

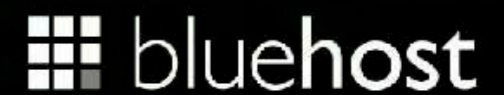


**WORDCAMP**  
B O G O T Á  
2 0 1 8



# JORGE TORRES WILLINGTON VEGA

Introducción al desarrollo de plugins  
para WordPress



# ¿Quiénes somos?



**Jorge A. Torres**

[jorge@prospress.com](mailto:jorge@prospress.com)



**Willington Vega**

[wvega@wvega.com](mailto:wvega@wvega.com)

# INTRODUCCIÓN



# ¿Qué es un *plugin*?



`/wp-content/plugins/`

# ¿Por qué crear un *plugin*?

No existe ya un *plugin* que haga lo que queremos...

... o no está siendo mantenido o no es compatible con las últimas versiones de WordPress

This plugin hasn't been tested with the latest 3 major releases of WordPress. It may no longer be maintained or supported and may have compatibility issues when used with more recent versions of WordPress.



## Real Estate Agent

By [RealtyBaron](#)

Download

A WordPress Plugin that conveniently displays an agent's AgentRank data — profile, sales, reviews, forecasts, etc — from RealtyBaron.

Last updated: 8 years ago

Active installations: Fewer than 10

# ¿Por qué crear un *plugin*?

- No tocar el *core* (regla “cardinal” de WP).
- No modificar o sobrecargar el tema.
- Reutilizar código en varios sitios web.
- Extender o cambiar el comportamiento del *core* o de otros *plugins*.
- Fama (y quizás dinero):
  - La oportunidad de ayudar a una comunidad de millones de usuarios.
  - El repositorio oficial de WordPress tiene más de 56,000 *plugins* gratuitos.

# ¿Qué puede hacer un *plugin*?

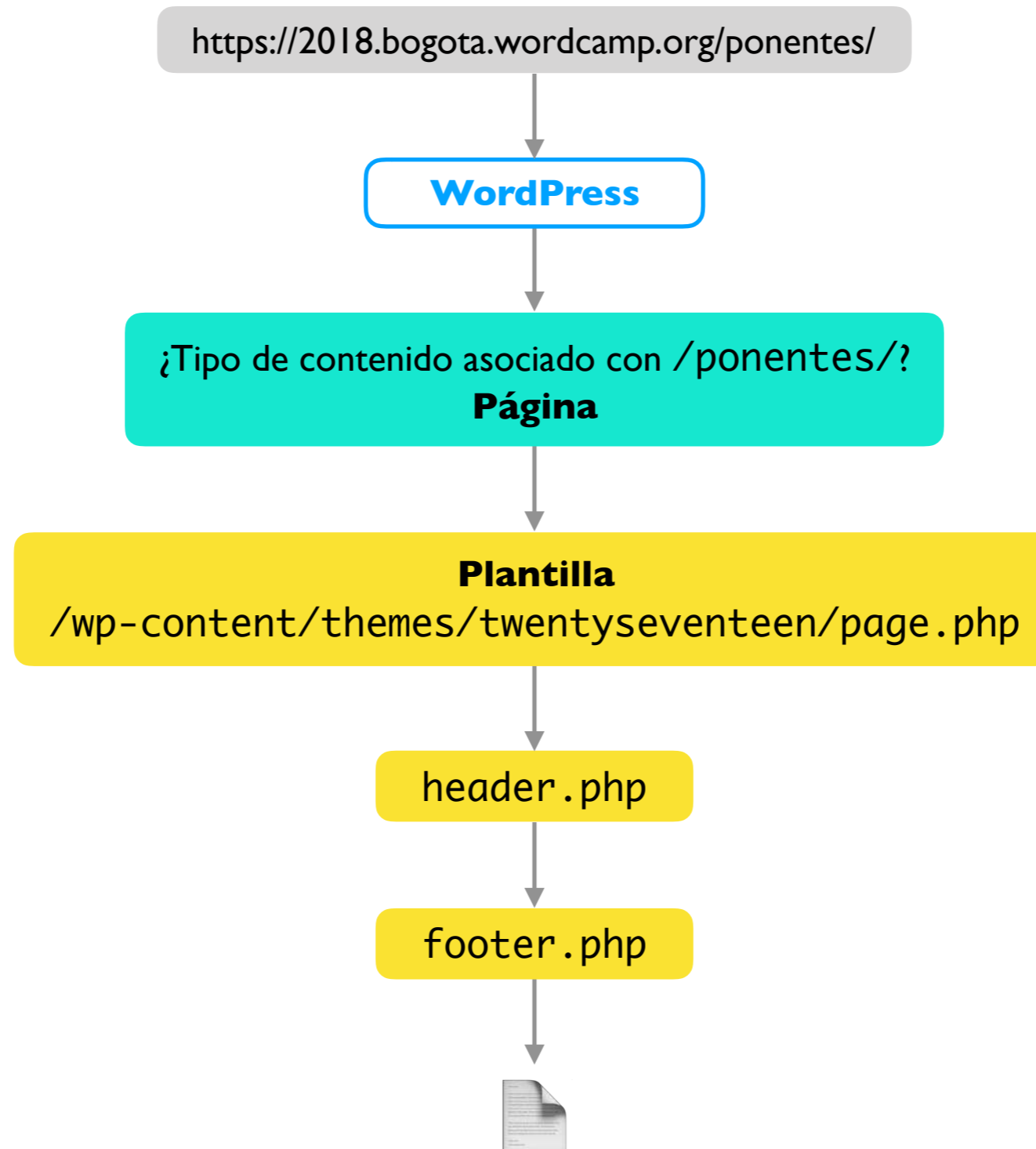
- Agregar nuevos tamaños de *thumbnails*.
- Registrar nuevos *shortcodes* para uso en las entradas.
- Personalizar los correos enviados por WordPress.
- Insertar contenido o scripts en los *posts* del sitio.
- Implementar soluciones completas como CMS, e-commerce, *help desk*.
- ... las posibilidades son innumerables.

