



I E S   A L B A R R E G A S

TÉCNICO SUPERIOR EN DESARROLLO DE  
APLICACIONES WEB

Departamento de Informática

## PROYECTO

CARTA WP. Plugin de Wordpress para carta de restaurante.

### **Manual Técnico**

## Índice

- 1. Introducción y objetivos**
- 2. Arquitectura de la aplicación**
  - 2.1. Frontend**
    - 2.1.1. Tecnologías usadas**
    - 2.1.2. Entorno de desarrollo**
    - 2.1.3. Estructura**
  - 2.2. Backend**
    - 2.2.1. Tecnologías usadas**
    - 2.2.2. Entorno de desarrollo**
    - 2.2.3. Estructura**
  - 2.3. Base de datos**
- 3. Documentación técnica**
  - 3.1. Análisis**
  - 3.2. Desarrollo**
  - 3.3. Pruebas realizadas**
- 4. Proceso de despliegue**
- 5. Propuesta de mejoras**
- 6. Bibliografía**

## 1. Introducción y objetivos

**Carta WP** es un plugin para Wordpress que permite al usuario de esta aplicación poner a disposición de sus clientes una **carta digital** dentro de la web de su restaurante, bar o cafetería.

El plugin se podrá descargar de forma gratuita desde el repositorio de Wordpress y el usuario podrá instalarlo para dotar a su página web de las siguientes funcionalidades:

- Organización de todos los elementos de la carta bajo un mismo tipo de contenido (**Custom Post Type**).

Wordpress por defecto organiza su contenido en “Entradas” y “Páginas”. Carta WP crea un nuevo tipo de contenido llamado “Platos”.

- Creación de los campos necesarios (**Custom Fields**) para ese contenido. Wordpress asocia por defecto a cada CPT los siguientes campos: título, extracto, contenido, imagen destacada y comentarios.

Con nuestro plugin eliminaremos algunos de estos campos y añadiremos otros personalizados, de forma que el CPT plato tendrá los siguientes campos: título, contenido, imagen destacada y precio.

- Creación de las taxonomías (**Custom taxonomies**) para categorizar el contenido.

De forma nativa, Wordpress permite organizar el contenido mediante categorías y etiquetas. Carta WP va a crear dos taxonomías personalizadas para el tipo de contenido “Plato”:

- Secciones: nos permite asociar a cada plato una o varias secciones a las que pertenece (Por ejemplo: bebidas, tapas, raciones, postres, sugerencias). El usuario podrá crear las secciones que necesite.
- Alérgenos: permite asociar a cada plato sus alérgenos. Al instalar el plugin, nos crea los 14 alérgenos obligatorios por ley. El usuario podrá añadir tanto más como necesite. A su vez, a esta taxonomía le vamos a crear un campo personalizado de tipo imagen que permite asociar a cada alérgeno un icono.

- Posibilidad de **crear un usuario** de Wordpress que solo tenga acceso en el panel de administración para editar la carta.

En muchas ocasiones vamos a necesitar que la persona que administra la carta no tenga acceso a hacer cambios en el resto de la web, para lo que nuestro plugin crea un rol de usuario llamado “editor de carta”. Al usuario de Wordpress que asignemos este rol solo va a tener acceso a los contenidos de la carta, lo que va a permitir por una parte una administración más sencilla de esta y por otra que no pueda realizar cambios al resto de la web.

- **Mostrar la carta** en el lugar de la web que deseemos mediante un shortcode.

El contenido que hemos introducido a la carta en el backend de Wordpress lo vamos a poder mostrar en el lugar deseado del frontend mediante el shortcode [wp-mostrar-carta]. La forma recomendada de cara al SEO sería crear una página

# Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto:



JUNTA DE EXTREMADURA

Consejería de Educación y Empleo

llamada “carta” y en ella introducir donde queramos mostrar todo el contenido de nuestra carta el shortcode [wp-mostrar-carta].

- La carta se muestra de forma **responsive** (diseño Mobile First) y con posibilidad de **filtrado de contenido**.

Se mostrará en la url elegida todo el contenido de nuestra carta junto a un listado de las secciones. Pinchando sobre cualquiera de ellas podremos filtrar el contenido.

El objetivo final de este plugin es dotar a las páginas web de los restaurantes de una **carta digital organizada, fácil de mantener para el usuario y atractiva, clara y rápida para el cliente**.

Carta WP va dirigido tanto a los **implementadores** Wordpress que creen webs de restaurantes a sus clientes y solo quieran darle acceso para poder editar la carta, como para **gerentes de restaurantes** que hayan creado su propia web en Wordpress y quieran añadir de forma sencilla y organizada una carta digital.

## 2. Arquitectura de la aplicación

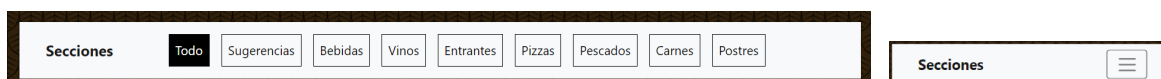
La decisión de resolver este proyecto como un plugin de Wordpress y no como una aplicación independiente son las siguientes:

- El 40% de las webs en la actualidad está desarrollada con este CMS y va en aumento cada año.
- Permite que cualquiera con pocos conocimientos pueda modificar la web.
- El tener la carta en nuestra propia web nos va a permitir asegurarnos tráfico, analizarlo y tomar decisiones en base a ese tráfico.
- En lo referente a monetización del proyecto, el realizarlo con esta tecnología me va a posibilitar dos cosas:
  - Actualmente soy desarrollador Wordpress y este plugin me va a permitir ahorrar tiempo y trabajo repetitivo a la hora de implementar cartas digitales en webs de clientes.
  - Subiré el plugin al repositorio de Wordpress para que esté disponible de forma gratuita y se me vaya conociendo en la comunidad de desarrollo de Wordpress. Además realizaré algunas extensiones de pago para ampliar las capacidades del plugin (Opciones de personalización, pedidos desde la web...) siguiendo el modelo freemium (parte básica gratis, parte ampliada de pago)

Wordpress es un CMS desarrollado en PHP y javascript, por lo que para este proyecto hará uso de las siguientes tecnologías: **HTML5**, **CSS3**, **Javascript** y **PHP**.

### 2.1. Frontend

La estructura de nuestra carta va a estar formada por una cabecera donde aparecerán las secciones que vamos a poder filtrar y el contenedor donde aparecen los platos de la sección seleccionada:



Al entrar nos aparece todo el contenido de la carta por si el cliente quiere verla entera y si pincha en alguno de los botones se muestra solo el contenido de esa sección

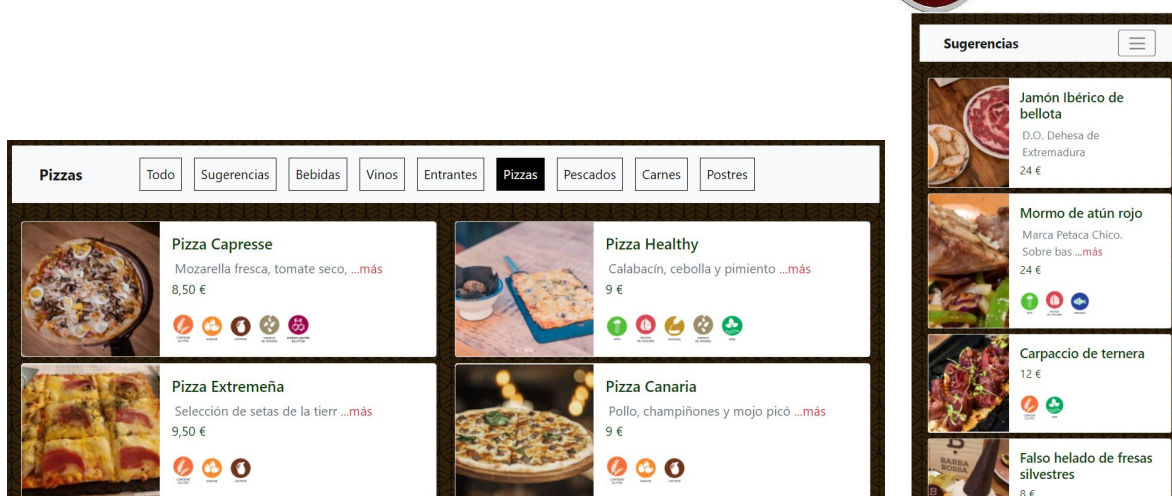
# Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto:



JUNTA DE EXTREMADURA

Consejería de Educación y Empleo

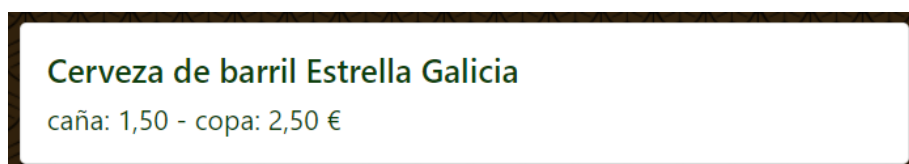


En el contenedor de cada plato vamos a incluir la siguiente información:

- Fotografía.
- Título.
- Descripción: una sola línea con posibilidad de leer más si interesa.
- Precio. Su información es obligatoria por ley.
- Iconos de alérgenos. Su información es obligatoria por ley.



