

## ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA



### TRABAJO FIN DE GRADO

## Plataforma Web Configurable basada en Plantillas y Plugins de WordPress para la Generación de Aplicaciones Móviles en iOS y Android

**Titulación: Grado en Ingeniería en Tecnologías de la  
Telecomunicación**

**Autor: Samuel Martín Rodríguez**

**Tutores: Dr. Luis Hernández Acosta**

**Fecha: Enero 2020**

## ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA



### TRABAJO FIN DE GRADO

### Plataforma Web Configurable basada en Plantillas y Plugins de WordPress para la Generación de Aplicaciones Móviles en iOS y Android

### HOJA DE EVALUACIÓN

**Calificación:** \_\_\_\_\_

**Presidente**

**Vocal**

**Secretario/a**

Fdo.:

Fdo.:

Fdo.:

**Fecha: Enero 2020**

# Índice

<b>1. INTRODUCCIÓN</b> .....	11
1.1. Objetivos .....	12
1.2. Estructura de la memoria .....	12
<b>2. TECNOLOGÍAS SOFTWARE</b> .....	15
2.1. WordPress .....	15
2.1.1. <i>Plugins</i> .....	15
2.1.2. <i>Temas</i> .....	15
2.1.3. <i>Widgets</i> .....	15
2.1.4. <i>Tema de WordPress, Escapade</i> .....	16
2.2. Toolset .....	16
2.2.1. <i>Toolset Types</i> .....	16
2.2.2. <i>Toolset Views</i> .....	16
2.2.3. <i>Toolset Forms</i> .....	16
2.2.4. <i>Toolset Layouts</i> .....	17
2.2.5. <i>Toolset Maps</i> .....	17
2.3. REST API Controller .....	17
2.4. Framework IUMATI .....	17
2.5. MySQL .....	18
2.6. Google My Maps .....	19
<b>3. DESCRIPCIÓN GENERAL</b> .....	23
3.1. Diagrama de bloques .....	23
<b>4. ANÁLISIS</b> .....	27
<b>5. DISEÑO</b> .....	31

5.1.	Diseño de la web.....	31
5.2.	Tablas MySQL de WordPress .....	40
5.2.1.	<i>Esquema de la base de datos de WordPress</i> .....	40
5.2.2.	<i>Estructura de las tablas de WordPress</i> .....	42
5.3.	Tablas MySQL de Toolset.....	50
5.3.1.	<i>Esquema de la base de datos de Toolset</i> .....	50
5.3.2.	<i>Estructura de las tablas de Toolset</i> .....	51
<b>6.</b>	<b>IMPLEMENTACIÓN</b> .....	<b>57</b>
6.1.	Modificaciones aplicadas sobre WordPress.....	57
6.1.1.	<i>Ajustes generales de WordPress</i> .....	57
6.1.2.	<i>Ajustes de lectura</i> .....	58
6.1.3.	<i>Ajustes de Enlaces permanentes</i> .....	58
6.2.	Modificaciones aplicadas sobre el tema de WordPress Escapade .....	59
6.2.1.	<i>Identidad del sitio</i> .....	59
6.2.2.	<i>Colores</i> .....	59
6.2.3.	<i>Fuentes</i> .....	60
6.2.4.	<i>Medios del encabezado</i> .....	61
6.2.5.	<i>Ajustes de portada</i> .....	61
6.3.	Implementación llevada a cabo en Toolset.....	61
6.3.1.	<i>Configuración de Toolset</i> .....	61
6.3.2.	<i>Creación de la estructura y taxonomías de la página web</i> .....	62
6.3.3.	<i>Generación de las rutas</i> .....	78
6.3.4.	<i>Generación de la página inicio</i> .....	79
6.3.5.	<i>Generación de la página ayuda</i> .....	80
6.3.6.	<i>Generación de la página añadir rutas</i> .....	81
6.3.7.	<i>Formulario para subir las rutas</i> .....	82

6.3.8.	<i>Layout para rutas</i> .....	85
6.3.9.	<i>Layout plantilla para subir rutas</i> .....	87
6.3.10.	<i>Modificaciones realizadas con el plugin REST API Controller</i> .....	89
<b>7.</b>	<b>Conclusiones y líneas futuras</b> .....	97
7.1.	Introducción .....	97
7.2.	Conclusiones .....	97
7.3.	Líneas futuras.....	97
<b>8.</b>	<b>Presupuesto</b> .....	101
8.1.	Recursos humanos .....	101
8.2.	Hardware.....	101
8.3.	Software .....	102
8.4.	Material fungible.....	102
8.5.	Presupuesto total del proyecto .....	103
<b>9.</b>	<b>Bibliografía</b> .....	107
10.1.	Guía de instalación de WordPress.....	113
10.2.	Instalación de Toolset.....	117
10.3.	Guía de usuario.....	119

# Índice de figuras

Figura 1. Diagrama de funcionamiento del Framework IUMATI .....	18
Figura 2. Diagrama de bloques.....	24
Figura 3. Página “Inicio” del sitio web.....	31
Figura 4. Página “Rutas” del sitio web.....	32
Figura 5. Menú “Islas” .....	32
Figura 6. Página “Isla - Fuerteventura” .....	33
Figura 7. Menú “Dificultad”.....	33
Figura 8. Página “Dificultad - Difícil” .....	34
Figura 9. Página “Dificultad – Experto” sin rutas .....	34
Figura 10. Página “Ayuda” 1.....	35
Figura 11. Página “Ayuda” 2.....	35
Figura 12. Formulario de subida, parte 1.....	36
Figura 13. Formulario de subida, parte 2.....	36
Figura 14. Formulario de subida, parte 3.....	37
Figura 15. Formulario de subida, parte 4.....	37
Figura 16. Ruta ya subida a la web, parte 1.....	38
Figura 17. Ruta ya subida a la web, parte 2.....	38
Figura 18. Ruta ya subida a la web, parte 3.....	39
Figura 19. Ruta ya subida a la web, parte 4.....	39
Figura 20. Ruta ya subida a la web, parte 5.....	39
Figura 21. Diagrama de la base de datos de WordPress.....	41
Figura 22. Estructura de la tabla wp_commentmeta .....	42
Figura 23. Estructura de la tabla wp_comments.....	42
Figura 24. Estructura de la tabla wp_options .....	44
Figura 25. Estructura de la tabla wp_postmeta.....	44

Figura 26. Estructura de la tabla wp_posts.....	45
Figura 27. Estructura de la tabla wp_termmeta.....	46
Figura 28. Estructura de la tabla wp_terms .....	47
Figura 29. Estructura de la tabla wp_term_relationships .....	47
Figura 30. Estructura de la tabla wp_term_taxonomy.....	48
Figura 31. Estructura de la tabla wp_usermeta.....	49
Figura 32. Estructura de la tabla wp_users.....	49
Figura 33. Esquema de la base de datos de Toolset .....	51
Figura 34. Estructura de la tabla wp_toolset_associations .....	51
Figura 35. Estructura de la tabla wp_toolset_post_guid_id .....	52
Figura 36. Estructura de la tabla wp_toolset_relationships.....	53
Figura 37. Estructura de la tabla wp_toolset_type_sets .....	54
Figura 38. Estructura de la tabla wp_toolset_maps_address_cache.....	54
Figura 39. Tabla resumen de los ajustes generales de WordPress .....	58
Figura 40. Icono del sitio web .....	59
Figura 41. Imagen de cabecera del sitio web.....	61
Figura 42. Editar el tipo de entrada Páginas.....	62
Figura 43. Editar el tipo de entrada Posts.....	63
Figura 44. Agregar nuevo tipo de entrada Rutas .....	63
Figura 45. Agregar nuevo tipo de entrada Ayuda .....	63
Figura 46. Agregar nuevo tipo de entrada Añadir Rutas.....	64
Figura 47. Panel de control de WordPress .....	64
Figura 48. Formulario de creación de taxonomías .....	65
Figura 49. Formulario de la taxonomía Islas.....	66
Figura 50. Formulario de la taxonomía Niveles de dificultad.....	66
Figura 51. Formulario para agregar las islas .....	67

Figura 52. Formulario para agregar los niveles de dificultad.....	67
Figura 53. Agregar campo actividad .....	68
Figura 54. Agregar campo descripción.....	69
Figura 55. Agregar campo Fotos .....	69
Figura 56. Agregar campo dirección .....	70
Figura 57. Agregar campo distancia.....	70
Figura 58. Agregar campo tiempo de duración .....	71
Figura 59. Agregar campo mapa ruta .....	72
Figura 60. Opciones de pantalla .....	72
Figura 61. Añadir “Inicio” al menú .....	73
Figura 62. Añadir “Rutas” al menú .....	74
Figura 63. Añadir “Islas” al menú .....	75
Figura 64. Añadir “Dificultad” al menú .....	75
Figura 65. Añadir ”añadir rutas” al menú.....	76
Figura 66. Añadir “Ayuda” al menú.....	76
Figura 67. Estructura del menú.....	77
Figura 68. Generar rutas .....	79
Figura 69. Generación de la página “Inicio”. .....	80
Figura 70. Generación de la página “Ayuda” .....	81
Figura 71. Generación de la página “Añadir ruta” .....	81
Figura 72. Configuración del formulario subir rutas .....	82
Figura 73. Editor del formulario subir rutas .....	83
Figura 74. Crear layout.....	85
Figura 75. Elección de celdas de diseño.....	86
Figura 76. Creación del layout “Plantilla para ruta individual” .....	86
Figura 77. Selección del tipo de celda Post Form .....	88

Figura 78. Creación de celda de formulario .....	88
Figura 79. Creación del layout “Plantilla para Subir Rutas” .....	89
Figura 80. REST API Controller – Posts Types .....	90
Figura 81. Configuración del post type ruta en REST API Controller .....	91
Figura 82. REST API Controller – Taxonomías .....	92
Figura 83. Configuración de la taxonomía isla en REST API Controller .....	92
Figura 84. JSON del post type ruta .....	93
Figura 85. JSON de la taxonomía isla .....	93
Figura 86. Pantalla de prueba de Apache .....	113
Figura 87. Página de Login en phpMyAdmin .....	115

## **Índice de tablas**

Tabla 1. Costes de recursos humanos .....	101
Tabla 2. Costes de los recursos hardware .....	102
Tabla 3. Costes de los recursos software .....	102
Tabla 4. Costes del material fungible .....	103
Tabla 5. Presupuesto total del proyecto .....	103



# **Introducción**



# 1. INTRODUCCIÓN

Hoy en día el mercado de las aplicaciones móviles (Apps) está en constante crecimiento, debido a que estas facilitan el trabajo en multitud de sectores y también han cambiado los hábitos de vida y de consumo de la población.

Numerosos emprendedores y empresas han enfocado su idea de negocio a través de estas plataformas, pero en muchos casos no se tienen conocimientos técnicos especializados en este campo, y se tiene que recurrir a los servicios de un programador o una empresa especializada. Teniendo en cuenta que el coste de desarrollo de una aplicación nativa, para un sistema operativo, oscila entre los 6.000€ y los 30.000€ se suelen abaratar costes desarrollando aplicaciones híbridas que son compatibles con varios sistemas operativos, ya que el coste de estas oscila entre un 65% y un 80% menos [1].

Es por lo que el Sr. Luis Hernández Acosta y el Sr. José María Quinteiro González pertenecientes a la división de Tecnologías de la Información del IUMA, desarrollaron el Framework IUMATI. Este framework genera aplicaciones para Android e iOS, simultáneamente, en base a una metodología propia. Esta metodología ha sido diseñada en base a la funcionalidad, por lo que su estructura y entorno de desarrollo no son cómodos para el usuario común. Debido a esto, el objetivo de este Trabajo Fin de Título (TFT) será generar un sitio web específico para una aplicación concreta (caso de uso) con una interfaz web desarrollada mediante plantillas, para que terceros puedan introducir la información y configuración necesaria para el framework de una manera cómoda e intuitiva.

Para llevar a cabo el desarrollo de dicho TFT, nos apoyaremos en el gestor de contenidos WordPress, los plugins de Toolset y el plugin de WordPress REST API Controller.

Para argumentar la elección de dichas herramientas nos apoyaremos en datos ofrecidos por W3Techs [2] que indican que WordPress, en 2017, se sitúa como el gestor de contenido (CMS, Content Management System) más utilizado. Además, también es el gestor de contenido con mayor crecimiento. Esto se debe a que es una herramienta sencilla y eficaz, extensible mediante plugins, dispone de multitud de temas y es una herramienta ligera, fiable y de software libre para diseñar sitios web [3].

Por otro lado, el plugin que usaremos será Toolset debido a que nos permite crear sitios web basados en tipos de datos específicos y campos personalizados, creados ambos por el usuario, sin tener que escribir código PHP. Es decir, nos permite dotar de funcionalidades

avanzadas a nuestro sitio web basado en WordPress prácticamente sin necesidad de programar. [4]

### **1.1. Objetivos**

El objetivo principal de este proyecto es el de crear una interfaz de usuario mediante plantillas de WordPress que permita que un usuario sin conocimientos de programación de aplicaciones móviles pueda diseñar aplicaciones para dispositivos móviles basados en Android e iOS. Esta herramienta será accesible a través de un navegador web, siendo presentada como páginas web de la plataforma WordPress integrando el plugin Toolset.

- 1) Generar un sitio web de una aplicación específica basada en WordPress con los plugins anteriormente citados.
- 2) Diseñar plantillas de WordPress en las que el usuario pueda añadir información y crear su propio sitio web de una forma sencilla.
- 3) Configurar el plugin REST API Controller, para hacer posible el envío de los datos introducidos por el usuario, en el sitio web, al framework IUMATI cuando este los solicite.

### **1.2. Estructura de la memoria**

Este trabajo fin de grado sigue la estructura clásica, que se divide en cuatro secciones principales:

- Memoria, es la parte principal del TFG donde se encuentra detallado el contenido del trabajo y se explica detalladamente el procedimiento a seguir para la realización de este.
- Pliego de Condiciones, en este apartado se definen los equipos y software necesarios para el desarrollo del TFG, así como los requerimientos para su correcto funcionamiento.
- Presupuesto, se detalla el coste monetario necesario para la implementación del TFG.
- Anexos, se encuentra información adicional de utilidad para el TFG.

# **TECNOLOGÍAS**

# **SOFTWARE**



## **2. TECNOLOGÍAS SOFTWARE**

### **2.1. WordPress**

WordPress es un gestor de contenido (CMS, Content Management System) creado en 2003 por Mike Little y Matt Mullenweg basándose en b2/cafelog [5], el cual es un CMS creado en 2001 que corría sobre PHP y MySQL, que también soportaba plantillas y permitía postear vía E-mail [6]. Sin embargo, la necesidad de crear un CMS más potente, elegante y mejor estructurado los llevó a la creación de WordPress, que también está creado con PHP y MySQL y sujeto a la licencia GPLv2 (Licencia Pública General versión 2) [5].

WordPress a través del tiempo, ha mutado de ser un sistema de publicaciones a un complejo gestor de contenido que nos permite crear prácticamente cualquier tipo de sitio Web. Esto se debe a que WordPress está conformado por tres componentes [7].

#### **2.1.1. Plugins**

Los plugins son complementos software usados para aumentar las capacidades de WordPress y permitir elevar su utilidad. Es decir, son un conjunto de archivos que siguiendo unas normas dotan de funcionalidades a un sitio web de WordPress. Estos son modulares, lo que nos permite añadirlos y eliminarlos sin que el proyecto se vea afectado.

#### **2.1.2. Temas**

Los temas son plantillas usadas en WordPress para modificar la apariencia del sitio web y establecer un control sobre como se muestra el contenido. Estos nos permiten adaptar WordPress a cualquier tipo de página web. Tienen carácter modular por lo que podemos cambiar de tema sin que el contenido del sitio web se vea modificado.

#### **2.1.3. Widgets**

Los widgets son pequeños bloques de información para ayudar al usuario a tener un mayor control sobre el diseño y el contenido de su sitio web. Se utilizan en las Sidebars de WordPress y algunos temas permiten insertarlos en el pie o en el encabezado de la página.

Gracias a todo esto, WordPress aporta una gran versatilidad y es por lo que se ha convertido en el gestor de contenidos más usado actualmente.

#### **2.1.4. Tema de WordPress, Escapade**

Esta plantilla es usada en WordPress para modificar el diseño y la apariencia del sitio Web, debido a que nos permite cambiar el color y tipo de la fuente usada en nuestra web, insertar una imagen de portada y un logo, también podemos modificar el menú de nuestra web, e insertar widgets en las barras laterales, en los pies de página o en el encabezado.

## **2.2. Toolset**

Toolset es un plugin premium para WordPress creado por la empresa OnTheGoSystems, que permite al usuario agregar tipos personalizados, campos y taxonomías a cualquier sitio web independientemente del tema que se esté utilizando [4].

Los tipos y campos personalizados nos permiten desarrollar sitios web con mayor versatilidad. Toolset nos permite presentarlos en pantalla de una forma sencilla sin escribir código PHP [8].

Toolset a su vez se compone de varios subplugins.

### **2.2.1. Toolset Types**

Nos permite crear y configurar los tipos de entrada personalizados, taxonomías personalizadas, campos personalizados, categorías y etiquetas. Además, nos permite relacionar y campos y tipos de post. Es decir, podemos personalizar la administración de WordPress como el front-end sin necesidad de escribir PHP.

### **2.2.2. Toolset Views**

Muestra el contenido en el front-end del sitio, de la manera que elija el usuario. Es decir, nos permite crear y personalizar las plantillas de contenido que necesitemos. Todo esto se hace usando HTML simple y eligiendo los campos de un menú.

### **2.2.3. Toolset Forms**

Con Forms podemos crear cualquier tipo de formulario en el front-end de tu web para generar y editar contenido y usuarios. Dichos formularios pueden contener cualquier campo que pertenezca a los contenidos que se necesiten mostrar y podremos maquetarlos con HTML y CSS a nuestra conveniencia. También nos permite validar campos y enviar alertas por email una vez el usuario ha enviado el formulario.

#### **2.2.4. *Toolset Layouts***

Este plugin nos permite generar diseños adaptables para páginas enteras, yendo desde el encabezado hasta el pie de página. Debido a esto nos permite crear diseños avanzados para webs completas, utilizando un editor para arrastrar y soltar.

#### **2.2.5. *Toolset Maps***

Nos permite visualizar múltiples contenidos en Google Maps. Se puede usar tanto para cosas sencillas como mostrar una dirección, o para cosas más complejas, como podría ser mostrar los resultados de una búsqueda en un mapa [9], [10].

### **2.3. REST API Controller**

REST API Controller es un plugin de WordPress, que permite al administrador cambiar la visibilidad de todos los posts types personalizados y taxonomías para hacerlos accesibles o inaccesibles a las solicitudes de la API. Todo esto lo haremos desde una sencilla interfaz gráfica [11].

Este plugin es el que nos permite enviar la información, introducida por el usuario en la web, al Framework IUMATI en formato JSON.

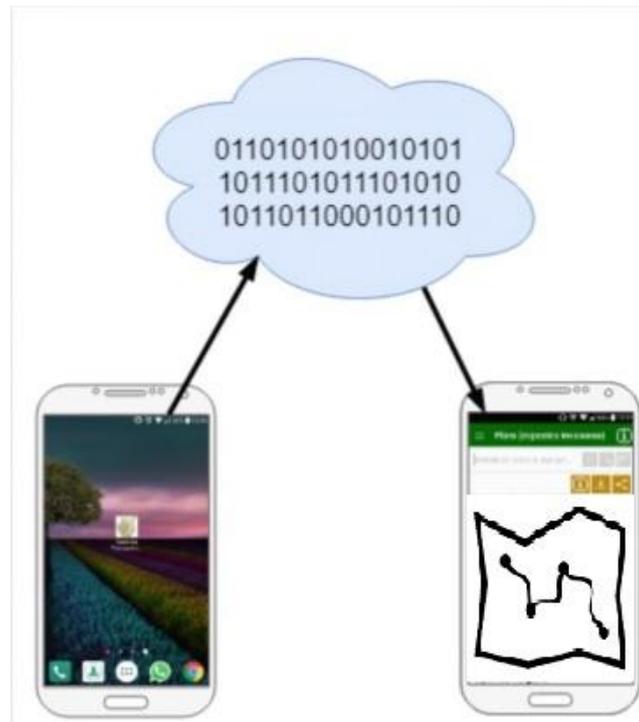
JavaScript Object Notation (JSON) es un formato basado en texto estándar, que se utiliza para transmitir datos en aplicaciones web (por ejemplo: enviar datos desde un servidor al cliente, así estos datos pueden ser mostrados en páginas web, o viceversa) [12], [13].

### **2.4. Framework IUMATI**

El framework IUMATI, creado por el Instituto Universitario de Microelectrónica Aplicada (IUMA), nos permite disponer de aplicaciones móviles actualizables en los dos principales sistemas operativos de la telefonía móvil actual, iOS y Android de forma simultánea.

Consiste en tener una aplicación móvil base para iOS y Android que se sube a la Apple Store o Google Play respectivamente. Configurada para acceder a una base de datos alojada en el Backend y así simplificar las funcionalidades necesarias para el diseño de aplicaciones. Esto nos hace posible la creación de aplicaciones de una forma más fácil y rápida para plataformas como iOS, Android o Windows [14].

Cuando el usuario inicia la aplicación accede a la base de datos, y automáticamente descarga contenido en formato JSON que incluye los datos y la configuración de la aplicación. Esto es especialmente útil para el administrador de la aplicación, debido a que podrá editar los datos cuando lo necesite, viéndose estos reflejados en la app la próxima vez que el usuario la inicie, sin necesidad de descargar ninguna actualización de la aplicación desde el store.



*Figura 1. Diagrama de funcionamiento del Framework IUMATI*

Ya existe una aplicación para dispositivos Android creada con este framework en la Play Store, llamada “Georruta Transgrancanaria” [15].

## 2.5. MySQL

MySQL es un sistema de gestión de bases de datos relacionales concebido originalmente por la compañía MySQL AB, adquirida en 2008 por Oracle. Los desarrolladores pueden seguir usando MySQL bajo la licencia pública general (GPL), pero las empresas deben obtener la licencia comercial de Oracle [16].

Gracias a su gran rendimiento, fiabilidad y facilidad de uso, es la base de datos líder en el mercado elegida por empresas como Facebook, Twitter, YouTube [17].

Una base de datos relacional archiva datos en tablas separadas en lugar de insertar todos los datos en un gran archivo. Esto le permite una mayor velocidad y flexibilidad. Dichas tablas están conectadas por relaciones definidas, que hacen que se pueda combinar datos de diferentes tablas sobre pedido.

Una de las características más interesantes de MySQL es que permite recurrir a bases de datos multi usuario a través de la web y en diferentes lenguajes de programación que se adaptan a diferentes necesidades y requisitos.

## **2.6. Google My Maps**

Google My Maps es un servicio puesto en marcha por Google en 2007. Nos permite crear mapas personalizados añadiendo puntos o dibujando formas, buscando lugares y guardándolos, importando datos desde hojas de cálculo para crear mapas y personalizarlos añadiendo iconos, fotos o videos.

Luego podemos compartir los mapas personalizados en sitios web o blogs y también consultarlos en la aplicación móvil Google Maps [18].