# EXTENDIENDO WORDPRESS



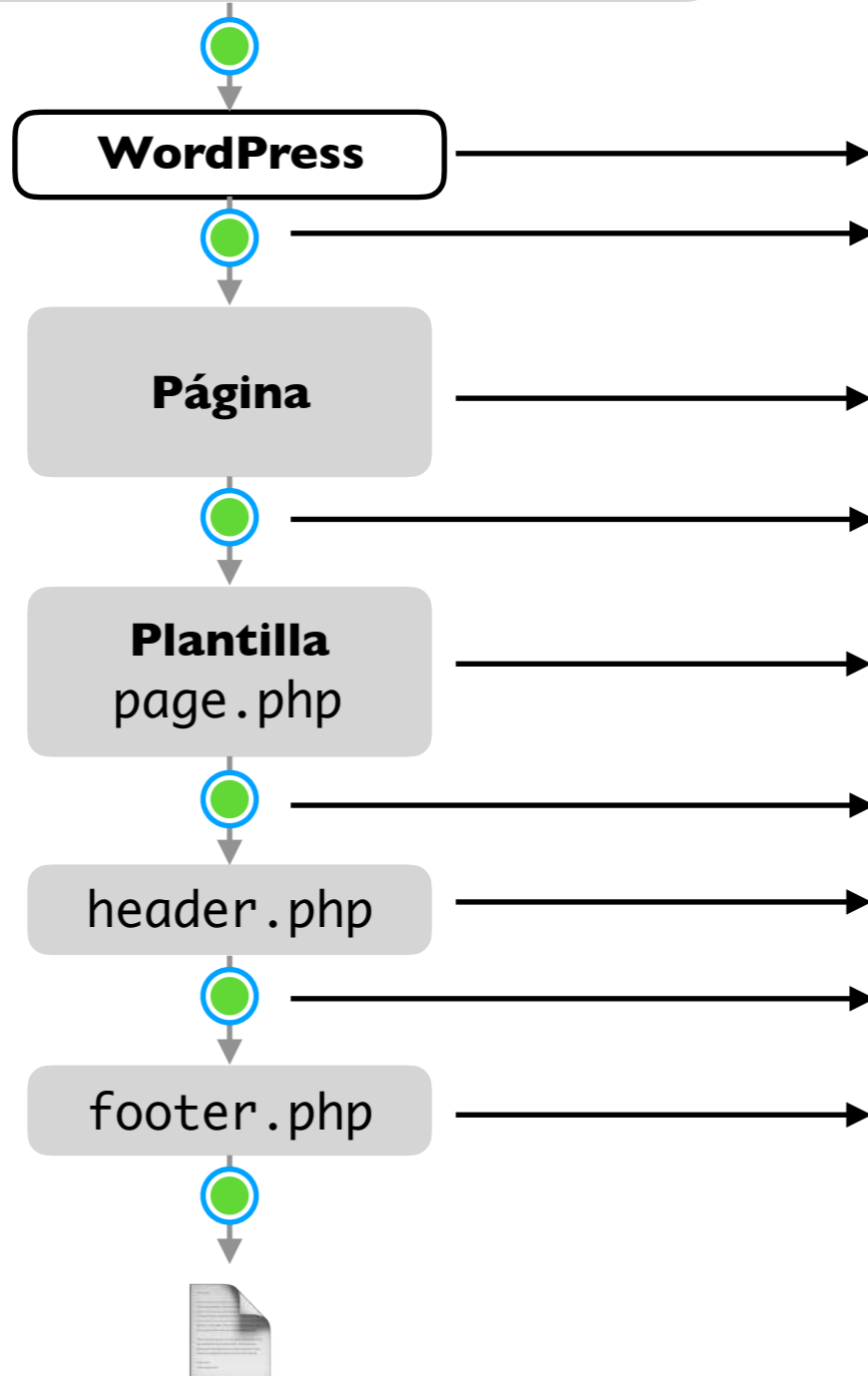


# Una petición típica de WP



# Extendiendo WordPress

<https://2018.bogota.wordcamp.org/ponentes/>



Los *hooks* son puntos de extensión que ofrece WP para cambiar cómo se comporta en ciertos momentos específicos.

# *Hooks: actions y filters*

Hay dos tipos de *hooks*: *actions* y *filters*.

## ***Actions:***

- Sirven para agregar o cambiar funcionalidad permitiendo llamar a una función en un momento específico de la ejecución de WP:
  - luego de guardar una entrada,
  - al inicio de una petición,
  - al cargar el *footer* del sitio,
  - etc.