En caso de no incluir fotografía del producto, la información se mostrará ocupando todo el contenedor



Si nos interesa más información sobre el plato, al pinchar sobre el nos aparecerá un pop-up con la descripción ampliada y la fotografía a mayor tamaño.

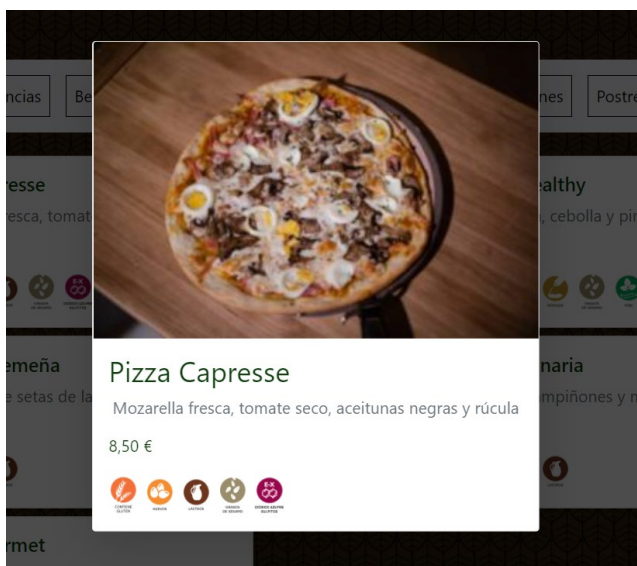
# Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto:



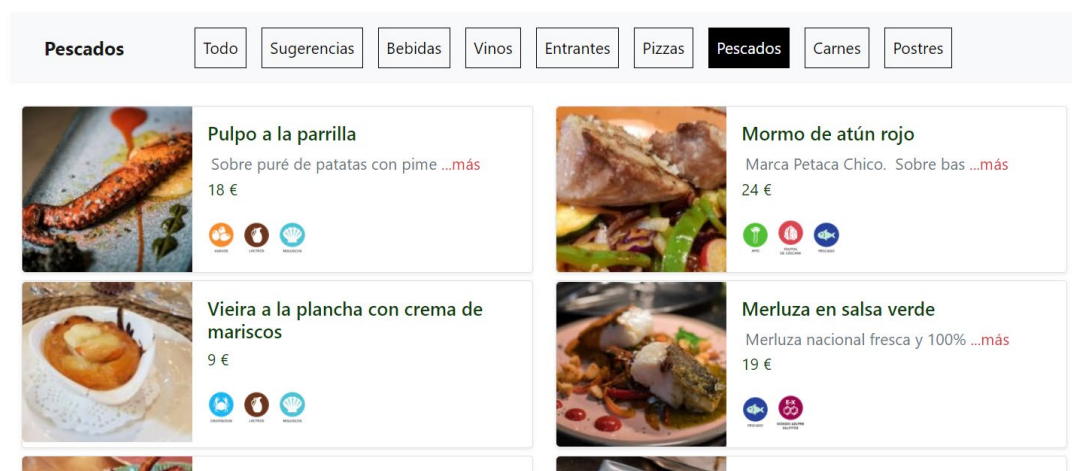
JUNTA DE EXTREMADURA

Consejería de Educación y Empleo



Para diseñar la interfaz del frontend hemos utilizado colores neutros para que el plugin pueda funcionar bien sobre cualquier diseño, ya que va a poder utilizarse sobre cualquier web.

- Ejemplo de uso sobre fondos claros:



- Ejemplo de uso sobre fondos oscuros e imágenes:



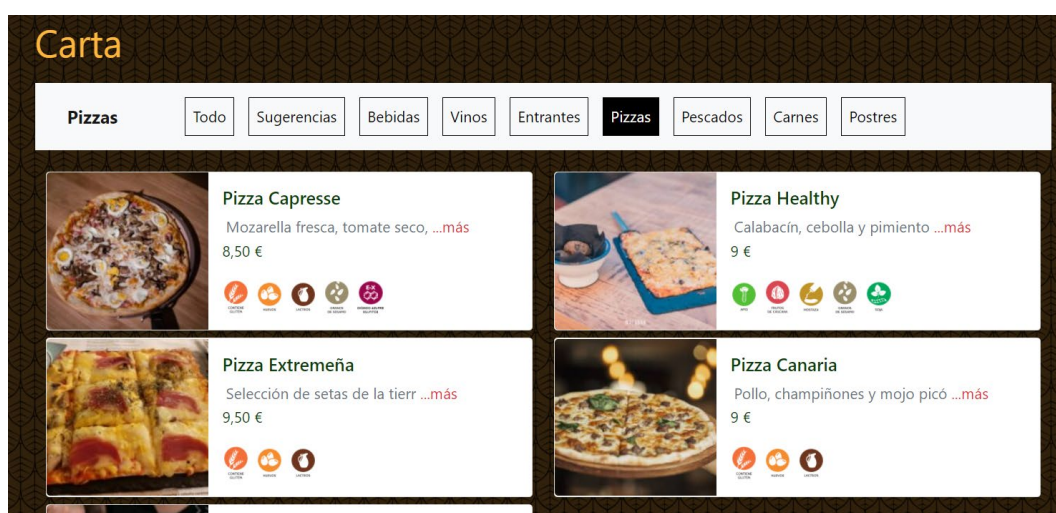
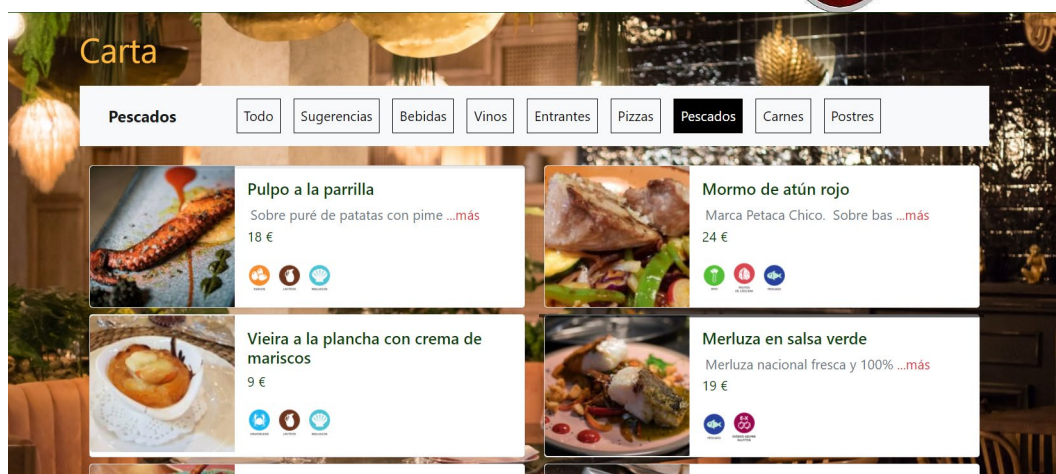
# Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto:



JUNTA DE EXTREMADURA

Consejería de Educación y Empleo



## 2.1.1. Tecnologías usadas

Para el diseño del frontend hemos utilizado las siguientes tecnologías:

- HTML 5
- CSS 3
- Bootstrap 5: nos permite realizar una maquetación responsive de manera rápida y efectiva.
- Javascript: para el filtrado del contenido he desarrollado un plugin con la librería jQuery que una vez recuperado todo los platos de la base de datos mediante una única consulta nos va a permitir mostrar u ocultar del DOM los platos según la categoría seleccionada.

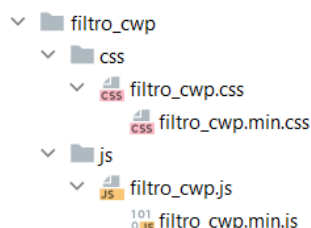
## 2.1.2. Entorno de desarrollo

Para todo el desarrollo del plugin he utilizado el IDE PHPStorm y para pruebas y depuración del frontend la consola de Google Chrome.



## 2.1.3. Estructura

Como ya hemos comentado anteriormente, para filtrar el contenido traído de la base de datos al frontend hemos desarrollado un plugin de jQuery que incluiremos como una librería del plugin Carta WP. Pasamos a explicar su estructura:



En la imagen podemos ver la estructura que va a tener el plugin de filtrado. Mencionar que lo desarrollamos en jQuery en vez de javascript puro porque es una librería que está incluida en el propio Wordpress, por lo que no va a aumentar el peso del mismo y nos va a facilitar el desarrollo.