# **DESCRIPCIÓN GENERAL**



### **3. DESCRIPCIÓN GENERAL**

La finalidad de este proyecto es crear una interfaz para permitir que un usuario, sin conocimientos de programación, pueda generar aplicaciones móviles en iOS y Android simultáneamente. Previamente se había desarrollado el Framework IUMATI que nos da la posibilidad de generar apps para iOS y Android simultáneamente en función de los datos de contenido y configuración introducidos mediante una interfaz que el framework lee desde la nube y así configura la app resultante. Entonces se nos plantea el desarrollo de una interfaz web más sencilla para el usuario común, para una aplicación específica, y que los datos introducidos por el usuario en dicha interfaz puedan ser leídos por el Framework IUMATI con el fin de generar las aplicaciones.

Para ello desarrollamos un sitio web de rutas en las Islas Canarias, donde los usuarios rellenando unas plantillas puedan agregar sus propias rutas a la web. Todo esto se ha desarrollado bajo WordPress y el plugin Toolset, ya que la unión de ambos nos permite, de una forma sencilla, la creación de sitios webs más versátiles.

Luego, para poder enviar los datos introducidos por el usuario en la web al Framework IUMATI, utilizamos el plugin de WordPress REST API Controller, donde seleccionamos que tipos de datos estarán disponibles para enviárselos al framework en formato JSON y que este se encargue de generar la aplicación de nuestra web en Android e iOS.

#### **3.1. Diagrama de bloques**

En el diagrama de bloques de la Figura 2 podemos observar las tecnologías fundamentales del proyecto:

- WordPress como gestor de contenidos principal.
- Toolset, plugin que dota a WordPress de funcionalidades para desarrollar sitios webs con mayor versatilidad y sin necesidad de escribir código PHP.
- REST API Controller, plugin que añade a WordPress la funcionalidad de poder seleccionar y enviar la información introducida por el usuario al Framework IUMATI en formato JSON.
- MySQL es un sistema de gestión de bases de datos. Es el punto en común entre WordPress y Toolset, ya que se encarga de manejar los datos de ambos bloques.

A continuación, veremos los pasos que sigue el sistema para su funcionamiento en su respectivo orden:

- **A:** Toolset lee de la base de datos MySQL de WordPress para cargar los datos necesarios en este.
- **B:** Después de diseñar las plantillas, los formularios y las rutas en Toolset, este lo almacena en la base de datos y publica la web de rutas.
- **C:** El usuario rellena los formularios para la creación de las rutas a través del navegador web y esta información (datos de la ruta) la recibe Toolset.
- **A:** Después de procesar los datos de la ruta enviados por el usuario Toolset los escribe en la base de datos.
- **D:** El plugin REST API Controller lee la información de la base de datos. Y permite al administrador del sistema elegir cual de esos datos estarán disponibles para las solicitudes de la API, a su vez estos datos serán los que necesite el Framework IUMATI para su correcto funcionamiento.
- **E:** El plugin REST API Controller envía la información al Framework IUMATI en formato JSON.
- **F:** Finalmente, el Framework IUMATI se encarga de generar las aplicaciones para iOS y Android en función de los datos JSON.

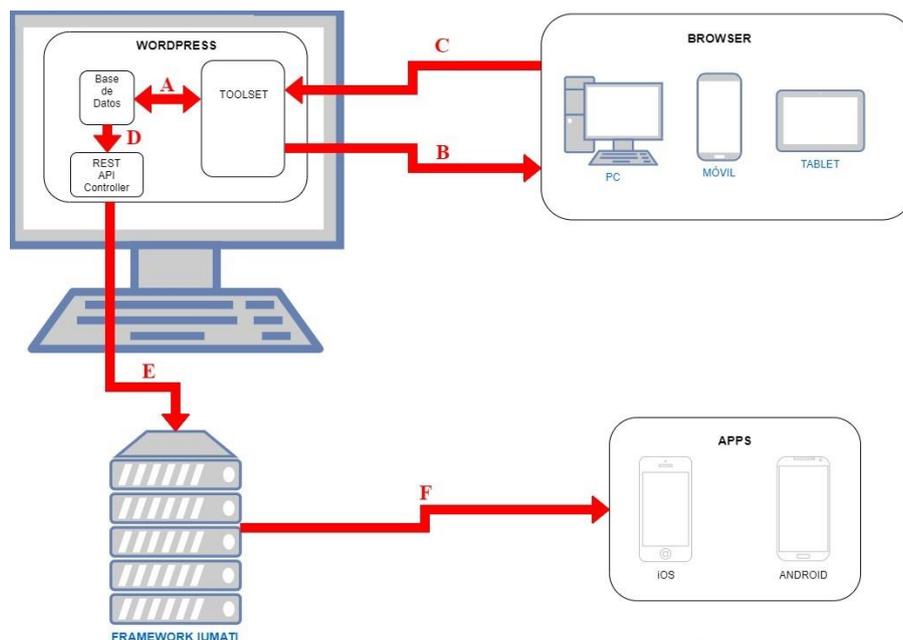


Figura 2. Diagrama de bloques

# ANÁLISIS



## 4. ANÁLISIS

En este apartado nos centraremos en el bloque principal de este proyecto, WordPress.

En sus inicios, solo servía para hacer blogs y por lo tanto solo existían los “Posts”. Con el tiempo, los desarrolladores vieron el potencial de esta herramienta y crearon las “Páginas” o lo que es lo mismo “*Post Type*”.

Las páginas son similares a los posts, pero con algunas diferencias. Como puede ser el que no tienen fecha, categorías, ni etiquetas.

Posteriormente se añadieron a WordPress los “*Custom Post Types*” (CPTs) en español “Tipos de Posts Personalizados”. Estos nos permiten agregar nuevos contenidos a WordPress, además de las entradas (*Posts*) y las páginas (*Post Types*).

Existen tres formas de crear un *Custom Post Types*:

- 1) A través de un plugin que ya contenga sus propios CPTs.

Solo tendremos que instalar y activar el plugin para tener los CPTs del plugin en nuestra instalación de WordPress.

- 2) A través de un plugin de creación de CPTs.

Este tipo de plugins nos suelen ofrecer una interfaz gráfica a través de la cual podemos crear nuestros propios CPT para WordPress y poder usarlos para añadir contenido a nuestro sitio web.

- 3) A través de código.

Aquí el usuario tendría que desarrollar su propio código para crear sus CPTs [19].

Es por todo esto que hemos decidido utilizar el conjunto de plugins Toolset para WordPress debido a que contiene un plugin de creación de CPTs como es Toolset Types y otros plugins que nos dotan de una mayor versatilidad para el desarrollo del proyecto, además de disminuir la dificultad a la hora de realizar varias tareas en el mismo.



# DISEÑO

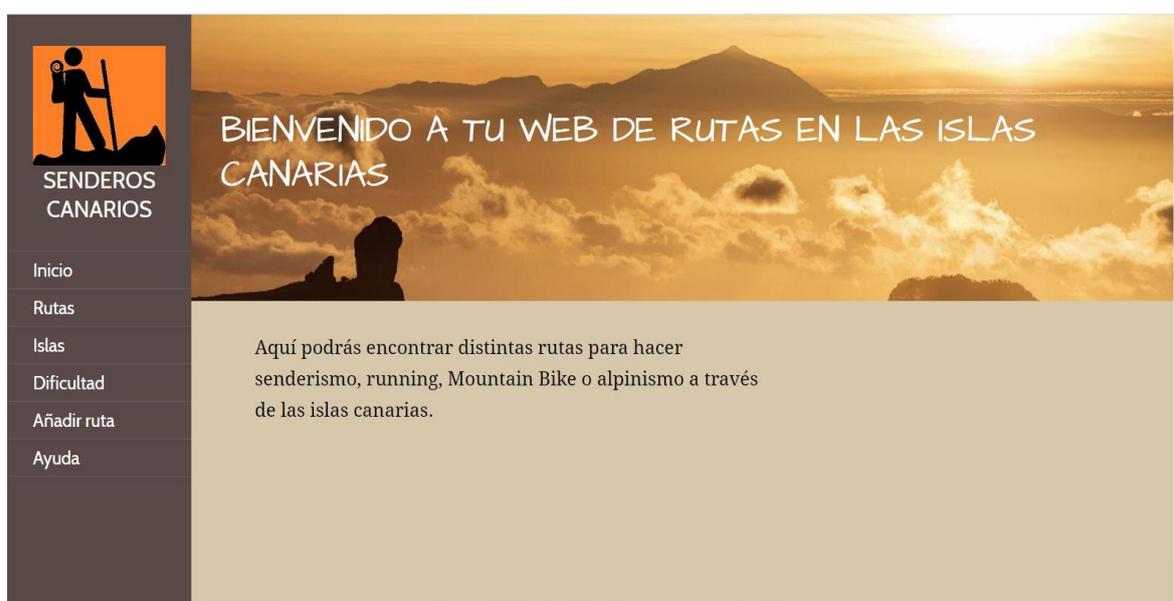


## 5. DISEÑO

### 5.1. Diseño de la web

En este punto, estudiaremos el diseño de nuestro sitio web. Para el desarrollo de la web hemos optado por un menú en forma de barra lateral izquierda donde podremos encontrar las distintas secciones de la web como son: “Inicio”, “Rutas”, “Islas”, “Dificultad”, “Añadir ruta” y “Ayuda”. A la derecha de dicho menú tendremos la sección principal, donde encontraremos la información suministrada en cada sección de la web. Encima de la sección principal hallaremos la sección “titulo” donde aparece el título de las diferentes secciones

Todo esto lo podremos ver con mayor detalle en la *Figura 3*.



*Figura 3. Página “Inicio” del sitio web*

A continuación, veremos el diseño de la web en todas sus secciones.

En la *Figura 4* podremos apreciar el diseño con el que se encuentra el usuario al entrar en el apartado “Rutas”. Y donde este puede seleccionar la ruta a la que quiere acceder.

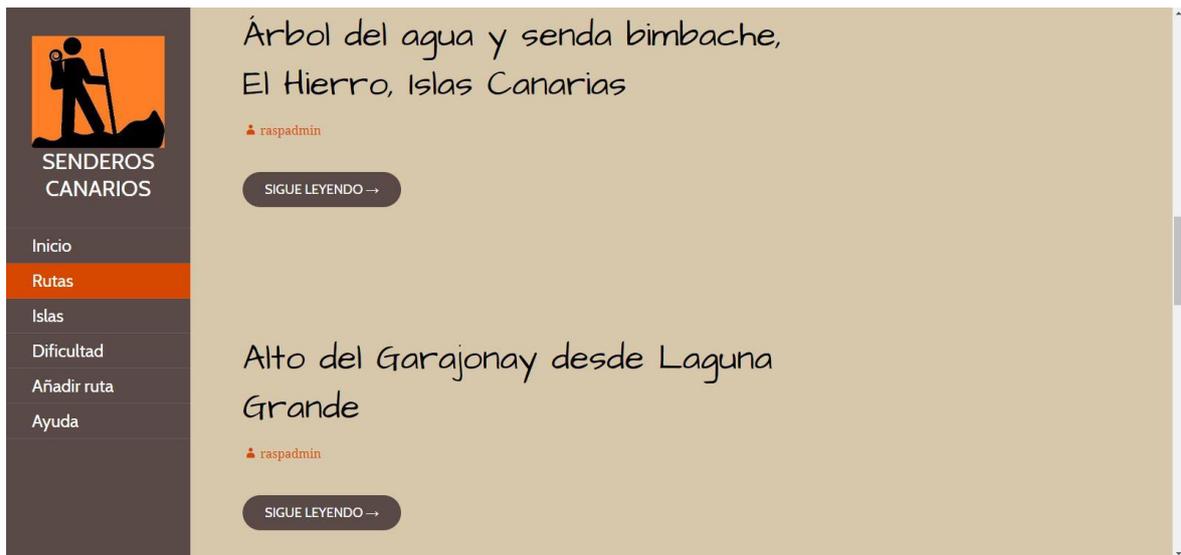


Figura 4. Página “Rutas” del sitio web

En la siguiente *Figura 5* vemos el submenú del apartado “Islas”, donde el usuario puede seleccionar la isla de la que quiere ver las rutas.



Figura 5. Menú “Islas”

En la *Figura 6* podemos apreciar en el título el nombre de la isla seleccionada y en la sección principal vemos las distintas rutas que se encuentran en dicha isla.



Figura 6. Página "Isla - Fuerteventura"

A continuación, podemos observar en la *Figura 7* el submenú del apartado "Dificultad", donde el usuario elige la dificultad de las rutas que desea visualizar.



Figura 7. Menú "Dificultad"

En la *Figura 8*, observamos que en el título se encuentra el nivel de dificultad seleccionado por el usuario. Y en el apartado principal vemos las rutas que tienen la dificultad elegida.



Figura 8. Página “Dificultad - Difícil”



Figura 9. Página “Dificultad – Experto” sin rutas

En la *Figura 9* vemos la pantalla que se encuentra el usuario a la hora de elegir una dificultad que no tiene asociada ninguna ruta. Esta pantalla es muy similar a la que se encontraría un usuario al seleccionar una isla que no tenga ninguna ruta asociada, en este caso solo cambiaría el título de la página.

En la *Figura 10*, veremos la pantalla de ayuda de la web. En la sección de cabecera visualizamos el título “Ayuda”, ya que es donde se encuentra el usuario en ese momento y en la sección principal observamos la información relativa a la ayuda de la web.



Figura 10. Página “Ayuda” 1

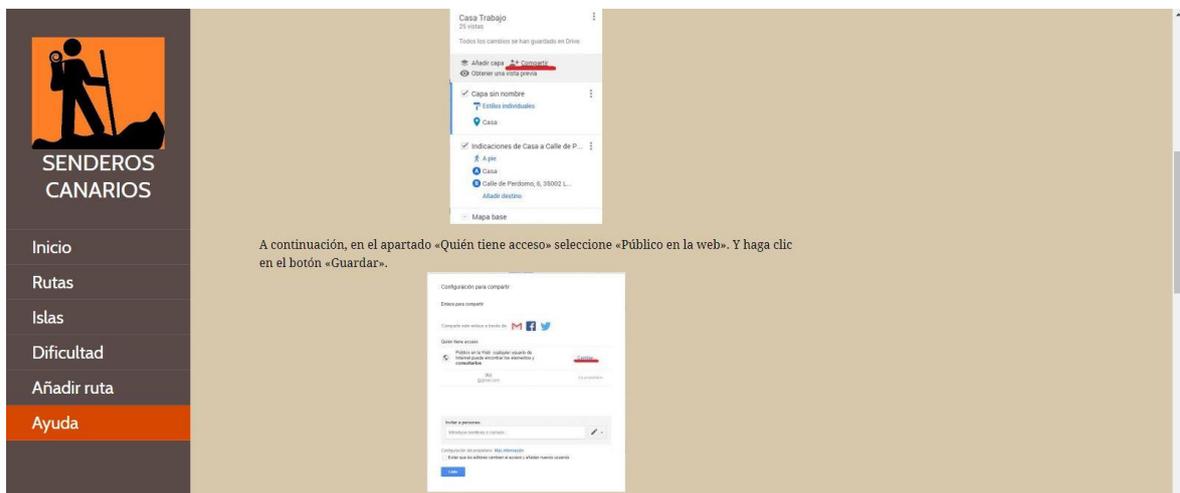


Figura 11. Página “Ayuda” 2

Como podemos observar en la *Figura 11*, también mostramos imágenes en la sección principal para una mayor información de ayuda al usuario.

En las *Figuras 12, 13, 14 y 15* podemos ver el diseño del formulario que se encuentra el usuario al ir a la sección “Añadir ruta”. Dicho formulario se compone de:

- Un campo de texto donde el usuario tiene que introducir el nombre de la ruta.
- Un desplegable donde el usuario tiene que seleccionar la isla en la que se encuentra situada la ruta.
- Un desplegable donde se tiene que seleccionar el nivel de dificultad de la ruta a cargar.

- Un *checkbox* donde se puede seleccionar la actividad a realizar en la ruta.
- Un campo de texto donde el usuario tiene que insertar una descripción de la ruta.
- Un campo para cargar fotos en la web, donde el usuario puede subir fotos de la ruta.
- Un campo de dirección donde el usuario tiene que introducir la dirección donde se encuentra el punto inicial de la ruta. Y a continuación se mostrará dicha dirección en un mapa.
- Un campo de texto donde se puede insertar la distancia de la ruta.
- Un campo de texto donde se puede insertar la duración de la ruta.
- Un campo de texto donde el usuario tiene que insertar la URL del sitio donde se encuentre alojado el mapa de la ruta a subir.

Figura 12. Formulario de subida, parte 1

Figura 13. Formulario de subida, parte 2

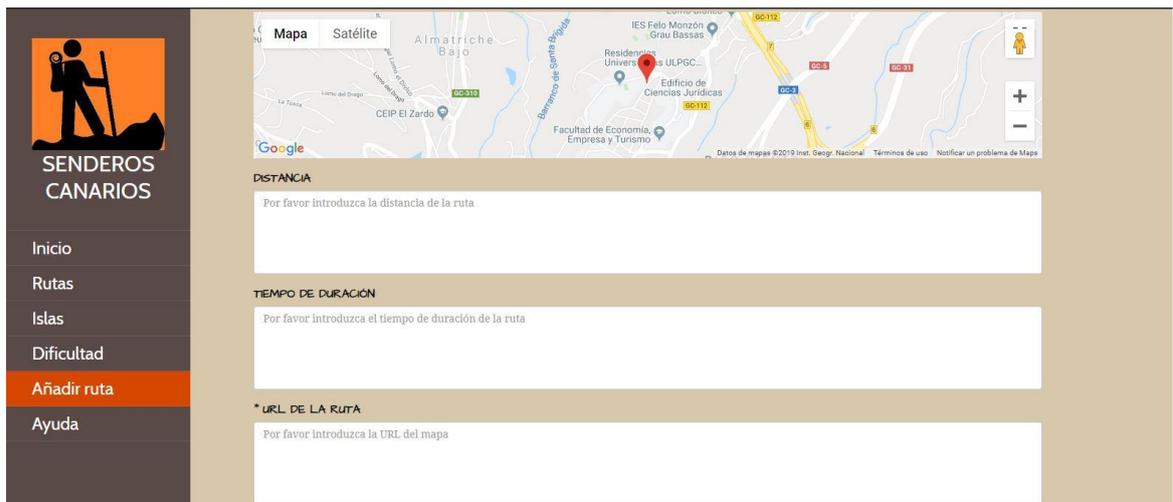


Figura 14. Formulario de subida, parte 3

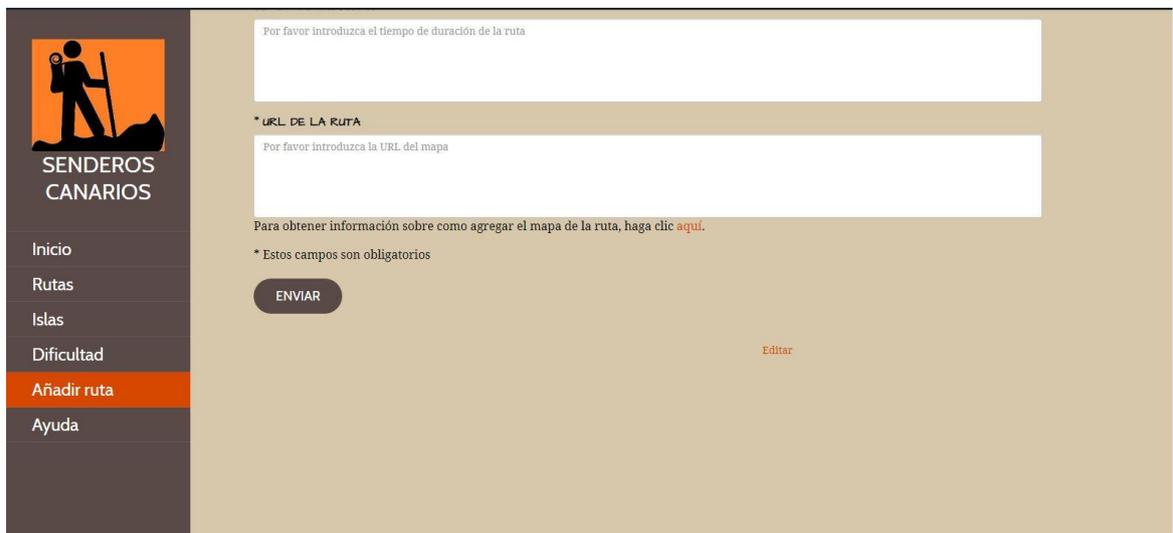


Figura 15. Formulario de subida, parte 4

En las Figuras 16, 17 y 18, procedemos a ver el diseño de una ruta ya subida al sitio web.



Figura 16. Ruta ya subida a la web, parte 1

Como se puede ver en la *Figura 16* se encuentra el “Nombre”, la “Actividad a realizar”, el “Nivel de dificultad”, la “Distancia”, el “Tiempo de duración”, la “Isla” y la “Dirección” de la ruta.

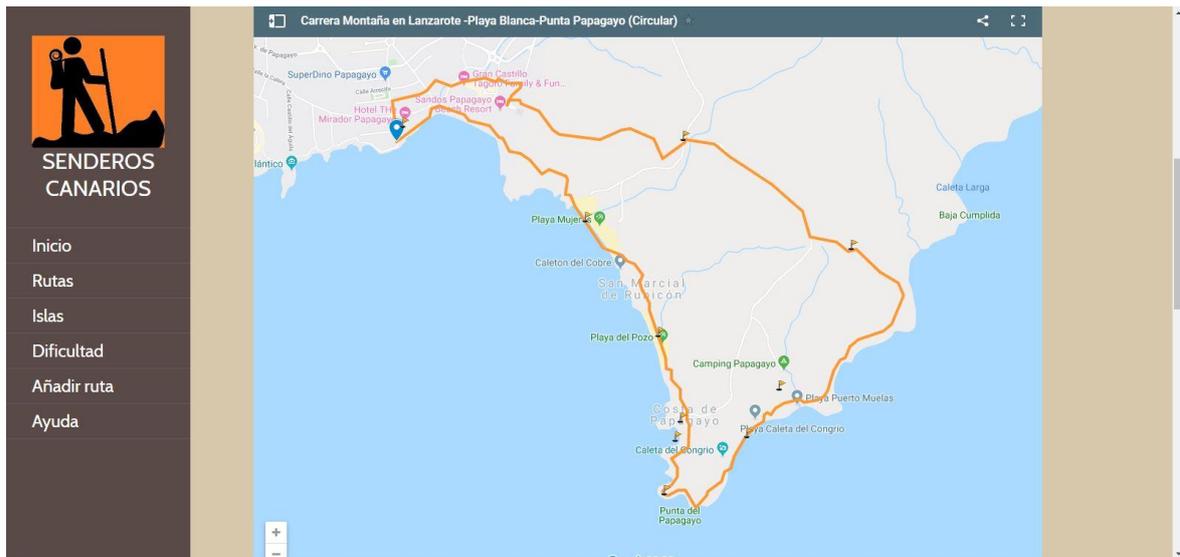


Figura 17. Ruta ya subida a la web, parte 2

En la *Figura 17* tenemos el mapa de la ruta y en la *Figura 18* hallamos el mismo mapa, pero esta vez, con un *checkpoint* activado.

