

# WORDPRESS DESACOPLADO.

## « MUEVE TU WORDPRESS AL ~~FUTURO~~ PRESENTE »

---

*Carlos Asín  
Joaquín Ruiz*

*WordCamp Zaragoza 2020 #WCZGZ*



<2020/>  
WORDCAMP  
ZARAGOZA

# SOBRE NOSOTROS

---

*Joaquín Ruiz*

CTO en Flat 101

Ingeniero Superior en Informática

Especializado en Internet : Web-Apps, optimización de Sistemas e eCommerce

tw://@JokiRuizLite <https://jokiruiz.com>

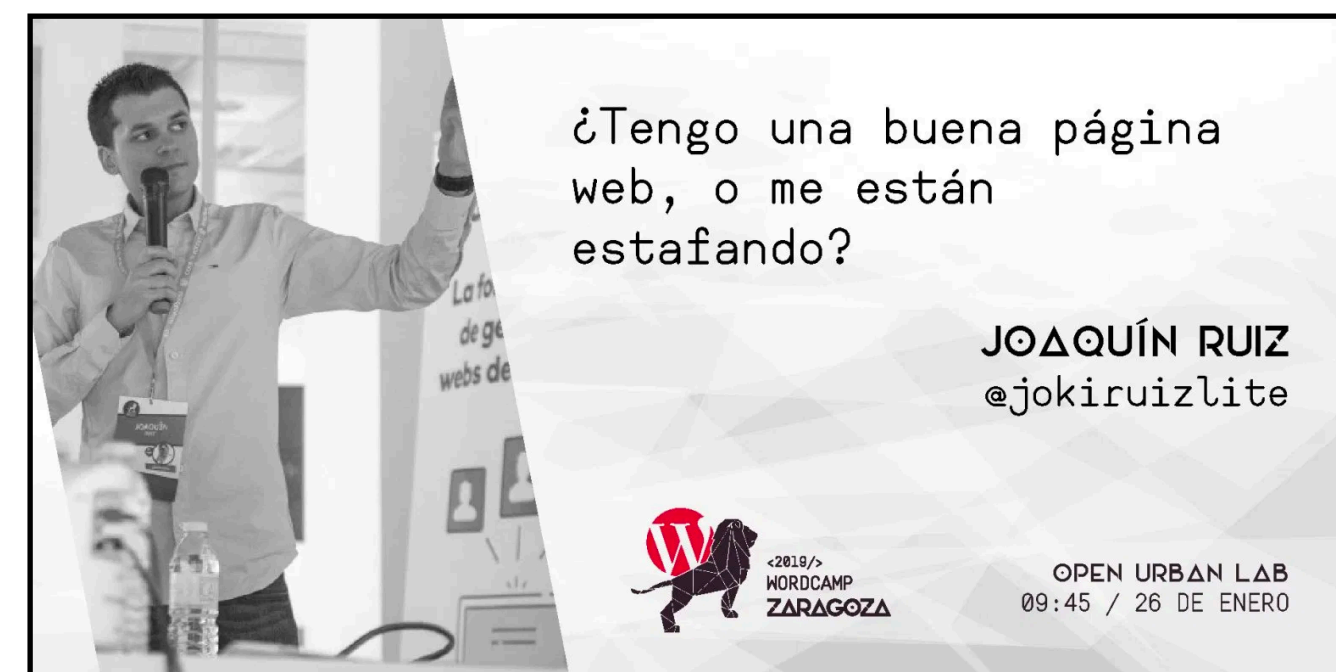
*Carlos Asín*

Tech Lead en Afición360

Ingeniero Superior en Informática

Especializado en Internet : PWA, SPA, Nodejs, sistemas de ticketing

tw://@CarlosAs1n



#WCZGZ

# WORDPRESS DESACOPLADO. « MUEVE TU WORDPRESS AL FUTURO PRESENTE »

---

## 1. Introducción

- Historia de WordPress
- Evolución.. a medias?

## 2. Arquitectura Desacoplada

- REST API
- UI

## 3. Show me the code!

- Live Demo (WP Headless)