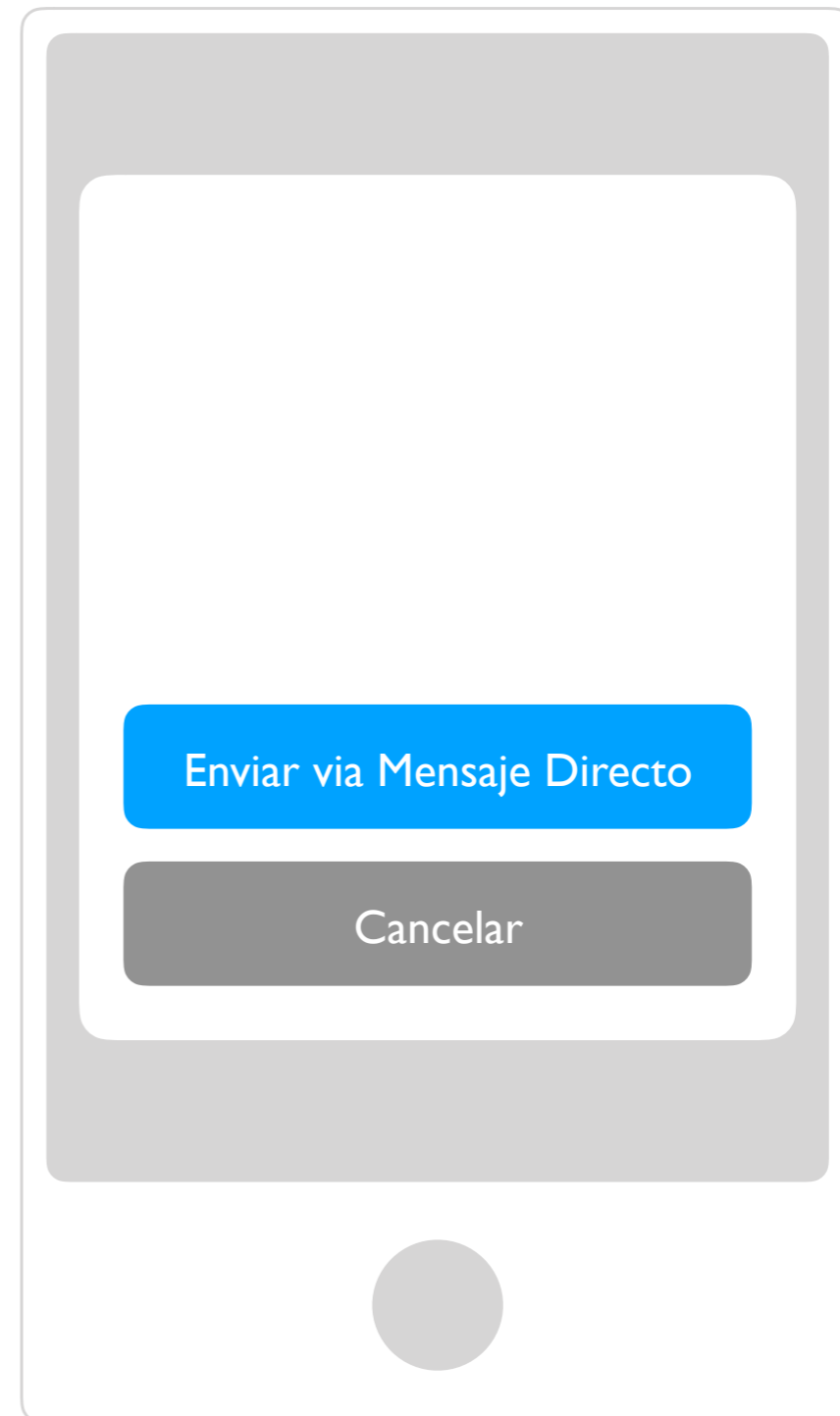
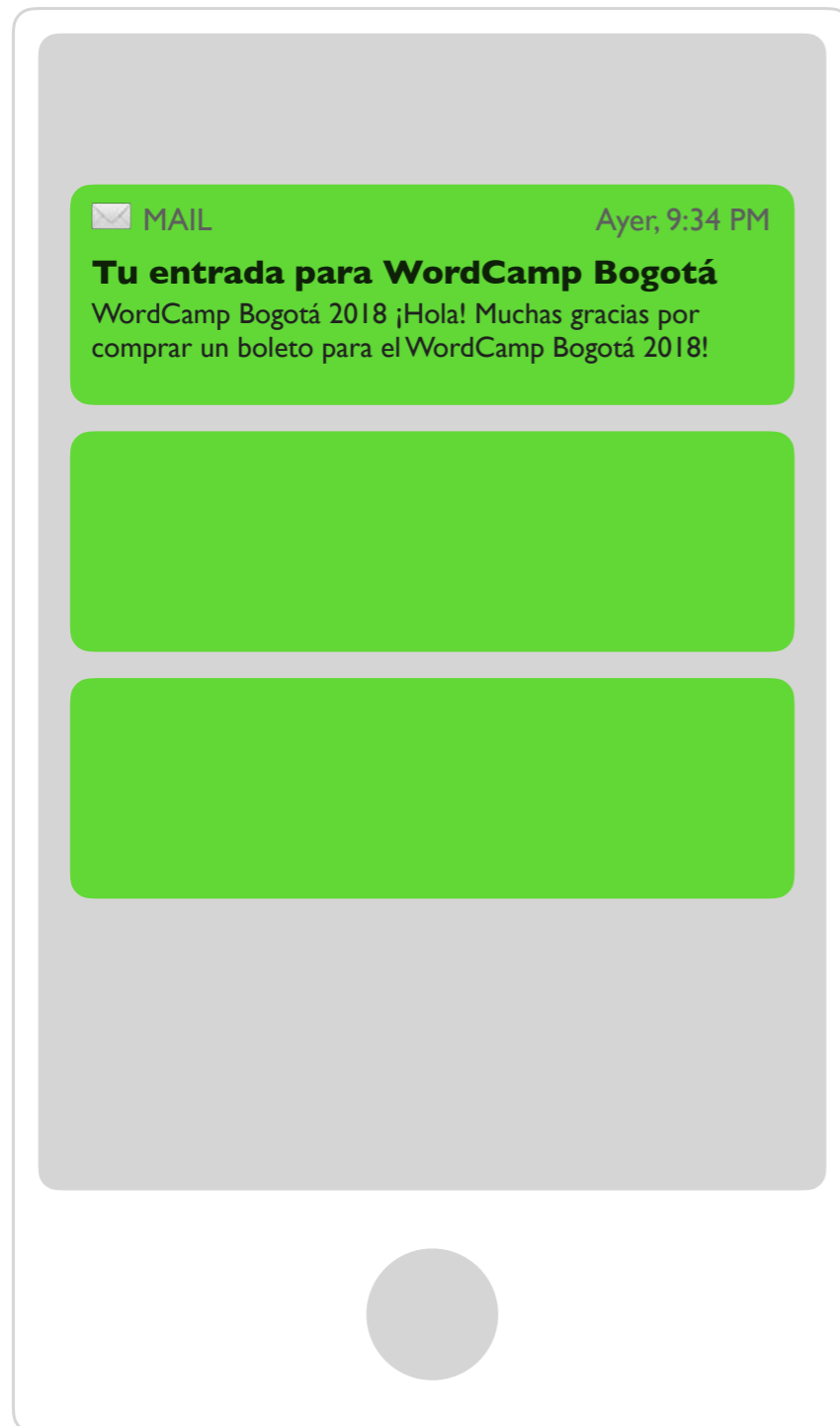
# *Hooks: actions y filters*

Hay dos tipos de *hooks: actions y filters*.