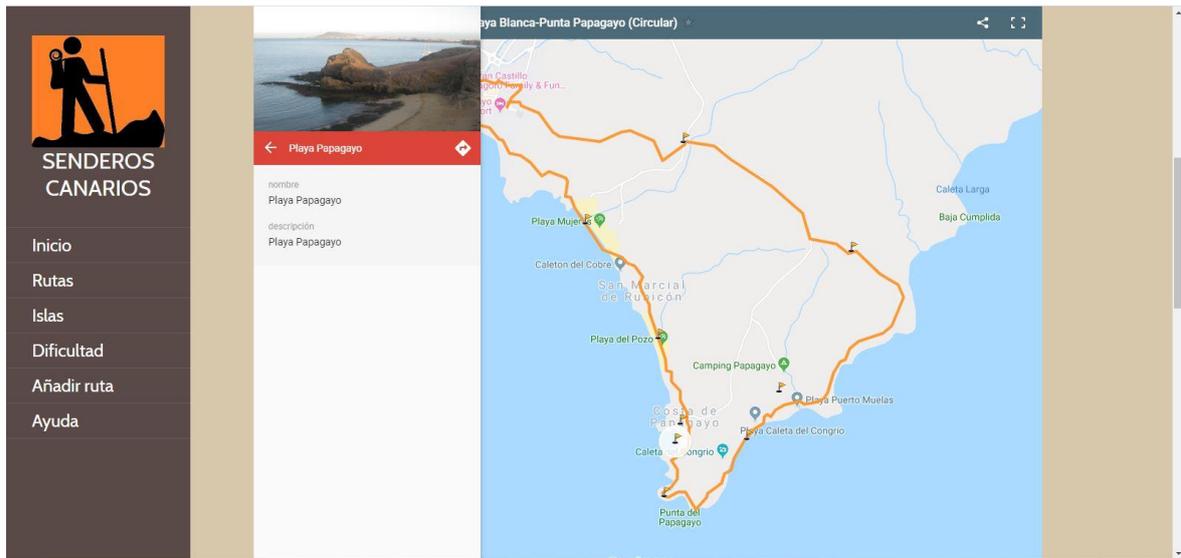


Figura 18. Ruta ya subida a la web, parte 3



Figura 19. Ruta ya subida a la web, parte 4

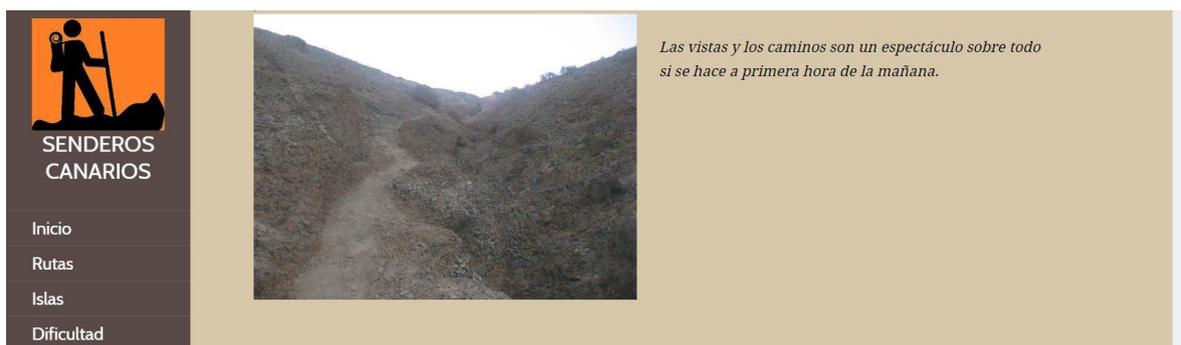


Figura 20. Ruta ya subida a la web, parte 5

En la Figura 19 y 20 podemos apreciar las fotos y la descripción de la ruta.

## **5.2. Tablas MySQL de WordPress**

### **5.2.1. Esquema de la base de datos de WordPress**

Al instalar WordPress, este nos crea por defecto 12 tablas iniciales en la base de datos. En la *Figura 21* podemos apreciar un esquema de las tablas que crea WordPress inicialmente y las relaciones entre estas [20].

Debemos tener en cuenta que los plugins y temas también pueden crear tablas en la base de datos. Como se verá en el apartado 5.3, el plugin Toolset, utilizado en este proyecto, ha creado tablas adicionales.

En WordPress, la estructura de las tablas de la base de datos está definida de tal forma que cada tabla tiene un identificador (ID) único, este hace de clave primaria de la tabla. Asimismo, una tabla puede contener campos de índice para mejorar el manejo de los datos en las peticiones a la base de datos. El valor más importante en dichas tablas es el ID que, aunque no siempre se presenta con el nombre ID, siempre es un valor que se autoincrementa para asegurar que cada nuevo valor insertado tenga un valor diferente a los anteriores, incluso a aquellos que han sido borrados.

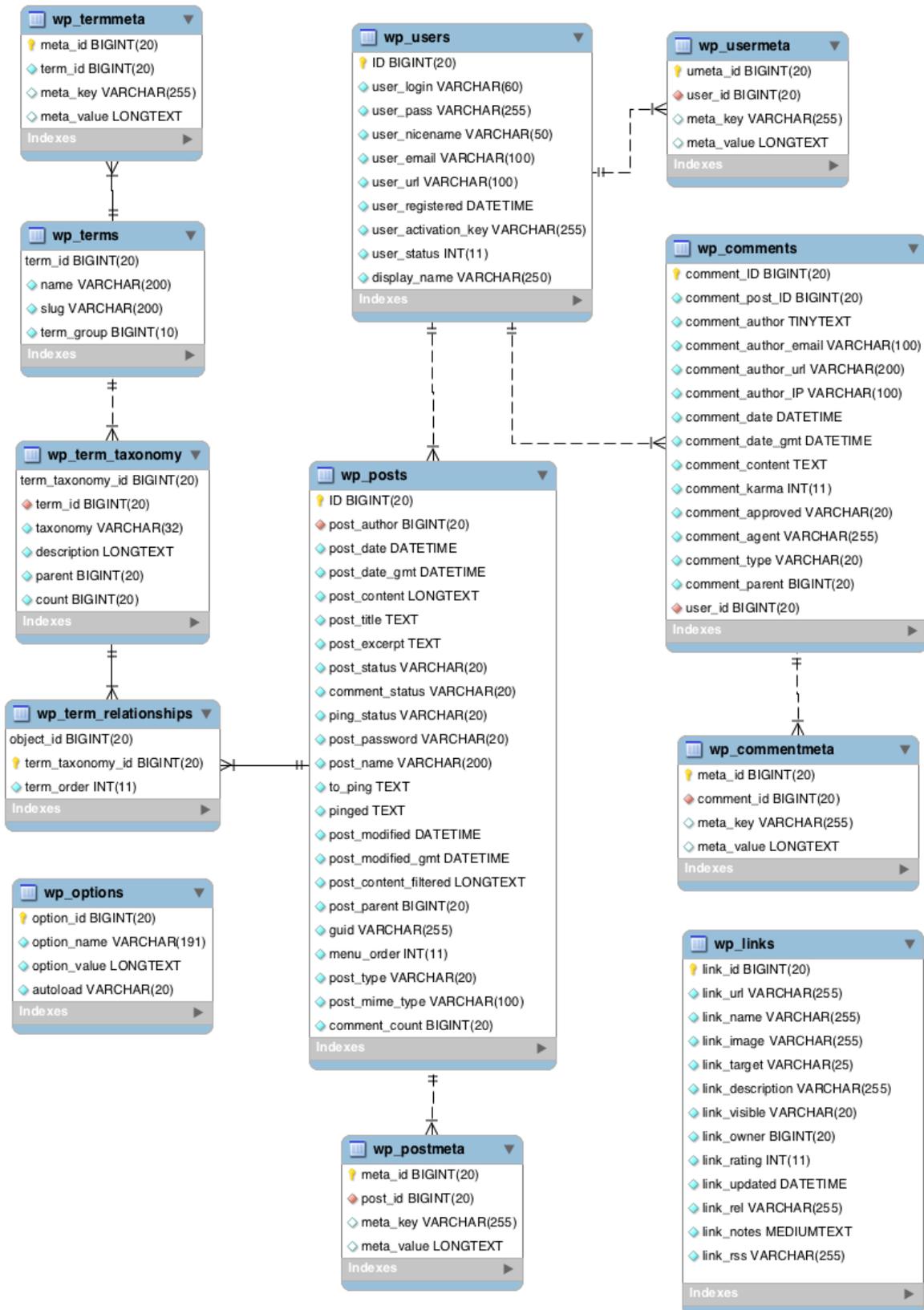


Figura 21. Diagrama de la base de datos de WordPress

## 5.2.2. Estructura de las tablas de WordPress

- **Tabla *wp\_commentmeta***

En esta tabla se almacenan los datos adicionales (metadatos) de los comentarios que se almacenan en WordPress utilizando la estructura clave-valor. Está pensada para ser utilizada por plugins o temas que añadan una funcionalidad extra a los comentarios.

#	Nombre	Tipo
<input type="checkbox"/>	1 <b>meta_id</b> 	bigint(20)
<input type="checkbox"/>	2 <b>comment_id</b>	bigint(20)
<input type="checkbox"/>	3 <b>meta_key</b>	varchar(255)
<input type="checkbox"/>	4 <b>meta_value</b>	longtext

Figura 22. Estructura de la tabla *wp\_commentmeta*

Como podemos apreciar en la *Figura 22*, esta tabla contiene 4 campos:

- *meta\_id*: ID principal de la tabla.
- *comment\_id*: ID del comentario (tabla *wp\_comments*) al que pertenece el metadato.
- *meta\_key*: Clave de identificación para el metadato.
- *meta\_value*: valor del metadato

Hay que destacar que el *comment\_id* hace referencia al id de la tabla *wp\_comments*, por lo que ambas tablas están relacionadas a través de este campo.

- **Tabla *wp\_comments***

Esta es la tabla donde se guardan los comentarios que se han realizado en nuestro sitio WordPress.

comment_ID	comment_post_ID	comment_author	comment_author_email	comment_author_url	comment_author_IP	comment_date	comment_date_gmt	comment_content
1	1	Un comentarista de WordPress	wapuu@wordpress.example	https://wordpress.org/		2017-12-08 09:26:49	2017-12-08 09:26:49	Hola, esto es un comentario. Para empezar a modera...

Figura 23. Estructura de la tabla *wp\_comments*

- *comment\_ID*: ID del comentario.
- *comment\_post\_ID*: ID del *post* (tabla *wp\_posts*) al que pertenece el comentario.
- *comment\_author*: Autor del comentario.
- *comment\_author\_email*: Email del autor del comentario.
- *comment\_author\_url*: URL del autor del comentario.
- *comment\_author\_IP*: IP del autor del comentario.
- *comment\_date*: Fecha y hora del comentario.
- *comment\_date\_gmt*: Fecha y hora del comentario (gmt).
- *comment\_content*: Contenido del comentario.
- *comment\_karma*: Meta o valor del campo. Actualmente no es usado por WordPress, pero puede ser utilizado por un plugin.
- *comment\_approved*: Valor (0 o 1) para especificar si el comentario ha sido aprobado o no.
- *comment\_agent*: Browser, sistema operativo, etc. Desde donde se hizo el comentario.
- *comment\_type*: Especifica el tipo de comentario (pingback o trackback).
- *comment\_parent*: Comentario padre del actual. Es decir, si este comentario es una respuesta al comentario especificado en este campo.
- *user\_id*: id del usuario si está registrado.

- **Tabla *wp\_links***

Esta tabla no se utiliza actualmente. Antes se utilizaba para la gestión de los enlaces, pero si queremos utilizar esta característica podemos utilizar un plugin para ello.

- **Tabla *wp\_options***

En esta tabla se guardan las opciones de configuración del sitio web de WordPress. Es donde normalmente los plugins y temas almacenan sus ajustes. Esta tabla no contiene relación con ninguna otra tabla de la base de datos.

#	Nombre	Tipo
<input type="checkbox"/> 1	<b>option_id</b> 	bigint(20)
<input type="checkbox"/> 2	<b>option_name</b>	varchar(191)
<input type="checkbox"/> 3	<b>option_value</b>	longtext
<input type="checkbox"/> 4	<b>autoload</b>	varchar(20)

Figura 24. Estructura de la tabla *wp\_options*

- *option\_id*: Id de la opción.
- *option\_name*: Nombre de la opción de configuración.
- *option\_value*: Valor de la opción de configuración.
- *autoload*: Si la opción de configuración se carga automáticamente.

- **Tabla *wp\_postmeta***

En esta tabla se almacenan los metadatos de un post. Algunos plugins pueden usar esta tabla para guardar su propia información.

#	Nombre	Tipo
1	<b>meta_id</b> 	bigint(20)
2	<b>post_id</b>	bigint(20)
3	<b>meta_key</b>	varchar(255)
4	<b>meta_value</b>	longtext

Figura 25. Estructura de la tabla *wp\_postmeta*

- *meta\_id*: Id de la información (meta).
- *post\_id*: Id del *post* asociado al que pertenece el metadato.
- *meta\_key*: Clave para identificar el metadato.
- *meta\_value*: valor asociado a un metadato (información).

- Tabla *wp\_posts*

Esta es una de las tablas más importantes de la base de datos. En ella se almacena la información de los Posts de WordPress (“Entradas”, “Páginas”, etc.).

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	E
1	<b>ID</b> 	bigint(20)		UNSIGNED	No	Ninguna	A
2	<b>post_author</b>	bigint(20)		UNSIGNED	No	0	
3	<b>post_date</b>	datetime			No	0000-00-00 00:00:00	
4	<b>post_date_gmt</b>	datetime			No	0000-00-00 00:00:00	
5	<b>post_content</b>	longtext	utf8mb4_unicode_520_ci		No	Ninguna	
6	<b>post_title</b>	text	utf8mb4_unicode_520_ci		No	Ninguna	
7	<b>post_excerpt</b>	text	utf8mb4_unicode_520_ci		No	Ninguna	
8	<b>post_status</b>	varchar(20)	utf8mb4_unicode_520_ci		No	publish	
9	<b>comment_status</b>	varchar(20)	utf8mb4_unicode_520_ci		No	open	
10	<b>ping_status</b>	varchar(20)	utf8mb4_unicode_520_ci		No	open	
11	<b>post_password</b>	varchar(255)	utf8mb4_unicode_520_ci		No		
12	<b>post_name</b>	varchar(200)	utf8mb4_unicode_520_ci		No		
13	<b>to_ping</b>	text	utf8mb4_unicode_520_ci		No	Ninguna	
14	<b>pinged</b>	text	utf8mb4_unicode_520_ci		No	Ninguna	
15	<b>post_modified</b>	datetime			No	0000-00-00 00:00:00	
16	<b>post_modified_gmt</b>	datetime			No	0000-00-00 00:00:00	
17	<b>post_content_filtered</b>	longtext	utf8mb4_unicode_520_ci		No	Ninguna	
18	<b>post_parent</b>	bigint(20)		UNSIGNED	No	0	
19	<b>guid</b>	varchar(255)	utf8mb4_unicode_520_ci		No		
20	<b>menu_order</b>	int(11)			No	0	
21	<b>post_type</b>	varchar(20)	utf8mb4_unicode_520_ci		No	post	
22	<b>post_mime_type</b>	varchar(100)	utf8mb4_unicode_520_ci		No		
23	<b>comment_count</b>	bigint(20)			No	0	

Figura 26. Estructura de la tabla *wp\_posts*

- *ID*: Id de la información.
- *post\_author*: Autor del post.

- *post\_date*: Fecha del post.
- *post\_date\_gmt*: Fecha del *post* (gmt)
- *post\_content*: Contenido del post.
- *post\_title*: Título del post.
- *post\_excerpt*: Resumen del post.
- *post\_status*: Estado del *post* (publicado, borrador, etc.).
- *comment\_status*: Estado de los comentarios (¿abiertos o no?).
- *ping\_status*: Estado de los pings (¿abiertos o no?).
- *post\_password*: Contraseña del post.
- *post\_name*: URL del post.
- *to\_ping*: URL a la que enviar el pingback cuando se modifique el contenido.
- *pinged*: URL a la que se ha enviado el pingback.
- *post\_modified*: Fecha y hora de la modificación del post.
- *post\_modified\_gmt*: Fecha y hora de la modificación del *post* (gmt).
- *post\_modified\_filtered*: utilizado por los plugin para almacenar en cache información relacionada a un contenido.
- *guid*: URL permanente al *post* (identificador global único).
- *menu\_order*: Orden en el que se muestra el contenido en los menús.
- *post\_type*: Tipo de *post* (post, page, etc.).
- *post\_mime\_type*: utilizado sólo para archivos adjuntos. Identifica cual es el media type del archivo adjunto.
- *comment\_count*: Número de comentarios del post.

- **Tabla *wp\_termmeta***

En ella se almacenan los metadatos relacionados con las categorías.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado
1	<b>meta_id</b> 	bigint(20)		UNSIGNED	No	Ninguna
2	<b>term_id</b>	bigint(20)		UNSIGNED	No	0
3	<b>meta_key</b>	varchar(255)	utf8mb4_unicode_520_ci		Sí	NULL
4	<b>meta_value</b>	longtext	utf8mb4_unicode_520_ci		Sí	NULL

Figura 27. Estructura de la tabla *wp\_termmeta*

- *meta\_id*: Id de la información (meta).
- *term\_id*: Id del término.
- *meta\_key*: Clave del meta.
- *meta\_value*: Valor del meta (información).

- **Tabla *wp\_terms***

En esta tabla se guardan tanto las categorías como las etiquetas para los Posts y Páginas.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado
1	<b>term_id</b> 	bigint(20)		UNSIGNED	No	Ninguna
2	<b>name</b>	varchar(200)	utf8mb4_unicode_520_ci		No	
3	<b>slug</b>	varchar(200)	utf8mb4_unicode_520_ci		No	
4	<b>term_group</b>	bigint(10)			No	0

Figura 28. Estructura de la tabla *wp\_terms*

- *term\_id*: Id de término.
- *name*: Nombre de la categoría.
- *slug*: URL de la categoría.
- *term\_group*: Agrupación de términos. puede ser utilizado por plugins o temas para crear grupos de taxonomías.

- **Tabla *wp\_term\_relationships***

En esta tabla se relaciona a un objeto de la tabla *wp\_post* a través de su identificador con un ítem de la tabla *wp\_term\_taxonomy*.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
1	<b>object_id</b> 	bigint(20)		UNSIGNED	No	0	
2	<b>term_taxonomy_id</b> 	bigint(20)		UNSIGNED	No	0	
3	<b>term_order</b>	int(11)			No	0	

Figura 29. Estructura de la tabla *wp\_term\_relationships*

- *object\_id*: Id del ítem correspondiente a la tabla *wp\_posts*.
- *term\_taxonomy\_id*: Nombre del término correspondiente a la tabla *wp\_term\_taxonomy*.
- *term\_order*: permite ordenar los términos de un objeto.

- **Tabla *wp\_term\_taxonomy***

Esta tabla asigna a cada ítem de la tabla *wp\_terms* una taxonomía definida para que WordPress pueda interpretarla.

#	Nombre	Tipo	Cotejamiento	Atributos
1	<b>term_taxonomy_id</b> 	bigint(20)		UNSIGNED
2	<b>term_id</b>	bigint(20)		UNSIGNED
3	<b>taxonomy</b>	varchar(32)	utf8mb4_unicode_520_ci	
4	<b>description</b>	longtext	utf8mb4_unicode_520_ci	
5	<b>parent</b>	bigint(20)		UNSIGNED
6	<b>count</b>	bigint(20)		

Figura 30. Estructura de la tabla *wp\_term\_taxonomy*

- *term\_taxonomy\_id*: Id del término de la taxonomía.
- *term\_id*: Id del término (tabla *wp\_terms*).
- *taxonomy*: Nombre de la taxonomía.
- *description*: Descripción de la taxonomía.
- *parent*: Id de la taxonomía superior. Usado si la estructura de la taxonomía es jerárquica.
- *count*: Número de objetos asignados a la taxonomía.

- **Tabla *wp\_usermeta***

En esta tabla se almacenan datos adicionales de los usuarios.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado
1	<b>umeta_id</b> 	bigint(20)		UNSIGNED	No	Ninguna
2	<b>user_id</b>	bigint(20)		UNSIGNED	No	0
3	<b>meta_key</b>	varchar(255)	utf8mb4_unicode_520_ci		Sí	NULL
4	<b>meta_value</b>	longtext	utf8mb4_unicode_520_ci		Sí	NULL

Figura 31. Estructura de la tabla *wp\_usermeta*

- *umeta\_id*: Id de la tabla.
- *user\_id*: Id del usuario (tabla *wp\_users*)
- *meta\_key*: Clave de la información o meta.
- *meta\_value*: Valor de la información o meta.

- **Tabla *wp\_users***

Aquí se almacenan los datos de los usuarios (nombre, email, etc.). Esta tabla está directamente relacionada con la tabla *wp\_usermeta*.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo
1	<b>ID</b> 	bigint(20)		UNSIGNED	No
2	<b>user_login</b>	varchar(60)	utf8mb4_unicode_520_ci		No
3	<b>user_pass</b>	varchar(255)	utf8mb4_unicode_520_ci		No
4	<b>user_nicename</b>	varchar(50)	utf8mb4_unicode_520_ci		No
5	<b>user_email</b>	varchar(100)	utf8mb4_unicode_520_ci		No
6	<b>user_url</b>	varchar(100)	utf8mb4_unicode_520_ci		No
7	<b>user_registered</b>	datetime			No
8	<b>user_activation_key</b>	varchar(255)	utf8mb4_unicode_520_ci		No
9	<b>user_status</b>	int(11)			No
10	<b>display_name</b>	varchar(250)	utf8mb4_unicode_520_ci		No

Figura 32. Estructura de la tabla *wp\_users*

- *ID*: Id del usuario.
- *user\_login*: Login o username para acceder.
- *user\_pass*: Contraseña del usuario (encriptada).
- *user\_nicename*: Nombre del usuario que se muestra.
- *user\_email*: Email del usuario.
- *user\_url*: URL del usuario (sitio web).
- *user\_registered*: Fecha y hora en la que se registró el usuario.
- *user\_activation\_key*: Este campo es usado para reestablecer la contraseña.
- *user\_status*: Es un campo que ya no se usa, anteriormente era usado para indicar a los usuarios baneados.
- *display\_name*: nombre del usuario elegido para mostrar (*user\_login*, *user\_nicename*).

### **5.3. Tablas MySQL de Toolset**

#### **5.3.1. Esquema de la base de datos de Toolset**

En la *Figura 33* tenemos el esquema de la base de datos de Toolset. En este caso, Toolset ha creado 5 tablas en nuestra base de datos, pero hay que destacar que, dependiendo de los subplugins de Toolset utilizados y las funcionalidades, este puede crear otras tablas en la base de datos.

