin_init();
```

# Inicialização

```
function someplugin_init () {  
    add_action( 'hook1', 'callback1' );  
    add_action( 'hook2', 'callback2' );  
    add_filter( 'filter', 'callback3' );  
}  
someplugin_init();
```

```
add_action( 'plugins_loaded', 'someplugin_init' );
```

# Inicialização

```
class Some_Plugin {  
  
    function SomePlugin() {  
        add_action( 'hook1', array($this, 'hook1' ));  
        add_action( 'hook1', array($this, 'hook2' ));  
    }  
  
    function hook1 { /* code */ }  
    function hook2 { /* code */ }  
  
}  
new Some_Plugin();
```

# Inicialização

```
class Some_Plugin {  
  
    function SomePlugin() {  
        add_action( 'hook1', array($this, 'hook1' ));  
        add_action( 'hook1', array($this, 'hook2' ));  
    }  
  
    function hook1 { /* code */ }  
    function hook2 { /* code */ }  
  
}  
new Some_Plugin();  
  
add_action( 'plugins_loaded',  
    array( 'Some_Plugin', 'Some_Plugin' ));
```



# Banco de dados

Usar sempre a wpdb:

# Banco de dados

Usar sempre a wpdb:

- query

# Banco de dados

Usar sempre a wpdb:

- query
- prepare

# Banco de dados

Usar sempre a wpdb:

- query
- prepare
- insert

# Banco de dados

Usar sempre a wpdb:

- query
- prepare
- insert
- update

# Banco de dados

Usar sempre a wpdb:

- query
- prepare
- insert
- update
- get\_var

# Tratando dados para consultas

# Tratando dados para consultas

```
$wpdb->get_var( $wpdb->prepare(
    "
    SELECT foo
    FROM {$wpdb->table}
    WHERE 1=1
        AND bar_id = %d
        AND bar_slug = %s
    ",
    $id, $slug
) );
```



# Uso de Ações e Filtros

# Uso de Ações e Filtros

- Base da construção de plugins no WordPress

# Uso de Ações e Filtros

- Base da construção de plugins no WordPress
- Certificar-se de agendar os eventos e tratar os dados adequadamente

# Implementação de Ações e Filtros

# Implementação de Ações e Filtros

- Oferecer extensibilidade aos dados gerados

# Implementação de Ações e Filtros

- Oferecer extensibilidade aos dados gerados
- Possibilitar a inserção de novos procedimentos à medida que eventos relevantes acontecem

# Implementação de Ações

```
function show_something() {  
    do_action( 'before_show_something' );  
    /* something */  
    do_action( 'after_show_something' );  
}
```

# Implementação de Ações

```
function show_something() {  
    do_action( 'before_show_something' );  
    /* something */  
    do_action( 'after_show_something' );  
}
```

```
function show_something_header() { /* header */ }  
add_action( 'before_show_something', 'show_something_header' );  
  
function show_something_footer() { /* footer */ }  
add_action( 'after_show_something', 'show_something_footer' );
```



# Implementação de Filtros

```
function get_something ( $param ) {  
  
    /* get $something */  
  
    return apply_filters( 'get_something', $something );  
}
```

# Implementação de Filtros

```
function get_something ( $param ) {  
  
    /* get $something */  
  
    return apply_filters( 'get_something', $something );  
}  
  
function parse_something ( $something ) {  
  
    return preg_replace( $regex, $replace, $something );  
}  
add_filter( 'get_something', 'parse_something' );
```

## Implementação de Ações e Filtros

```
<form>
<?php foreach ( $field as $f ) : ?>
    <?php $field = apply_filters( 'display_field', $field ); ?>
    <?php do_action( 'after_display_field', $field ); ?>

    <input
        name="<?php echo $f['name']; ?>"
        type="<?php echo $f['type']; ?>"
        value="<?php echo $f['value']; ?>" />

    <?php do_action( 'after_display_field', $field ); ?>
<?php endforeach; ?>

<?php do_action( 'before_submit' ); ?>
<input type="submit" value="Enviar" />

</form>
```

# Adicionando scripts

# Adicionando scripts

Errado

# Adicionando scripts

Errado

No tema:

```
<html>
<head>
    ...
    <script type="text/javascript" src="script.js" />
</head>
```

# Adicionando scripts

# Adicionando scripts

Errado



# Adicionando scripts

## Errado