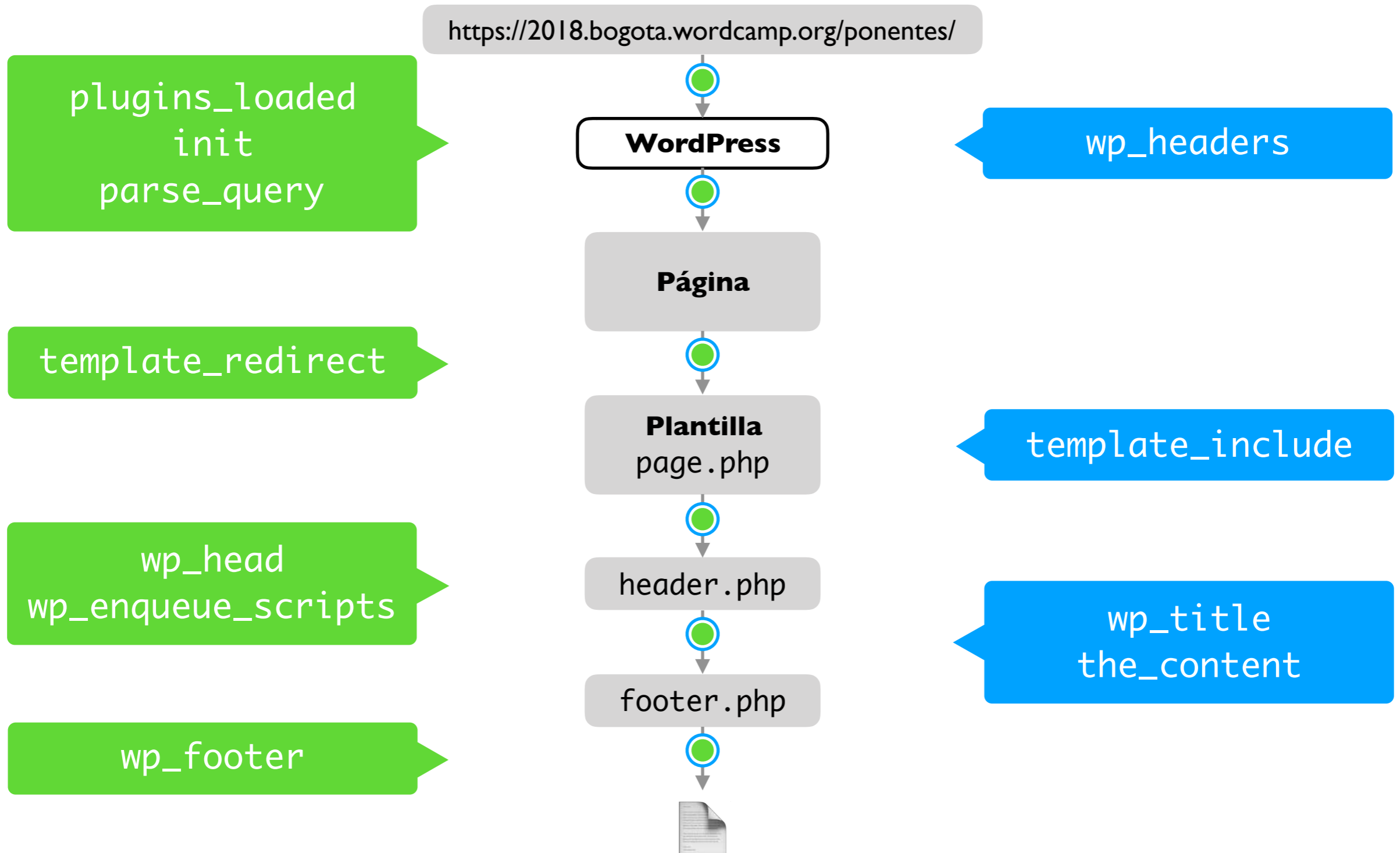
## ***Filters:***

- Permiten alterar los datos y el contenido con el que trabajan otras funciones de WP:
  - Cambiar el valor de una opción obtenida de la base de datos.
  - Agregar contenido adicional a la entrada.
  - Modificar las variables de la solicitud SQL antes de realizarla.

# Hooks: actions y filters



# Una petición típica de WWP



# Actions

```
<!DOCTYPE html>
<html class="no-js">
  <head>
    <meta http-equiv="Content-type" content="text/html; charset=utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0,
    <title>WordPress Medellín &#8211; Comunidad WordPress de Medellín
    <link rel="stylesheet" href="http://.../rams/style.css" type="text/css">
    <link rel="stylesheet" href="http://.../rams-child/style.css" type="text/css">
    <script type="text/javascript" src="http://.../jquery.js"></script>
  </head>
```

# Actions

```
<!DOCTYPE html>
<html class="no-js">
  <head>
    <meta http-equiv="Content-type" content="text/html; charset=utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0,">
    <title>WordPress Medellín &#8211; Comunidad WordPress de Medellín
    wp_head
    <link rel="stylesheet" href="http://.../rams/style.css" type="text/css">
    <link rel="stylesheet" href="http://.../rams-child/style.css" type="text/css">
    <script type="text/javascript" src="http://.../jquery.js"></script>
  </head>
```