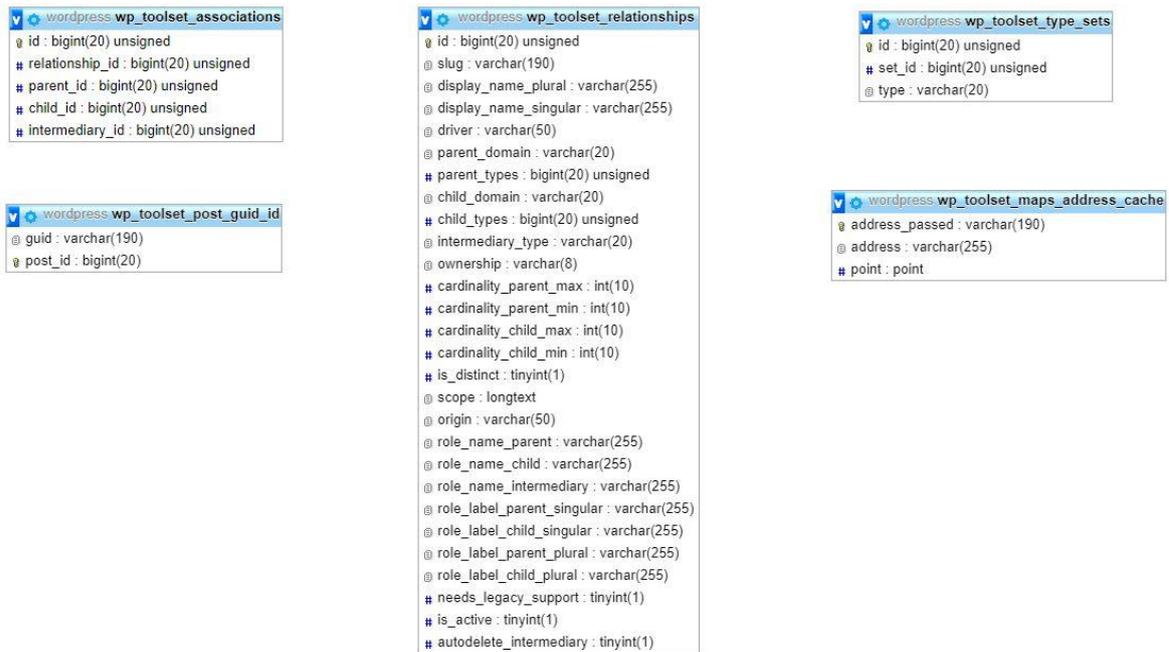


Figura 33. Esquema de la base de datos de Toolset

### 5.3.2. Estructura de las tablas de Toolset

- **Tabla *wp\_toolset\_associations***

Esta tabla contiene información sobre cada relación establecida entre posts.

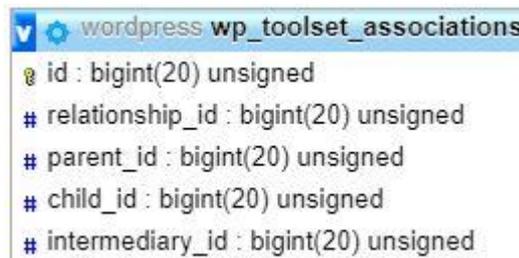


Figura 34. Estructura de la tabla *wp\_toolset\_associations*

- *id*: Es el identificador único asignado a cada fila de la tabla (autoincremento).
- *relationship\_id*: Id de la relación de publicaciones.
- *parent\_id*: Id de la publicación principal.
- *child\_id*: Id de la publicación secundaria.
- *intermediary\_id*: Id de la publicación intermedia. Si no existe este campo es igual a cero.

- **Tabla *wp\_toolset\_post\_guid\_id***

Esta tabla se usa para optimizar el trabajo con los campos de imagen. Es decir, relaciona el *guid* (URL) de una imagen con su id de publicación, y poder admitir valores de campo de imagen personalizados.

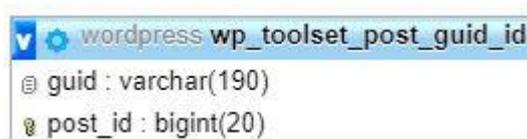


Figura 35. Estructura de la tabla *wp\_toolset\_post\_guid\_id*

- *guid*: Identificador único global de la imagen.
- *post\_id*: Id de la publicación a la que pertenece la imagen.

- **Tabla *wp\_toolset\_relationships***

En esta tabla se guarda la información de cada relación de publicación, como el nombre y el *slug*, que tipos de publicaciones están involucrados, los nombres de los roles asignados a las publicaciones en dicha relación e información sobre el número máximo permitido de cada publicación.

```

wordpress wp_toolset_relationships
id : bigint(20) unsigned
slug : varchar(190)
display_name_plural : varchar(255)
display_name_singular : varchar(255)
driver : varchar(50)
parent_domain : varchar(20)
# parent_types : bigint(20) unsigned
child_domain : varchar(20)
# child_types : bigint(20) unsigned
intermediary_type : varchar(20)
ownership : varchar(8)
# cardinality_parent_max : int(10)
# cardinality_parent_min : int(10)
# cardinality_child_max : int(10)
# cardinality_child_min : int(10)
# is_distinct : tinyint(1)
scope : longtext
origin : varchar(50)
role_name_parent : varchar(255)
role_name_child : varchar(255)
role_name_intermediary : varchar(255)
role_label_parent_singular : varchar(255)
role_label_child_singular : varchar(255)
role_label_parent_plural : varchar(255)
role_label_child_plural : varchar(255)
# needs_legacy_support : tinyint(1)
# is_active : tinyint(1)
# autodelete_intermediary : tinyint(1)

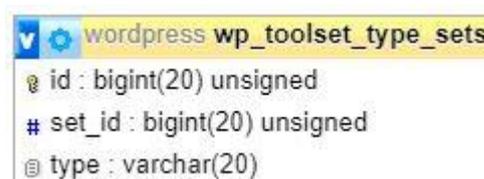
```

Figura 36. Estructura de la tabla *wp\_toolset\_relationships*

- *id*: Identificador único de la relación.
- *slug*: *slug* de la relación.
- *display\_name\_plural*: Nombre de la relación en plural.
- *display\_name\_singular*: Nombre de la relación en singular.
- *driver*: Por quien está hecha la relación. En nuestro caso el valor de este campo es “toolset”.
- *parent\_domain*: Tipo del padre de la relación. En nuestro caso el valor es “posts”
- *child\_domain*: Tipo del hijo de la relación. En nuestro caso el valor es “posts”
- *is\_active*: Si la relación esta activa.

- **Tabla *wp\_toolset\_type\_sets***

Esta tabla es utilizada por la API para las relaciones de muchos a muchos entre publicaciones, y así admitir referencias a los *slugs* de los *Posts Types*.

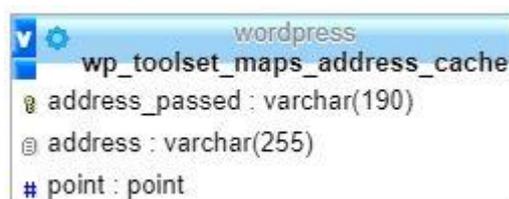


wordpress wp_toolset_type_sets	
id	bigint(20) unsigned
set_id	bigint(20) unsigned
type	varchar(20)

Figura 37. Estructura de la tabla *wp\_toolset\_type\_sets*

- **Tabla *wp\_toolset\_maps\_address\_cache***

Después de que una ubicación se geocodifica, su latitud, longitud y dirección formateada se almacenaran en esta tabla, para ahorrar costosos accesos a la API en referencias futuras.



wordpress wp_toolset_maps_address_cache	
address_passed	varchar(190)
address	varchar(255)
point	point

Figura 38. Estructura de la tabla *wp\_toolset\_maps\_address\_cache*

# **IMPLEMENTACIÓN**



## 6. IMPLEMENTACIÓN

### 6.1. Modificaciones aplicadas sobre WordPress

WordPress nos permite configurar una serie de ajustes para así poder personalizar el funcionamiento de nuestra instalación. A continuación, especificaremos los cambios realizados en ellos.

#### 6.1.1. Ajustes generales de WordPress

- **“Título del sitio”**: En este apartado introducimos el nombre que le queremos asignar a nuestro sitio web. En nuestro caso “SENDEROS CANARIOS”.
- **“Dirección de WordPress (URL)”**: Ponemos la dirección de la carpeta del servidor donde está instalado WordPress. Normalmente coincide con el dominio ya que solemos instalar WordPress en la raíz del dominio.
- **“Dirección del sitio (URL)”**: Es la dirección de nuestra página principal. Lo más común es utilizar tu dominio, pero también se puede personalizar la URL si queremos que nuestro dominio tenga un *slug* adicional. En nuestro caso hemos puesto la misma dirección que en el apartado anterior.
- **“Dirección de correo electrónico”**: Aquí insertamos la dirección de correo electrónico del administrador de la web. Ya que, a través de esta, el administrador recibirá notificaciones del sitio, por ejemplo, el aviso del registro de un nuevo usuario.
- **“Perfil predeterminado para nuevos usuarios”**: Este campo define que rol de administración y que capacidades tendrán los nuevos usuarios que se registren. En nuestro caso le asignamos el rol de “Suscriptor”.
- **“Idioma del sitio”**: Idioma principal de la web y el idioma en el que se muestra el panel de administración de WordPress. En este caso hemos elegido el idioma “Español”.
- **“Zona Horaria”**: Elegimos la zona horaria en la que queremos que trabaje WordPress. Para nosotros “UTC+1”.
- **“Formato de fecha”**: Elegimos el formato de fecha con el que trabajara WordPress. En este caso elegimos “d/m/Y”.
- **“Formato de hora”**: Formato de hora con el que trabajara WordPress. Seleccionamos “H:i”.

- **“La semana comienza el”**: Opción que nos permite elegir qué día empieza la semana. En este caso seleccionamos el “Lunes”.

<b>Título del sitio</b>	SENDEROS CANARIOS
<b>Dirección de WordPress (URL)</b>	http://xxx.xxx.x.xx
<b>Dirección del sitio (URL)</b>	http://xxx.xxx.x.
<b>Dirección de correo electrónico</b>	xxxxxx@ulpgc.es
<b>Perfil predeterminado para nuevos usuarios</b>	Suscriptor
<b>Idioma del sitio</b>	Español
<b>Zona horaria</b>	UTC+1
<b>Formato de fecha</b>	d/m/Y
<b>Formato de hora</b>	H:i
<b>La semana comienza el</b>	Lunes

Figura 39. Tabla resumen de los ajustes generales de WordPress

### 6.1.2. Ajustes de lectura

En los ajustes de lectura de WordPress solo cambiaremos el apartado “Tu portada muestra” donde seleccionamos la opción “Una página estática” frente a “Tus últimas entradas”, para que cuando un usuario ingrese a nuestro sitio web, visualice una página de bienvenida y en el subapartado portada seleccionamos dicha página, que ha de estar previamente creada.

Todos los demás ajustes de esta sección se dejarán como vienen por defecto.

### 6.1.3. Ajustes de Enlaces permanentes

Aquí es donde se configuran los enlaces permanentes, que son las URL de las publicaciones (post/páginas) individuales, así como de las categorías y otras listas de publicaciones. Un enlace permanente es lo que otro blog utilizará para vincular a cada Post, o para enviar un enlace a un *post* en un mensaje de correo electrónico o publicar un enlace al *post* en redes sociales. La URL de cada *post* debe ser permanente y no cambiar nunca.

En estos ajustes hemos configurado el apartado “Ajustes comunes” de la siguiente forma. Hemos seleccionado la opción de Estructura personalizada y en etiquetas disponibles seleccionamos “%year%”, “%monthnum%”, “%day%”, “%postname%”. Para que la URL

de nuestras páginas o *post* se compongan de la dirección del sitio web, el año de la publicación, el mes de la publicación (en formato numérico), el día de la publicación y el nombre del post. Al realizar estos cambios podemos observar en el campo de texto la estructura resultante: “/index.php/%year%/%monthnum%/%day%/%postname%”.

Todos los demás ajustes de esta sección los dejamos como vienen configurados por defecto.

## 6.2. Modificaciones aplicadas sobre el tema de WordPress Escapade

A continuación, especificaremos los ajustes que hemos seleccionado en el tema de WordPress para así darle la apariencia deseada al sitio web.

### 6.2.1. Identidad del sitio

En este apartado se eligen los ajustes sobre la identidad del sitio web que desarrollamos. A continuación, veremos los cambios realizados en dicho apartado:

- **“Título del sitio”**: Introducimos el título que queremos que tenga nuestro sitio web, en nuestro caso “SENDEROS CANARIOS”.
- **Seleccionamos el *checkbox*** que nos indica si queremos “mostrar el título y la descripción corta del sitio”.
- **“Icono del sitio”**: En este apartado podemos seleccionar y subir el icono que se ve en las pestañas del navegador, barra de favoritos y en las aplicaciones móviles de WordPress. En nuestro caso hemos subido la siguiente figura.



Figura 40. Icono del sitio web

### 6.2.2. Colores

Esta sección nos permite modificar los colores de los componentes del tema seleccionado.

Los cambios realizados en este apartado son los siguientes:

- **“Esquemas de color”**: Seleccionamos predeterminado.
- **“Paleta de color”**: Lo dejamos como viene configurado por defecto.
- **“Cabecera”**: En este apartado seleccionamos los colores deseados para la cabecera de nuestro sitio web. En nuestro caso:
  - En el color del texto del título del sitio insertamos #ffffff.
  - En el color del texto de la sección Hero insertamos #ffffff.
  - En el color de fondo de la sección Hero insertamos #ddb34f.
  - Y seleccionamos la cantidad de “transparencia del color superpuesto” deseada cuando utilizamos una imagen de cabecera personalizada.
- **“Menú”**: En este apartado podemos modificar el color del texto y del fondo del menú del sitio web. En nuestro caso seleccionamos como color del texto el #ffffff y como color de fondo #594a47.
- **“Botones”**: En esta sección podemos elegir el color del texto y del fondo de los botones del sitio web. Elegimos como color de fondo el #594a47 y como color de texto #ffffff.
- **“Contenido”**: A continuación, veremos los colores elegidos para el contenido del sitio web:
  - El color del texto de la cabecera elegido es #000000.
  - El color del texto principal es #000000.
  - El color del texto secundario es #d64700.
  - El color del texto del enlace es #d64700.
  - El color de fondo de la página es #d8c8ab.
  - El color del tagline\_text\_color es #757575

### 6.2.3. Fuentes

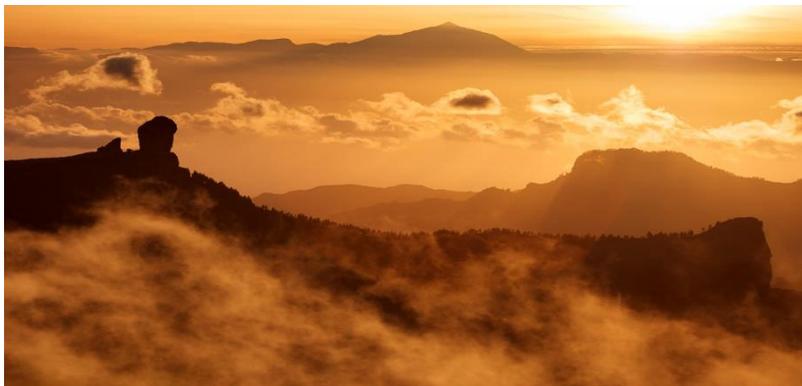
En este apartado debemos elegir el tipo de fuente que queremos para el sitio web.

- **“Título del sitio”**: Elegimos la fuente Cabin para el texto del título del sitio mostrado en la cabecera.
- **“Navegación”**: Seleccionamos la fuente Cabin para los enlaces del menú principal y de los botones.
- **“Encabezados”**: La fuente utilizada para los títulos de las entradas es Architects Daughter.

- **“Primario”**: Para los párrafos, listas, enlaces, citas y tablas utilizamos la fuente Droid Serif.
- **“Secundario”**: Optamos también por la fuente Droid Serif para los nombres de autor, enlaces de respuestas y pies de las entradas.

#### 6.2.4. *Medios del encabezado*

En este apartado podemos añadir un video o una imagen de cabecera para mostrarla en el sitio web. En nuestro mostramos una imagen de cabecera, por lo que seleccionamos el *checkbox* que nos indica si queremos usar una imagen destacada. Y en la sección imagen de cabecera añadimos la imagen que vemos en la *Figura 41*.



*Figura 41. Imagen de cabecera del sitio web*

#### 6.2.5. *Ajustes de portada*

En estos ajustes podemos elegir que mostrar en la portada del sitio. Pueden ser entradas en orden cronológico inverso o una página fija/estática. En este proyecto optamos por elegir una página estática, para ello en el apartado “Tu portada muestra” elegimos la opción “Una página estática” y en el apartado “Página de inicio” seleccionamos la página que queremos mostrar, que ha de estar previamente creada.

### 6.3. **Implementación llevada a cabo en Toolset**

#### 6.3.1. *Configuración de Toolset*

En este apartado veremos los cambios realizados en la configuración de Toolset para el desarrollo de este proyecto.

- En el apartado “**Contenido personalizado**” marcamos el *checkbox* “**REST API**” para así mostrar los campos personalizados administrados por Toolset Types, los usuarios y términos a través de la API REST (una interfaz entre sistemas que usa HTTP para obtener datos o generar operaciones sobre esos datos en todos los formatos posibles como XML y JSON [21]) y poder utilizar como veremos más adelante el plugin REST API Controller.
- En el apartado “Mapas” elegimos la API que queremos usar, en nuestro caso seleccionamos la “API de Google Maps”. En el siguiente apartado debemos introducir la clave API de Google Maps que nos proporciona Google.

### 6.3.2. Creación de la estructura y taxonomías de la página web

En primer lugar, vamos a modificar los tipos de entrada que vienen en WordPress por defecto: “Páginas”, “Post” y “Multimedia”. En nuestro caso el tipo de entrada “Páginas” lo renombramos a “Inicio”, que es donde más adelante estará la página de bienvenida (portada) de nuestra web. Para ello dentro de Toolset vamos a “Tipos de entrada” (este apartado de Toolset es manejado por la extensión Toolset Types) y nos aparece un listado con todos tipos de entrada que tenemos definidos en nuestro WordPress, donde seleccionamos el tipo de entrada “Páginas” y en la ventana que se nos abre cambiamos el nombre plural y el nombre en singular a “Inicio”.

Figura 42. Editar el tipo de entrada Páginas

Los tipos “Post” y “Multimedia” no los vamos a utilizar para este proyecto, debido a que, al ser un tipo de entrada integrado de WordPress, no nos da la posibilidad de modificarlo totalmente a nuestra conveniencia. Entonces hemos optado por deshabilitar su visibilidad en el panel de control de WordPress. Para ello, los editamos como hemos hecho anteriormente

con el tipo de entrada “Páginas” y en la opción estado seleccionamos “Borrador”, como podemos ver en la *Figura 43*.

The screenshot shows the 'Edit Post Type' interface for 'Posts'. On the left, under 'Nombre y descripción', the 'Nombre plural (obligatorio)' is 'Posts', 'Nombre singular (obligatorio)' is 'post', and 'Slug (obligatorio)' is 'post'. The 'Editor (obligatorio)' is set to 'Bloque'. Below, there is a 'Description' field with a link to 'Agregar descripción' and an 'Ícono' field with a 'Cambiar ícono' button. A note states: 'Esta opción no está disponible para tipos de entrada integrados.' On the right, the 'Guardar' panel shows the 'Estado' set to 'Borrador' (selected), with options for 'Publicada' and 'Borrador (no visible en menús de administración, sin interfaz de usuario para administrar entradas, no consultable en el front-end)'. There are 'OK' and 'Cancel' buttons. Below, the 'Posición del menú de administración después de:' is set to 'Inicio'. At the bottom right is a 'Guardar tipo de entrada' button.

*Figura 43. Editar el tipo de entrada Posts*

Una vez modificados los tipos de entrada predefinidos en WordPress, procedemos a crear nuestros propios tipos de entrada, donde más adelante se introducirá el contenido de la web. En nuestro caso “Rutas”, “Añadir Rutas” y “Ayuda”, para ello volvemos a “Tipos de entrada” y seleccionamos “Agregar nuevo”, donde rellenamos los campos solicitados para cada uno de los tipos de entrada.

The screenshot shows the 'Add New Post Type' interface for 'Rutas'. On the left, under 'Nombre y descripción', the 'Nombre plural (obligatorio)' is 'Rutas', 'Nombre singular (obligatorio)' is 'Ruta', and 'Slug (obligatorio)' is 'ruta'. The 'Editor (obligatorio)' is set to 'Clásico'. Below, there is a 'Description' field with a link to 'Agregar descripción' and an 'Ícono' field with a 'Cambiar ícono' button. On the right, the 'Guardar' panel shows the 'Estado' set to 'Publicada' with an 'Edit' link. The 'Posición del menú de administración después de:' is set to 'Escritorio'. There is a checkbox for 'Mostrar cantidad de entradas en el widget administrativo "Vista rápida"'. At the bottom right is a 'Guardar tipo de entrada' button.

*Figura 44. Agregar nuevo tipo de entrada Rutas*

The screenshot shows the 'Add New Post Type' interface for 'Ayuda'. On the left, under 'Nombre y descripción', the 'Nombre plural (obligatorio)' is 'Ayuda', 'Nombre singular (obligatorio)' is 'Ayudas', and 'Slug (obligatorio)' is 'ayuda'. The 'Editor (obligatorio)' is set to 'Clásico'. Below, there is a 'Description' field with a link to 'Agregar descripción' and an 'Ícono' field with a 'Cambiar ícono' button. On the right, the 'Guardar' panel shows the 'Estado' set to 'Publicada' with an 'Edit' link. The 'Posición del menú de administración después de:' is set to 'Añadir Rutas'. There is a checkbox for 'Mostrar cantidad de entradas en el widget administrativo "Vista rápida"'. At the bottom right is a 'Guardar tipo de entrada' button.

*Figura 45. Agregar nuevo tipo de entrada Ayuda*

Figura 46. Agregar nuevo tipo de entrada Añadir Rutas

Como podemos observar en las Figuras 44, 45 y 46, los campos a rellenar son: el nombre del tipo de entrada en plural, el nombre en singular, el “*Slug*” que se rellena automáticamente al insertar el nombre que, en nuestro caso, no hemos modificado y el icono (en cada uno de los tipos hemos elegido el adecuado).

En este punto ya hemos finalizado la creación de los tipos de entrada y en la Figura 47 podemos ver como se nos muestran en el panel de control de WordPress.

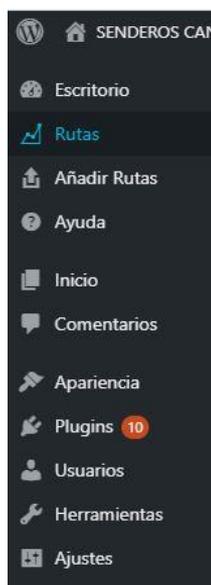


Figura 47. Panel de control de WordPress

En segundo lugar, vamos a crear las taxonomías para así poder agrupar el contenido de la web de una forma más intuitiva y efectiva para el usuario. En este proyecto se han creado dos taxonomías, “Islas” y “Niveles de dificultad”. De este modo, el usuario puede visualizar fácilmente todas las rutas que hay en una isla o todas las rutas que tengan el nivel de dificultad elegido. Para crear dichas taxonomías nos vamos al apartado “Taxonomías”

(también manejado por la extensión Toolset Types) dentro de Toolset y vamos a “Agregar nuevo”. A continuación, rellenamos el formulario que podemos ver en la *Figura 48*.