<2020/>  
WORDCAMP  
ZARAGOZA

#WCZGZ

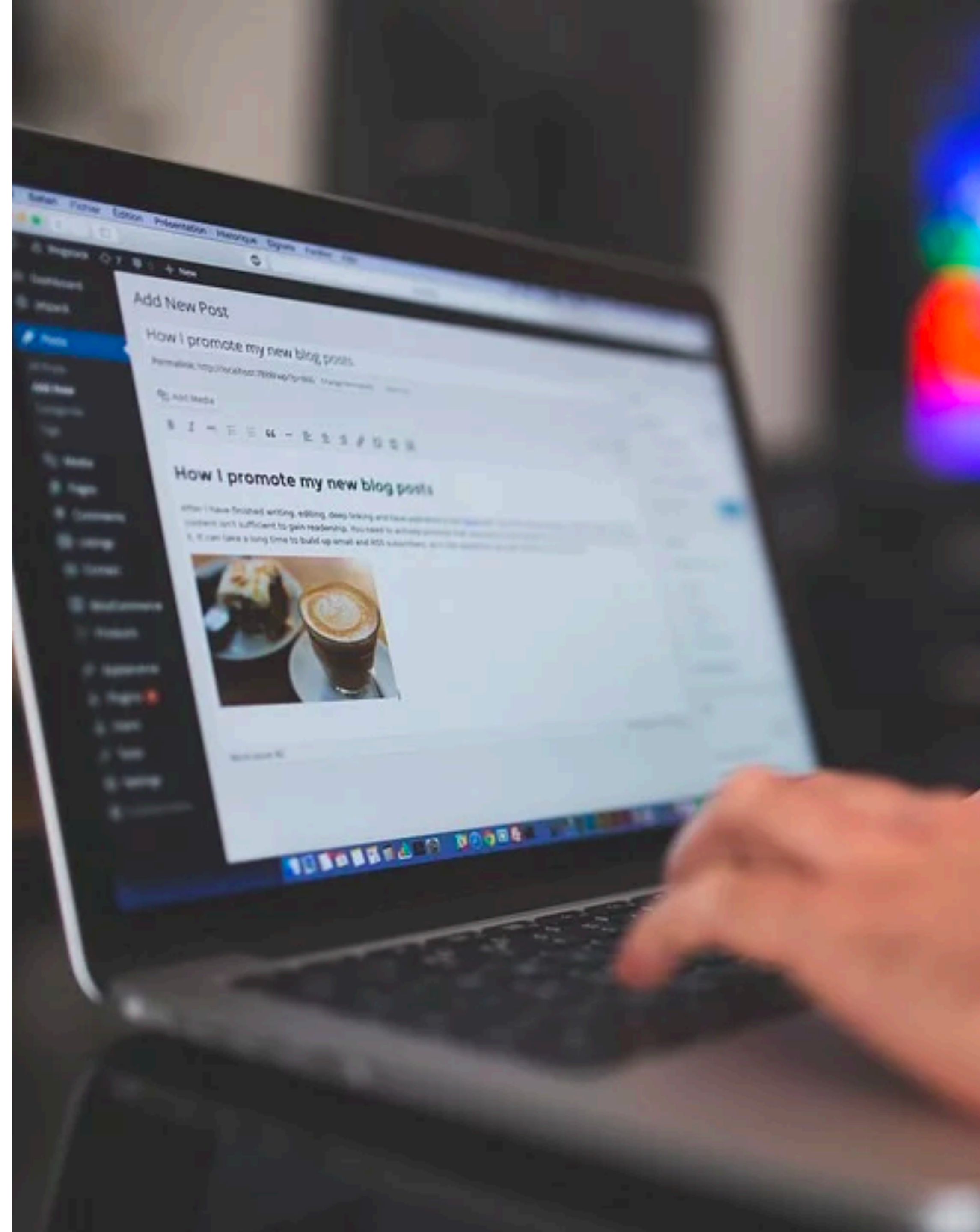


#WCZGZ

# INTRODUCCIÓN

---

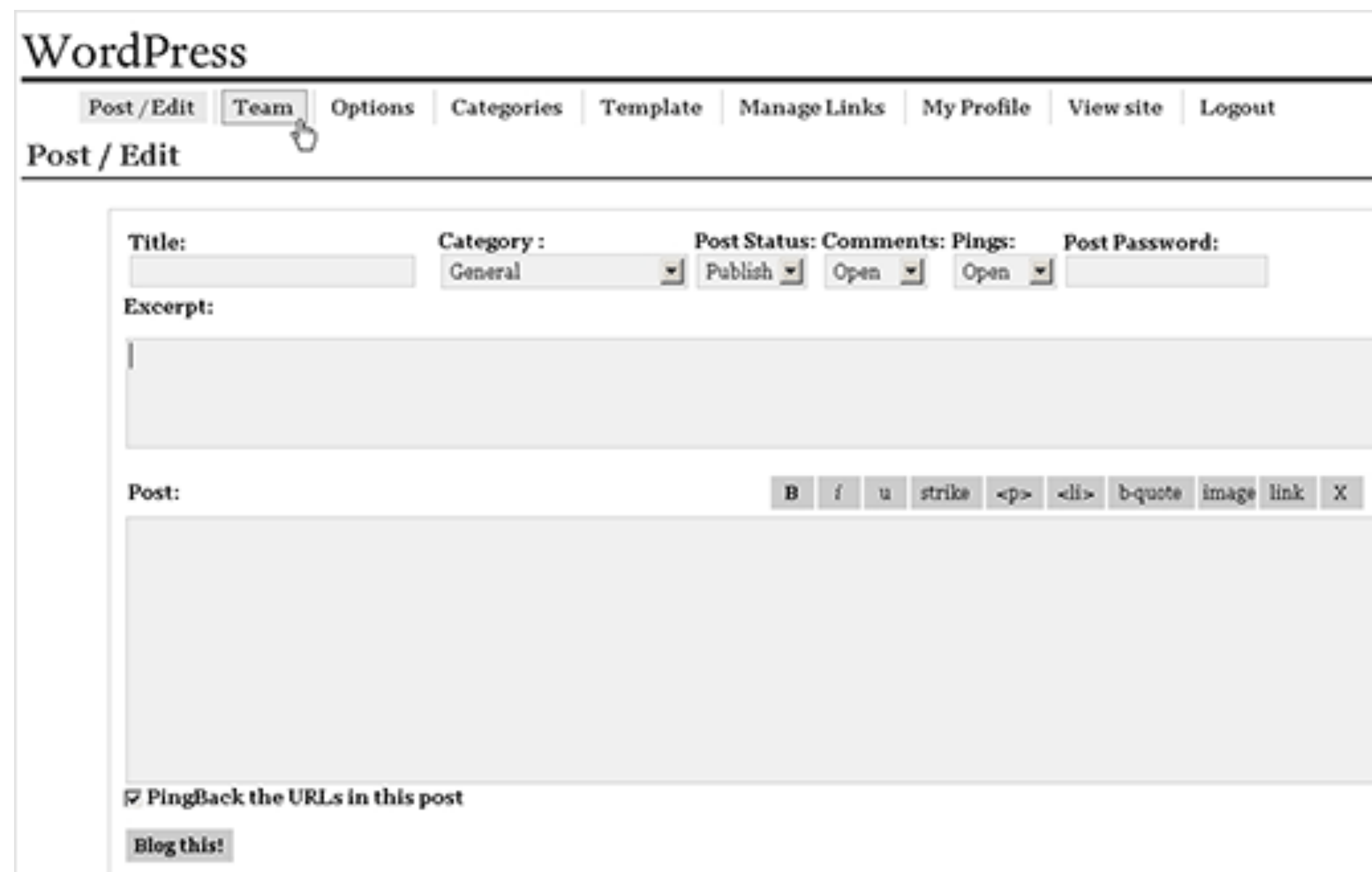
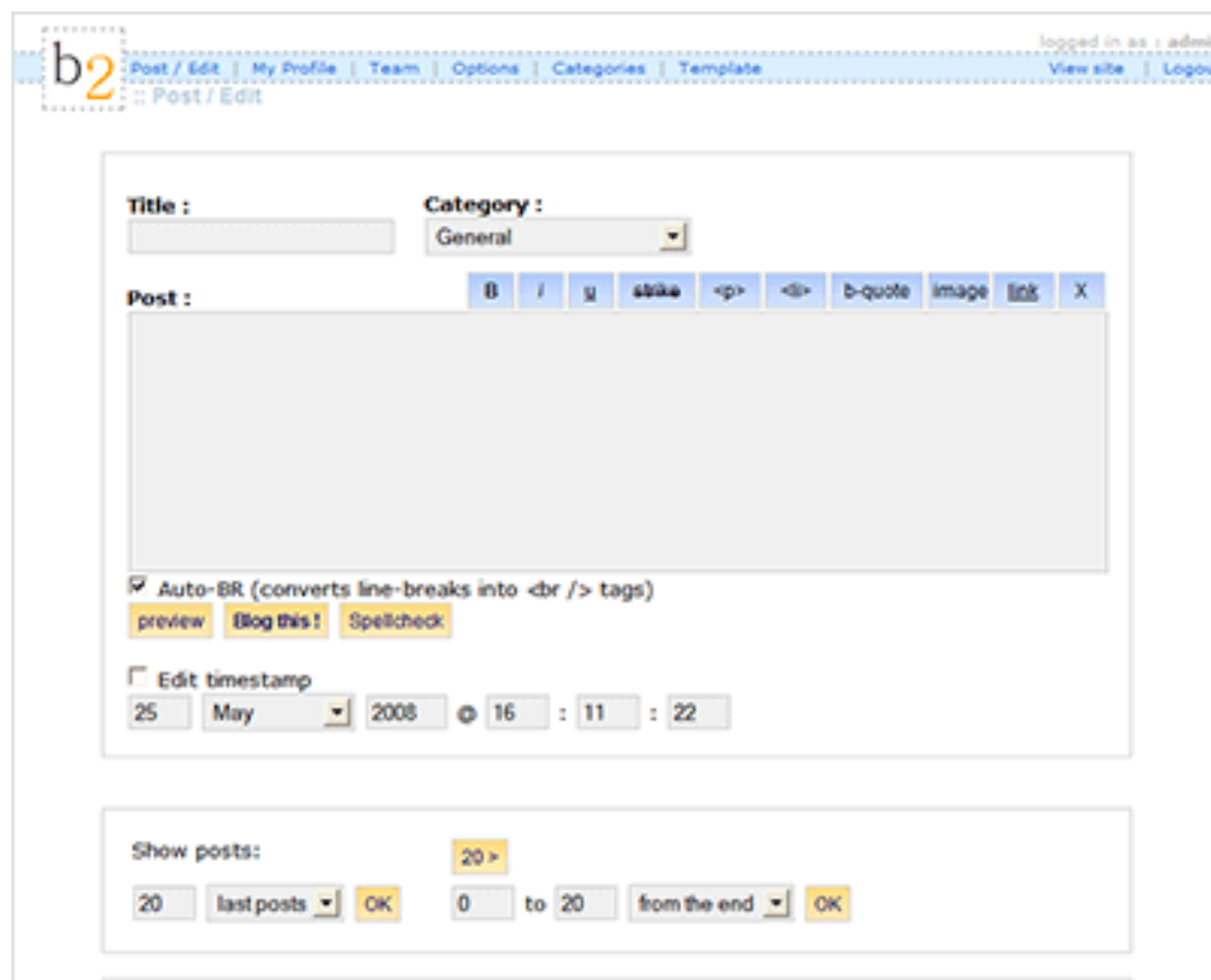
- *Historia de WordPress*
- *Evolución.. a medias?*