El desarrollo se va a centrar en mostrar u ocultar contenido según la sección seleccionada por el usuario y ampliar la información de cada plato y su imagen al hacer clic sobre él.

Pasamos a explicar algunos puntos interesantes del desarrollo:

- Especificaremos en el plugin mediante selectores cuales van a ser los botones de filtrado y cuales van a ser los elementos a filtrar:

```
TJ_CartaWp.Defaults = {  
  filter : '.cwp-categorias li', // selector de cada botón de sección  
  item : '.cwp-item', // selector los item a filtrar (clase que tendrá el contenedor de cada plato)  
  callback : null  
}
```

- Definimos un método para el filtrado. Cada botón (filter) va a tener un atributo 'data-filter', cuyo valor será el nombre de la sección y cada plato (item) va a tener un atributo 'data-f' cuyo valor o valores van a ser las secciones a la que pertenezca el plato.

Al pulsar sobre el botón almacenamos el valor de su data-filter y mostramos solo los que coincidan con el atributo data-f de los platos.

```
else {  
  // Si el elemento que pulsamos no tiene la clase cwp-categoria-activa vamos a agrársela y quitársela a los hermanos  
  if (!$this.hasClass('cwp-categoria-activa')) {  
    $this.addClass('cwp-categoria-activa').siblings().removeClass('cwp-categoria-activa');  
    // Ocultamos todos los items y una vez ocultos ejecutamos la función  
    $item.hide(0, function () {  
      // Mostramos solo los items que tengan como valor del atributo data-f el valor de filtro (data-filter del botón en este caso)  
      // Utilizamos *= para indicar que data-f tiene que contener el valor de filtro, por si hay más de un filtro aplicado  
      $('['+data-f+ '*="'+filter+'"]').show();  
    });  
  }
```

- Definimos un método para ampliar la información del plato. Su función será seleccionar el contenido del contenedor pequeño (donde viene la descripción acortada a una sola línea y la imagen en tamaño reducido) y mostrarla ampliada en el grande con la maquetación que definamos.

```
this.options = null;
this.zoomfull = '<!-- INICIO - Maquetación al pinchar sobre un ítem -->\n' +
  '<div class="card shadow-lg cwp-zoom" id="">\n' +
  '  <img src="" class="card-img-top cwp-main-image" alt="">\n' +
  '  <div class="card-body">\n' +
  '    <h3 class="card-title cwp-title"></h3>\n' +
  '    <p class="card-text text-secondary m-1 cwp-description"></p>\n' +
  '    <h6 class="card-text cwp-price mt-3"></h6>\n' +
  '    <ul class="list-inline m-0 cwp-alergenos">\n' +
  '      </ul>\n' +
  '    </div>\n' +
  '  </div>\n' +
  '<!-- FIN - Maquetación al pinchar sobre un ítem -->';
this.avantank = '<div class="cwp-fondo-zoom"></div>'; // Máscara negra al hacer zoom
```

## 2.2. Backend

Wordpress nos proporciona un framework con librerías propias creadas en PHP y un conjunto de filtros y acciones que nos van a resultar de gran ayuda a la hora de desarrollar plugins para este entorno.

Estas funciones están testeadas y optimizadas para funcionar con Wordpress y debemos utilizarlas siempre que sea posible (manual buenas prácticas de desarrollo en Wordpress). En caso de necesitar alguna funcionalidad que no podamos realizar con ellas recurriremos a desarrollar una a medida en PHP.

En el desarrollo de plugins, themes y funcionalidades debemos evitar en lo posible modificar el core de Wordpress. La buena práctica es utilizar los filtros y acciones disponibles para ejecutar y modificar el contenido mediante funciones. Esto evitará conflictos en futuras actualizaciones, ya que Wordpress está en continua evolución y se actualiza frecuentemente.

Pasamos a ver la estructura del backend (entre paréntesis el archivo en el que han sido creados):

# Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto:



JUNTA DE EXTREMADURA

Consejería de Educación y Empleo



Añadimos al menú de administración de Wordpress:

- Tipo de contenido 'Plato'.  
(includes/schema/cpt-platos/cpt.platos.php).
  - Taxonomía 'Secciones' que nos va a servir para categorizar los platos. Por ejemplo: bebidas, tapas, entrantes...  
(includes/schema/cpt-platos/taxonomies.platos.php)
  - Taxonomía 'Alérgenos' para asignarle a cada plato sus alérgenos. Le vamos a crear también un campo personalizado para subir el icono del alérgeno.  
(includes/schema/cpt-platos/taxonomies.platos.php)
- Esta taxonomía va a tener por defecto al instalar el plugin los 14 alérgenos de obligada información por ley con sus correspondientes iconos.  
(includes/schema/cpt-platos/alergeno.terms.php)

Cuando entramos en añadir plato nos aparecerán los campos que hemos creado para dicho CPT:

Hemos registrado el CPT platos para que solo tenga asociados los campos de Wordpress que vamos a utilizar:

- Título.
- Descripción.
- Imagen.

Las taxonomías que hemos comentado en el punto anterior:

- Secciones.
- Alérgenos.

Y un campo personalizado:

- Precio.  
(includes/schema/cpt-platos/cf.platos.php)

Estos serán los campos que guardaremos en la base de datos de Wordpress cada vez que creamos un nuevo plato para después poderlos mostrar en el frontend.

Otras de las opciones añadidas en el backend es la posibilidad de que cuando creamos un nuevo usuario en Wordpress le podemos asignar un perfil 'Editor de carta', que hemos creado y que solo tiene permisos para editar la carta.  
(includes/admin/add.rol.carta.php)

# Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto:



JUNTA DE EXTREMADURA

Consejería de Educación y Empleo

El usuario que tenga asignado este perfil en vez de ver el menú del administrador verá el de este perfil. En la imagen se puede ver en primer lugar el menú de administrador seguido del menú de ‘Edito de carta’. Este último solo tiene permisos para editar y subir nuevos platos, editar medios para subir imágenes de los platos y a su perfil.

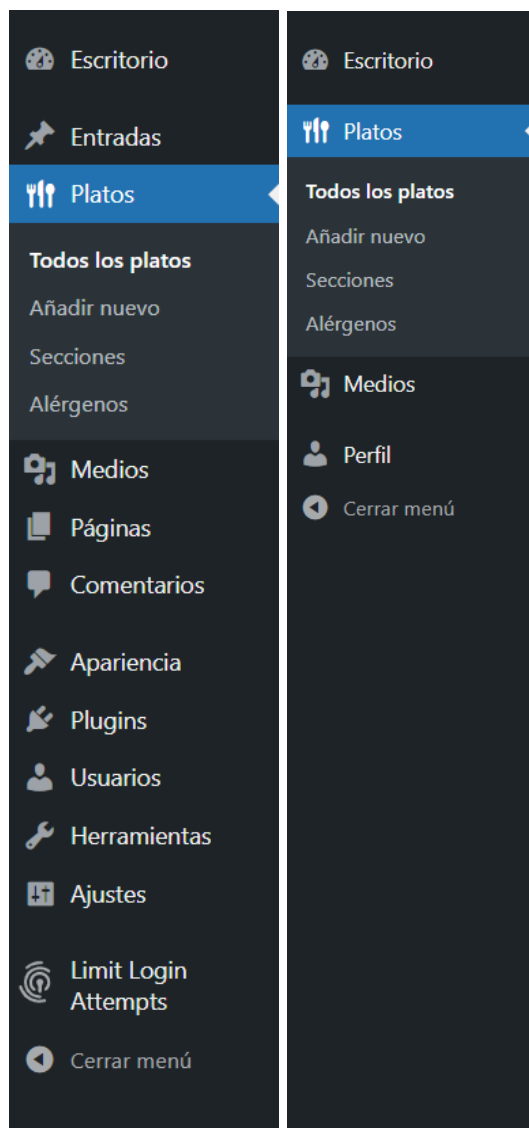
# Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto:



JUNTA DE EXTREMADURA

Consejería de Educación y Empleo



## 2.2.1. Tecnologías usadas

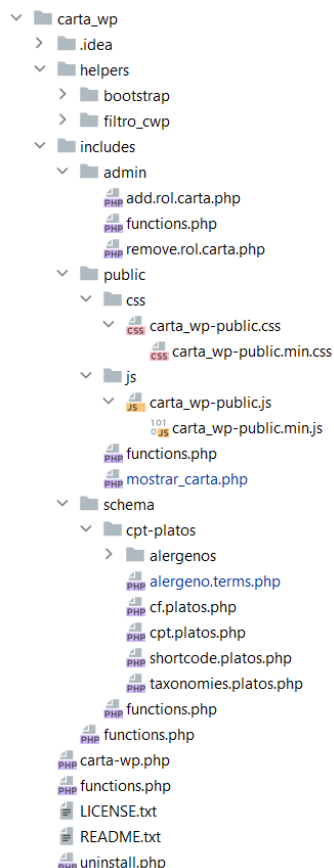
Para el diseño del backend hemos utilizado las siguientes tecnologías:

- HTML 5
- CSS 3
- PHP: funciones propias y de Wordpress

## 2.2.2. Entorno de desarrollo

Para todo el desarrollo del plugin he utilizado el IDE PHPStorm y para pruebas y depuración del backend, el modo DEBUG de Wordpress y el plugin Query monitor.

