



# Plugins de WordPress

Meetup WP Nicaragua – Julio 2016

# Acerca de

- @leogg
- Desarrollador web @monchitonet
- Miembro comunidad @wpnicaragua

# ¿Qué es un plugin?

- Un plugin es un script de PHP que extiende o altera las funcionalidades de WordPress
- Se instalan para agregar una función o grupo de funciones a WordPress

# Plugins populares

- Akismet
- Yoast
- Contact Form 7
- WooCommerce

# Estructura de un plugin

- Se hace de uno o mas archivos PHP
- El archivo principal es un archivo PHP
- Puede (o no) contener otro tipo de archivos (imagenes, archivos de traduccion, etc.)
- Ubicado en un sub directorio dentro de wp-content/plugins

# Antes de comenzar

- Sigue los estándares!

[https://codex.wordpress.org/WordPress\\_Coding\\_Standards](https://codex.wordpress.org/WordPress_Coding_Standards)

[https://codex.wordpress.org/Writing\\_a\\_Plugin](https://codex.wordpress.org/Writing_a_Plugin)

# Creando un plugin

- Crea un archivo php en blanco
- mi-plugin/plugin.php
- *(dentro de wp-content/plugins)*

# Encabezado

- Lo minimo:

```
<?php
```

```
/*
```

```
Plugin Name: Mi Plugin
```

```
*/
```



# Versión completa (1/2)

```
<?php
```

```
/*
```

```
Plugin Name: Mi Plugin
```

```
Plugin URI: http://Pagina_del_plugin
```

```
Description: Descripcion de mi plugin
```

```
Version: 1.0
```

```
...
```

# Versión completa (2/2)

...

Author: Juan Perez

Author URI: [http://Pagina\\_del\\_autor](http://Pagina_del_autor)

License: GPL2

License URI: <https://www.gnu.org/licenses/gpl-2.0.html>

Domain Path: /languages

Text Domain: mi-plugin

\*/

# Más información

- Consulta la documentación completa en:  
[https://codex.wordpress.org/es:API\\_Plugin](https://codex.wordpress.org/es:API_Plugin)

# Shortcodes

- Función para agregar shortcodes:

```
add_shortcode( $etiqueta, $funcion_a_llamar );
```

# Shortcodes

```
function linea_shortcode() {  
    return '<hr/>';  
}  
  
add_shortcode( 'linea', 'linea_shortcode' );
```

# Hooks

- **Hooks** es un término genérico en WordPress que se refiere a inyectar código propio y agregar comportamientos o cambiar el funcionamiento por defecto de WordPress
- Dos tipos de hooks; *Action Hooks* y *Filter Hooks*

# Action Hooks

- Una acción te permite ejecutar tus propias funciones en un lugar específico dentro de la línea de tiempo de ejecución de WordPress, Plugin o Tema para que puedas agregar nuevas funcionalidades o personalizaciones
- O, dicho de modo muy simple, *“Cuando llegues a este punto haz esto”*

# Action Hooks

```
add_action( $hook, $funcion );
```

- `add_action` es la función nativa de WordPress para agregar una acción
- `$hook` es el nombre de la acción en donde inyectaremos nuestro código
- `$funcion` es el nombre de la función que deseamos ejecutar



# Action Hooks

- Usaremos el hook `wp_head` que se utiliza para agregar informacion en el encabezado de la pagina, entre las etiquetas `<head>`

# Action Hooks

```
add_action( 'wp_head',  
'custom_add_meta_description' );
```

```
function custom_add_meta_description(){  
echo "<meta name='description' content='Esta página  
es generada usando un hook para WordPress' />\n\r";  
}
```

# Filter Hooks

- Un filtro te permite manipular textos y contenidos antes de que sean utilizados... por ejemplo, modificar el título de un artículo antes de ser mostrado en pantalla
- O, dicho de modo simple, *“Dame esta información y cámbiala por esta otra”*

# Filter Hooks

```
add_action( $hook, $funcion );
```

- `add_action` es la función nativa de WordPress para agregar una acción
- `$hook` es el nombre de la acción en donde inyectaremos nuestro código
- `$funcion` es el nombre de la función que deseamos ejecutar

# Filter Hooks

- Usaremos la funcion wp\_title que muestra el titulo de nuestro sitio

<title>Mi sitio web | Just another WordPress  
site</title>

# Filter Hooks

```
add_filter( 'wp_title', 'custom_change_site_title' );  
  
function custom_change_site_title(){  
    return $title . " | Un sitio de la comunidad WP  
    Nicaragua";  
}
```

# Incluir CSS/JS

```
wp_register_script();
```

- Registra un script que se cargara posteriormente

```
wp_enqueue_script();
```

- Forma segura de cargar un script en WordPress

# Incluir CSS/JS

```
function custom_init_method() {  
  
    wp_deregister_script( 'jquery' );  
    wp_register_script( 'jquery',  
'http://ajax.googleapis.com/ajax/libs/jquery/1.4/jquery.min.js');  
}  
  
add_action('init', 'custom_init_method');
```



# Registrar un script (1/2)

```
function custom_scripts_basic()  
{  
    // Registrar el script para un plugin:  
    wp_register_script( 'custom-script',  
plugins_url( '/js/custom-script.js', __FILE__ ) );  
    // Registrar el script para un tema:  
    wp_register_script( 'custom-script',  
get_template_directory_uri() . '/js/custom-script.js' );  
}
```

# Registrar un script (2/2)

...

```
wp_enqueue_script( 'custom-script' );  
}  
  
add_action( 'wp_enqueue_scripts',  
'custom_scripts_basic' );
```