Nombre y descripción

Nombre plural (obligatorio) Introduzca el nombre de taxonomía plural

Nombre singular (obligatorio) Introduzca el nombre de taxonomía singular

Slug (obligatorio) Introduzca el slug de la taxonomía

Description [Agregar descripción](#)

Guardar

Estado: [Publicada](#) [Edit](#)

[Guardar taxonomía](#)

Etiquetas

Tipo de taxonomía

Jerárquico - como categorías de entrada, con relación superior/inferior y cuadros de selección para elegir taxonomías

Sin formato - como etiquetas de entrada, con una entrada de texto para introducir términos

Tipos de entrada para usar con esta taxonomía

Ayuda  Añadir Rutas  Entradas

Inicio  Medios  Rutas

*Figura 48. Formulario de creación de taxonomías*

Como podemos ver, el formulario consta del nombre de la taxonomía en plural, el nombre en singular, el “*Slug*” que se rellena automáticamente al introducir el nombre, el tipo de taxonomía que puede ser “Jerárquico” o “Sin formato” y el tipo de entrada para usar con esa taxonomía. En nuestro caso el tipo de taxonomía es “Jerárquico” ya que queremos que tenga una relación superior/inferior, y el tipo de entrada a usar es “Rutas” en ambos casos. A continuación, podremos ver en la *Figura 49* y *50* estos formularios ya rellenos para crear dichas taxonomías.

The screenshot shows the WordPress taxonomy form for 'Islas'. The 'Nombre y descripción' section contains the following fields: 'Nombre plural (obligatorio)' with the value 'Islas', 'Nombre singular (obligatorio)' with 'Isla', 'Slug (obligatorio)' with 'isla', and a 'Description' field with the text 'Isla en la que se encuentra situada la ruta'. The 'Etiquetas' section is collapsed. The 'Tipo de taxonomía' section has 'Jerárquico' selected. The 'Tipos de entrada para usar con esta taxonomía' section has 'Rutas' checked. On the right, the 'Guardar' section shows 'Estado: Publicada' and a 'Guardar taxonomía' button. The 'Module Manager' section is also visible.

Figura 49. Formulario de la taxonomía Islas

The screenshot shows the WordPress taxonomy form for 'Niveles de dificultad'. The 'Nombre y descripción' section contains the following fields: 'Nombre plural (obligatorio)' with the value 'Niveles de dificultad', 'Nombre singular (obligatorio)' with 'Nivel de dificultad', 'Slug (obligatorio)' with 'nivelde-dificultad', and a 'Description' field with the text 'Agregar descripción'. The 'Etiquetas' section is collapsed. The 'Tipo de taxonomía' section has 'Jerárquico' selected. The 'Tipos de entrada para usar con esta taxonomía' section has 'Rutas' checked. On the right, the 'Guardar' section shows 'Estado: Publicada' and a 'Guardar taxonomía' button. The 'Module Manager' section is also visible.

Figura 50. Formulario de la taxonomía Niveles de dificultad

Ahora que ya tenemos las taxonomías creadas, vamos a agregar los tipos de taxonomías, es decir, las “Islas” y los “Niveles de dificultad”. Para agregar las “Islas” nos vamos a “Rutas”, en el panel de control de WordPress, seleccionamos “Islas” y, posteriormente, rellenamos los campos solicitados para ir añadiendo las islas una a una. Dichos campos son el “Nombre”, el “Slug”, un campo para seleccionar el padre, que en nuestro caso no tiene

sentido y por lo tanto no lo modificamos, una descripción y un campo para añadir una imagen.

**Add New Isla**

Nombre  
El nombre es cómo aparecerá en tu sitio.

Slug  
El «slug» es la versión amigable de la URL del nombre. Suele estar en minúsculas y contiene solo letras, números y guiones.

Parent Isla  
Ninguna  
Asigna un término superior para crear una jerarquía. El término jazz, por ejemplo, sería el superior de bebop y big band.

Descripción  
La descripción no suele mostrarse por defecto, sin embargo hay algunos temas que puede que la muestren.

Imagen  
Imagen de la isla  
Seleccionar imagen  
Add New Isla

Acciones en lote ▼ Aplicar 7 elementos

<input type="checkbox"/> Nombre	Descripción	Slug	Cantidad	Imagen de la isla
<input type="checkbox"/> El Hierro	—	el-hierro	1	
<input type="checkbox"/> Fuerteventura	—	fuerteventura	1	
<input type="checkbox"/> Gran Canaria	—	gran-canaria	3	
<input type="checkbox"/> La Gomera	—	la-gomera	1	
<input type="checkbox"/> La Palma	—	la-palma	1	
<input type="checkbox"/> Lanzarote	—	lanzarote	1	
<input type="checkbox"/> Tenerife	—	tenerife	1	

Acciones en lote ▼ Aplicar 4 elementos

Figura 51. Formulario para agregar las islas

Como podemos observar en la *Figura 51* se han añadido las siete Islas Canarias, cada una con su respectivo nombre en plural y singular, su “Slug” y una imagen identificativa.

Para agregar los “Niveles de dificultad” realizamos el mismo proceso que seguimos anteriormente, aunque sin añadir imágenes a cada uno de los niveles de dificultad.

**Add New Nivel de dificultad**

Nombre  
El nombre es cómo aparecerá en tu sitio.

Slug  
El «slug» es la versión amigable de la URL del nombre. Suele estar en minúsculas y contiene solo letras, números y guiones.

Parent Nivel de dificultad  
Ninguna  
Asigna un término superior para crear una jerarquía. El término jazz, por ejemplo, sería el superior de bebop y big band.

Descripción  
La descripción no suele mostrarse por defecto, sin embargo hay algunos temas que puede que la muestren.

Add New Nivel de dificultad

Acciones en lote ▼ Aplicar 4 elementos

<input type="checkbox"/> Nombre	Descripción	Slug	Cantidad
<input type="checkbox"/> Difícil	—	difícil	1
<input type="checkbox"/> Experto	—	experto	0
<input type="checkbox"/> Fácil	—	facil	4
<input type="checkbox"/> Moderado	—	moderado	4

Acciones en lote ▼ Aplicar 4 elementos

Figura 52. Formulario para agregar los niveles de dificultad

En tercer lugar, vamos a crear el grupo de campos para las rutas, es decir, todos aquellos campos de los que se compone una ruta. En nuestro diseño, las rutas deben tener un *checkbox* de actividad donde el usuario elige la actividad adecuada para la ruta, un campo de texto donde se añada una breve descripción de la ruta, un campo para añadir fotos de la ruta, un campo de dirección donde el usuario añada la dirección del punto de partida de la ruta, un campo de texto para indicar la distancia de la ruta, otro campo de texto donde poner el tiempo de duración aproximado de la ruta y un campo donde poder insertar el mapa de la ruta. Para crear estos campos, utilizamos el apartado “Campos personalizados” dentro de Toolset. Y seleccionamos “Agregar nuevo”, después rellanamos el campo “Nombre” con el nombre que le queremos dar a este grupo de campos de entrada, en nuestro caso el nombre que le dimos fue “Características de las rutas”. Seguidamente, vamos agregando campos nuevos con las características adecuadas para nuestro diseño. En las Figuras 53, 54, 55, 56, 57, 58 y 59 podemos ver el formulario para crear cada uno de estos campos con su respectiva información.

The screenshot shows the 'Campos personalizados' (Custom Fields) interface in Toolset. The title is 'Actividad (Casillas de verificación)'. The form contains the following fields:

- Nombre de campo:** 'Actividad'
- Slug de campo:** 'actividad'
- Tipo de campo:** 'Casillas de verificación' (dropdown menu)
- Description:** 'Actividad para practicar en la ruta' (text area)

Below the main form, there are two radio button options for 'Guardar opción':

- Si no está marcado, guardar 0 en la base de datos
- Si no está marcado, no guardar nada en la base de datos

The 'Casillas de verificación' (checkboxes) section contains four items, each with a plus icon, a minus icon, and a right-pointing arrow:

- Mountain Bike
- Senderismo
- Running
- Alpinismo

At the bottom, there is a button labeled 'Agregar opción'.

Figura 53. Agregar campo actividad

**Descripción (Líneas múltiples) - requerido**

Nombre de campo ? Descripción

Slug de campo ? descripcion

Tipo de campo Líneas múltiples

Description ? Descripción de la ruta

Marcador ? Por favor introduzca una breve descripción de la ruta

Valor predeterminado Introducir el valor predeterminado

¿Campo simple o repetitivo?  Permitir múltiples instancias de este campo  
 Este campo puede tener sólo un valor

Validación  Obligatorio

Mensaje de error de validación Campo obligatorio - Por favor introduzca una breve descripción de la ruta

Incluir en la búsqueda  Incluir este campo en las búsquedas de texto en todo el sitio web

Visualización condicional Establecer condiciones

Figura 54. Agregar campo descripción

**Fotos (Imagen)**

Only the users with proper permissions can use the media uploader. Others will see a simple input field. Use Toolset Access to manage users' permissions.

Nombre de campo ? Fotos

Slug de campo ? fotos

Tipo de campo Imagen

Description ? Fotos de la ruta

Marcador ? Introducir un marcador

Valor predeterminado ? Introducir el valor predeterminado

¿Campo simple o repetitivo?  Permitir múltiples instancias de este campo  
 Este campo puede tener sólo un valor

Validación  Obligatorio

Mensaje de error de validación Campo obligatorio

Visualización condicional Establecer condiciones

Figura 55. Agregar campo Fotos

**Dirección** (*Dirección*) - requerido ▲

Nombre de campo ? Dirección

Slug de campo ? direccion

Tipo de campo Dirección ▼

Description ? Por favor introduzca la dirección de la ruta

¿Campo simple o repetitivo?  Permitir múltiples instancias de este campo  
 Este campo puede tener sólo un valor

Obligatorio   
 Campo obligatorio

Visualización condicional Establecer condiciones

Figura 56. Agregar campo dirección

**Distancia** (*Líneas múltiples*) ▲

Nombre de campo ? Distancia

Slug de campo ? distancia

Tipo de campo Líneas múltiples ▼

Description ? Distancia de la ruta

Marcador ? Por favor introduzca la distancia de la ruta

Valor predeterminado Introducir el valor predeterminado

¿Campo simple o repetitivo?  Permitir múltiples instancias de este campo  
 Este campo puede tener sólo un valor

Validación  Obligatorio

Mensaje de error de validación Campo obligatorio

Incluir en la búsqueda  Incluir este campo en las búsquedas de texto en todo el sitio web

Visualización condicional Establecer condiciones

Figura 57. Agregar campo distancia.

☰ **Tiempo de duración** *(Líneas múltiples)* ▲

Nombre de campo ?

Slug de campo ?

Tipo de campo  ▼

Description ?

---

Marcador ?

Valor predeterminado

---

¿Campo simple o repetitivo?  Permitir múltiples instancias de este campo  
 Este campo puede tener sólo un valor

---

Validación  Obligatorio

Mensaje de error de validación

---

Incluir en la búsqueda  Incluir este campo en las búsquedas de texto en todo el sitio web

---

Visualización condicional

*Figura 58. Agregar campo tiempo de duración*

**Mapa Ruta (Líneas múltiples) - requerido**

Nombre de campo [?](#) Mapa Ruta

Slug de campo [?](#) mapa-ruta

Tipo de campo Líneas múltiples ▼

Description [?](#) Introducir una descripción del campo

Marcador [?](#) Por favor introduzca la URL del mapa

Valor predeterminado Introducir el valor predeterminado

¿Campo simple o repetitivo?  Permitir múltiples instancias de este campo  
 Este campo puede tener sólo un valor

Validación  Obligatorio

Mensaje de error de validación Campo obligatorio - Por favor introduzca el mapa de la ruta

Incluir en la búsqueda  Incluir este campo en las búsquedas de texto en todo el sitio web

Visualización condicional

*Figura 59. Agregar campo mapa ruta*

Para finalizar con el desarrollo de la estructura del sitio web, debemos crear el menú de este. Para ello, nos vamos al menú “Apariencia” en el panel de control de WordPress y seleccionamos “Menús”. Luego elegimos que queremos que nos aparezca en el menú al desplegar las opciones de pantalla.

Cajas

Inicio  Entradas  Rutas  Añadir Rutas  Ayuda  Enlaces personalizados  Etiquetas  Etiquetas  Formatos  Islas

Niveles de dificultad

Mostrar propiedades avanzadas de menú

Destino del enlace  Atributos del título  Clases CSS  Relación con el enlace (XFN)  Descripción

Opciones de pantalla ▲

*Figura 60. Opciones de pantalla*

Como podemos observar en la *Figura 60*, hemos seleccionado “Inicio”, “Rutas”, “Añadir rutas”, “Ayuda”, “Islas” y “Niveles de dificultad”, ya que estos son los apartados de nuestro menú.

A continuación, veremos detalladamente como añadir los elementos al menú de nuestra web.

- “Inicio”: Aquí queremos que se muestre la página de inicio con el mensaje de bienvenida a los usuarios de la web. Para ello seleccionamos “Enlaces personalizados” en el apartado “Añadir elementos al menú” y rellenamos los campos solicitados, en “URL” ponemos la dirección que tiene la página inicio de nuestra web y en “Texto del enlace” ponemos “Inicio” ya que es el nombre que queremos que aparezca en el menú.



The image shows a web interface titled "Añadir elementos al menú". It features a vertical list of menu items: "Inicio", "Rutas", "Añadir Rutas", "Ayuda", "Enlaces personalizados", "Islas", and "Niveles de dificultad". The "Enlaces personalizados" item is highlighted with a grey background and an upward-pointing arrow. Below this list, there are two input fields: "URL" with the value "http://xxx.xxx.x.x" and "Texto del enlace" which is empty. A button labeled "Añadir al menú" is positioned below the second input field. At the bottom of the form, there are two more dropdown menus: "Islas" and "Niveles de dificultad".

Figura 61. Añadir “Inicio” al menú

- “Rutas”: Aquí queremos que se muestren todas las rutas, para ello desplegamos “Rutas” en “Añadir elementos al menú” y seleccionamos “Ver todo” y en el *checkbox* hacemos clic en “Todos los elementos” y le damos a “Añadir”. Posteriormente, en la “Etiqueta de navegación”, cambiamos el nombre “Todos los elementos” por el de “Rutas”

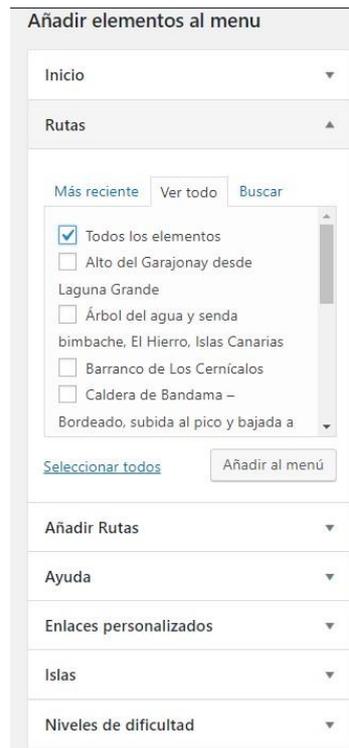


Figura 62. Añadir "Rutas" al menú

- "Islas": En este apartado queremos que se nos despliegue un submenú donde el usuario pueda elegir la isla que desee. Para ello, debemos insertar un enlace personalizado, pero como queremos que nos muestre un *dropdown* de islas, debemos de rellenar el campo "URL" con una almohadilla (#) y en "Texto del enlace" le damos el nombre "Islas" y hacemos clic en "Añadir al menú". A continuación, nos vamos a "Islas" y seleccionamos las islas una a una y las añadimos al menú, donde posteriormente las ordenamos después de "Islas" en el orden deseado.

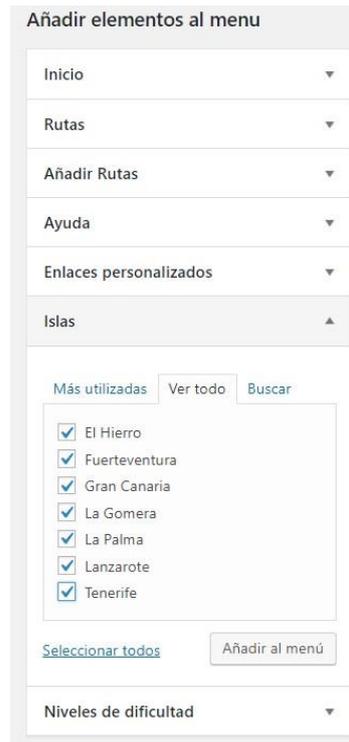


Figura 63. Añadir "Islas" al menú

- "Nivel de dificultad": En este caso queremos mostrar un *dropdown* con los niveles de dificultad. Para ello seguimos los mismos pasos que en el apartado anterior, pero en "Texto del enlace" ponemos "Dificultad", añadimos y ordenamos los niveles después de "Dificultad".



Figura 64. Añadir "Dificultad" al menú

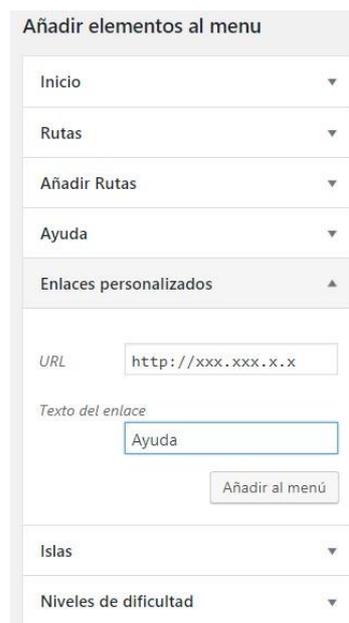
- “Añadir ruta”: Aquí se nos mostrará la plantilla que debe rellenar el usuario cuando quiera subir una ruta al sitio web. Seleccionamos “Añadir rutas” y añadimos al menú.



The screenshot shows a web interface titled "Añadir elementos al menú". It features a vertical list of menu items: "Inicio", "Rutas", "Añadir Rutas", "Ayuda", "Enlaces personalizados", "Islas", and "Niveles de dificultad". The "Añadir Rutas" item is selected and expanded, revealing a sub-section with a search bar containing "Más reciente", "Ver todo", and "Buscar". Below the search bar are two checkboxes: "Todos los elementos" (unchecked) and "Añadir ruta" (checked). At the bottom of this sub-section are two buttons: "Seleccionar todos" and "Añadir al menú".

Figura 65. Añadir "Añadir rutas" al menú

- “Ayuda”: En esta sección se le mostrará al usuario una página donde aparecerá la información sobre como añadir una ruta a nuestro sitio web. Para llegar a ello, seleccionamos “Enlaces personalizados” y procedemos al igual que con el apartado “Inicio” cambiando la “URL” por la dirección de nuestra página ayuda y en el “Texto del enlace” dándole el nombre “Ayuda”.



The screenshot shows the same "Añadir elementos al menú" interface, but now the "Enlaces personalizados" item is selected and expanded. It displays a form with two input fields: "URL" with the placeholder text "http://xxx.xxx.x.x" and "Texto del enlace" with the text "Ayuda". A button labeled "Añadir al menú" is positioned below the "Texto del enlace" field. The other menu items remain visible in the background.

Figura 66. Añadir "Ayuda" al menú

En la *Figura 67* veremos cómo nos queda la estructura del menú después de haber realizado los cambios anteriores.



*Figura 67. Estructura del menú*

En este punto ya hemos creado la estructura del sitio web, es decir, los tipos de entrada “Rutas”, “Añadir rutas”, “Ayuda” e “Inicio”, donde se introducirán los datos de la web. También hemos creado las taxonomías, en nuestro caso “Islas” y “Niveles de dificultad”, luego construimos el menú del sitio web y para finalizar creamos el “Grupo de campo” para el tipo de entrada “Rutas”, donde se definen los campos necesarios que debe tener una ruta.

### **6.3.3. Generación de las rutas**

En este apartado vamos a explicar cómo se introducen los datos para generar las rutas de nuestra web desde WordPress utilizando el plugin Toolset. Para ello debemos ir a “Rutas” en el panel de control de Toolset y seleccionar la opción “Agregar nuevo”, seguidamente se nos muestran los campos que debemos rellenar para añadir una ruta a nuestro sitio web. Estos campos son:

- “Título”: se introduce el título de la ruta a agregar.
- “Actividad”: se selecciona la actividad para la que está indicada la ruta.
- “Descripción”: es un campo de texto en el que introducir una descripción de la ruta.
- “Fotos”: en este campo se pueden agregar fotos de la ruta.
- “Dirección”: este es un campo de dirección donde el usuario debe poner la dirección del punto de partida de la ruta.
- “Distancia”: campo de texto para indicar la distancia aproximada de la ruta.
- “Tiempo de duración”: campo de texto para indicar la duración aproximada de la ruta.
- “Mapa Ruta”: campo de texto donde se introduce la URL del mapa de la ruta a subir.
- “Islas”: *checkbox* donde se debe seleccionar la isla (taxonomía) en la que se encuentra ubicada la ruta.
- “Niveles de dificultad”: campo para seleccionar el nivel de dificultad (taxonomía) que tiene la ruta.

Introducir el título aquí

**Características de las rutas**

**Actividad**  
Actividad para practicar en la ruta

Mountain Bike  
 Senderismo  
 Running  
 Alpinismo

**Descripción\***  
Descripción de la ruta

Por favor introduzca una breve descripción de la ruta

**Fotos**  
Fotos de la ruta

Agregar nuevo

Select image(s)

**Dirección\***  
Por favor introduzca la dirección de la ruta

Introducir dirección

Mostrar/ocultar coordenadas

**Distancia**  
Distancia de la ruta

Por favor introduzca la distancia de la ruta

**Tiempo de duración**  
Por favor introduzca el tiempo de duración de la ruta

**Mapa Ruta\***  
Por favor introduzca la URL del mapa

**Template Layout**

Plantilla para Ruta individual OK

OR

Create a new template layout for Rutas

**Publicar**

Guardar borrador Vista previa

Estado: Borrador [Editar](#)

Visibilidad: Público [Editar](#)

[Publicar inmediatamente](#) [Editar](#)

[Publicar](#)

**Islas**

All Islas Más utilizadas

El Hierro  
 Fuerteventura  
 Gran Canaria  
 La Gomera  
 La Palma  
 Lanzarote  
 Tenerife

[+ Add New Isla](#)

**Niveles de dificultad**

All Niveles de dificultad Más utilizadas

Difícil  
 Experto  
 Fácil  
 Moderado

[+ Add New Nivel de dificultad](#)

**Disposición**

Predeterminado  Personalizado

Figura 68. Generar rutas

Para insertar las rutas que se encuentran en nuestra web, realizamos este proceso para cada una de las rutas.

#### 6.3.4. Generación de la página inicio

Para generar la página de inicio de nuestra web, nos dirigimos a “Inicio” en el panel de control de WordPress y “Agregar nuevo”. Donde nos da la opción de añadir un título a la página y un campo de texto para escribir la información necesaria.

# BIENVENIDO A TU WEB DE RUTAS EN LAS ISLAS CANARIAS

Aquí podrás encontrar distintas rutas para hacer senderismo, running, Mountain Bike o alpinismo a través de las islas canarias.

*Figura 69. Generación de la página “Inicio”.*

Al ser esta una página estática, rellenamos los campos como podemos ver en la *Figura 69* y al guardar ya tenemos nuestra página de inicio creada correctamente.

### **6.3.5. Generación de la página ayuda**

Esta es la página donde se muestra la información de ayuda al usuario de nuestra web. Es decir, una página estática que tendrá un título y un apartado donde se indica la información necesaria. Para lograr esto, nos dirigimos al apartado “Ayuda” en el panel de control de Toolset y en este seleccionamos “Agregar nuevo”, posteriormente introducimos el título de esta página, en nuestro caso “INSTRUCCIONES SOBRE COMO AGREGAR EL MAPA A NUESTRA WEB” y en el campo de texto siguiente seleccionamos “HTML”, ya que queremos introducir la información y links que redireccionen a otras webs, todo esto con nuestro propio diseño.