## 2.2.3. Estructura



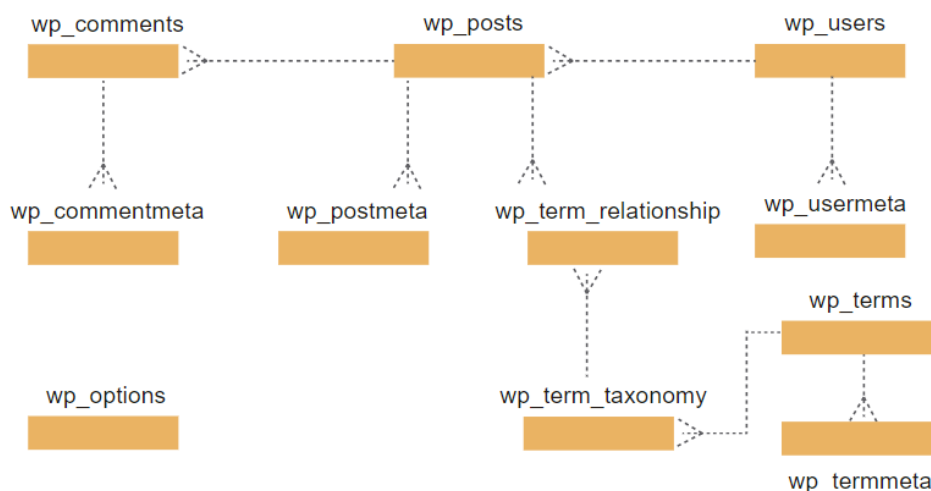
El plugin presenta la siguiente estructura:

- Raíz: en el situamos el archivo principal carta-wp.php, en el que va la cabecera para que Wordpress reconozca que se trata de un plugin. Desde aquí cargamos los scripts y estilos y hacemos los includes de los demás archivos.
- Helpers: incluye las librerías externas que vamos a utilizar, en nuestro caso Bootstrap y el plugin de filtrado que hemos desarrollado.
- Includes: que lo dividimos en admin (archivos que afectan a la parte del admin), public (archivos que afectan a la parte pública del sitio) y schema (archivos que afectan a la estructura interna de wordpress)

## 2.3. Base de datos

Para almacenar los datos de nuestra carta vamos a utilizar las tablas ya existentes en la base de datos de Wordpress.

Es una base de datos relacional formada inicialmente por 11 tablas





# Proyecto “Desarrollo de Aplicaciones Web”

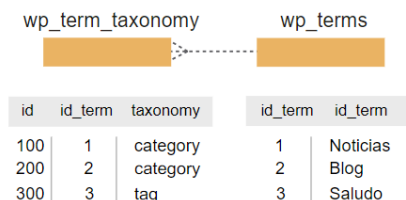
Título del Proyecto:



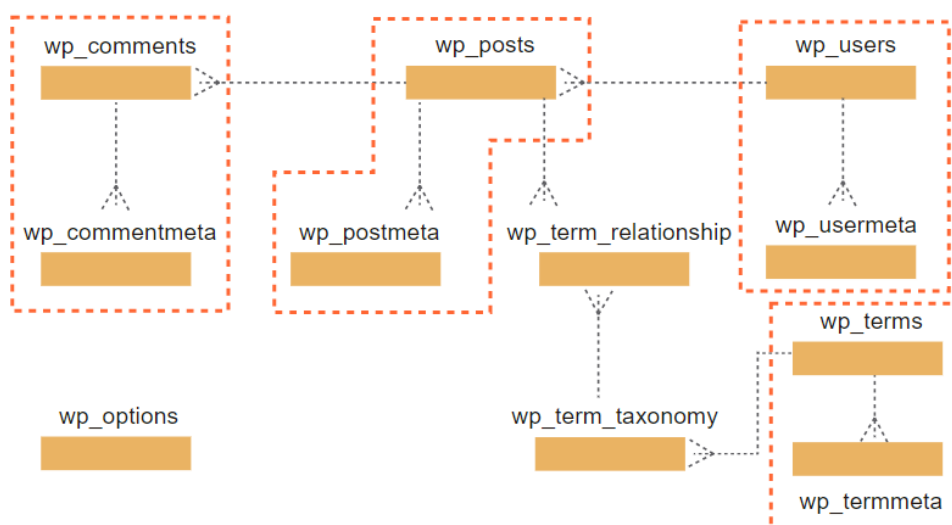
JUNTA DE EXTREMADURA

Consejería de Educación y Empleo

Podemos ver un ejemplo de cómo se relacionan unas tablas con otras:



Cada tabla principal va a tener una asociada de metadatos, que va a utilizarse para ampliar la cantidad de campos que puede tener un registro particular (además de los que trae Wordpress por defecto)



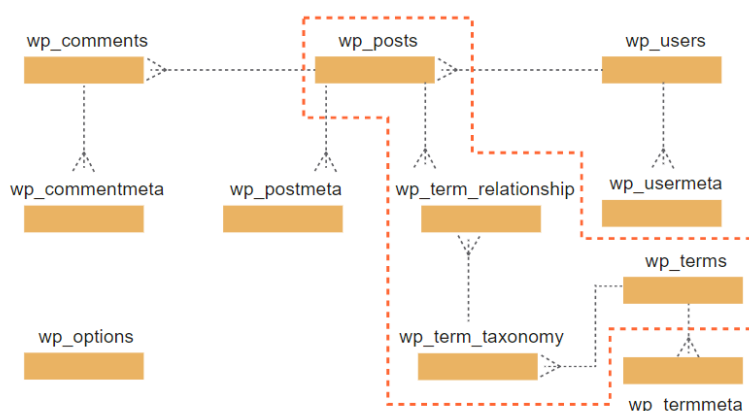
Haremos un repaso de los datos que se almacenan en cada tabla:

- **wp\_options:** configuraciones generales tanto de Wordpress como de plugins. En nuestro caso almacenaremos los permisos de usuarios relativos al CPT ‘platos’ y del rol ‘Editor de carta’.
- **wp\_comments:** los comentarios del sitio, además incluye campos de monitoreo, como lp del usuario, fecha, etc.
- **wp\_commentmeta:** metadatos de los comentarios, por ejemplo datos relacionados a la aprobación del comentario.
- **wp\_users:** los usuarios (correo, clave, usuario, etc).
- **wp\_usermeta:** metadatos del usuario: información personal adicional, personalizaciones del backend, configuraciones de plugins a nivel de usuario, etc.).
- **wp\_posts:** todo el contenido de nuestro sitio: entradas, páginas, menús, etc., tiene campos como: título, contenido, tipo, etc. En nuestro plugin utilizamos esta tabla para almacenar la información de los platos relativa al título, contenido e imagen destacada. Y las propias imágenes (tanto de los platos como las de los iconos de alérgenos) que es un tipo de contenido ‘attachment’.
- **wp\_postmeta:** información adicional para cada registro de contenido. En

nuestro caso almacenaremos el valor del campo precio.

- **wp\_terms:** nombre de las taxonomías del contenido. Aquí guardaremos los nombres de los términos de cada taxonomía. Al instalar el plugin almacenaremos el nombre de los 14 alérgenos (correspondientes a la taxonomía alérgenos) y las secciones que vaya creando el usuario para clasificar los platos (taxonomía secciones)
- **wp\_termmeta:** datos adicionales para cada nombre de taxonomía. Situaremos los datos referentes al campo personalizado ‘alergeno-imagen’ de la taxonomía alérgenos correspondiente al icono de cada uno.

Para relacionar la tabla wp\_post donde se almacena todo el contenido del sitio, con su término de taxonomía (tabla wp\_terms) existen **tablas intermedias**.