```
function custom_scripts() {  
    <script type="text/javascript" src="script.js" />  
    <script type="text/javascript">  
        someCall();  
    </script>  
}  
add_action( 'custom_scripts', 'wp_head' );
```

# Enfileiradores de scripts

# Enfileiradores de scripts

Funções:

# Enfileiradores de scripts

Funções:

- `wp_enqueue_script`

# Enfileiradores de scripts

Funções:

- `wp_enqueue_script`
- `wp_enqueue_style`

# Enfileiradores de scripts

Funções:

- `wp_enqueue_script`
- `wp_enqueue_style`
- `wp_localize_script`

# Enfileiradores de scripts

Funções:

- `wp_enqueue_script`
- `wp_enqueue_style`
- `wp_localize_script`

Hooks:

# Enfileiradores de scripts

Funções:

- `wp_enqueue_script`
- `wp_enqueue_style`
- `wp_localize_script`

Hooks:

- `wp_enqueue_scripts`



# Enfileiradores de scripts

Funções:

- `wp_enqueue_script`
- `wp_enqueue_style`
- `wp_localize_script`

Hooks:

- `wp_enqueue_scripts`
- `admin_enqueue_scripts`

# Incluindo scripts

# Incluindo scripts

```
function custom_scripts() {  
    wp_enqueue_script( 'jquery' );  
    wp_enqueue_script( 'my_script',  
        'my_script.js', array( 'jquery' ) );  
}  
add_action( 'wp_enqueue_scripts', 'custom_scripts' );
```

# Incluindo scripts com variáveis

# Incluindo scripts com variáveis

```
function custom_scripts() {  
    wp_enqueue_script( 'jquery' );  
    wp_enqueue_script( 'my_script',  
        'my_script.js', array( 'jquery' ) );  
    $params = array(  
        'key1' => 'val1',  
        'key2' => 'val2'  
    );  
    wp_localize_script( 'my_plugin', 'params', $params );  
}  
add_action( 'wp_enqueue_scripts', 'custom_scripts' );
```

## Incluindo scripts com variáveis: resultado

```
<script .. jquery.js .. ></script>

<script type="text/javascript">
/*  */
var params = {"key1":"val1","key2":"val2"};
/* ]]&gt; */
&lt;/script&gt;

&lt;script .. my_script.js .. &gt;&lt;/script&gt;</pre></div><div data-bbox="646 922 988 945" data-label="Page-Footer"><img alt="Navigation icons"/></div><div data-bbox="315 962 477 988" data-label="Page-Footer"><p>Vinicius Massuchetto</p></div><div data-bbox="517 962 686 989" data-label="Page-Footer"><p>Plugins no WordPress</p></div>
```



# Internacionalização



# Internacionalização

- Usar funções `__()` e `_e()`

# Internacionalização

- Usar funções `__()` e `_e()`
- Carregar o arquivo MO

# Internacionalização

```
echo __( 'String a ser traduzida', 'plugin_textdomain' );  
  
_e( 'String a ser traduzida', 'plugin_textdomain' );  
  
printf(  
    __( 'String com um valor %s qualquer no meio',  
        'plugin_textdomain' ),  
    $string  
);
```

# Internacionalização

```
function plugin_textdomain() {  
    load_plugin_textdomain( 'plugin_textdomain', false,  
        dirname( plugin_basename( __FILE__ ) ) . '/lang/' );  
}  
add_action( 'plugins_loaded', 'plugin_textdomain' );
```

# Tratamento de Erros



# Tratamento de erros

# Tratamento de erros

- Instanciações da WP\_Error

# Tratamento de erros

- Instanciações da `WP_Error`
- Verificação com `is_wp_error`



# Tratamento de erros

- Instanciações da `WP_Error`
- Verificação com `is_wp_error`
- Utilizar a `wp_die` para morrer elegantemente

# Tratamento de erros

```
function some_func( $param ) {  
  
    $error = false;  
  
    if ( ! empty( $param ) )  
        $error = new WP_Error( 'plugin_name',  
                                sprintf( __( 'Parametro %1$s invalido no arquivo  
%2$s linha %3$s.',  
                                'plugin_textdomain' ),  
                                '$param', __FILE__, __LINE__ ) );  
  
    if ( is_wp_error ( $error ) )  
        wp_die ( $error, 'Argumento invalido.' );  
  
    /* func */  
  
}
```

# Tratamento de erros com objetos

```
class Some_Plugin {  
  
    var $error;  
  
    function Some_Plugin() {  
        $this->error = new WP_Error ( 'plugin_name' );  
  
        /* hooks */  
    }  
}
```

# Adicionando erros com objetos

```
function some_func( $param ) {  
  
    if ( empty( $param ) )  
        $this->error->add( 'plugin_name',  
            sprintf( __( 'Parametro %1$s invalido no arquivo  
%2$s linha %3$s.',  
                'plugin_textdomain' ),  
                '$param', __FILE__, __LINE__ ) );  
  
    /* func */  
  
}
```

# Mostrando erros

```
function dump_errors() {  
    if ( ! $error_codes = $this->error->get_error_codes() )  
        return true;  
  
    foreach ( $error_codes as $code ) {  
        $message = $this->error->get_error_message( $code );  
        $data = $this->error->get_error_data( $code );  
  