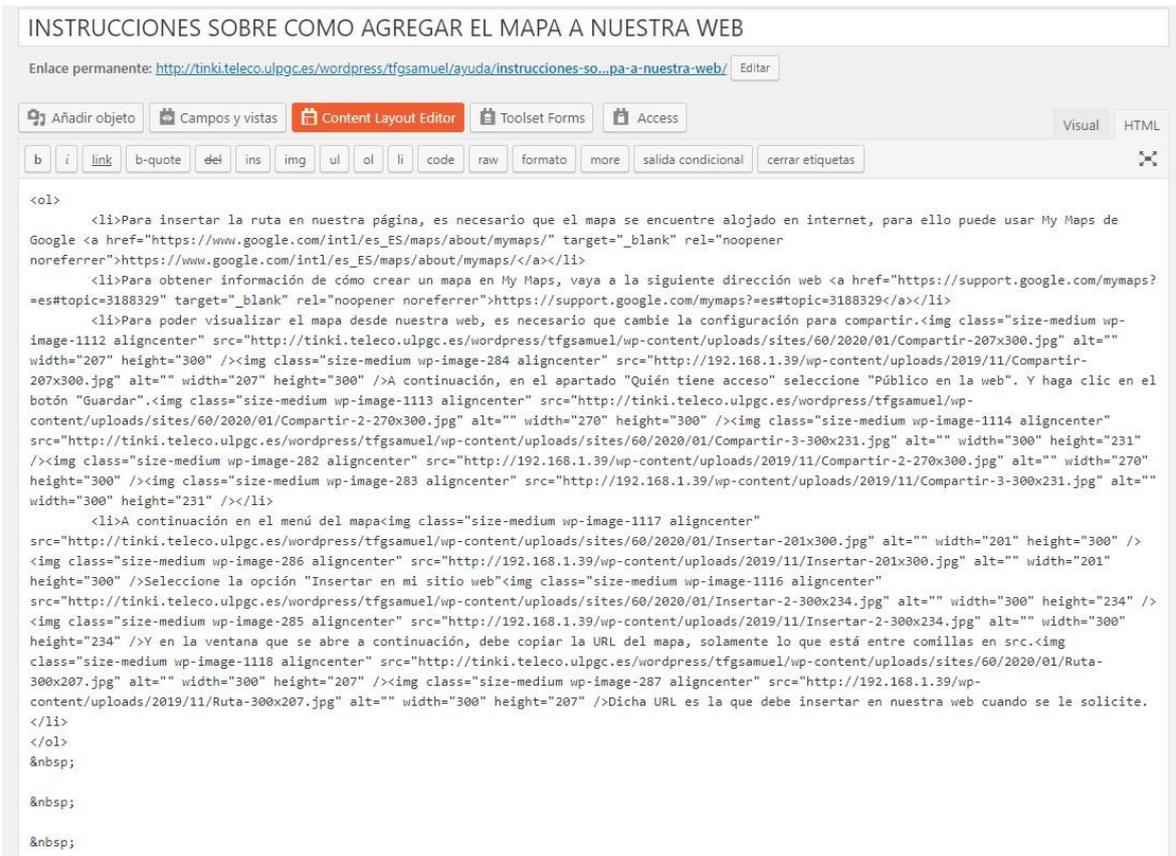


Figura 70. Generación de la página "Ayuda"

### 6.3.6. Generación de la página añadir rutas

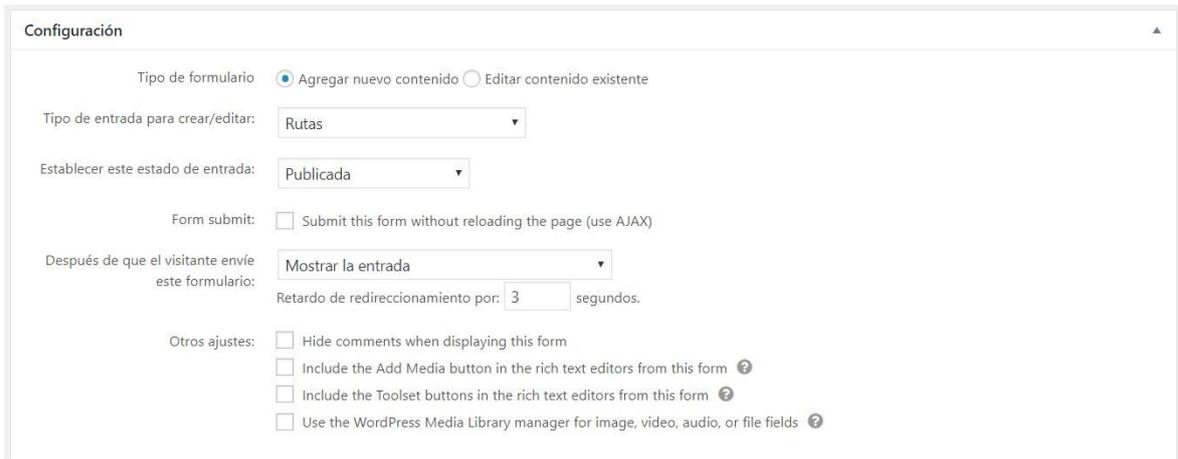
Para generar esta página debemos ir al panel de control de Toolset y seleccionar "Añadir rutas" para luego hacer clic en "Agregar nuevo". Una vez hecho esto, solo nos da la posibilidad de insertar un título, ya que en esta página es donde se encuentra el formulario de entrada que hay que rellenar para subir una ruta al sitio web. Para ello se le asociara un "Template layout", que veremos más adelante, a esta página. Por lo tanto, en nuestro caso le ponemos el título "Añadir ruta".



Figura 71. Generación de la página "Añadir ruta"

### 6.3.7. Formulario para subir las rutas

En este apartado veremos cómo se ha construido el formulario de entrada para que el usuario pueda subir rutas a nuestro sitio. Para tal fin, dentro de Toolset nos dirigimos a “Formularios de entrada” y “Agregar nuevo”. Donde se nos pide que insertemos el nombre del formulario, en nuestro caso ese nombre es “Formulario Subir Ruta”, seguidamente se nos piden unos ajustes de configuración que podemos ver en la *Figura 72*.



The screenshot shows the configuration window for a form. The title is 'Configuración'. Under 'Tipo de formulario', the radio button for 'Agregar nuevo contenido' is selected. 'Tipo de entrada para crear/editar:' is set to 'Rutas'. 'Establecer este estado de entrada:' is set to 'Publicada'. 'Form submit:' has the checkbox 'Submit this form without reloading the page (use AJAX)' unchecked. 'Después de que el visitante envíe este formulario:' is set to 'Mostrar la entrada'. Below this, 'Retardo de redireccionamiento por:' is set to '3' segundos. Under 'Otros ajustes:', there are four unchecked checkboxes: 'Hide comments when displaying this form', 'Include the Add Media button in the rich text editors from this form', 'Include the Toolset buttons in the rich text editors from this form', and 'Use the WordPress Media Library manager for image, video, audio, or file fields'.

*Figura 72. Configuración del formulario subir rutas*

Como podemos observar en la *Figura 72*, hemos elegido que se agregue nuevo contenido cada vez que se envíe este formulario. Este contenido será del tipo de entrada “Rutas” y se publicará automáticamente. También vemos que lo configuramos para que, al enviar el formulario, se muestre la entrada (“Ruta”) después de 3 segundos.

A continuación, en el editor de formulario activamos el modo experto, ya que en este podremos modificar e insertar los campos necesarios a través de código. Esto es muy importante ya que, si no lo hiciéramos así, no podríamos insertar el mapa de la ruta con una URL en el formulario.

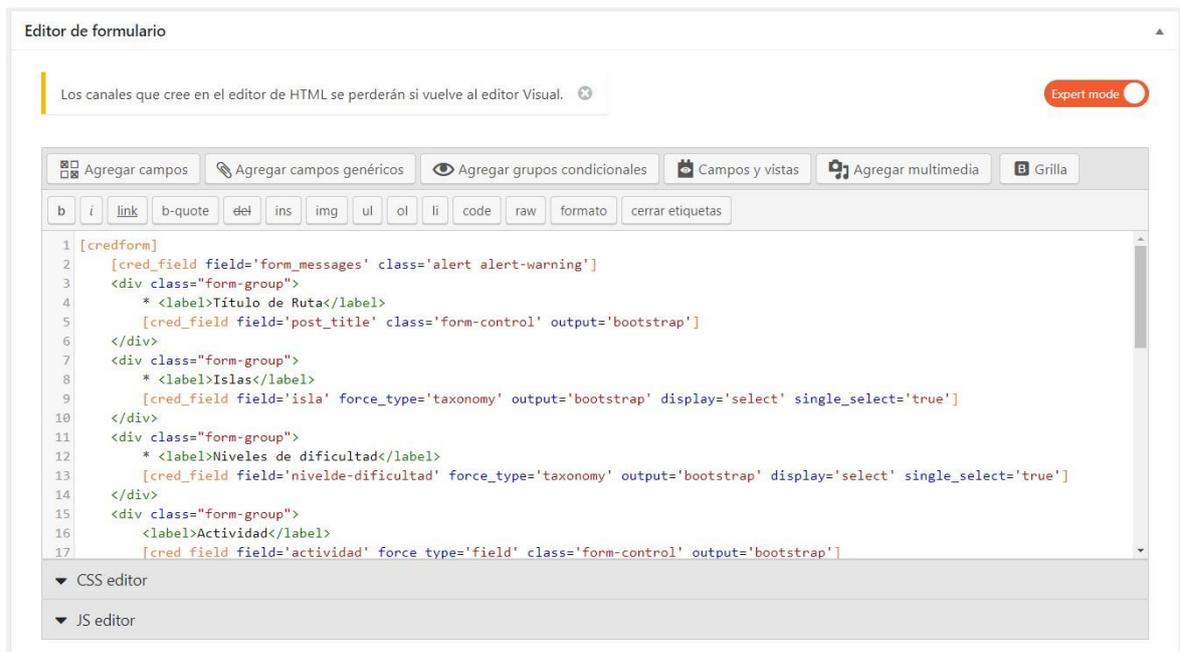


Figura 73. Editor del formulario subir rutas

A continuación, veremos el código introducido en el editor:

```
[credform]
[cred_field field='form_messages' class='alert alert-warning']
<div class="form-group">
  * <label>Título de Ruta</label>
  [cred_field field='post_title' class='form-control' output='bootstrap']
</div>
<div class="form-group">
  * <label>Islas</label>
  [cred_field field='isla' force_type='taxonomy' output='bootstrap' display='select' single_select='true']
</div>
<div class="form-group">
  * <label>Niveles de dificultad</label>
  [cred_field field='nivelde-dificultad' force_type='taxonomy' output='bootstrap' display='select' single_select='true']
</div>
<div class="form-group">
  <label>Actividad</label>
  [cred_field field='actividad' force_type='field' class='form-control' output='bootstrap']
</div>
<div class="form-group">
  * <label>Descripción</label>
  [cred_field field='descripcion' force_type='field' class='form-control' output='bootstrap']
</div>
<div class="form-group">
  <label>Fotos</label>
  [cred_field field='fotos' force_type='field' class='form-control' output='bootstrap']
</div>
<div class="form-group">
  * <label>Dirección</label>
  [cred_field field='direccion' force_type='field' class='form-control' output='bootstrap']
</div>
```

```

<div class="form-group">
  <label>Distancia</label>
  [cred_field field='distancia' force_type='field' class='form-control' output='bootstrap']
</div>
<div class="form-group">
  <label>Tiempo de duración</label>
  [cred_field field='tiempo-de-duracion' force_type='field' class='form-control'
output='bootstrap']
</div>
<div class="form-group">
  * <label>URL DE LA RUTA</label>
  [cred_field field='mapa-ruta' force_type='field' class='form-control' output='bootstrap']
  Para obtener información sobre como agregar el mapa de la ruta, haga clic <a
href="http://192.168.1.39/index.php/ayuda/instrucciones-sobre-como-agregar-el-mapa-a-nuestra-web/"
target="_blank" rel="noopener noreferrer">aquí</a>.
</div>
<div>
  * Estos campos son obligatorios
</div>
[cred_field field='form_submit' output='bootstrap' value='Enviar' class='btn btn-primary btn-lg']
[/credform]

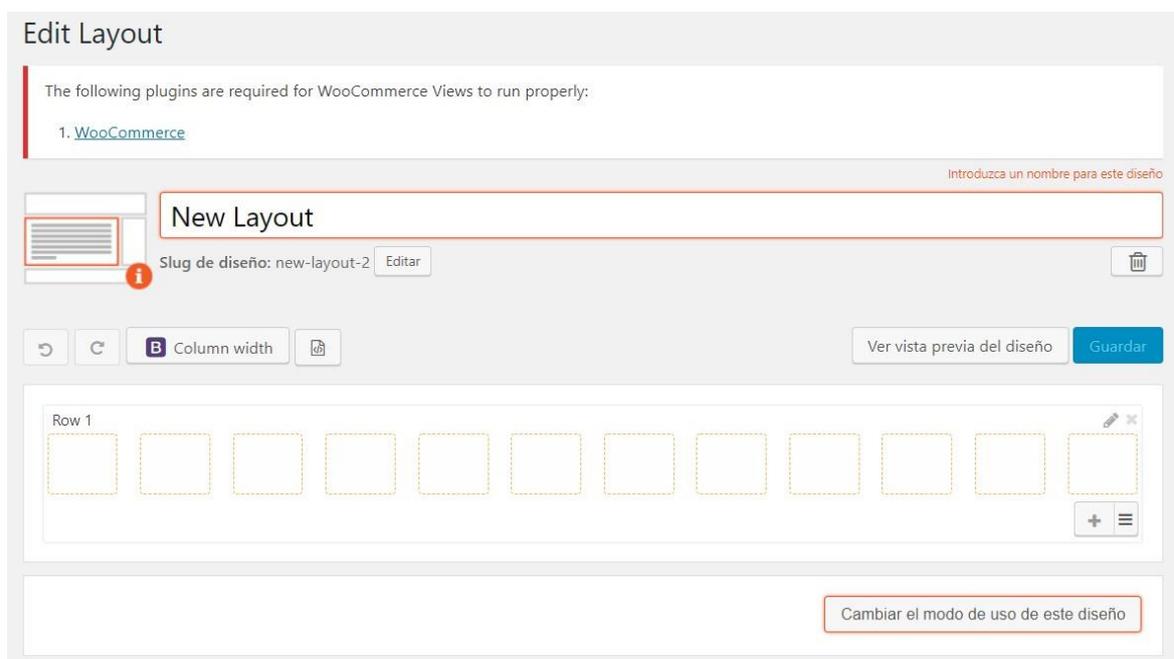
```

Como podemos ver en el código, el formulario se compone de:

- Un campo de texto llamado “Título de Ruta”.
- Un *dropdown* llamado “Islas” donde únicamente podemos elegir una opción ya que es una taxonomía.
- Un *dropdown* llamado “Niveles de dificultad” donde solo podemos elegir una opción ya que es una taxonomía.
- Un *checkbox* con el nombre “Actividad” donde se puede seleccionar más de una opción.
- Un campo para la introducción de texto nombrado “Descripción”.
- Un campo llamado “Fotos” que nos permite agregar fotos. Se permite agregar más de una imagen.
- Un campo de dirección que llamamos con el mismo nombre “Dirección”.
- Un campo de texto para introducir la distancia que lo llamamos “Distancia”.
- Un campo de texto llamado “Tiempo de duración” para introducir la duración de la ruta.
- Un campo de texto llamado “URL de la ruta” donde introducir la URL del mapa de la ruta.
- También vemos como insertamos texto aclaratorio y un link a la página ayuda de nuestra web.

### 6.3.8. *Layout para rutas*

En este apartado se mostrará cómo se creó el *layout* para las rutas, es decir, como se va a mostrar el contenido de cada una de las rutas en nuestra web. Para ello hacemos uso de Toolset, seleccionando la opción “Layouts” y seguidamente “Agregar nuevo”. Donde se nos muestra la ventana que podemos ver en la *Figura 74*.



*Figura 74. Crear layout*

Como vemos, tenemos que añadir un nombre al *layout* para luego poder identificarlo fácilmente a la hora de asignarlo. Y, a través de filas, se pueden agregar distintos tipos de celdas de diseño para mostrar el contenido.

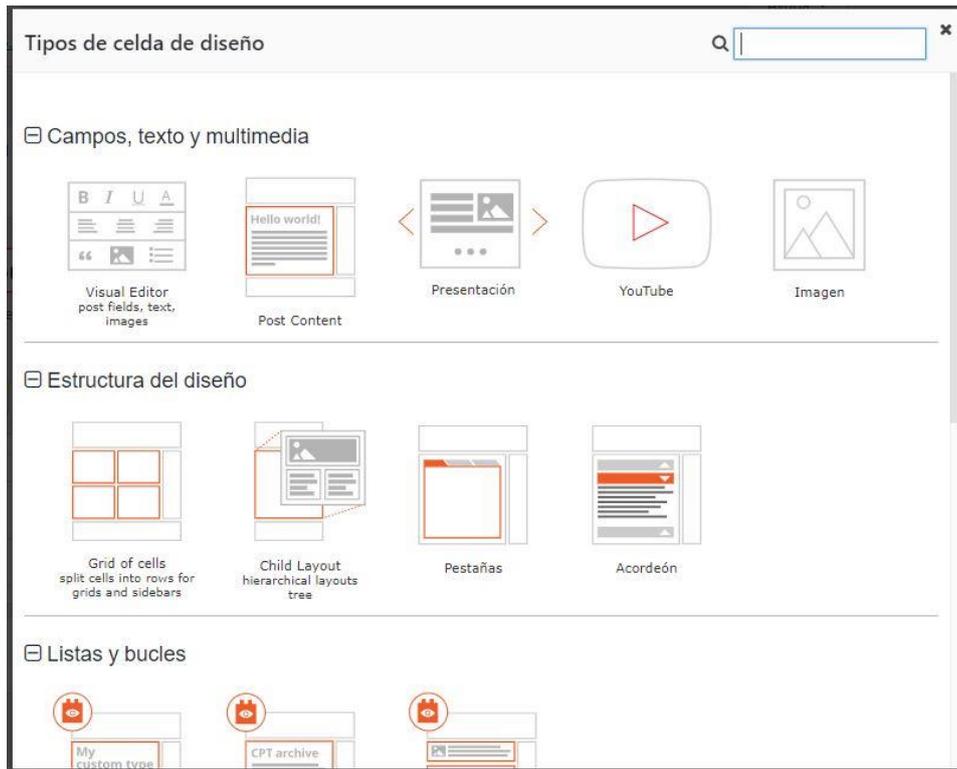


Figura 75. Elección de celdas de diseño

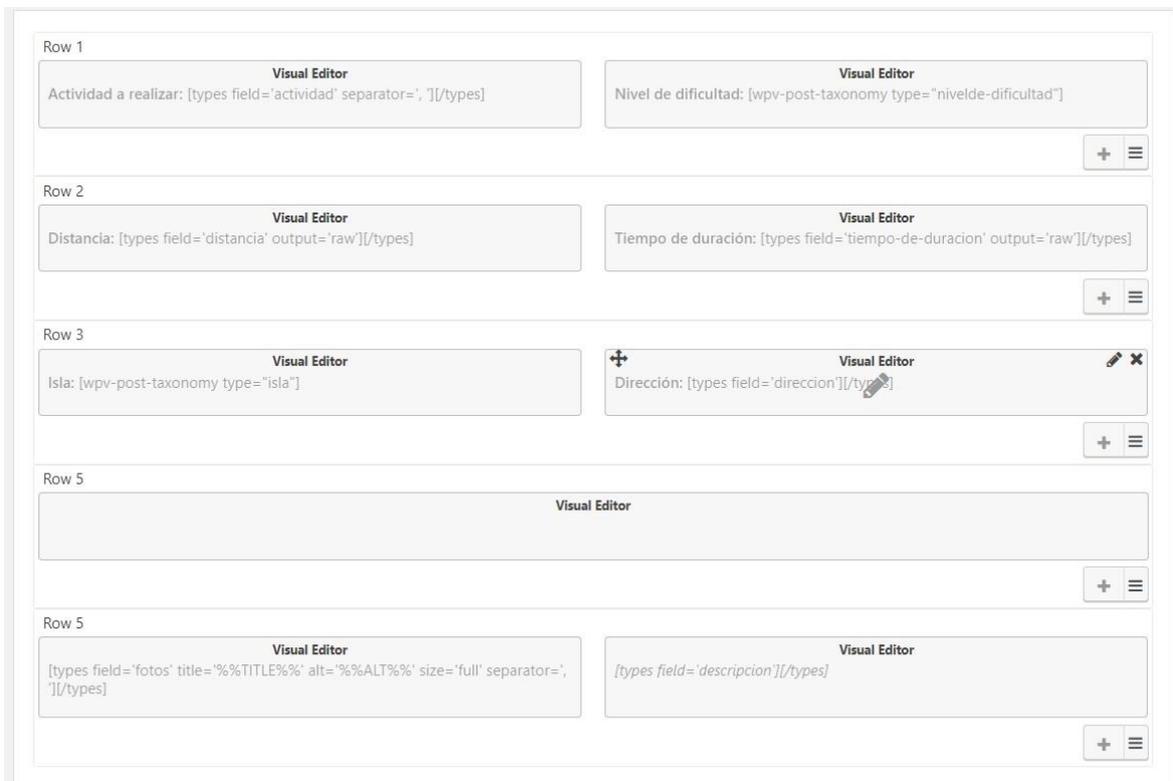


Figura 76. Creación del layout "Plantilla para ruta individual"

Como observamos en la *Figura 76*, en la parte izquierda de la fila 1 nos encontraremos un campo de texto donde aparece la “Actividad a realizar” y en la parte derecha otro campo de texto donde aparece el “Nivel de dificultad”. En la fila 2 tenemos un campo de texto para la “Distancia” y otro campo de texto para el “Tiempo de duracion”. La fila 3 la encontramos dividida en dos partes, en la parte izquierda nos encontramos con un campo de texto donde va el nombre de la “Isla” y en la parte derecha un campo de texto para la “Dirección”. En la fila 4, es donde nos aparece el mapa, para que esto ocurra, debemos construir un *iframe* dentro de esta celda debido a que los formularios de Toolset no admiten insertar un *iframe* directamente, por lo tanto, tenemos que insertar la URL en el formulario de entrada y manejarlo como texto y en este layout construir el *iframe* obteniendo la ruta del formulario. Para ello insertamos el siguiente código HTML en esta celda:

```
<p><iframe src="[types field='mapa-ruta' output='raw'][/types]" width="600" height="450"
frameborder="0" style="border:0;" allowfullscreen=""></iframe></p>
```

A continuación, tenemos la fila 5 donde se encuentra un campo de imagen para mostrar las fotos y un campo de texto donde va ubicada la descripción. Para finalizar con el desarrollo de este layout, debemos elegir para que *Post Types* se utiliza dicho layout. En nuestro caso seleccionamos “Rutas”.