# UN POCO DE HISTORIA (INICIOS)

---

- Sistema de gestión de contenidos basado en b2/cafeolog
- 27 Mayo de 2003 -> ¡Hace casi 17 años!
- PHP + MySQL

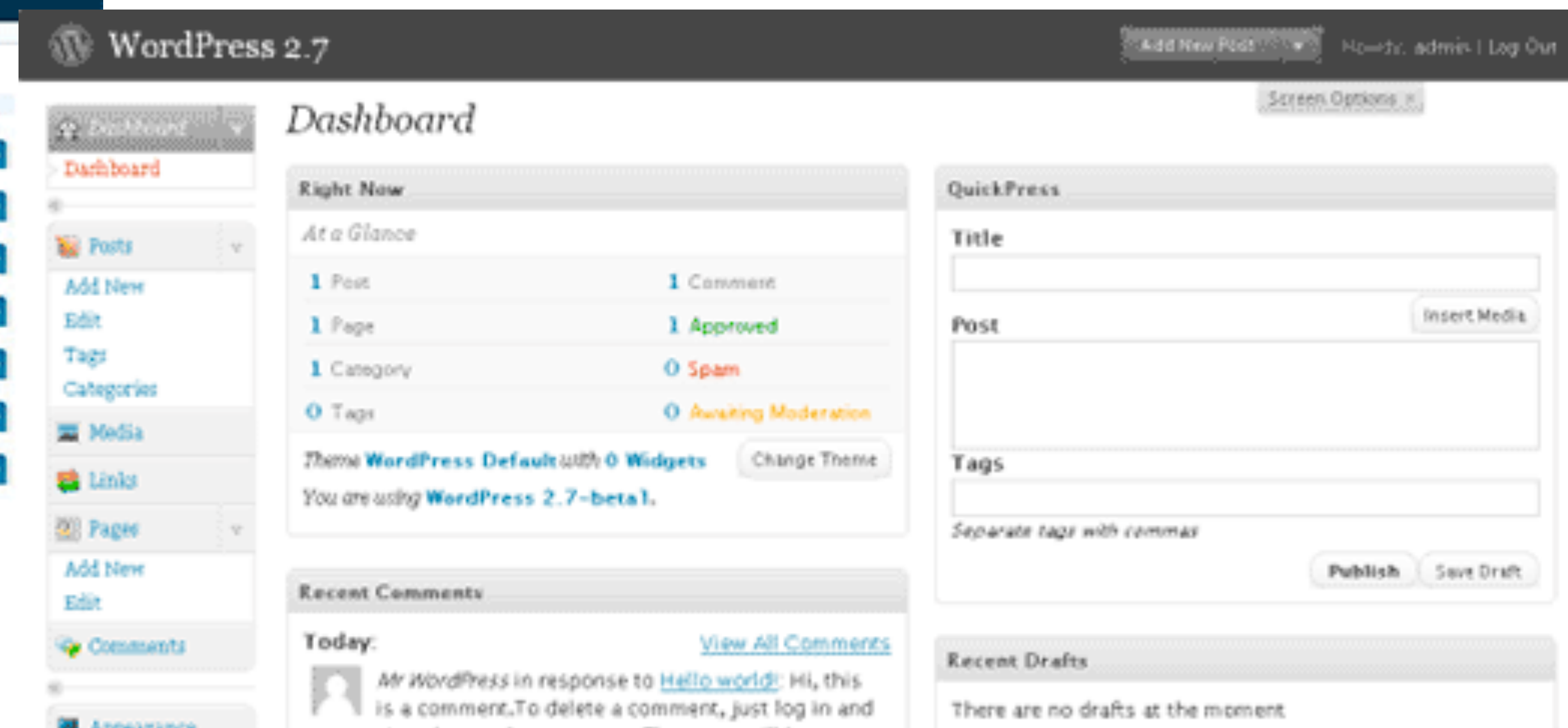
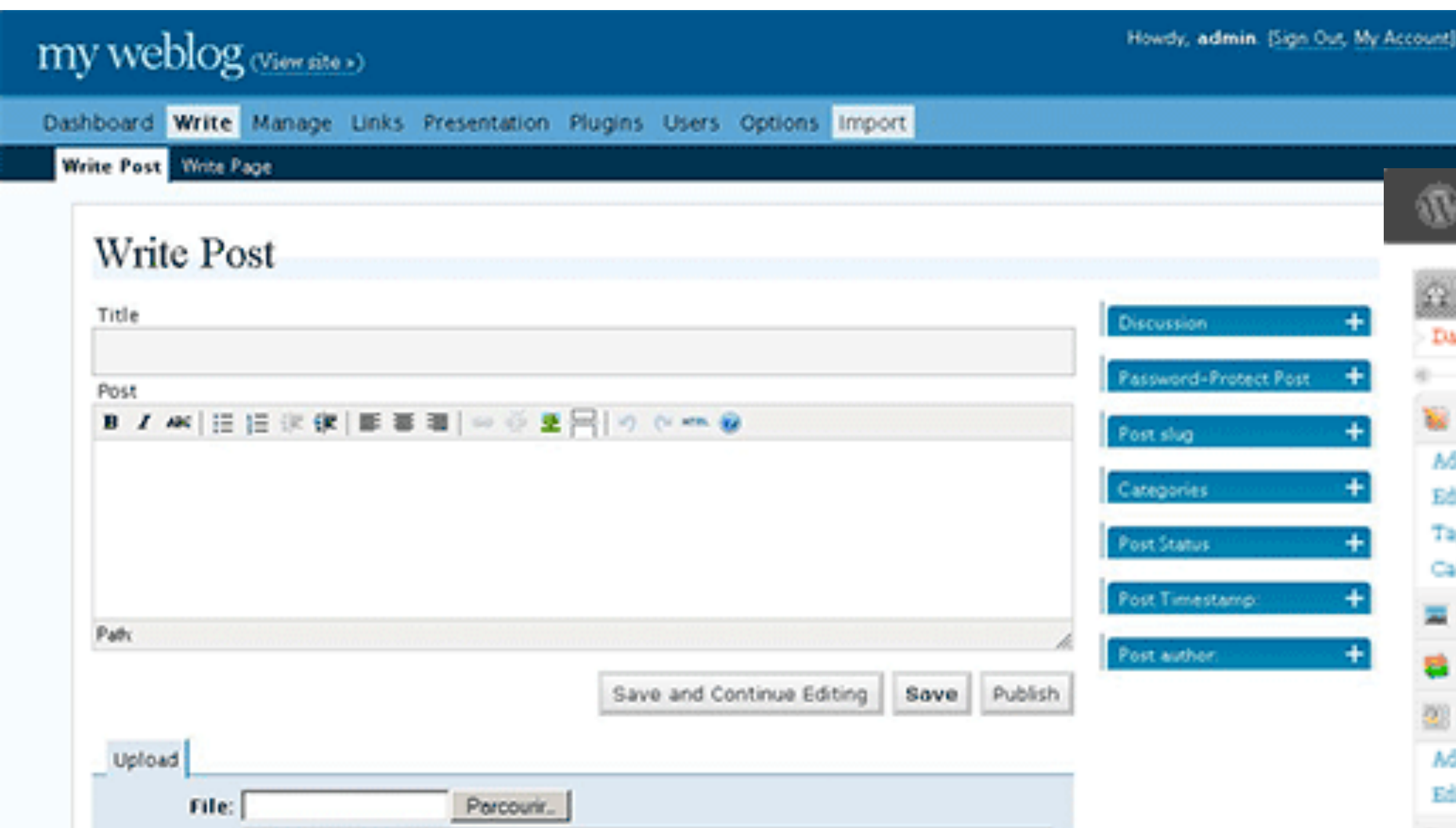




