

Trabajo de Fin de Grado

Grado en Ingeniería Informática

Herramienta de automatización en WordPress para la web PeriodismoULL.

*Automation tool in WordPress for the website
PeriodismoULL*

Juan Eduardo Flores González

Dña. **Carina Soledad González González**, con N.I.F. 54064251Z catedrática de Universidad del Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutora

D. **Eduardo Nacimiento García**, con N.I.F. 78628112J investigador FPI adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como cotutor

C E R T I F I C A (N)

Que la presente memoria titulada:

“Herramienta de automatización en WordPress para la web PeriodismoULL”

ha sido realizada bajo su dirección por D. **Juan Eduardo Flores González**, con N.I.F. 78648184Y.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 7 de Septiembre de 2021

Agradecimientos

Quisiera agradecer a todas aquellas personas que me han apoyado en este largo viaje tanto a familiares, amigos, pareja, como a mis dos tutores que me han acompañado.

Licencia



© Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional.

Resumen

En la actualidad no se puede ignorar el gran foco de importancia que tienen las redes sociales en todos los negocios y vida cotidiana de nuestra sociedad. En los últimos años la fuerza que lleva acumulando estos gigantes ha sido imparable y cada vez a más. Por ello en la web PeriodismoULL es aconsejable que se automaticen las tareas de publicación en redes y con esto llegar al máximo número de usuarios.

El TFG propuesto va a consistir en el desarrollo de un plugin de WordPress para la automatización de publicaciones en redes sociales con distintos parámetros configurables, como mensajes formateados, cronómetro, planificador, etc. Este se ha desarrollado con PHP, distintas API de redes sociales, y apoyado por funciones y hooks propios de WordPress.

Se abre la posibilidad de extenderlo a más redes sociales en un futuro, y también añadiendo nuevas funcionalidades y/o parámetros configurables.

Palabras claves: WordPress, plugin, automatización, redes, sociales, publicación, planificador, PHP.

Abstract

Currently, the great focus of importance that social networks have in all businesses and daily life of our society cannot be ignored. In recent years, the force that these giants have accumulated has been unstoppable. For this reason, on the website PeriodismoULL it is recommended to automate the tasks of publication in networks and with this reach the maximum number of users.

The proposed TFG will consist of the development of a WordPress plugin for the automation of publications on social networks with different configurable parameters, such as formatted messages, stopwatch, planner, etc. This has been developed with PHP, different social media APIs, and supported by WordPress own functions and hooks.

It opens the possibility of extending it to more social networks in the future, and also adding new functionalities and / or configurable parameters.

Keywords: WordPress, plugin, networks, social, publication, planner, PHP.

Índice general

Introducción	11
Descripción y objetivos	11
Justificación del proyecto	12
Antecedentes	13
Herramientas y Tecnología	14
WordPress	14
XAMPP	14
PHP	15
Codebird	16
Facebook SDK	16
CURL	17
Desarrollo del proyecto	18
Funcionamiento de un plugin para WordPress	18
Hooks en WordPress	19
Creación de menús y submenús	20
Creación de Settings	21
Clase de Twitter	24
Clase de Facebook	25
Clase de Instagram	27
Clase de Telegram	28
Funcionamiento del cronómetro	29
Funcionamiento del planificador	31
Guía de Uso	33
Instalación y activación del plugin	33
Página principal del plugin	34
Configuración de Twitter	34
Configuración de Facebook	36

Configuración de Instagram	37
Configuración de Telegram	39
Configuración del cronómetro	40
Configuración del planificador	41
Conclusiones y líneas futuras	43
Summary and Conclusions	45
Presupuesto	47
Tabla de presupuesto	47
Obtención de ingresos con el plugin	48
Apéndice del repositorio	49
BIBLIOGRAFÍA	50