### **6.3.9. Layout plantilla para subir rutas**

Este layout es el encargado de mostrar el contenido del formulario online para subir rutas a nuestro sitio web. A continuación, veremos el procedimiento realizado para su desarrollo.

Para comenzar seleccionamos la opción “Layouts” dentro de Toolset y seguidamente “Agregar nuevo” para así obtener la pantalla de la *Figura 74* donde introducimos el nombre del layout, en este caso “Plantilla para Subir Rutas”. Posteriormente agregamos una sola columna con el ancho máximo permitido, donde introducimos una celda de tipo formulario, para ello seleccionamos “Post Form” dentro del apartado “Formularios” como podemos ver en la *Figura 77*.

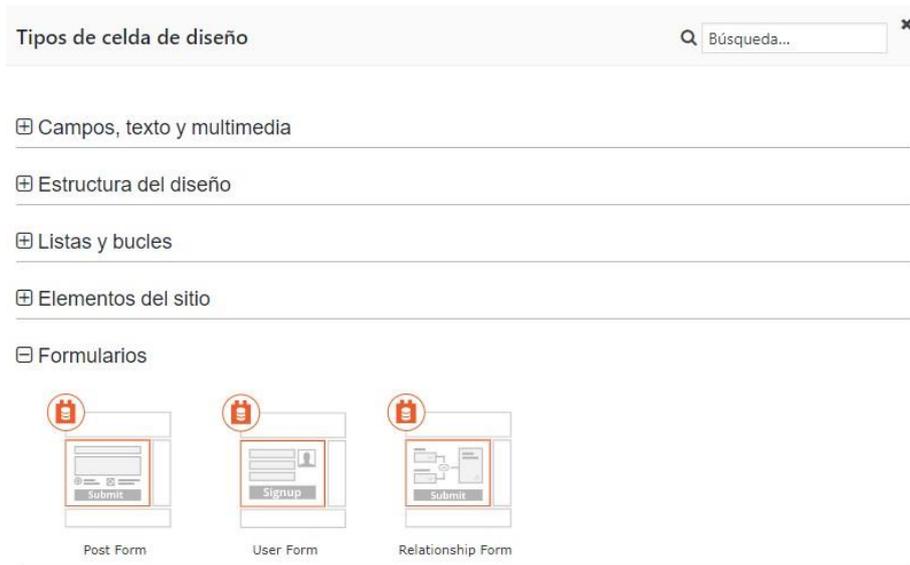


Figura 77. Selección del tipo de celda Post Form

A continuación, se nos abre la ventana que veremos en la *Figura 78*, donde debemos insertar los valores para la creación de la celda. En nuestro caso seleccionamos la opción “*Use an existing Post Form*” y en el desplegable seleccionamos el formulario adecuado, en este caso “Formulario Subir Ruta” y le damos a “Crear celda”.

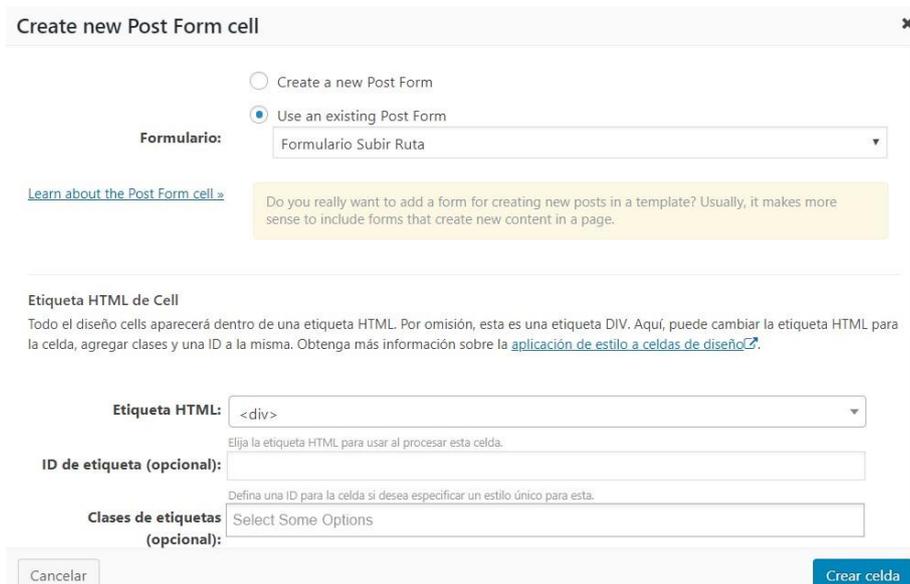


Figura 78. Creación de celda de formulario

Y para finalizar asignamos este layout al *post* “Añadir ruta”.

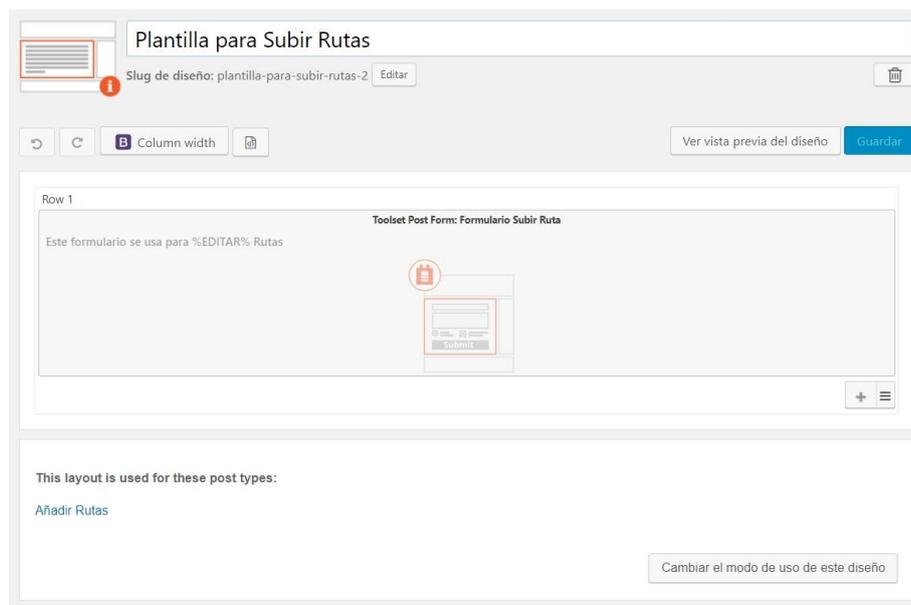


Figura 79. Creación del layout “Plantilla para Subir Rutas”

### 6.3.10. Modificaciones realizadas con el plugin REST API Controller

Debido a que necesitamos poder enviar los datos de nuestra web al framework IUMATI hemos utilizado el plugin de WordPress REST API Controller. Este nos da la posibilidad de elegir qué datos estarán disponibles para las solicitudes de la API REST y así el framework puede solicitar los datos necesarios a través de los JSON generados.

Para empezar a utilizar este plugin debemos irnos al menú “Herramientas” en el panel de control de Toolset y, dentro de este, seleccionar “REST API Controller” para que se nos abra la configuración del plugin. Una vez dentro, podremos ver dos apartados, uno con los “*Posts Types*” de nuestro sitio y otro con las “*Taxonomías*”.

En primer lugar, vamos al apartado de los “*Posts Types*” y seleccionamos aquellos que queremos que estén disponibles para la API. En nuestro caso el framework IUMATI solo necesita el *Post Type* “Rutas” para generar las apps. Por lo tanto, solo ponemos en “Enabled” el *Post Type* “Ruta” y dentro de este podemos ir activando o desactivando los campos que componen la ruta (“Actividad”, “Descripción”, “Fotos”, etc.) según nuestra necesidad. En este caso dejamos todos los campos en ON.

En las Figuras 80 y 81 podremos observar los cambios realizados en el apartado “*Posts Types*”.

## WP REST API Controller Settings

Post Types
Taxonomies

Toggle visibility of post types and select meta data to the REST API.

<b>oembed_cache</b>	<input type="checkbox"/> DISABLED	Expose the <b>Respuesta oEmbed</b> post type to the REST API.
<b>user_request</b>	<input type="checkbox"/> DISABLED	Expose the <b>Petición de usuario</b> post type to the REST API.
<b>wp_block</b>	<input checked="" type="checkbox"/> ENABLED	<a href="http://tinki.teleco.ulpgc.es/wordpress/tfgsamuel/wp-json/wp/v2/wp_block">http://tinki.teleco.ulpgc.es/wordpress/tfgsamuel/wp-json/wp/v2/wp_block</a> <span>Edit Endpoint</span> Expose the <b>Bloque</b> post type to the REST API.
<b>dd_layouts</b>	<input type="checkbox"/> DISABLED	Expose the <b>Layout</b> post type to the REST API.
<b>cred-form</b>	<input type="checkbox"/> DISABLED	Expose the <b>Post Form</b> post type to the REST API.
<b>cred-user-form</b>	<input type="checkbox"/> DISABLED	Expose the <b>User Form</b> post type to the REST API.

Figura 80. REST API Controller – Posts Types

ruta ENABLED

<http://tinkiteleco.ulpgc.es/wordpress/hgasamuel/wp-json/wp/v2/ruta> Edit Endpoint

<span>DISABLE ALL</span>	Meta Key	Custom Meta Key
<input checked="" type="checkbox"/>	_layouts_template	<input type="text" value="_layouts_template"/>
<input checked="" type="checkbox"/>	_wp_page_template	<input type="text" value="_wp_page_template"/>
<input checked="" type="checkbox"/>	_edit_lock	<input type="text" value="_edit_lock"/>
<input checked="" type="checkbox"/>	_edit_last	<input type="text" value="_edit_last"/>
<input checked="" type="checkbox"/>	_wpcf-fotos-sort-order	<input type="text" value="_wpcf-fotos-sort-order"/>
<input checked="" type="checkbox"/>	wpcf-actividad	<input type="text" value="wpcf-actividad"/>
<input checked="" type="checkbox"/>	wpcf-descripcion	<input type="text" value="wpcf-descripcion"/>
<input checked="" type="checkbox"/>	wpcf-fotos	<input type="text" value="wpcf-fotos"/>
<input checked="" type="checkbox"/>	wpcf-direccion	<input type="text" value="wpcf-direccion"/>
<input checked="" type="checkbox"/>	wpcf-distancia	<input type="text" value="wpcf-distancia"/>
<input checked="" type="checkbox"/>	wpcf-tiempo-de-duracion	<input type="text" value="wpcf-tiempo-de-duracion"/>
<input checked="" type="checkbox"/>	wpcf-mapa-ruta	<input type="text" value="wpcf-mapa-ruta"/>
<input checked="" type="checkbox"/>	_wp_trash_meta_status	<input type="text" value="_wp_trash_meta_status"/>
<input checked="" type="checkbox"/>	_wp_trash_meta_time	<input type="text" value="_wp_trash_meta_time"/>
<input checked="" type="checkbox"/>	_wp_desired_post_slug	<input type="text" value="_wp_desired_post_slug"/>

Expose the **Ruta** post type to the REST API.

anadir-ruta DISABLED

Expose the **Añadir Ruta** post type to the REST API.

ayuda DISABLED

Expose the **Ayudas** post type to the REST API.

widget-area DISABLED

Expose the **Widget Area** post type to the REST API.

Save Settings Clear Cache

Figura 81. Configuración del Post Type “Ruta” en REST API Controller

Ahora nos queda configurar el apartado “Taxonomías”, para ello nos vamos a dicho apartado y activamos la taxonomía “isla”, debido a que es la única que nos interesa que reciba el framework. Dentro de esta solo nos permite habilitar el campo imagen de la isla, pero como no nos interesa enviar dicha imagen lo dejaremos desactivado.

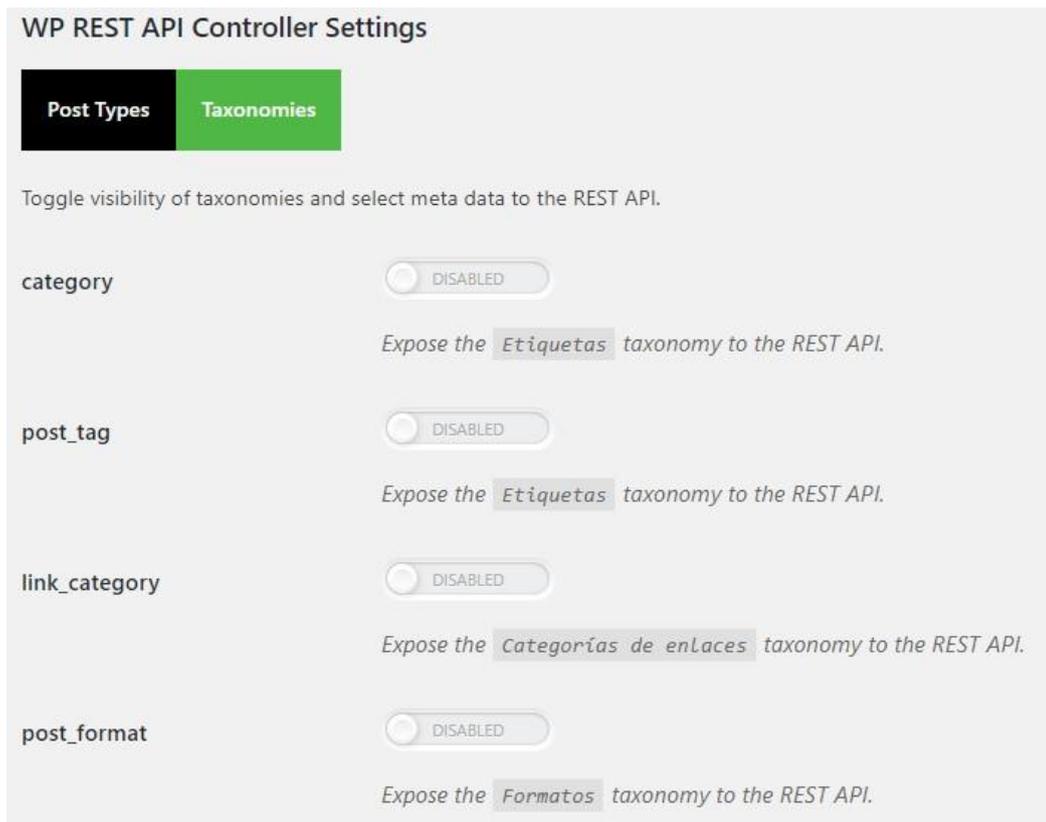


Figura 82. REST API Controller – Taxonomías

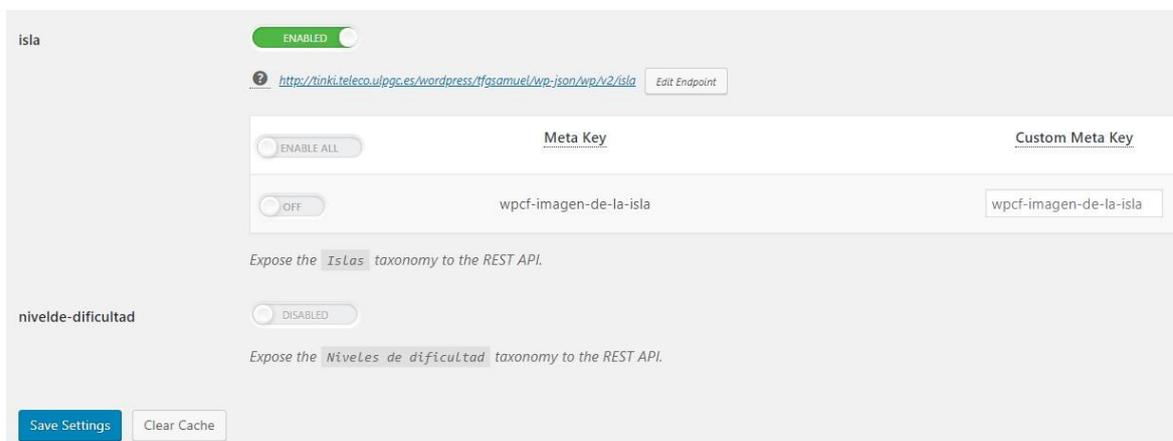


Figura 83. Configuración de la taxonomía isla en REST API Controller

Como podemos observar en la *Figura 81* y en la *Figura 83*, el plugin nos da una URL para acceder al *Post Type* “Ruta” o a la taxonomía “Islas” respectivamente a través de la API REST.

En la *Figura 84* y *85* vemos los JSON que se muestran al acceder a dichas URLs.





# **CONCLUSIONES Y LÍNEAS FUTURAS**



## **7. Conclusiones y líneas futuras**

### **7.1. Introducción**

Este proyecto surge ante la necesidad de crear una interfaz web cómoda e intuitiva para que el usuario pueda introducir los datos en el Framework IUMATI sin necesidad de tener conocimientos de este. Para ello, se optó por desarrollar un sitio web donde el usuario pueda insertar los datos necesarios y estos sean recibidos en el framework.

Tras la finalización del proyecto, ha llegado el momento de sacar las conclusiones correspondientes. En los siguientes apartados haremos un repaso a los objetivos planteados en un principio y si realmente hemos obtenido el resultado esperado. También reflexionaremos sobre las posibles líneas futuras que se pueden seguir para la mejora de este proyecto.

### **7.2. Conclusiones**

Si revisamos los objetivos planteados al comienzo del proyecto, podemos concluir afirmando que hemos alcanzado el objetivo principal de este: desarrollar un sitio web específico, para una aplicación concreta, con una interfaz web para que terceros puedan introducir la información necesaria para el framework IUMATI de una manera cómoda e intuitiva. En este caso se ha desarrollado una página web de rutas en las Islas Canarias, utilizando WordPress y el plugin Toolset, donde los usuarios pueden ver las distintas rutas y agregar nuevas rutas a la web insertando los datos de estas en una plantilla. Además de esto, hemos utilizado el plugin REST API Controller para configurar y habilitar el envío de los datos de la web al Framework IUMATI, que se encargará de generar las apps para iOS y Android de la web.

### **7.3. Líneas futuras**

Como posibles mejoras en el diseño de nuestro sitio web, podríamos introducir mensajes de alerta o información para el usuario a la hora de añadir una ruta. Insertar más campos de información en las rutas, como podría ser la altitud máxima y mínima de la ruta, la temperatura adecuada para la realización de la ruta, materiales necesarios para hacer la ruta (cuerdas, bastones de senderismo), etc. También se podría desarrollar un formulario de registro y login para que los usuarios puedan registrarse o loguearse en la web.

Otra línea para seguir podría ser la de dotar a nuestro sitio web de rutas, de la posibilidad de que un usuario pueda subir el mapa de la ruta que desee añadir a través de un fichero. Esto es muy útil debido a que hoy en día existen multitud de dispositivos con GPS integrados (smartwatches, smartband, smartphones) que van guardando la información en un fichero .GPX, para luego poder importarlo a un PC. Dicho fichero sería el que pudiéramos introducir directamente en el apartado del "Mapa de la ruta" y evitarle al usuario el tener que estar creando, con otras herramientas, el mapa de la ruta.

Otro punto interesante, sería el de implementar diferentes sitios web con temáticas distintas (tienda online, administración de un centro escolar, etc.) y sus correspondientes plantillas. Donde el usuario pueda insertar la información e ir creando su sitio web sin tener conocimientos de programación y, a su vez, a través del framework IUMATI se le generen las apps iOS y Android de su sitio web.

# **PRESUPUESTO**



## 8. Presupuesto

En este apartado se verá el presupuesto de este proyecto. El cual estructuraremos en los siguientes apartados:

- Recursos humanos.
- Recursos hardware.
- Recursos Software.
- Material fungible.
- Presupuesto total del proyecto.

### 8.1. Recursos humanos

Para el cálculo de los costes en recursos humanos, consideraremos que son necesarios tres meses para la realización del proyecto. Este periodo de tiempo se divide en dos partes: un mes para el estudio del proyecto y formación en las herramientas utilizadas y dos meses para el desarrollo. Si tenemos en cuenta que el ingeniero trabajará 8 horas al día, de lunes a viernes, y tiene un sueldo de 1500 € al mes [22].

Actividad	Tiempo empleado	Coste mensual	Importe final
Estudio y formación	1 mes	1500 €	1500 €
Desarrollo	2 meses	1500 €	3000 €
			<b>4500 €</b>

*Tabla 1. Costes de recursos humanos*

Como podemos observar en la tabla 1, el coste asociado a los recursos humanos es de cuatro mil quinientos euros (4500 €).

### 8.2. Hardware

Para el desarrollo de este proyecto se necesita un ordenador personal, un servidor, y conexión de banda ancha a internet. En el caso del equipamiento, se considera un periodo de amortización de dos años.

<b>Hardware</b>	<b>Coste de adquisición</b>	<b>Periodo de amortización</b>	<b>Tiempo empleado</b>	<b>Coste mensual</b>	<b>Importe final</b>
Ordenador personal	1200 €	24 meses	3 meses	50 €	150 €
Servidor	2500 €	24 meses	3 meses	104,16 €	312,5 €
Conexión a Internet	-----		3 meses	74,40 €	223,2 €
					<b>685,7 €</b>

*Tabla 2. Costes de los recursos hardware*

Según la tabla 2 el coste total de los recursos hardware necesarios es de seiscientos ochenta y cinco euros con setenta céntimos de euro (685,7 €).

### **8.3. Software**

Algunos de los recursos software que utilizamos son de licencia libre, por lo que no representan coste alguno. Para los demás productos software se tienen en cuenta los precios de las licencias para empresas.

<b>Software</b>	<b>Coste de adquisición</b>	<b>Tiempo empleado</b>	<b>Importe final</b>
WordPress	0 €	3 meses	0 €
Toolset	149 € / año	3 meses	149 €
Windows	0 €	3 meses	0 €
Google Chrome	0 €	3 meses	0 €
Microsoft Word	8,80 € / mes	3 meses	26,4 €
Google Maps Platform	0 €	3 meses	0 €
			<b>175,4 €</b>

*Tabla 3. Costes de los recursos software*

Como vemos en la tabla 3, el coste asociado a los recursos software de este proyecto asciende a un total de ciento setenta y cinco euros con cuarenta céntimos de euro (175,4 €).

### **8.4. Material fungible**

Se incluye todo aquel material gastado durante la realización del proyecto.

Concepto	Cantidad	Coste Unidad	Importe final
Paquete de papel DIN-A4	1	3,90 €	3,90 €
Cartuchos de tinta para impresora	2	25 €	50 €
Encuadernación	1	10 €	10 €
CD-ROM	1	0,80 €	0,80 €
			<b>64,7 €</b>

Tabla 4. Costes del material fungible

Según la tabla 4, el coste del material fungible de este proyecto es de sesenta y cuatro euros con setenta céntimos de euro (64,7 €).

### 8.5. Presupuesto total del proyecto

En la siguiente tabla observaremos el cálculo del coste total del proyecto teniendo en cuenta el 7% del Impuesto General Indirecto Canario (IGIC).

Concepto	Coste
Recursos Humanos	4500 €
Recursos Hardware	685,7 €
Recursos Software	175,4 €
Material Fungible	64,7 €
Total, antes de impuestos	5425,8 €
IGIC (7%)	379,81 €
<b>TOTAL</b>	<b>5805,61 €</b>

Tabla 5. Presupuesto total del proyecto

Por tanto, el presupuesto total del proyecto “Plataforma Web Configurable basada en Plantillas y Plugins de WordPress para la Generación de Aplicaciones Móviles en iOS y Android” es de **cinco mil ochocientos cinco euros con sesenta y un céntimo de euro (5805,61 €)**.

