

Plugins para Wordpress

Álvaro González Sotillo

30 de enero de 2025

Índice

1. ¿Qué es un <i>plugin</i>	1
2. <i>Hooks</i>	1
3. <i>Plugins</i>	2
4. Ejercicios	3
5. Referencias	4

1. ¿Qué es un *plugin*

- Un *plugin* permite añadir funcionalidad a un *software*
- El *software* debe tener soporte para ello
- En el caso de Wordpress
 - Se utiliza PHP
 - Se extiende con *hooks* de acción y de filtro
 - Expone un conjunto de **funciones (API)**

2. *Hooks*

- Similares a *listeners* de Javascript, o *triggers* de Oracle
- Registran funciones que se llamarán en momentos adecuados

[Saber más...](#)

2.1. *Hooks* de acción

- Llamados antes de realizar una acción:
 - `save_post`, `get_footer`, `setup_theme` ([lista completa](#)) ([otra lista](#))
 - Internamente, Wordpress los llama con `do_action` o `do_action_ref_array`
- Una acción
 - Recibe información sobre el objeto que actúa
 - Modifica ese objeto, u otra parte de Wordpress, y no devuelve nada
 - Es como un *procedure* de Oracle

2.2. Uso de *hooks* de acción

```
function envia_cabecera_http_personalizada() {
    header('X-Custom-Header: El_Valores_Personalizado');
}

add_action('send_headers', 'envia_cabecera_http_personalizada');
```

2.3. *Hooks* de filtro

- Sirven para modificar datos devueltos por algunas funciones.
 - `get_pages`, `body_class`, `sanitize_title`, `get_comment_excerpt` ([lista completa](#))
 - Internamente, WordPress los llama con `apply_filters`

2.4. Uso de *hooks* de filtro

```
function pon_prefijo_a_titulo($title) {
    if (is_single()) {
        $title = 'Este es mi blog: ' . $title;
    }
    return $title;
}

add_filter('the_title', 'pon_prefijo_a_titulo');
```

3. *Plugins*

- Hay varias [formas de hacerlos](#)
- Usaremos
 - Un directorio en `/wp-content/plugins`
 - Dentro de ese directorio, un fichero `.php` con el nombre del directorio

3.1. Cabecera del *plugin*

- En un comentario de bloque `/* ... */`, se incluyen [datos del plugin](#)
- Nombre del plugin, sitio de actualización, página web, autor, dependencias...

```
/**
 * Plugin Name:      Plugin Name
 * Plugin URI:       https://example.com/plugin-name
 * Description:      Description of the plugin.
 * Version:          1.0.0
 * Requires at least: 5.2
 * Requires PHP:     7.2
 * Author:           Your Name
 * Author URI:       https://example.com
 * Text Domain:      plugin-slug
 * License:          GPL v2 or later
 * License URI:      http://www.gnu.org/licenses/gpl-2.0.txt
 * Update URI:       https://example.com/my-plugin/
 * Requires Plugins: my-plugin, yet-another-plugin
 */
```

3.2. Fichero principal del *plugin*

- Fichero `.php` dentro del directorio del plugin, del mismo nombre
- Registra las acciones y filtros necesarios
- Opcionalmente

- Define las funciones llamadas por los *hooks*
- Incluye otros ficheros php (con `plugin_dir_path`)
- Registrar funciones para activar y desactivar el plugin (con `register_activation_hook`)

```
function miplugin_agrega_una_pagina_al_menu_admin() {  
  
    add_menu_page(  
        'Página del plugin', // Title of the page  
        'Página informativa del plugin de disclaimer', // Text to show on the menu link  
        'manage_options', // Capability requirement to see the link  
        plugin_dir_path(__FILE__) . 'pagina-admin.php' // se añade a /wp-admin  
    );  
  
    add_action( 'admin_menu', 'miplugin_agrega_una_pagina_al_menu_admin' );  
}
```

3.3. Normas básicas a seguir

- Las funciones y constantes son globales: hay que ponerles un prefijo para que no *colisionen*
- Sin efectos laterales en los filtros: son funciones

Saber más...

4. Ejercicios

4.1. Crea un *plugin* vacío

- Crea un plugin vacío
 - Elige un nombre (ejemplo: `iaw_test_plugin`)
 - Elige un prefijo para tus funciones y opciones (ejemplo: `iaw_test_plugin`)
 - Incluye comentario con los datos del *plugin*
- El *plugin* debe aparecer al visitar `http://SERVIDOR/wordpress/wp-admin/plugins.php`

4.2. *widget* en el *dashboard*

- Añade un *widget* que simplemente diga "¡Hola, mundo!"
 - Acción `wp_dashboard_setup`
 - Función `wp_add_dashboard_widget`

4.3. Página nueva en el menú de administración

- Acción `admin_menu`
- Función `add_menu_page`
- La página no tendrá funcionalidad: solo mostrará un texto *lorem ipsum*

4.4. Filtro de contenido

- Filtro `the_content`
- Quita todas las ocurrencias de la pseudopalabra WTF, las cambia por ###

4.5. Página de *settings*

- En vez de ###, se podrá censurar la palabra prohibida con una cadena configurable
- Acción `admin_init`
 - Registrar una opción con la función `register_setting`
 - El nombre de la opción:
- Acción `admin_menu`
 - Función `add_options_page`
- En la función de opciones de página
 - Usar `settings_fields` dentro del formulario
 - Los campos `input` tendrán los nombres definidos en `register_settings`
 - El valor actual se puede conseguir con `get_option`

```
echo '<form method="post" action="options.php">';
settings_fields( 'wpdocs-plugin-settings-group' );
$value = get_option("...");
echo '<input type="..." name="..." value="$value">'
...
```

4.6. Más posibilidades

- Haz que el filtro solo se active en chrome, pero no en firefox
 - `$is_chrome, $is_gecko`
- Haz que el filtro se pueda activar o desactivar
 - Con otro *setting*
 - En el *dashboard* se verá si está activado, y qué palabra se utiliza para censurar.
- Empaqueta el plugin y **publicalo**

5. Referencias

- Formatos:
 - [Transparencias](#)
 - [PDF](#)
 - [Página web](#)
 - [EPUB](#)
- Creado con:
 - [Emacs](#)
 - [org-re-reveal](#)
 - [Latex](#)
- Alojado en [Github](#)

<https://www.wpbeginner.com/wp-tutorials/how-to-create-a-wordpress-plugin/> <https://www.hostinger.es/tutoriales/tutorial-wordpress-plugins-crear-plugin-wordpress> <https://developer.wordpress.org/apis/>