# UN POCO DE HISTORIA (EVOLUCION - FUNCIONALIDADES)

---

- 2005: WordPress 2.0: Nuevo Dashboard (JavaScript)
- 2007: functions.php
- 2010: Custom Post Types, Custom Taxonomies

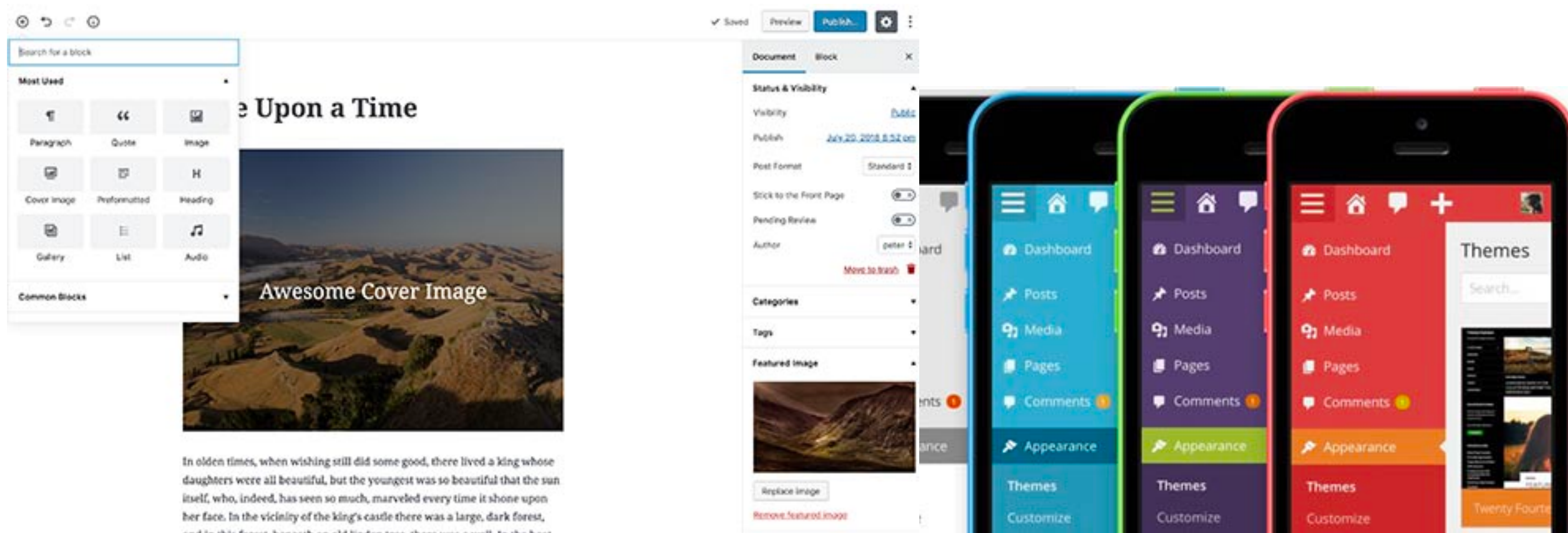




# UN POCO DE HISTORIA (EVOLUCION II - DESACOPLAMIENTO)

---

- 2013: Automatic Updates, New Admin Interface
- 2016: WordPress REST API
- 2018: Gutenberg