# Actions

```
add_action( string $tag, callable $function_to_add, int $priority = 10,  
int $accepted_args = 1 )
```

# Actions

```
add_action( 'wp_head', 'myplugin_print_meta_tags' );  
  
function myplugin_print_meta_tags() {  
    echo '<meta property="og:type" content="article" />';  
}
```

# Filters

```
<!DOCTYPE html>
<html class="no-js">
  <head>
    <meta http-equiv="Content-type" content="text/html; charset=utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0,
    <title>WordPress Medellín &#8211; Comunidad WordPress de Medellín
    <link rel="stylesheet" href="http://.../rams/style.css" type="text/css">
    <link rel="stylesheet" href="http://.../rams-child/style.css" type="text/css">
    <script type="text/javascript" src="http://.../jquery.js"></script>
  </head>
```

# Filters

```
<!DOCTYPE html>
<html class="no-js">
  <head>
    <meta http-equiv="Content-type" content="text/html; charset=utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0,
  <title> wp_title WordPress Medellín &#8211; Comunidad WordPressera de
  <link rel="stylesheet" href="http://.../rams/style.css" type="text/cs
  <link rel="stylesheet" href="http://.../rams-child/style.css" type="t
  <script type="text/javascript" src="http://.../jquery.js"></script>
</head>
```

# Filters

```
add_filter( string $tag, callable $function_to_add, int $priority = 10,  
int $accepted_args = 1 )
```

# Filters

```
add_filter( 'wp_title', 'myplugin_halloween_title' );  
  
function myplugin_halloween_title( $title ) {  
    return '🎃' . $title;  
}
```

# TALLER PRÁCTICO





**¿Qué vamos a construir?**



# LISTA DE CHEQUEO

1)

2)

3)

4)

# 0. Crear el archivo del *plugin*.

```
<?php
/**
 * Plugin Name: WordCamp Bogotá 2018
 * Description: Plugin de prueba para el Taller de introducción al
desarrollo de plugins para WordPress.
 * Version: 1.0
 */
```

 wp-content/plugins/

wcbog2018/wcbog2018.php

wcbog2018.php

# 0. Crear el archivo del *plugin*.

```
<?php
/**
 * Plugin Name: WordCamp Bogotá 2018
 * Description: Plugin de prueba para el Taller de introducción al
desarrollo de plugins para WordPress.
 * Version: 1.0
 */
```

 wp-content/plugins/

wcbog2018/wcbog2018.php

wcbog2018.php

# I. Agregar un filtro para modificar el contenido del post.

```
function wcbog2018_add_like_box_to_content( $content ) {  
    return $content;  
}  
add_filter( 'the_content', 'wcbog2018_add_like_box_to_content' );
```

# I. Agregar un filtro para modificar el contenido del post.

```
function wcbog2018_add_like_box_to_content( $content ) {  
    if ( ! is_singular() ) {  
        return $content;  
    }  
  
    $like_box = '';  
    $like_box .= '<p>';  
    $like_box .= '-- Mi like box --';  
    $like_box .= '</p>';  
  
    return $like_box . $content;  
}  
add_filter( 'the_content', 'wcbog2018_add_like_box_to_content' );
```

# I. Agregar un filtro para modificar el contenido del post.

```
function wcbog2018_add_like_box_to_content( $content ) {  
    if ( ! is_singular() ) {  
        return $content;  
    }  
    is_singular( array $post_types )  
$1 Retorna true cuando un post individual se está mostrando.  
$1  
$1 Si se incluye el parámetro opcional $post_types, entonces  
$1 solo retorna true cuando el tipo del post es uno de los  
tipos de posts especificados.  
re  
}  
add_filter( 'the_content', 'wcbog2018_add_like_box_to_content' );
```

## 2. Reemplazar el mensaje por un *link* que funcionará como botón *Like*.

```
function wcbog2018_add_like_box_to_content( $content ) {  
    if ( ! is_singular() ) {  
        return $content;  
    }  
  
    $like_box = '';  
    $like_box .= '<p>';  
    $like_box .= wcbog2018_generate_button();  
    $like_box .= '</p>';  
  
    return $like_box . $content;  
}
```

## 2. Reemplazar el mensaje por un *link* que funcionará como botón *Like*.

```
function wcbog2018_generate_button() {  
    $button = '';  
    $button .= '<a href="" . add_query_arg( 'wcbog2018-like', '1' ) .  
'">Like</a>';  
  
    return $button;  
}
```



## 2. Reemplazar el mensaje por un *link* que funcionará como botón *Like*.

```
function wcbog2018_generate_button() {  
    $button = '';  
    $button .= '<a href="" . add_query_arg( 'wcbog2018-like', '1' ) .  
    '>Like</a>';  
    return $button;  
}
```

```
add_query_arg( string $param, string $value, string $uri )
```

Retorna la URL que resulta de agregar el parámetro `$param` con value `$value` a la URL `$uri`. Si `$uri` no es especificado, entonces agrega `$param` a la URL de la página actual.