- **wp\_term\_taxonomy:** almacena la relación de cada término con su taxonomía. En nuestro caso almacenamos a qué taxonomía (alérgenos, secciones) pertenecen los términos (gluten, huevos, sulfitos, bebidas, tapas, raciones...)
- **wp\_term\_relationship:** relaciona cada post con sus términos. Nosotros almacenaremos la relación entre los platos y sus términos (Por ejemplo un plato puede tener 3 términos (gluten, huevos y tapas) pertenecientes a distintas taxonomías)

En la siguiente página podemos ver el diagrama completo de la base de datos de wordpress:

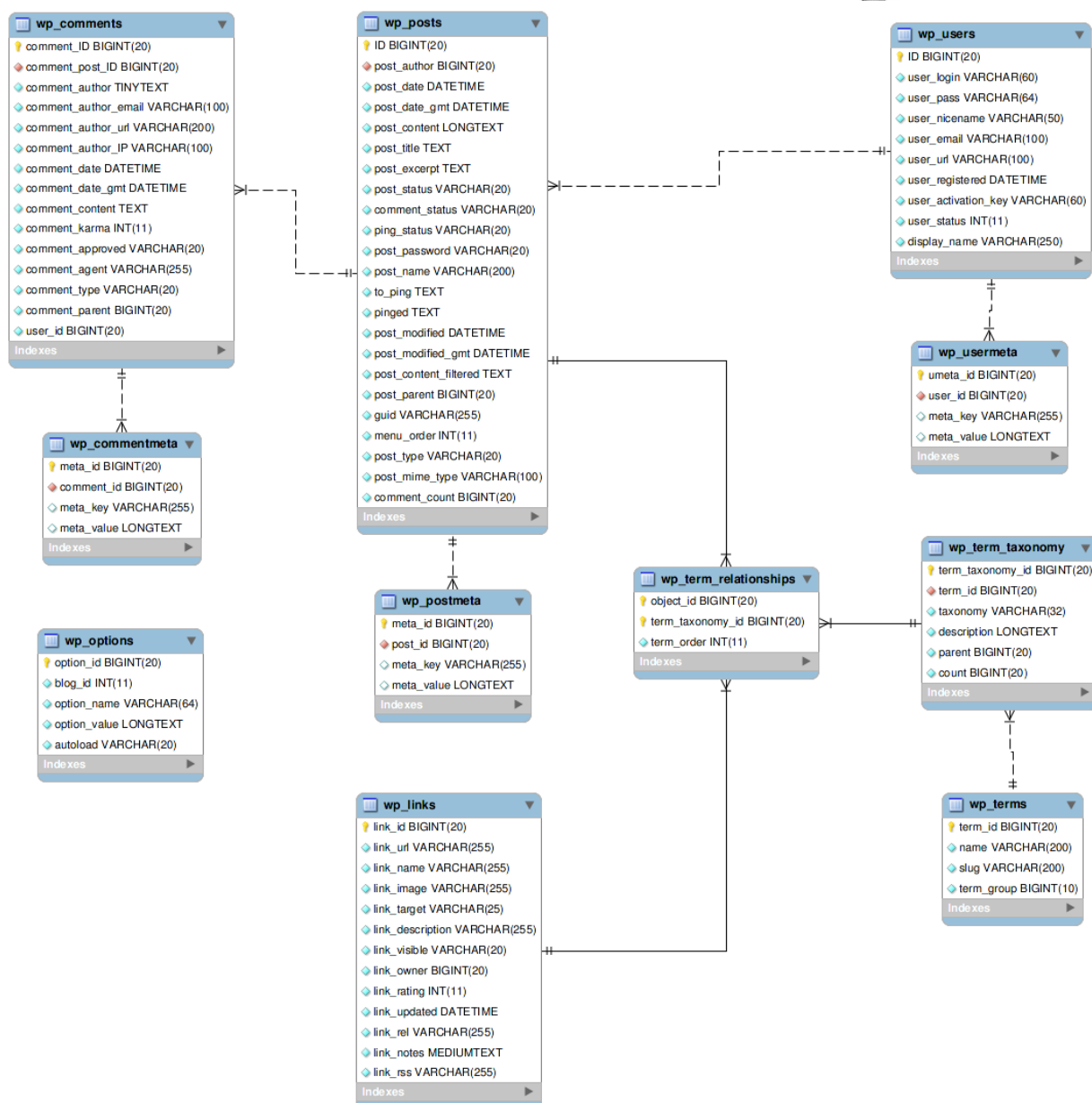
# Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto:



JUNTA DE EXTREMADURA

Consejería de Educación y Empleo



## 3. Documentación técnica

### 3.1. Análisis

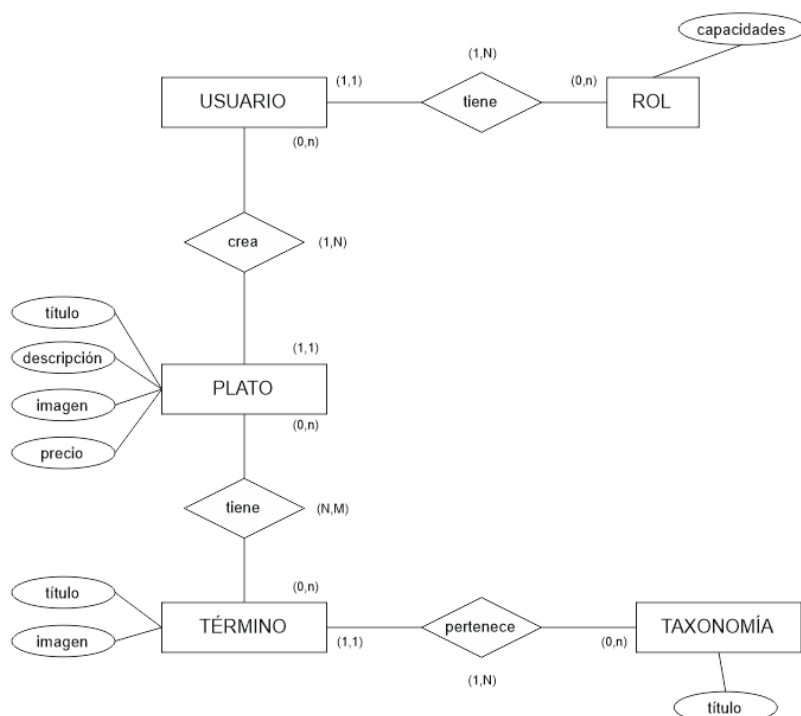
#### Requisitos funcionales:

- El plugin creará en Wordpress un Custom Post Type “Plato”.
- El plugin debe crear una taxonomía “secciones” para el CPT Plato.
- El plugin debe crear una taxonomía “alérgenos” para el CPT Plato.
- El plugin debe crear un campo personalizado de tipo imagen para los términos de la taxonomía “alérgenos”.
- El plugin creará 14 términos de la taxonomía ‘alérgenos’ correspondientes a los 14 alérgenos de obligada información con sus correspondientes iconos.
- El CPT “Plato” tendrá 1 campos personalizados (precio) solo tres de los propios de Wordpress (título, descripción e imagen).
- Debe haber un rol de usuario que al identificarse solo pueda acceder a modificar la carta.
- El usuario de Wordpress debe poder introducir, borrar y modificar tantos platos como quiera.
- La carta se tiene que poder insertar en la página de Wordpress que queramos mediante un shortcode.
- El usuario final debe poder filtrar los platos mostrados en la web por secciones.

#### Requisitos no funcionales:

- El plugin tendrá un diseño mobile first (diseñado para móviles).
- Debe seguir la guía de estilos de programación de Wordpress.
- Debe seguir las directrices y buenas prácticas sobre programación para Wordpress.

El diagrama E/R para nuestro caso sería el siguiente:



Que tendremos que adaptar a la base de datos de Wordpress como hemos visto en el apartado 2.3

## 3.2. Desarrollo

El **ciclo de vida** de este plugin va a seguir el modelo iterativo incremental, donde las fases se repiten y refinan, y van propagando su mejora a las fases siguientes. Usaremos este modelo porque nos va a permitir que el sistema no esté completo para ir haciendo uso de él y a lo largo de su vida podemos ir realizando mejoras y ampliaciones.

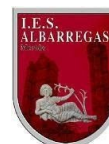
## 3.3. Pruebas realizadas

Durante el desarrollo del plugin hemos utilizado las siguientes herramientas para la realización de pruebas y detección de errores:

- **Consola de Google Chrome** para ver errores de javascript y peticiones http, así como para hacer pruebas en el código html y css antes de implementarlas.
- **Modo debug de Wordpress**, que nos va a permitir mostrar errores de PHP e imprimir datos de depuración y prueba. Todo esto aparecerá en el fichero debug.log.

# Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto:

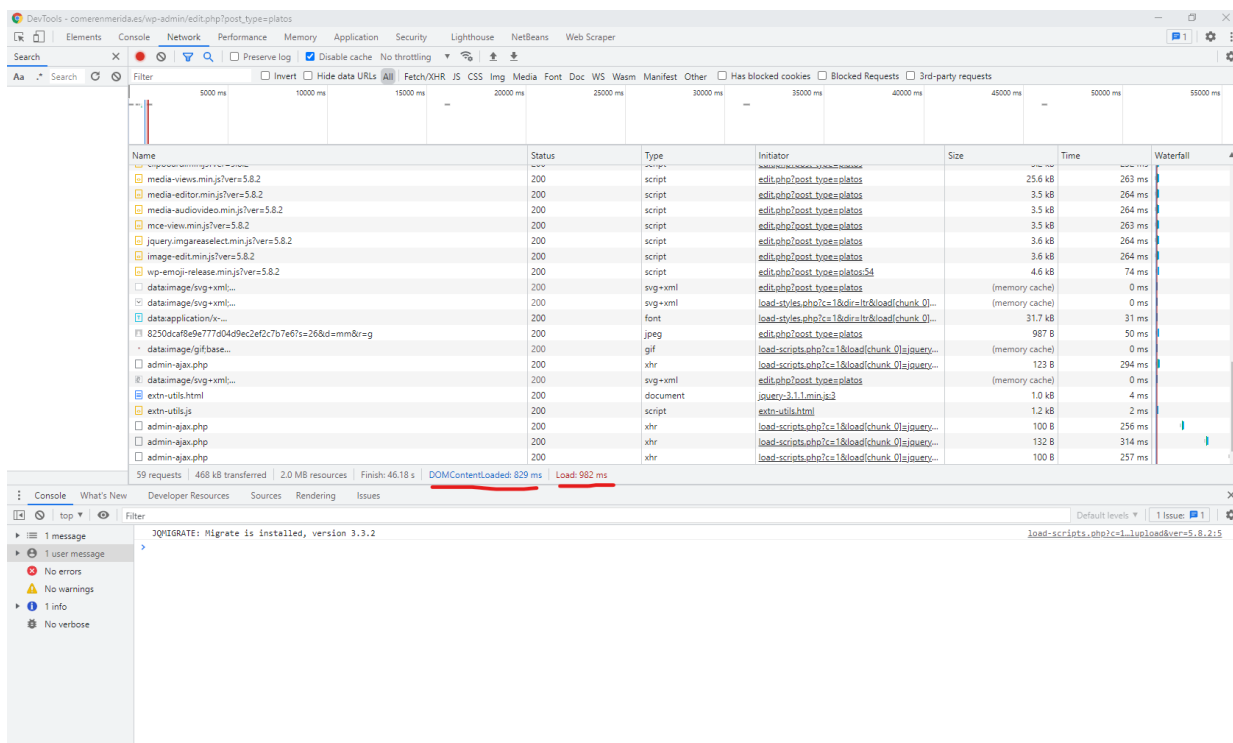


JUNTA DE EXTREMADURA

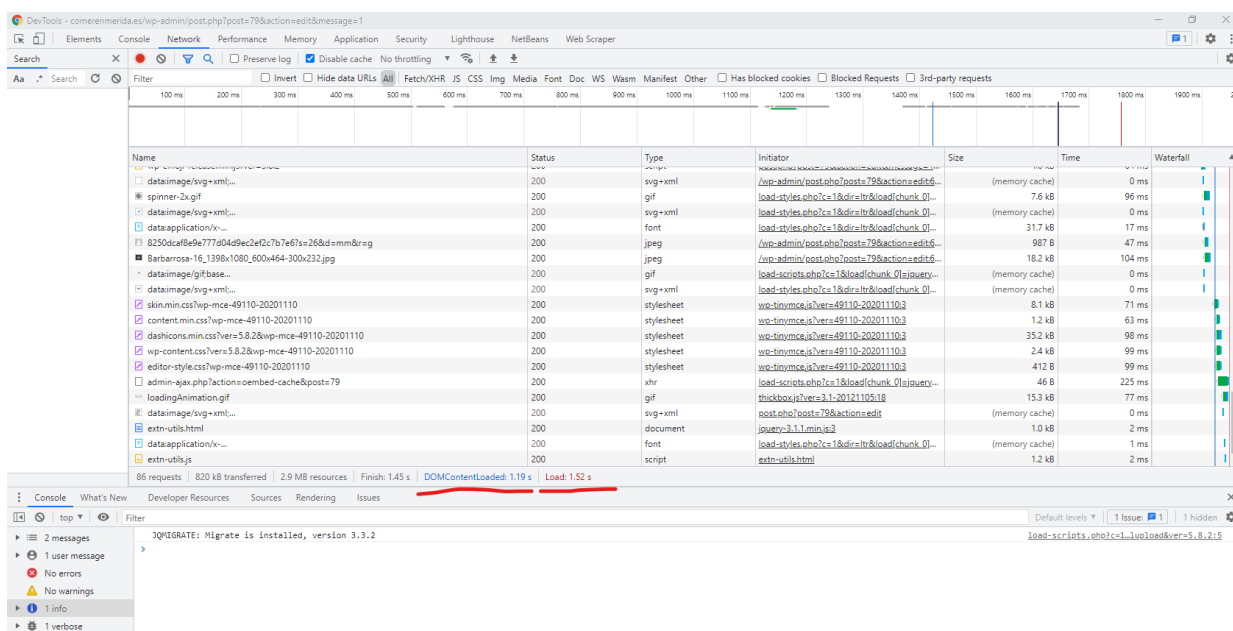
Consejería de Educación y Empleo

Una vez instalado el plugin en un servidor remoto, introduzco 50 platos con sus correspondientes datos y realizo diferentes pruebas para detectar posibles errores y tiempos de carga con la consola de Chrome

1.- En la parte de administración, en el listado de todos los platos: no aparecen errores ni advertencias, todas las peticiones tienen un status 200 y el tiempo de carga es inferior a 1s.



2.- En la parte de administración, probamos a introducir un plato y a editar un plato y tampoco nos aparece ningún error, todas las peticiones tienen status 200 y el tiempo de carga es ejecución es de 1,5s





# Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto:



JUNTA DE EXTREMADURA

Consejería de Educación y Empleo

3.- En la parte del frontend, probamos la carga de la página donde mostramos la carta e igualmente vemos que no nos aparecen errores js y las peticiones se realizan correctamente. El DOM completo tarda 1s en estar visible para el visitante y la carga total de la página 1,5s. Realizamos diferentes filtrados y tampoco nos aparece ningún error.

Seguidamente utilizo un plugin de Wordpress “**Query monitor**”, muy usado para la depuración de desarrollos en esta plataforma. Nos permite inspeccionar todas las consultas SQL, errores PHP, hooks y filtros usados, solicitudes http, scripts, estilos, datos del servidor php y mysql y las condicionales usadas para mostrar la página

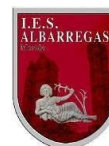
Realizamos diferentes pruebas en el backend y el frontend y vemos que tampoco nos aparece ningún error y los tiempos de realización de las consultas son buenos.

The screenshot displays the 'Comer en Mérida' website's 'Carta' (Menu) page. The page features a dark green header with the site name and a navigation bar with tabs for 'Secciones', 'Todo', 'Sugerencias', 'Bebidas', 'Vinos', 'Entrantes', 'Pizzas', 'Pescados', 'Carnes', and 'Postres'. Below the navigation bar, the 'Query Monitor' plugin is open, showing performance metrics for the page load. The metrics include: Tiempo de generación de página (0.1771s, 0.6% of 30s limit), Pico de uso de memoria (33.726 kB, 6.4% of 524.288 kB limit), Consultas de base de datos (0.0088s, SELECT: 137, SHOW: 1, UPDATE: 2, Total: 140), Llamadas API HTTP (Ninguno), and Caché de objetos (91.7% hit rate, 3.645 hits, 328 misses). The bottom part of the screenshot shows the 'wp-admin/edit.php?post\_type=platos' page, which lists the menu items. The table has columns for 'Titulo', 'secciones', 'alergenos', and 'Fecha'. The items listed are 'Cerveza de barril Estrella Galicia' and 'Cerveza SIN alcohol Heineken 1/3', both categorized as 'Bebidas'.

Titulo	secciones	alergenos	Fecha
Cerveza de barril Estrella Galicia	Bebidas	—	Publicada 07/12/2021 a las 16:25
Cerveza SIN alcohol Heineken 1/3	Bebidas	—	Publicada

# Proyecto “Desarrollo de Aplicaciones Web”

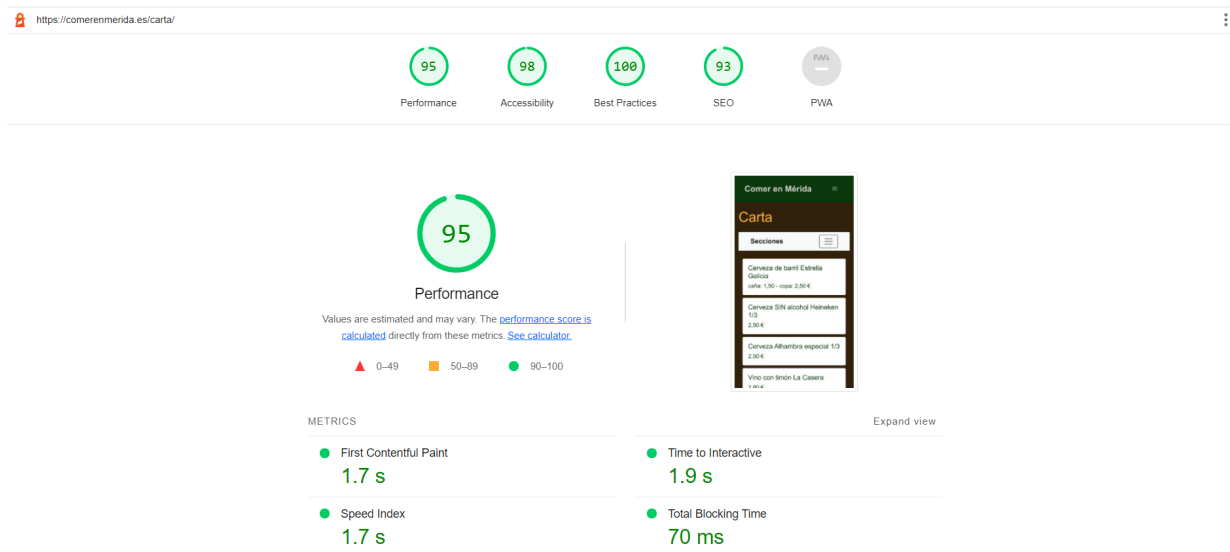
Título del Proyecto:



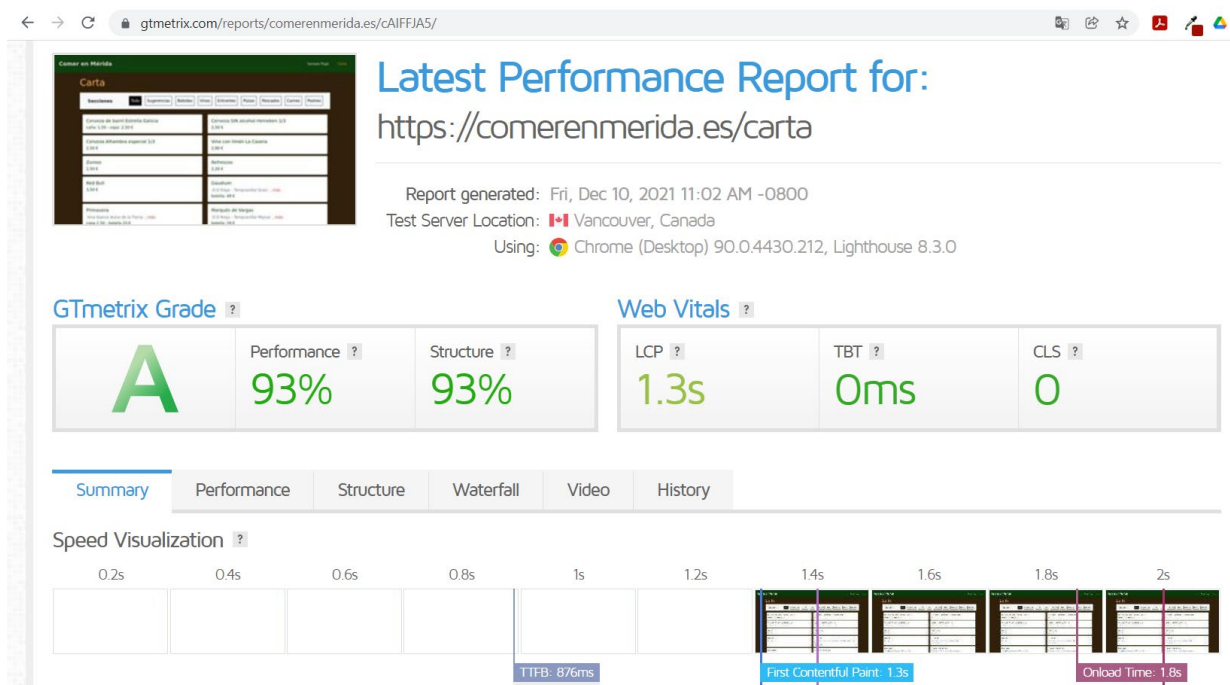
JUNTA DE EXTREMADURA

Consejería de Educación y Empleo

A continuación procedemos a **medir el rendimiento** y optimización de la página donde se muestra la carta con la herramienta de Google Lighthouse. Y comprobamos muy buenos resultados sin aplicar ningún tipo de herramienta de optimización adicional



Realizamos otra prueba de rendimiento con GTmetrix y vemos también buenos datos



# Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto:

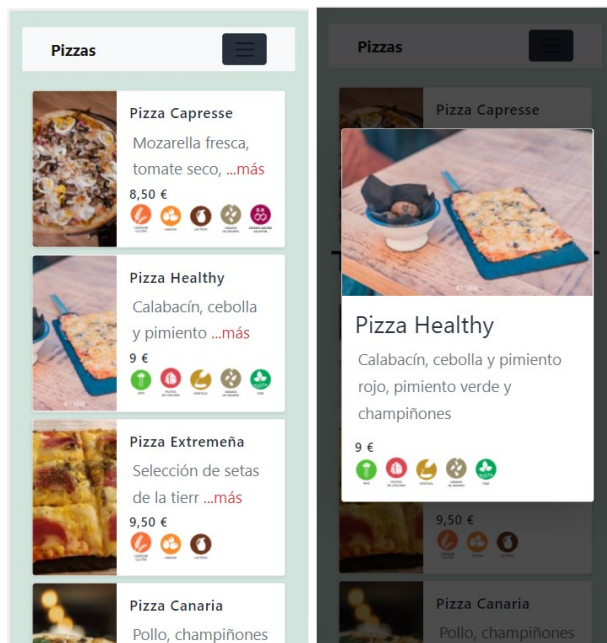


JUNTA DE EXTREMADURA

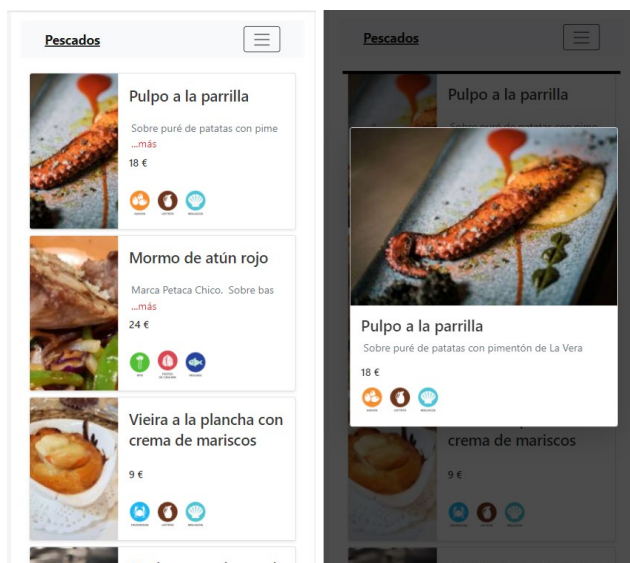
Consejería de Educación y Empleo

Todas las pruebas anteriores las hemos realizado con el tema que utilizo siempre para crear webs en Wordpress: Generatepress. Ahora procedemos a **probar nuestro plugin con los temas más populares de Wordpress** para ver si va todo bien:

## Twenty Twenty-one



## Astra



# Proyecto “Desarrollo de Aplicaciones Web”

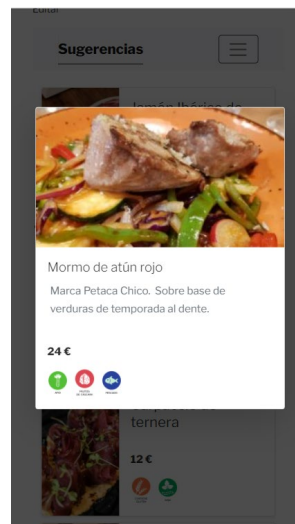
Título del Proyecto:



JUNTA DE EXTREMADURA

Consejería de Educación y Empleo

## Twenty seventeen



## OceanWP



## Kadence



# Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto:



JUNTA DE EXTREMADURA

Consejería de Educación y Empleo

Por último, realizo la **instalación del plugin en varios sitios en producción** donde tengo instalados varios plugins y tampoco observo ningún error.

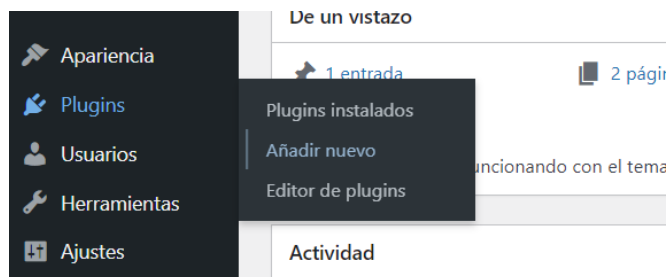
## 4. Proceso de despliegue

Para realizar la **instalación** de nuestro plugin tenemos que contar con una instalación de wordpress. Está probado con versiones a partir de la 5.0.