# EVOLUCION...A MEDIAS??

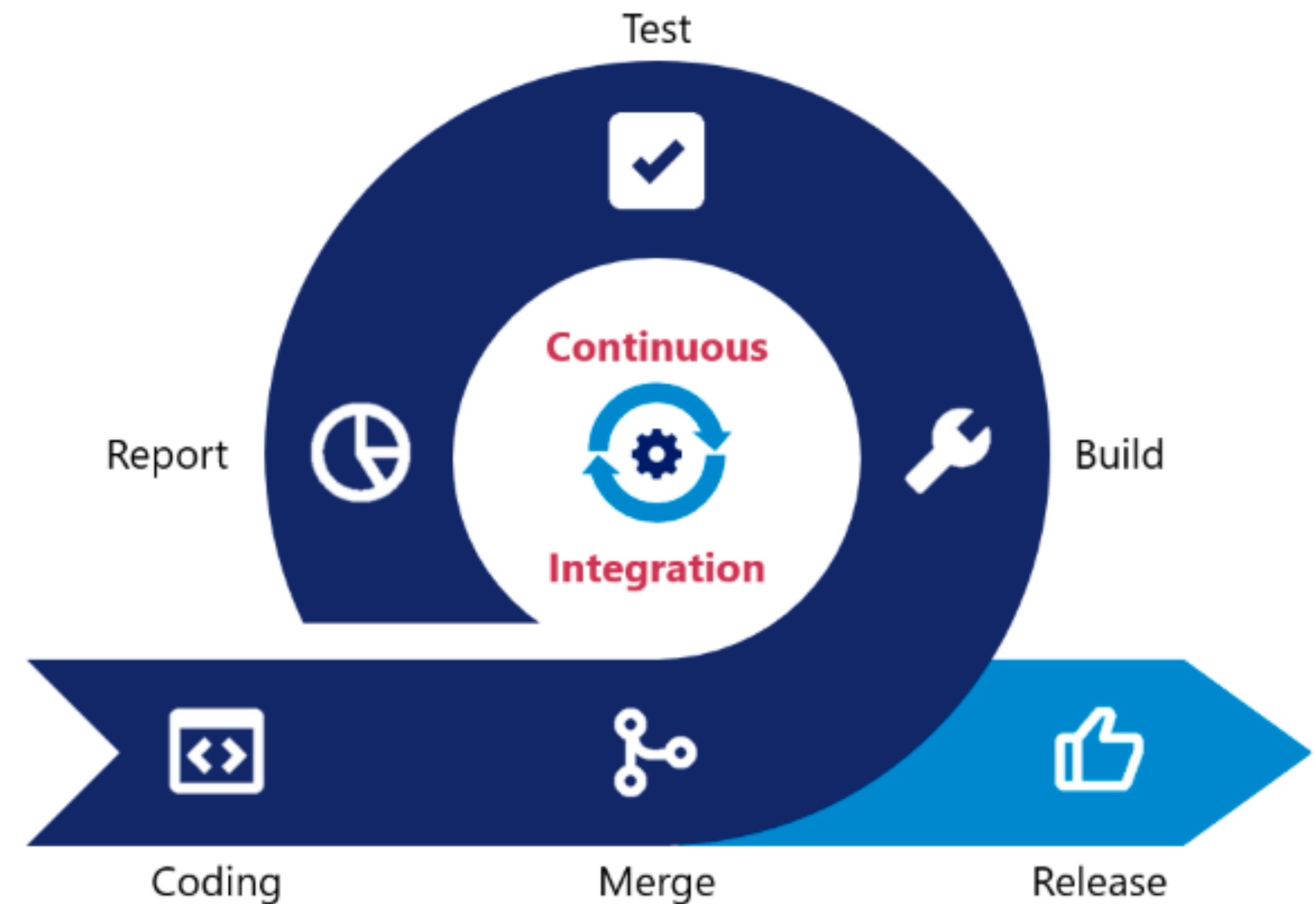
---

Paradigmas de programación:

- Procedural
- Object-Oriented (PHP 5.3)
- Functional

Integración Continua (CI/CD)

Testing, Code Sniffing, Mess Detection





# EVOLUCION...A MEDIAS??

---

WordPress fue diseñado antes de que el paradigma de programación orientada a objetos llegara a PHP.

- Proyecto WP-CLI (versionado de plugins)
- WP core como paquete composer
- Plugins propios -> Orientada a Objetos (<https://jokiruiz.com/wordpress/wordcamp-zaragoza-2018/>)
- Temas?? La jerarquía de templates en los temas, no está orientado a objetos



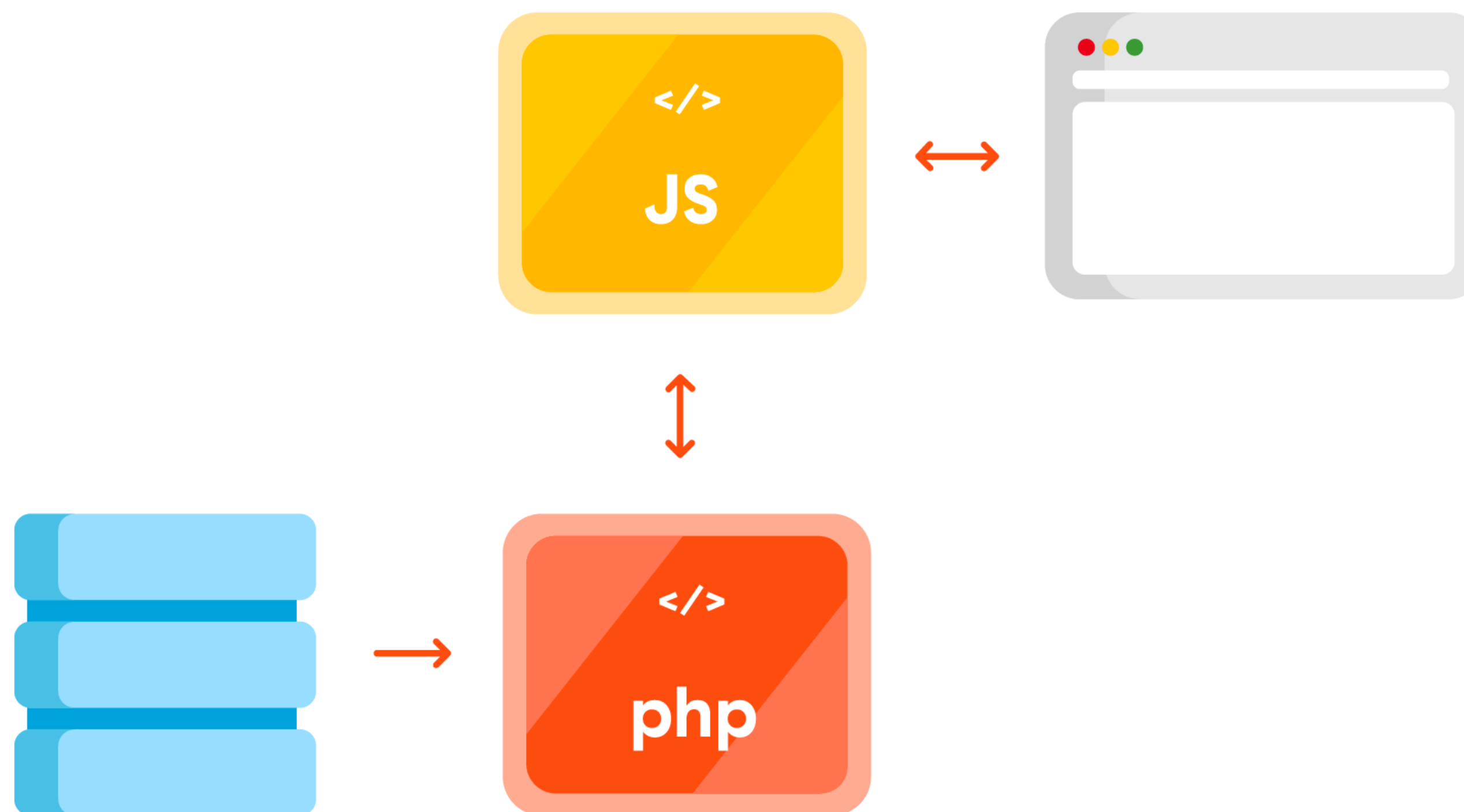
**Muy complicado desarrollar grandes y complejos temas con funcionalidades testeables, independientes y escalables.**

# EVOLUCION...A MEDIAS??

---

Pero tranquilos... Tenemos la solución!

¡Desacoplar el front-end!