Por ejemplo:

```
add_query_arg( 'foo', 'bar', 'https://example.org' );
```

Devuelve `https://example.org?foo=bar`

### 3. Mostrar un mensaje para indicar el número de *likes* que tiene un *post*, sin calcular la cantidad.

```
function wcbog2018_add_like_box_to_content( $content ) {
    if ( ! is_singular() ) {
        return $content;
    }

    $like_box = '';
    $like_box .= '<p>';
    $like_box .= wcbog2018_generate_button();
    $like_box .= ' / ';
    $like_box .= wcbog2018_generate_like_count();
    $like_box .= '</p>';

    return $like_box . $content;
}

function wcbog2018_generate_like_count() {
    return 'Este post tiene <b>0</b> likes';
}
```

## 4. Calcular el número de *likes* que tiene un *post*.

```
function wcbog2018_generate_like_count() {  
    $likes = get_post_meta( get_the_ID(), '_wcbog2018_liked_by' );  
    return 'Este post tiene <b>' . count( $likes ) . '</b> likes';  
}
```

```
add_post_meta( $post_id, '_wcbog2018_liked_by', $user_id );
```

## 4. Calcular el número de *likes* que tiene un *post*.

```
function wcbog2018_generate_like_count() {  
    $likes = get_post_meta( get_the_ID(), '_wcbog2018_liked_by' );  
    return 'Este post tiene <b>' . count( $likes ) . '</b> likes';  
}
```

```
get_post_meta( int $post_id, string $key = '', bool $single = false )
```

```
add_post_meta( $post_id, '_wcbog2018_liked_by', $user_id );
```

```
add_post_meta( int $post_id, string $meta_key, $meta_value, bool $unique = false )
```

## 5. Registrar en la base de datos cuando un usuario da *like*.

```
function wcbog2018_maybe_save_like() {  
}  
add_action( 'wp', 'wcbog2018_maybe_save_like' );
```

## 5. Registrar en la base de datos cuando un usuario da *like*.

```
function wcbog2018_maybe_save_like() {
    if ( ! is_user_logged_in() ) {
        return;
    }

    if ( empty( $_GET['wcbog2018-like'] ) ) {
        return;
    }
}
add_action( 'wp', 'wcbog2018_maybe_save_like' );
```

## 5. Registrar en la base de datos cuando un usuario da *like*.

```
function wcbog2018_maybe_save_like() {
    if ( ! is_user_logged_in() ) {
        return;
    }

    if ( empty( $_GET['wcbog2018-like'] ) ) {
        return;
    }

    $post_id = get_the_ID();
    $user_id = get_current_user_id();
    $likes    = get_post_meta( $post_id, '_wcbog2018_liked_by' );

    if ( ! in_array( $user_id, $likes ) ) {
        add_post_meta( $post_id, '_wcbog2018_liked_by', $user_id );
    }
}

add_action( 'wp', 'wcbog2018_maybe_save_like' );
```

## 5. Registrar en la base de datos cuando un usuario da *like*.

```
function wcbog2018_maybe_save_like() {
    if ( ! is_user_logged_in() ) {
        return;
    }

    if ( empty( $_GET['wcbog2018-like'] ) ) {
        get_post_meta( int $post_id, string $key = '', bool $single = false )

        $post_id = get_the_ID();
        $user_id = get_current_user_id();
        $likes   = get_post_meta( $post_id, '_wcbog2018_liked_by' );

        if ( ! in_array( $user_id, $likes ) ) {
            add_post_meta( $post_id, '_wcbog2018_liked_by', $user_id );
        }
    }
}
```

```
add_post_meta( int $post_id, string $meta_key, $meta_value, bool $unique = false )
```



## 6. Mostrar el número de *likes* utilizando un mensaje gramaticalmente correcto.

```
function wcbog2018_generate_like_count() {
    $likes = get_post_meta( get_the_ID(), '_wcbog2018_liked_by' );
    $no_likes = count( $likes );

    if ( 1 === $no_likes ) {
        return __( 'Este post tiene <b>1</b> like', 'wcbog2018' );
    } else {
        return sprintf( __( 'Este post tiene <b>%d</b> likes',
'wcbog2018' ), $no_likes );
    }
}
```

## 6. Mostrar el número de *likes* utilizando un mensaje gramaticalmente correcto.

```
function wcbog2018_generate_like_count() {
    $likes = get_post_meta( get_the_ID(), '_wcbog2018_liked_by' );
    $no_likes = count( $likes );

    if ( 1 === $no_likes ) {
        return __( 'Este post tiene <b>1</b> like', 'wcbog2018' );
    } else {
        return sprintf( __( 'Este post tiene <b>%d</b> likes',
'wcbog2018' ), $no_likes );
    }
}
```