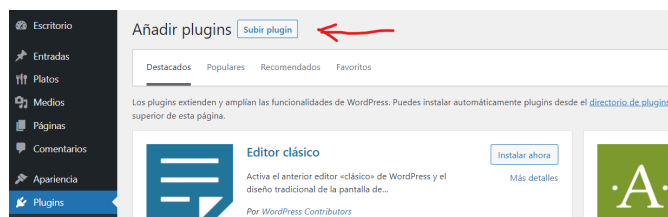
Debemos seguir el mismo proceso que para instalar cualquier otro plugin. Podemos hacerlo de dos formas:

- Desde el panel de administración de Wordpress:

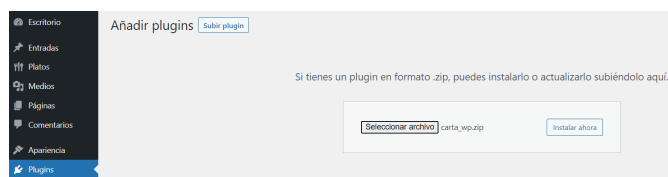
1. En el menú lateral vamos a Plugins > añadir nuevo



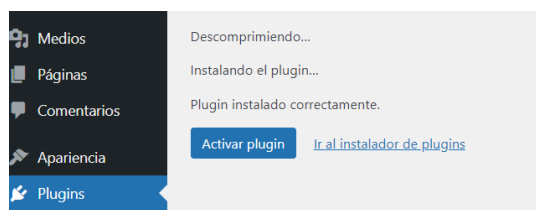
2. Seleccionamos “Subir plugin”



3. Subimos el archivo carta\_wp.zip



4. Activamos el plugin



# Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto:

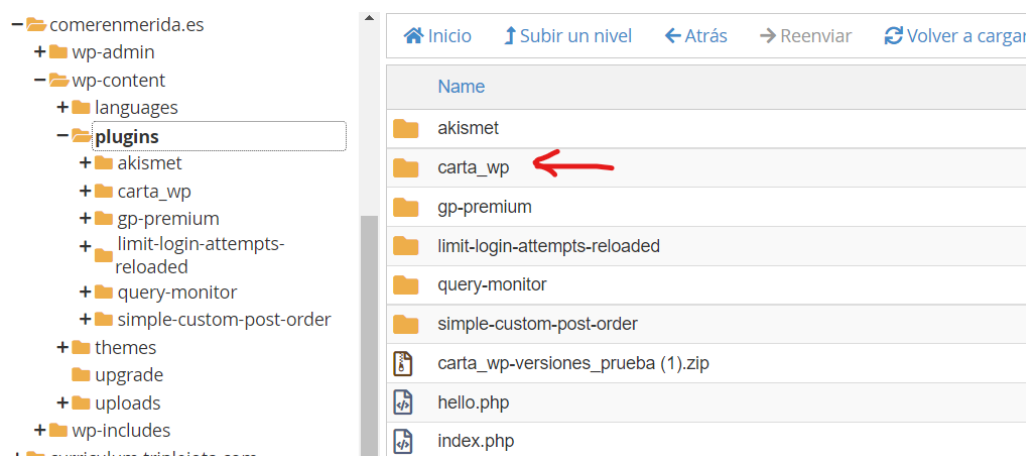


JUNTA DE EXTREMADURA

Consejería de Educación y Empleo

- **Mediante ftp:**

1. Descomprimos el archivo carta\_wp y subimos la carpeta carta\_wp al directorio de la instalación de wordpress: wp.content/plugins



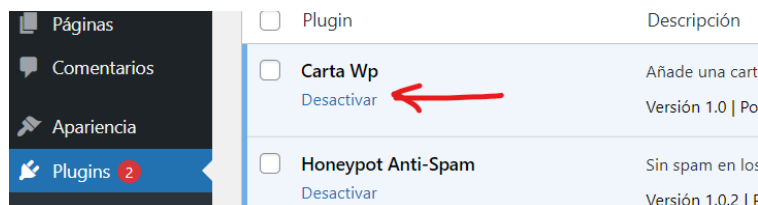
2. Una vez instalado tenemos que activarlo. En el menú de administración de wordpress entramos en plugins y pulsamos sobre “activar”



**Desinstalacion:** para desinstalar el plugin se recomienda hacer desde el administrador de wordpress, ya que si lo hacemos borrando directamente la carpeta por ftp no se van a ejecutar las acciones que hemos programado para borrar datos relativos al plugin de la base de datos.

Procederemos de la siguiente forma:

1. Vamos al listado de plugins y lo desactivamos





# Proyecto “Desarrollo de Aplicaciones Web”

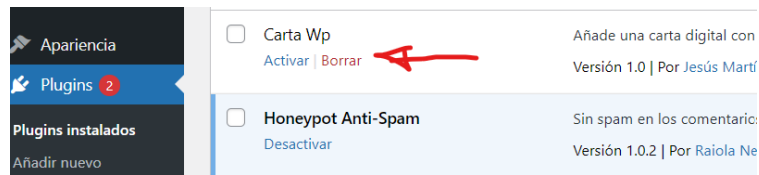
Título del Proyecto:



JUNTA DE EXTREMADURA

Consejería de Educación y Empleo

2. Una vez desactivado ya nos aparece la opción de borrarlo.



## 5. Propuestas de mejoras

Hasta esta parte del desarrollo el plugin ya estaría funcional y preparado para subir al repositorio de plugins de Wordpress para poder ser usado en cualquier instalación de manera gratuita.

Para la monetización de nuestro plugin vamos a usar el modelo freemium, es decir, base 100% funcional gratuita y extensiones con opciones adicionales de pago.

Todas **las mejoras que vamos a realizar irán destinadas a realizar extensiones de pago**. Pasamos a detallar:

- **Ordenación personalizada** de todos los platos y secciones para controlar de forma totalmente personalizada el orden en el que aparecen en la carta.

Hay plugins gratuitos dedicados específicamente a la ordenación del contenido que funcionan con el nuestro (personalmente utilizo ‘Simple Custom Post Order’), pero sería interesante incorporarlo de forma nativa.

- **Personalización de aspecto**. Número de columnas que queremos mostrar en la carta en cada dispositivo, color de las secciones, fuentes y tamaños de texto, tamaño de la imagen, etc.
- **Personalizar las acciones realizadas al borrar el plugin**. Las buenas prácticas de desarrollo Wordpress indican que cuando se borra un plugin hay que eliminar de la base de datos todo el contenido creado, aunque en la actualidad la mayoría no cumplen este estándar, ya que si borramos el plugin por error, o nos arrepentimos, al volver a instalarlo tendríamos de nuevo todo el contenido disponible sin tener que volver a introducirlo.

Para evitar los dos extremos anteriores, la forma de hacerlo sería que al desinstalar el plugin nos vaya preguntando los datos relativos a éste que queremos eliminar de la base de datos.

La versión gratuita de nuestro plugin solo elimina algunas cosas de la base de datos, como los roles y capacidades creados, ya que al instalarlo se volverían a crear y podríamos seguir utilizando el plugin. Pero no borra otros datos, como los platos que hemos introducido.

- **Opción de multicarta**. Sería útil para directorios de restaurantes o cadenas y franquicias. Nos permitiría crear diferentes cartas que podríamos mostrar mediante diferentes shortcodes en diferentes urls.

Pongamos el caso de una franquicia “Supertaco”, podría tener en el mismo dominio las cartas de todos sus restaurantes. P.ej: <https://supertaco.es/merida-carta> mostraríamos la carta del restaurante de Mérida, en <https://supertaco.es/badajoz-carta> mostraríamos la carta del restaurante de Badajoz.

# Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto:



Consejería de Educación y Empleo

- **Módulo de pedidos** dentro del propio restaurante para que la comanda llegue directamente a la barra.
- **Módulo de pedidos para servicio a domicilio.**

## 6. Bibliografía

<https://developer.wordpress.org/>

<https://developer.wordpress.org/plugins/>

<https://studiofrea.com/wordpress/how-to-upload-images-to-wordpress-programmatically/>

<https://webdesign.tutsplus.com/es/tutorials/create-a-custom-wordpress-plugin-from-scratch--net-2668>

<https://code.tutsplus.com/es/tutorials/create-a-wordpress-image-gallery-code-the-plugin--cms-32093>

<https://wordpress.stackexchange.com/questions/3396/create-custom-page-templates-with-plugins>

[https://codex.wordpress.org/Database Description](https://codex.wordpress.org/Database_Description)

<https://wordpress.org/support/topic/delete-unused-capabilities-left-over-from-old-plugin/>

<https://developer.wordpress.org/plugins/users/roles-and-capabilities/>

<https://developer.wordpress.org/plugins/plugin-basics/uninstall-methods/>

<https://ayudawp.com/uninstall-php/>

<https://decodecms.com/>

<https://www.php.net/manual/es/index.php>

<https://getbootstrap.com/docs/5.1/getting-started/introduction/>

<https://jquery.com/>