#WCZGZ

# ARQUITECTURA DESACOPLADA

---

- *REST API*
- *UI*





# ARQUITECTURA DESACOPLADA

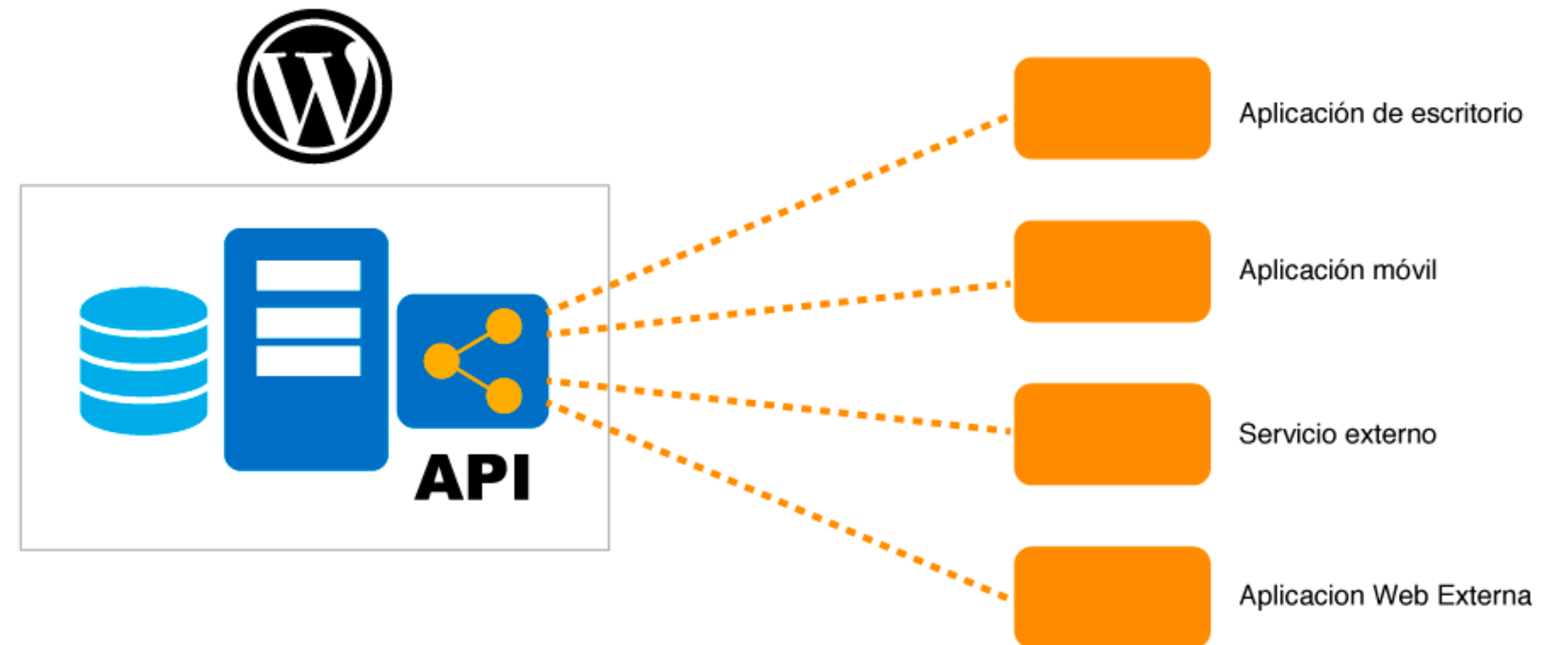
---

Una parte front-end con tecnologías de cliente como JavaScript, CSS, etc.

El Back-End (Controller+Modelo) proporcione un conjunto de servicios REST

## Beneficios:

- Liberar al servidor del trabajo de generar pantallas
- La transmisión de datos cliente y servidor el estrictamente necesario.
- Formularios Interactivos, sin validación por parte de servidor
- **Mejor Rendimiento y ahorro en Servidores**



# WORDPRESS REST API

---

La API REST de WordPress es un conector universal desde cualquier WordPress a cualquier aplicación en un servidor web o sistema operativo del usuario.

<https://developer.wordpress.org/rest-api/>

Aprender a trabajar con esta herramienta puede abrir las puertas a posibilidades casi infinitas para extender tu sitio web de WordPress.

Incorporación al core a finales de 2015, principios de 2016.

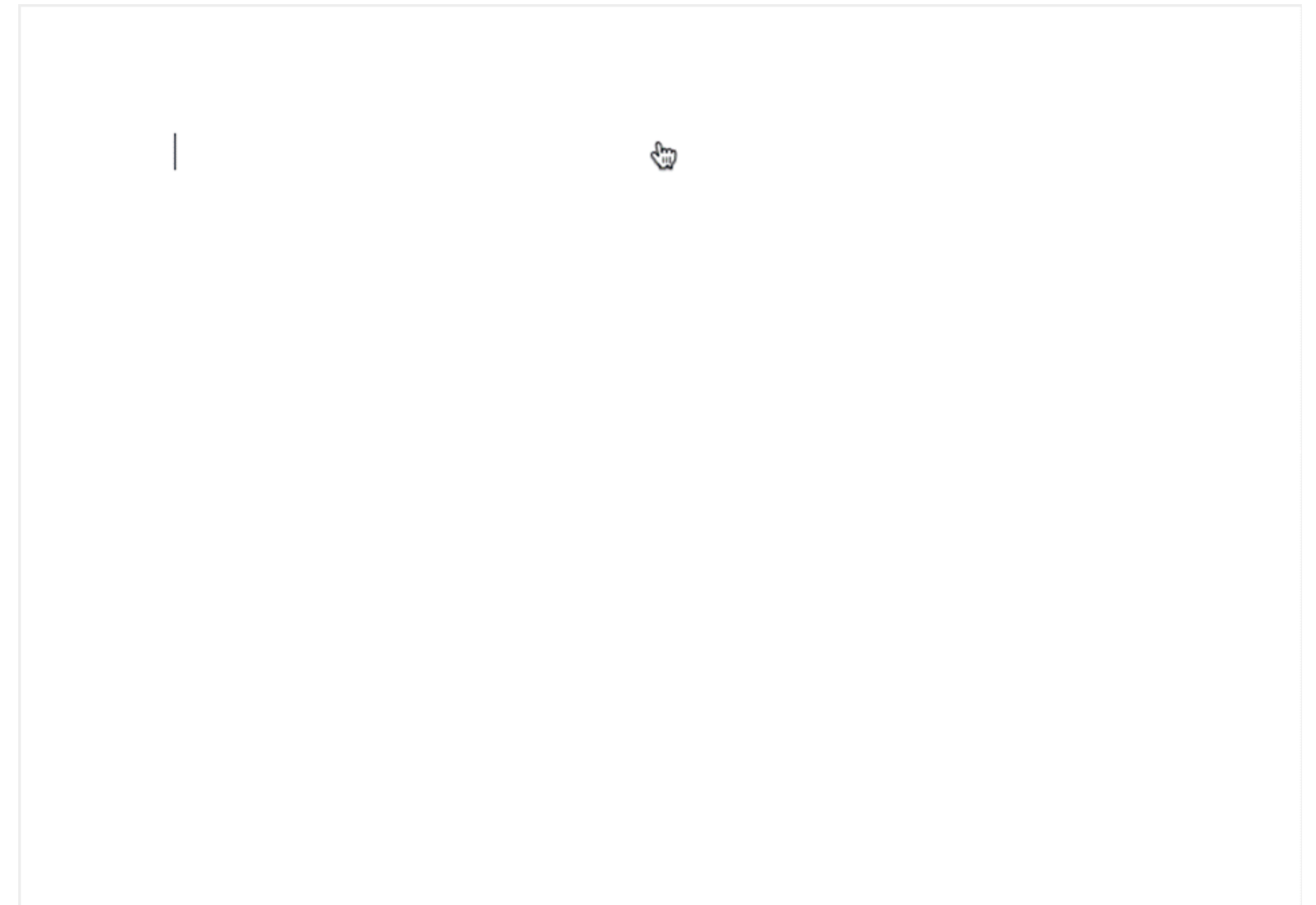
Por defecto con tu WordPress: <http://jokiruiz.com/wp-json/wp/v2>

# REST API

---

Ejemplos:

- El **Editor Gutenberg** utiliza la API REST para visualizar cambios en la interfaz.
- Permite a los desarrolladores de **Aplicaciones móviles** acceder a la API para cargar contenido de WordPress.
- Permite el desarrollo de **widgets “desacoplados”** utilizando la API REST, olvidándonos de los pesados y anticuados `iframe`.