Fdo.: D. Samuel Martín Rodríguez

Las Palmas de Gran Canaria, enero 2020



# **BIBLIOGRAFÍA**



## 9. Bibliografía

- [1] M. Porras, «Cuánto cuesta una App móvil y cómo desarrollarla. Precio y tipos,» 31 Julio 2017. [En línea]. Available: <http://aulacm.com/precio-desarrollar-app-aplicacion-movil/>.
- [2] Q-Success, «W3Techs - World Wide Web Technology Surveys,» 2017. [En línea]. Available: <https://w3techs.com/>.
- [3] C. Aubry, WordPress 3 - un CMS para crear su sitio Web, Ediciones Eni, 2011.
- [4] OnTheGoSystems Limited, «Toolset,» 2018. [En línea]. Available: <https://wp-types.com/es/>.
- [5] WordPress, «WORDPRESS.ORG,» [En línea]. Available: <https://es.wordpress.org/about/>. [Último acceso: 13 12 2019].
- [6] G. Mayoraz, «Ayuda WordPress,» 12 01 2008. [En línea]. Available: <https://ayudawp.com/el-padre-de-wordpress/>. [Último acceso: 13 12 2019].
- [7] «WP Avanzado,» 16 11 2017. [En línea]. Available: <https://wpavanzado.com/que-es-wordpress/>. [Último acceso: 13 12 2019].
- [8] OnTheGoSystems, «WPML,» 26 04 2016. [En línea]. Available: <https://wpml.org/es/documentation-2/proyectos-relacionados/toolset/>. [Último acceso: 14 12 2019].
- [9] OnTheGoSystems Limited, «Componentes de Toolset,» 24 10 2016. [En línea]. Available: <https://toolset.com/es/nueva-Página-de-inicio/componentes-de-toolset/>. [Último acceso: 14 12 2019].
- [10] F. Vázquez, «Toolset para WordPress. Una herramienta que lleva cualquier theme y WordPress al siguiente nivel,» 05 03 2016. [En línea]. Available: <https://fabianvazquez.es/toolset-plugin-custom-posts-taxonomias-wordpress/>. [Último acceso: 14 12 2019].
- [11] WORDPRESS.ORG, «WORDPRESS.ORG,» [En línea]. Available: <https://es.wordpress.org/plugins/wp-rest-api->

- controller/#how%20do%20i%20retrieve%20repeating%20postmeta%20fields%3F. [Último acceso: 16 12 2019].
- [12] D. Crockford, «Introducción a JSON,» [En línea]. Available: <https://www.json.org/json-es.html>. [Último acceso: 16 12 2019].
- [13] MDN contributors, «Trabajando con JSON,» 02 09 2019. [En línea]. Available: <https://developer.mozilla.org/es/docs/Learn/JavaScript/Objects/JSON>. [Último acceso: 16 12 2019].
- [14] Develapps, «Creación de Apps mediante un Servidor Parse,» 28 12 2017. [En línea]. Available: <http://www.develapps.com/es/noticias/creacion-de-apps-mediante-un-servidor-parse>. [Último acceso: 16 12 2019].
- [15] L. H. Acosta, «Georruta Transgrancanaria,» 03 03 2016. [En línea]. Available: <https://play.google.com/store/apps/details?id=es.ulpgc.iumati.android.georutatransgc&hl=es>. [Último acceso: 16 12 2019].
- [16] M. Rouse, «MySQL,» 01 2015. [En línea]. Available: <https://searchdatacenter.techtarget.com/es/definicion/MySQL>. [Último acceso: 24 12 2019].
- [17] Oracle, «Oracle MySQL,» 03 03 2014. [En línea]. Available: <https://www.oracle.com/es/mysql/>. [Último acceso: 24 12 2019].
- [18] Google, «MY MAPS,» [En línea]. Available: [https://www.google.com/intl/es\\_ES/maps/about/mymaps/](https://www.google.com/intl/es_ES/maps/about/mymaps/). [Último acceso: 16 12 2019].
- [19] Boluda.com, «Uso de los Custom Post Types en WordPress,» 30 10 2016. [En línea]. Available: <https://boluda.com/tutorial/uso-de-los-custom-post-types-en-wordpress/>. [Último acceso: 24 12 2019].
- [20] O. A. Folgueira, «Guía de la base de datos de WordPress,» 08 12 2017. [En línea]. Available: <http://www.oscarabadfolgueira.com/guia-la-base-datos-wordpress/>. [Último acceso: 26 12 2019].

- [21] BBVAOPEN4U, «API REST: qué es y cuáles son sus ventajas en el desarrollo de proyectos,» 23 03 2016. [En línea]. Available: <https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos>. [Último acceso: 24 12 2019].
- [22] [cuantosecobra.com](http://cuantosecobra.com), «¿Cuánto cobra un ingeniero en telecomunicaciones?,» [En línea]. Available: <https://www.cuantosecobra.com/blog/61377-cuanto-cobra-un-ingeniero-en-telecomunicaciones/>. [Último acceso: 07 01 2020].
- [23] R. Prieto, «Cómo Instalar WordPress en Raspberry Pi,» 12 08 2018. [En línea]. Available: <https://www.silocreativo.com/instalar-wordpress-raspberry-pi/>. [Último acceso: 12 02 2019].



# **ANEXOS**



## 10.1. Guía de instalación de WordPress

Para desarrollar un sitio WordPress lo ideal es instalar este en un servidor local LAMP (Linux + Apache + MySQL + PHP), en nuestro caso usamos una Raspberry pi bajo el sistema operativo Raspbian. Luego, cuando el sitio web a desarrollar este listo podremos instalarlo en un servidor remoto.

Ya tenemos Linux al tener el sistema operativo Raspbian instalado en nuestra Raspberry pi, por lo tanto, el siguiente paso es instalar Apache. Para ello abrimos la línea de comandos en nuestra Raspberry y realizamos un update de la lista de paquetes introduciendo lo siguiente:

```
sudo apt-get update
```

Y posteriormente también un upgrade de todos los paquetes ya instalados:

```
sudo apt-get dist-upgrade
```

Ahora instalamos Apache introduciendo:

```
sudo apt-get install apache2 -y
```

Para comprobar que todo funciona, introduce en el navegador de tu Raspberry Pi la dirección localhost: `http://localhost/`. Deberías ver la página de prueba de Apache.

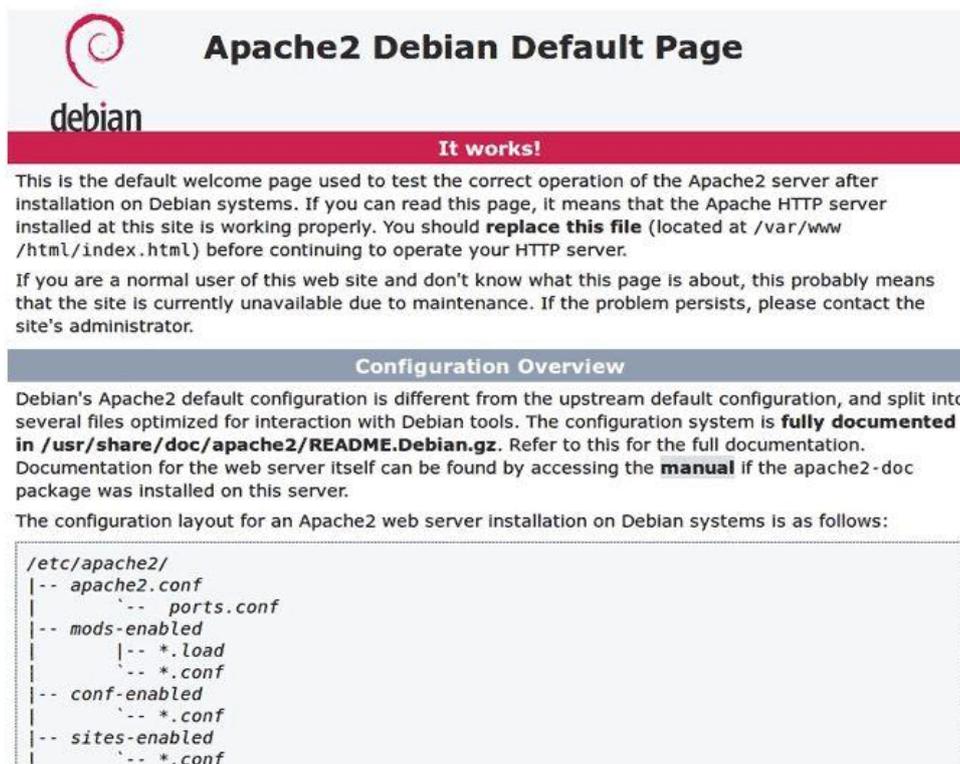


Figura 86. Pantalla de prueba de Apache

Ahora procederemos a instalar PHP introduciendo lo siguiente en la línea de comandos:

```
sudo apt-get install php5 libapache2-mod-php5 -y
```

y para finalizar con la instalación de PHP reiniciamos Apache:

```
sudo service apache2 restart
```

El último paso para crear nuestro servidor LAMP es instalar MySQL. Para ello, en la consola:

```
sudo apt-get install mysql-server php5-mysql -y
```

Durante la instalación te pedirán que introduzcas una contraseña para el usuario root. Es importante que guardes esta contraseña ya que puede ser útil en el futuro para permitir que WordPress acceda a la Base de datos.

Finalmente, cuando termine todo el proceso, reinicia Apache:

```
sudo service apache2 restart
```

A continuación, instalamos phpMyAdmin para posteriormente editar, crear o eliminar la base de datos de una forma más visual:

```
sudo apt-get install phpmyadmin
```

Durante la instalación nos preguntará en que servidor queremos instalarlo, seleccionaremos apache2, y con qué base de datos queremos conectar. Elegiremos la creada anteriormente con MySQL. Igualmente indicaremos YES cuando nos pregunte sobre configurar la base de datos para phpMyAdmin con dbconfig-common.

Cuando nos pregunte por la contraseña, usaremos la que hemos establecido anteriormente cuando instalamos MySQL. Después nos preguntarán una contraseña para phpMyAdmin. Podemos usar la misma u otra distinta, pero es importante que la recordemos. Finalmente, cuando haya terminado necesitamos editar una línea para indicarle a Apache que hemos instalado PHPMyAdmin. Para ello:

```
sudo nano /etc/apache2/apache2.conf
```

y añadiremos el final de este archivo lo siguiente:

```
Include /etc/phpmyadmin/apache.conf
```

Una vez realizado, salvamos presionado CTRL + X y pulsamos Y para aceptar los cambios. Reiniciamos Apache:

**sudo service apache2 restart**

y comprobamos que se ha instalado correctamente accediendo a la siguiente URL desde nuestro navegador en la Raspberry Pi <http://localhost/phpmyadmin> o bien desde otro equipo de la misma red . Debíamos ver algo similar a:



phpMyAdmin

Bienvenido a phpMyAdmin

Idioma - Language

Español - Spanish

Iniciar sesión

Usuario:

Contraseña:

Continuar

*Figura 87. Página de Login en phpMyAdmin*

Ahora ya tenemos instalado nuestro servidor LAMP, continuaremos con la instalación de WordPress. Para ello descargamos la última versión de WordPress desde la línea de comandos escribiendo:

**cd /var/www/html/**

con esto vamos a la carpeta donde queremos descargar WordPress. Ahora vamos a limpiar todo lo que haya (ojo, si tienes otra web o archivos se perderán, guárdalos antes si quieres conservarlos):

```
sudo rm *
```

y descargamos WordPress.

```
sudo wget http://wordpress.org/latest.tar.gz
```

Una vez terminemos, vamos a descomprimir el archivo descargado:

```
sudo tar xzf latest.tar.gz
```

Finalmente borramos el archivo comprimido ya que no lo vamos a necesitar:

```
sudo rm -rf wordpress latest.tar.gz
```

Perfecto. Un último paso, vamos a dar la propiedad de estos archivos a Apache, para ello:

```
sudo chown -R www-data: .
```

Ahora que ya tenemos instalado nuestro servidor LAMP y WordPress vamos a configurar la base de datos para WordPress.

Recuerda tu nombre de usuario y contraseña (normalmente root y tu contraseña elegida) y ejecuta con tus datos lo siguiente:

```
mysql -uroot -pcontraseña
```

(-u para usuario y -p para contraseña). Una vez hayamos conectado con MySQL, ya podemos crear nuestra Base de datos usando

```
mysql> create database wordpress;
```

Tras esto, debemos obtener la confirmación con algo similar a Query OK, 1 row affected (0.00 sec). Ya podemos salir de MySQL pulsando Ctrl + D

Por último debemos configurar Wordpress, para ello accedemos desde nuestro navegador a nuestra IP para iniciar la configuración de nuestro sitio. Nos preguntará por la información básica de nuestro servidor, que en nuestro caso es:

Nombre de la Base de Datos: wordpress

Nombre de usuario: root

Password: (tu contraseña elegida)

Database Host: localhost

Prefijo de tabla: wp\_

Para finalizar, le damos a siguiente y ya tendremos WordPress configurado y listo para trabajar en nuestra red local [23].

## 10.2. Instalación de Toolset

Primero, vamos a la página web <https://toolset.com/es/adquieralo-ahora-mismo/> donde debemos de registrarnos y realizar la compra de nuestro producto. Una vez realizada la compra debemos ir a la Página de descargas de Toolset <https://toolset.com/account/downloads/> y descargar el complemento types.

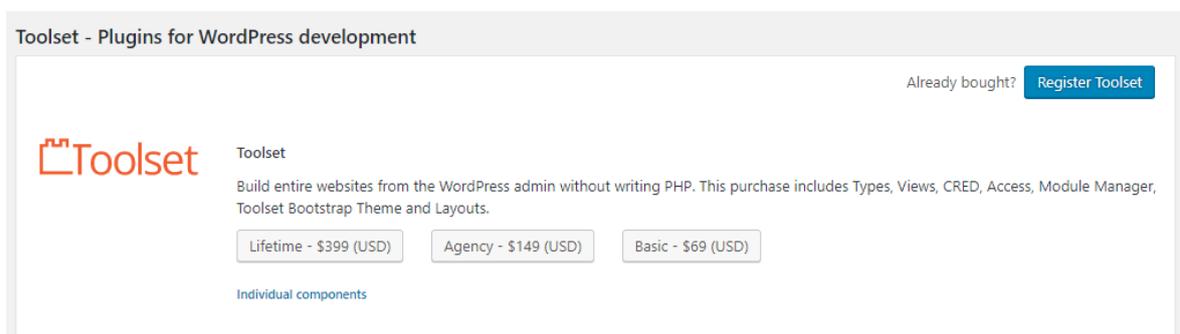
En segundo lugar, en el administrador de WordPress vamos la pestaña plugins y seleccionamos añadir nuevo, posteriormente hacemos clic en el botón subir plugin.

En tercer lugar, se nos abre un cuadro de dialogo en el que tenemos que elegir el archivo .ZIP con el complemento que descargamos anteriormente y hacer clic en instalar ahora.

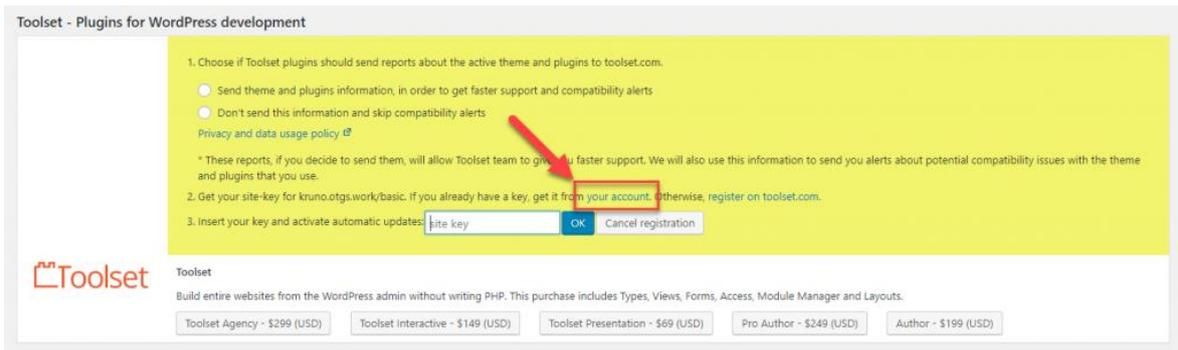
Por ultimo debemos activar el complemento.

Ahora para instalar de forma automática los demás complementos, debemos seguir los siguientes pasos:

1. Ir a plugins y agregar nuevo, posteriormente hacemos clic en la pestaña Comercial.
2. Hacemos clic en Register Toolset en la esquina superior derecha.



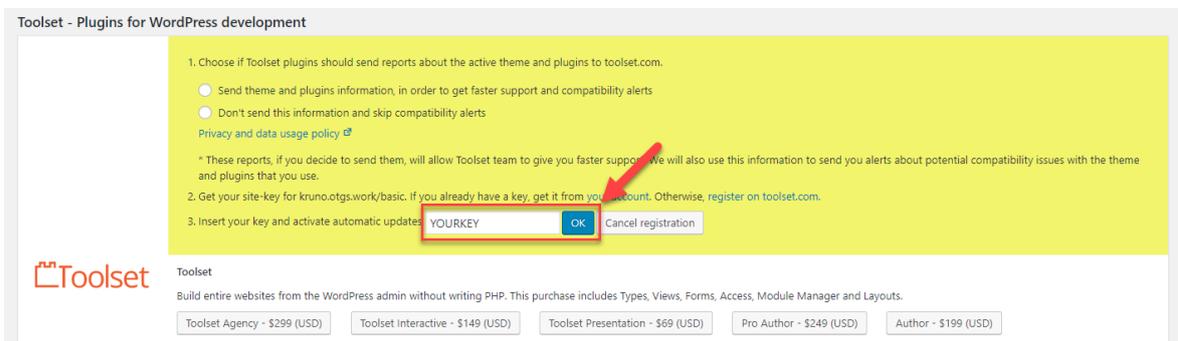
3. Hacemos clic en el enlace de su cuenta para registrar automáticamente este sitio.



4. Copiamos la clave del sitio. Seleccionándolo con el ratón (doble clic) y presionando CTRL-C.



5. Volvemos a WordPress y pegamos la clave del sitio para registrar Toolset en este sitio.



6. Seleccionamos los componentes del conjunto de herramientas para descargar. Para activarlos, marcamos activar después de la descarga.

**Toolset**

Build entire websites from the WordPress admin without writing PHP. This purchase includes Types, Views, Forms, Access, Module Manager and Layouts.

Updates channel: Production To select different update channels (beta, development) you must update your existing products to their most recent stable versions.

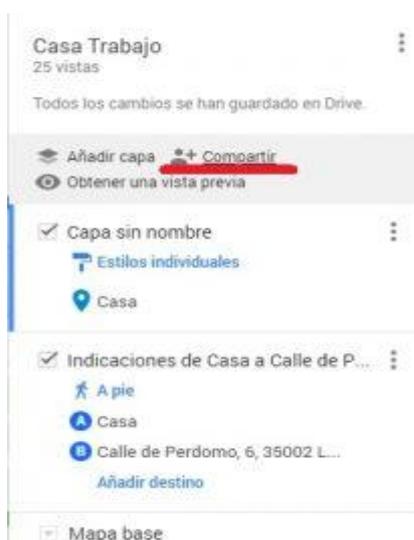
Plugin	Installed	Current	Released	
<input checked="" type="checkbox"/> Toolset Types	<b>3.3.1</b>	3.3.2	June 26, 2019	<a href="#">Release notes</a>
<input checked="" type="checkbox"/> Toolset Views		2.8.4	July 2, 2019	<a href="#">Release notes</a>
<input checked="" type="checkbox"/> Toolset Blocks		0.9.2	June 10, 2019	
<input type="checkbox"/> Toolset Layouts		2.5.2	January 28, 2019	<a href="#">Release notes</a>
<input checked="" type="checkbox"/> Toolset Forms		2.4	June 26, 2019	<a href="#">Release notes</a>
<input checked="" type="checkbox"/> Toolset Access		2.7.5	June 19, 2019	<a href="#">Release notes</a>
<input type="checkbox"/> Toolset WooCommerce Views		2.7.9	April 8, 2019	<a href="#">Release notes</a>
<input type="checkbox"/> Toolset Forms Commerce		1.8.1	August 8, 2018	<a href="#">Release notes</a>
<input checked="" type="checkbox"/> Toolset Maps		1.8.1	June 28, 2019	<a href="#">Release notes</a>
<input type="checkbox"/> Toolset Module Manager		1.8.6	March 26, 2019	<a href="#">Release notes</a>
<input type="checkbox"/> Toolset Advanced Export		1.0	January 24, 2017	<a href="#">Release notes</a>
<input type="checkbox"/> Toolset Layouts Migration		1.0	March 16, 2017	<a href="#">Release notes</a>
<input checked="" type="checkbox"/> Toolset Framework Installer		3.1.5.2	June 26, 2019	<a href="#">Release notes</a>

Activate after download

### 10.3. Guía de usuario

En este apartado se explicaran los pasos necesarios que debe seguir un usuario para subir una ruta a la página web.

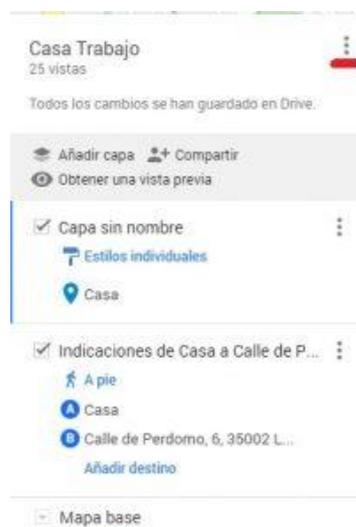
1. Para insertar la ruta en nuestra página, es necesario que el mapa se encuentre alojado en internet, para ello puede usar My Maps de Google [https://www.google.com/intl/es\\_ES/maps/about/mymaps/](https://www.google.com/intl/es_ES/maps/about/mymaps/)
2. Para obtener información de cómo crear un mapa en My Maps, vaya a la siguiente dirección web <https://support.google.com/mymaps?es#topic=3188329>
3. Para poder visualizar el mapa desde nuestra web, es necesario que cambie la configuración para compartir.



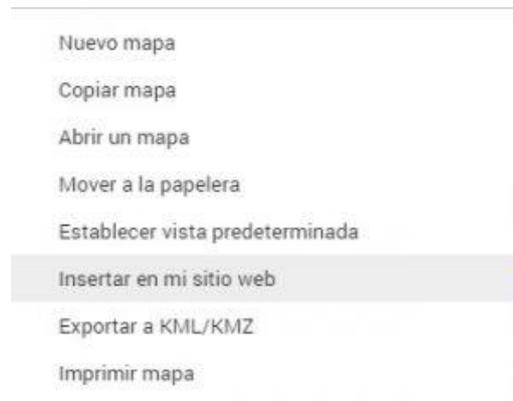
A continuación, en el apartado "Quién tiene acceso" seleccione "Público en la web". Y haga clic en el botón "Guardar".



4. A continuación, en el menú del mapa



Seleccione la opción "Insertar en mi sitio web"



Y en la ventana que se abre a continuación, debe copiar la URL del mapa, solamente lo que está entre comillas en src.



Dicha URL es la que debe insertar en nuestra web cuando se le solicite.