        /* display error */  
  
        echo "Error: {$message}\n\t{$data}\n\n";  
    }  
}
```

# Mostrando erros

```
function dump_errors() {  
    if ( ! $error_codes = $this->error->get_error_codes() )  
        return true;  
  
    foreach ( $error_codes as $code ) {  
        $message = $this->error->get_error_message( $code );  
        $data = $this->error->get_error_data( $code );  
  
        /* display error */  
  
        echo "Error: {$message}\n\t{$data}\n\n";  
    }  
}
```

```
add_action( 'shutdown', array( $this, 'dump_errors' ) );
```

# Classes e Funções Úteis



# Manipulação de Dados



# Manipulação de Dados

- `wp_parse_args`

# Manipulação de Dados

- `wp_parse_args`
- `wp_list_filter`

# Formatação

# Formatação

- `is_email`

# Formatação

- `is_email`
- `remove_accents`

# Formatação

- `is_email`
- `remove_accents`
- `sanitize_title`

# Formatação

- `is_email`
- `remove_accents`
- `sanitize_title`
- `sanitize_email`

# Formatação

- `is_email`
- `remove_accents`
- `sanitize_title`
- `sanitize_email`
- `seems_utf8`



# Formatação

- `is_email`
- `remove_accents`
- `sanitize_title`
- `sanitize_email`
- `seems_utf8`
- `zeroise`

# Formatação

- `is_email`
- `remove_accents`
- `sanitize_title`
- `sanitize_email`
- `seems_utf8`
- `zeroise`
- `wptexturize`

# Transients API

# Transients API

- `set_transient`

# Transients API

- `set_transient`
- `get_transient`

# Transients API

- `set_transient`
- `get_transient`
- `delete_transient`

# HTTP API

# HTTP API

- `wp_remote_get`



# HTTP API

- `wp_remote_get`
- `wp_remote_retrieve_body`

# HTTP API

- `wp_remote_get`
- `wp_remote_retrieve_body`
- `wp_remote_retrieve_headers`

# Object Cache

# Object Cache

- `wp_cache_add`

# Object Cache

- `wp_cache_add`
- `wp_cache_set`

# Object Cache

- `wp_cache_add`
- `wp_cache_set`
- `wp_cache_get`

# Object Cache

- `wp_cache_add`
- `wp_cache_set`
- `wp_cache_get`
- `wp_cache_delete`

# Object Cache

- `wp_cache_add`
- `wp_cache_set`
- `wp_cache_get`
- `wp_cache_delete`
- `wp_cache_flush`



# Cron

# Cron

- `wp_schedule_event`

# Cron

- `wp_schedule_event`
- `wp_schedule_single_event`

# Cron

- `wp_schedule_event`
- `wp_schedule_single_event`
- `wp_unschedule_event`

# Cron

- `wp_schedule_event`
- `wp_schedule_single_event`
- `wp_unschedule_event`
- `wp_next_scheduled`

# Classes Úteis

# Classes Úteis

- SimplePie

# Classes Úteis

- SimplePie
- PHPMailer



# Funções Úteis

# Funções Úteis

- `wp_mail`

# Funções Úteis

- `wp_mail`
- `fetch_feed`

# Funções Úteis

- `wp_mail`
- `fetch_feed`
- `human_time_diff`



# readme.txt de um plugin

```
=== Plugin Name ===  
  
== Description ==  
  
== Installation ==  
  
== Frequently Asked Questions ==  
  
== Screenshots ==  
  
== Changelog ==  
  
== Arbitrary section ==
```

## HyperDB

HyperDB is an advanced database class that supports replication, failover, load balancing, and partitioning.

[Download Version](#)

♥ Favorite

[Description](#) [Installation](#) [FAQ](#) [Changelog](#) [Stats](#) [Support](#) [Developers](#)

# Considerações Finais



# Considerações Finais

- Interfaces altamente flexíveis

# Considerações Finais

- Interfaces altamente flexíveis
- Conjunto de ferramentas amplo e disponível

# Considerações Finais

- Interfaces altamente flexíveis
- Conjunto de ferramentas amplo e disponível
- Fácil acesso às informações pelo desenvolvedor

# Considerações Finais

- Interfaces altamente flexíveis
- Conjunto de ferramentas amplo e disponível
- Fácil acesso às informações pelo desenvolvedor
- Não tem desculpa para não codificar com qualidade



# Referências

- Codex: Writing a Plugins  
[http://codex.wordpress.org/Writing\\_a\\_Plugin](http://codex.wordpress.org/Writing_a_Plugin)
- WordPress Answers  
<http://wordpress.stackexchange.com/questions/715/objective-best-practices-for-plugin-development>