# REST API

---

Las **API REST** se entregan a través de puntos finales de HTTP, utilizando el formato de **JavaScript Object Notation (JSON)**. En términos más simples, estas tecnologías proporcionan acceso a la API usando direcciones web únicas para entregar datos que se comportan como **objetos de JavaScript**.

La petición empieza con un verbo (o método). Los verbos definen la acción que se quiere realizar sobre el recurso. Los más comunes:

- **GET**: se utiliza para solicitar un recurso.
- **POST**: se utiliza para publicar un recurso.
- **PUT**: se utiliza para reemplazar un recurso.
- **DELETE**: se utiliza para eliminar un recurso.

# REST API

#WCZGZ  
@JokiRuizLite

## Template PHP:

```
<?php $the_query = new WP_Query( $args ); ?>

<?php if ( $the_query->have_posts() ) : ?>

    <?php while ( $the_query->have_posts() ) :
    <?php $the_query->the_post(); ?>
        <h2><?php the_title(); ?></h2>
    <?php endwhile; ?>

    <?php wp_reset_postdata(); ?>
<?php endif; ?>
```

## JSON REST API:

```
add_action( 'rest_api_init', function () {
    register_rest_route( 'custom-api/v1', '/wordcamp/', array(
        'methods' => 'GET',
        'callback' => 'wczgz2020_custom_call'
    ));
});

function wczgz2020_custom_call() {
    $the_query = new WP_Query( $args );
    $posts = $the_query->posts;
    return $posts;
}
```

# REST API (AUTENTICACION)

---

Basic Auth (development site): <https://github.com/WP-API/Basic-Auth>

```
curl -X GET --user username:password -i https://jokiruiz.com/wp-json/wp/v2/posts?status=draft
```

JWT Authentication (live site): <https://github.com/Tmeister/wp-api-jwt-auth>

```
https://jokiruiz.com/wp-json/jwt-auth/v1/token
```

```
{  
  "token": "eyJ0eXA...5wJ0O6qQ8QFcYizimDmBy6hCH_8",  
  "user_display_name": "admin",  
  "user_email": "admin@localhost.dev",  
  "user_nicename": "admin"  
}
```



.....

Es posible un tema de WordPress, desarrollado por un FrontEnd developer...

... que no sabe de PHP?

Llevamos años haciendo esto, incluso a veces sin saberlo. **Una llamada a una API desde un framework javascript no es tan distinto a las llamadas Ajax que hemos hecho durante años para pintar contenido dinámico.**

Los principales frameworks actuales de javascript (Angular, React, VueJs,..) están pensados para generar interfaces a medida que se nutren de datos recogidos mediante APIs ya sean por TCP o por sockets de comunicación UDP.

# VENTAJAS DE UNA UI JAVASCRIPT

---

- Total independencia del backend
- Ahorro en recursos de servidores
- Posibilidad de servir nuestro interfaz de forma serverless
- Posibilidad de generar PWA, aplicaciones embebidas en pantallas táctiles (kioskos), aplicaciones encapsuladas para las App stores, ...
- Posibilidad de integrar nuestro interfaz en otro sistema sin preocuparnos por incompatibilidades

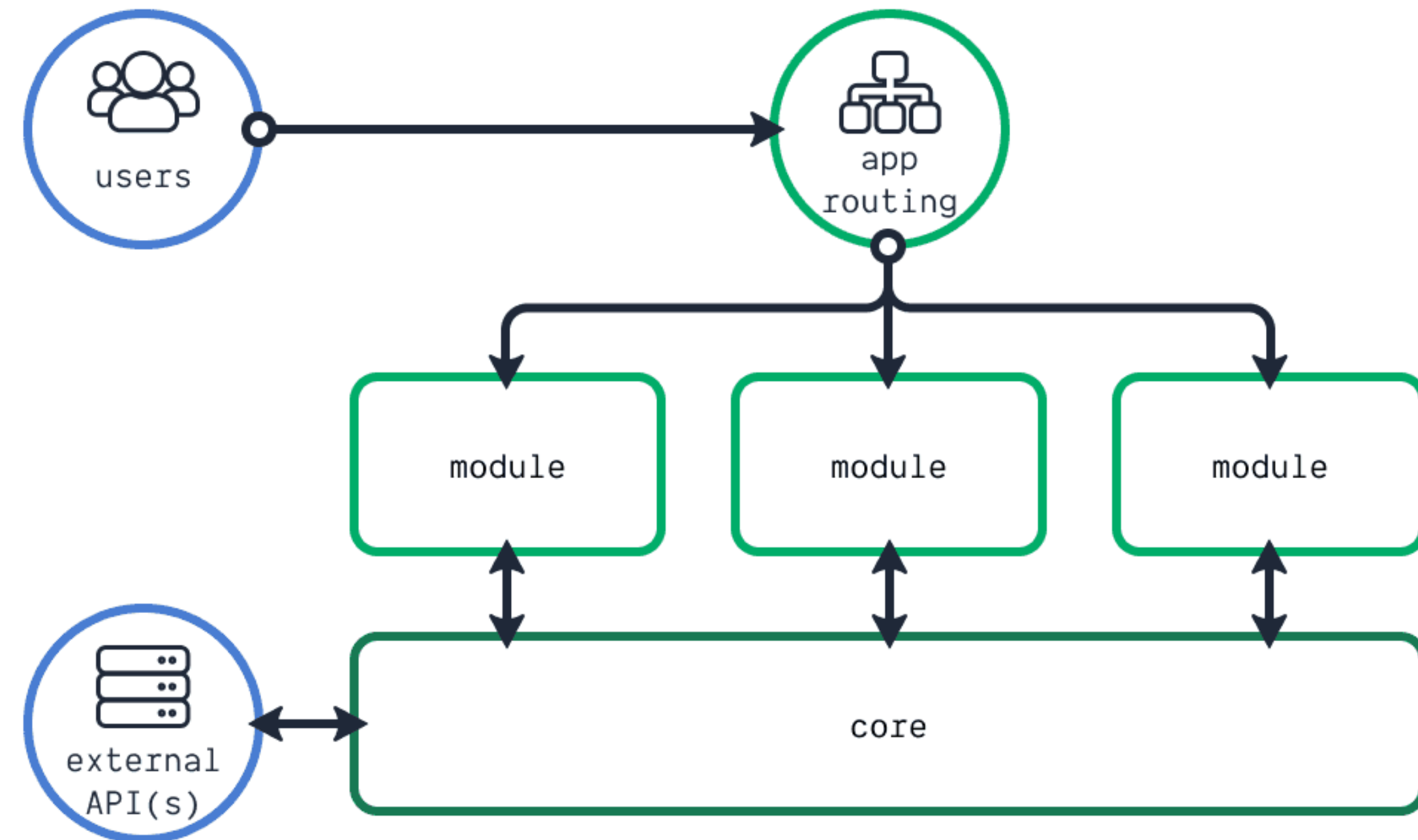
# ESTRUCTURA UI JAVASCRIPT

#WCZGZ  
@Carlosas1n

Toda web/app javascript está compuesta de:

- **Assets:** Recursos estáticos como tipografías, imágenes, ...
- **Modulos:** Agrupaciones de funcionalidades para facilitar el desarrollo y reutilizar funcionalidades.
- **Componentes:** Datos y lógica que en asociación con template HTML define una vista.
- **Estilos:** Conjunto de estilos css de la web.

```
src/  
├── assets/  
├── components/  
├── core/  
├── lib/  
├── modules/  
└── styles/
```





# JS UI PARA WORDPRESS

#WCZGZ  
@Carlosas1n

## - Página cabecera

/wp/v2/pages

/wp/v2/pages/:id - /wp/v2/pages?slug=pagina-ejemplo

## - Posts portada

/wp/v2/posts

/wp/v2/posts/:id - /wp/v2/posts/slug=wordcamp-zaragoza-2020

## - Categorías

/wp/v2/categories

## - Búsqueda

/wp/v2/search

The screenshot shows a WordPress website with the following content:

- Header:** "wczgz2020 Otro sitio realizado con WordPress" on the left, "WordCamp Zaragoza 2020" on the right, and a search icon.
- Post:** A post titled "WordCamp Zaragoza 2020" by "admin" on "8 enero 2020". The content includes the text: "17 y 18 de enero | Edificio ETOPIA" and "Dos días de WordPress repletos de innovación, diseño, desarrollo, marketing, contenidos, negocios y networking. Si disfrutas aprendiendo y compartiendo lo que sabes con los demás y cualquier excusa te parece buena para conocer gente nueva... ¡te esperamos!".
- Category Page:** A category page titled "¡Hola, mundo!" by "admin" on "7 enero 2020" with "1 comentario". The content includes the text: "Bienvenido a WordPress. Esta es tu primera entrada. Edítala o bórrala, ¡luego empieza a escribir!".
- Footer:** "© 2020 wczgz2020 Powered by WordPress" on the left and "To the top ↑" on the right.

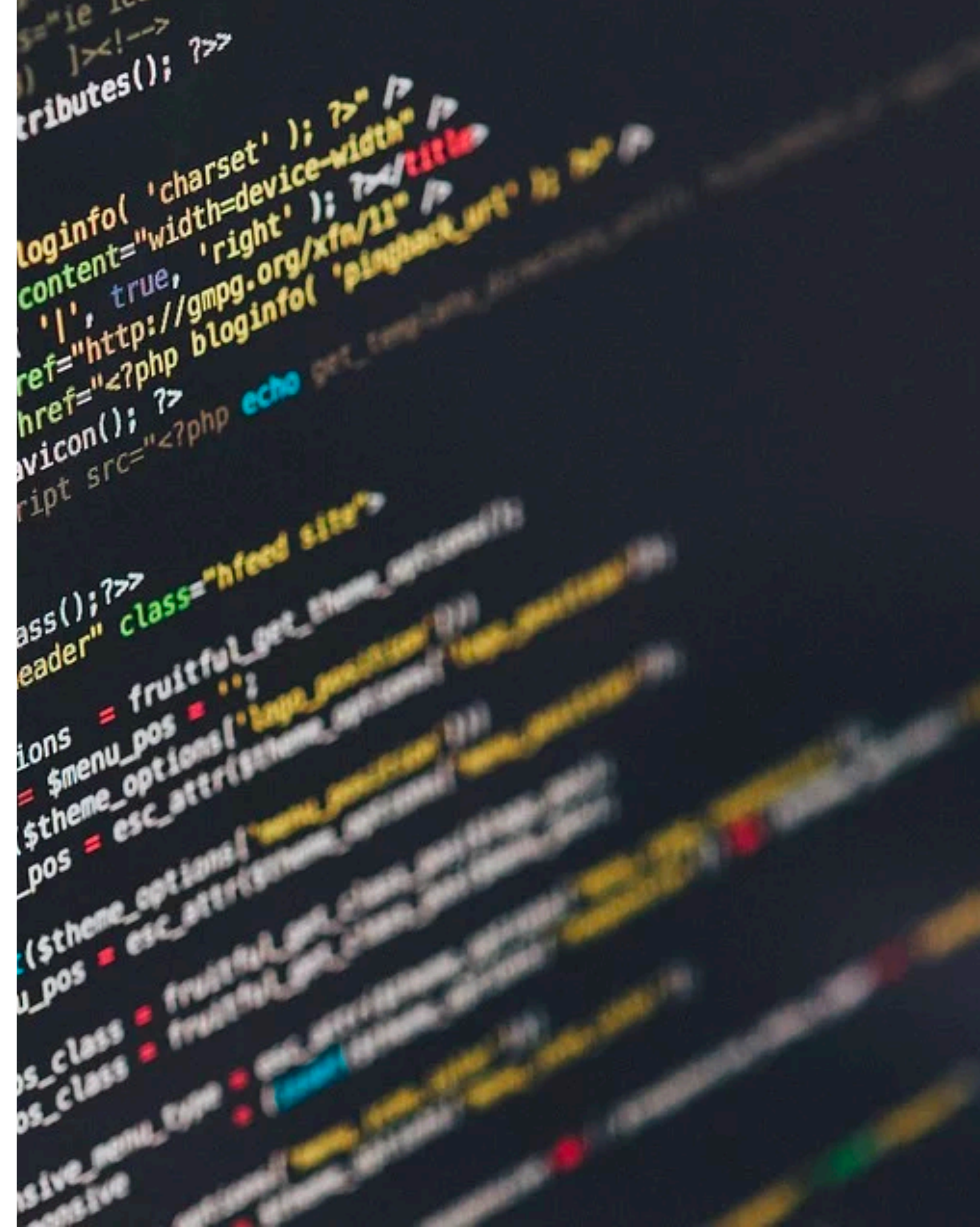


#WCZGZ

# SHOW ME THE CODE! :)

---

- *WordPress*
- *JavaScript Framework*





# RECURSOS

---

*WordPress usado para la demo*

<http://wczgz2020.jokiruiz.com/>

*Angular usado para la demo*

<http://wczgz2020.carlosasin.com/>

*Código WordPress: Rest API, Custom Post*

<https://github.com/JoaquinRuiz/wczgz2020>

*Código Frontend: Angular*

<https://github.com/carlosasin/wczgz2020>

```
    "35",
    "http://wczgz2020.jokiruiz.com/?p=17"
  ],
  "date": "2020-01-13T19:13:38",
  "slug": "wordcamp-zaragoza-2020",
  "title": "WordCamp Zaragoza 2020"
},
{
  "date": "2020-01-13T19:13:38",
  "slug": "17-y-18-de-enero-edificio-etopia",
  "title": "17 y 18 de enero | Edificio ETOPIA",
  "excerpt": "Dos días de WordPress repletos de networking. Si disfrutas aprendiendo y compartiendo lo que sabes con los demás, no te excusa te parece buena para conocer gente nueva... ¡ite esperamos! WordCamp Zaragoza 2020",
  "status": "published",
  "type": "post",
  "author": "carlosasin",
  "comment_status": "closed",
  "ping_status": "closed",
  "sticky": false,
  "template": "",
  "format": "standard",
  "meta": [],
  "categories": [
    2
  ],
  "tags": [],
  "next": null,
  "previous": {
    "id": 1,
    "slug": "hola-mundo"
  }
}
```



# GRACIAS!



<2020/>  
WORDCAMP  
ZARAGOZA

WORDPRESS DESACOPLADO. « MUEVE TU WORDPRESS AL ~~FUTURO~~ PRESENTE »

# ¿PREGUNTAS?