`__( string $text, string $domain )`

## 6. Mostrar el número de *likes* utilizando un mensaje gramaticalmente correcto.

```
function wcbog2018_generate_like_count() {
    $likes = get_post_meta( get_the_ID(), '_wcbog2018_liked_by' );
    $no_likes = count( $likes );

    return sprintf(
        _n(
            'Este post tiene <b>%d</b> like',
            'Este post tiene <b>%d</b> likes',
            $no_likes,
            'wcbog2018'
        ),
        $no_likes
    );
}
```

## 6. Mostrar el número de *likes* utilizando un mensaje gramaticalmente correcto.

```
function wcbog2018_generate_like_count() {
    $likes = get_post_meta( get_the_ID(), '_wcbog2018_liked_by' );
    $no_likes = count( $likes );

    return sprintf(
        _n(
            'Este post tiene <b>%d</b> like',
            'Este post tiene <b>%d</b> likes',
            $no_likes,
            'wcbog2018'
        ),
        _n( string $single, string $plural, int $number, string $domain )
    )
}
```

## 7. Mostrar el botón *like* solo para usuarios registrados.

```
function wcbog2018_generate_button() {
    if ( ! is_user_logged_in() ) {
        return '';
    }

    $button = '';
    $button .= '<a href="' . add_query_arg( 'wcbog2018-like', '1' ) .
'">';
    $button .= __( 'Like', 'wcbog2018' );
    $button .= '</a>';

    return $button;
}
```

## 8. Reemplazar el botón por un mensaje si el usuario actual ya dio *like*.

```
function wcbog2018_generate_button() {
    [...]
    $user_id = get_current_user_id();
    $likes    = get_post_meta( get_the_ID(), '_wcbog2018_liked_by' );

    $button = '';

    if ( in_array( $user_id, $likes ) ) {
        $button .= __( 'Ya diste like a este post', 'wcbog2018' );
    } else {
        $button .= '<a href="' . add_query_arg( 'wcbog2018-like', '1' ) .
'">';
        $button .= __( 'Like', 'wcbog2018' );
        $button .= '</a>';
    }

    return $button;
}
```

# GDPR

## General Data Protection Regulation



# Registrar *Data Exporters* y *Data Erasers*

```
add_filter(  
    'wp_privacy_personal_data_exporters',  
    'wcbog2018_register_personal_data_exporters'  
);
```

```
add_filter(  
    'wp_privacy_personal_data_erasers',  
    'wcbog2018_register_personal_data_erasers'  
);
```



# Registrar *Data Exporters* y *Data Erasers*

```
function wcbog2018_register_personal_data_exporters( $exporters ) {  
    $exporters['wordcamp-bogota-2018'] = array(  
        'exporter_friendly_name' => __( 'WordCamp Bogotá 2018', 'wcbog2018' ),  
        'callback'               => 'wcbog2018_export_personal_data',  
    );  
  
    return $exporters;  
}
```

# Registrar *Data Exporters* y *Data Erasers*

```
function wcbog2018_export_personal_data( $email_address, $page = 1 ) {
    return array(
        'data' => [
            [
                'group_id'    => 'wcbog2018-liked-entries',
                'group_label' => __( 'Publicaciones Favoritas', 'wcbog2018' ),
                'item_id'     => 1,
                'data'        => [
                    [
                        'name' => __( 'Título', 'wcbog2018' ),
                        'value' => 'Hello world!',
                    ],
                ],
            ],
        ],
        'done' => true,
    );
}
```

# Recursos Adicionales

- Otras formas de extensión
  - Shortcodes
  - Widgets
- Documentación
  - Developer Resources
  - Plugin Developer Handbook
- Plugins Útiles
  - Query Monitor
  - AIR Adminer
  - Rewrite Rules Inspector
  - WP Mail Logging



**¿Preguntas?**

The image features a vibrant teal background filled with numerous small, colorful confetti pieces in shades of pink, orange, yellow, and blue. Soft, out-of-focus light spots (bokeh) are scattered across the scene, creating a festive and celebratory atmosphere. In the center, the word "GRACIAS" is prominently displayed within a white rectangular box.

**GRACIAS**

# Imágenes gracias a

Luca Bravo en Unsplash

Thought Catalog en Unsplash

User 27707 en pixabay

rawpixel en pixabay

Ricardo Gomez Angel en Unsplash

Suzy Hazelwood en Pexels

Andrew Worley en Unsplash

Camylla Battani en Unsplash

Jason Leung en Unsplash

# Introducción al desarrollo de *plugins* para WordPress