Índice de figuras

2.2 Panel de Control de XAMPP	15
3.1 Cabecera de plugin	18
3.3 Creación de menú padre	20
3.4.1 Creación Settings Twitter	22
3.4.2 Formulario settings Twitter	23
3.5.1 Método init de Twitter	24
3.5.2 Método post de Twitter	24
3.6.1 Método init de Facebook	25
3.6.2 Método post de Facebook	26
3.7.1. Método init de la clase Instagram	27
3.7.2. Método post de la clase Instagram, contenedor	27
3.7.3. Método post de la clase Instagram, publicar	28
3.8.1 BotFather	28
3.8.2 Método post de Telegram	29
3.9.1 Método schedule del cronómetro	30
3.10.1. Función my_cron_schedules del planificador	31
3.10.2. Comprobación del planificador	31
3.10.3. Función recop del planificador	32
4.1.1 Casilla del plugin	33
4.1.2 Menús y submenús.	33
4.2 Página principal	34
4.3.1 Credenciales de Twitter	34
4.3.2 Parámetros configuración Twitter	35
4.3.3 Prueba final de Twitter	36
4.4.1. Credenciales de Facebook	36
4.4.2. Parámetros configurables de Facebook	37
4.4.3. Prueba final de Facebook	37
4.5.1 Credenciales de Instagram	38
4.5.2 Parámetros configurables de Instagram	38
4.5.3 Prueba final de Instagram	39
4.6.1 Credencial del bot de Telegram	39
4.6.2 Parámetros configurables de Telegram	40
4.6.3 Prueba final de Telegram	40
4.7.1 Parámetros del cronómetro	41
4.7.2 Prueba de cron del cronómetro	41
4.8.1 Parámetros configurables del planificador	42
4.8.2 Pruebas cron del planificador	42

Índice de tablas

Cuadro 8.1. Tabla de presupuesto 47

Capítulo 1 Introducción

1.1 Descripción y objetivos

El proyecto de TFG consiste en el desarrollo de un plugin de WordPress[1] dedicado a la automatización en redes sociales para la web PeriodismoULL [15]. Después de un estudio exhaustivo de redes sociales referidas a los usuarios de PeriodismoULL, se llegó a la conclusión de que toman importancia, las siguientes:

- **Twitter:** es una de las redes sociales que tiene un crecimiento más lento, su fortaleza es el tiempo real y permitir la escucha social a través de ella. Twitter es una herramienta muy importante para los portales de noticias.
- **Facebook:** es la red social con más usuarios activos lejos de la segunda, ambiente familiar que admite la presencia de empresas, y organizaciones en creación de páginas para esto.
- **Instagram:** una de las redes sociales con más crecimiento en los últimos años, mayormente entre los jóvenes. Comprada por Facebook en 2012, centrada principalmente en el formato imagen y donde cada vez el marketing es más importante dentro de ella debido a su crecimiento.
- **Telegram:** es una red social y aplicación de mensajería instantánea con crecimiento estable y aumento de reputación por su gran seguridad. Además de esto permite la creación de canales de información y uso de bots muy fácilmente que ayudan a estas y más tareas.

El objetivo del plugin es facilitar y automatizar directamente estos métodos publicitarios de oferta de PeriodismoULL en distintas redes sociales, para ello se apoya de diferentes parámetros configurables a la hora de realizar esta tarea como son:

- **Autopublicación:** casilla seleccionable que permite con cada entrada en la web, de manera inmediata, se cree una publicación en la red social seleccionada.
- **Mensaje:** nos permite personalizar el tipo de mensaje que se publicará en tu red social, con algunos comandos para usar partes de la respectiva entrada en tu página web como link, títulos, etc.

- **Imagen:** permite incluir la imagen destacada del post en tu publicación es otra casilla marcable para representar gráficamente la entrada publicada.
- **Cronómetro:** permite atrasar una publicación una cierta cantidad de horas y minutos seleccionados.
- **Planificador:** permite planificar la red social seleccionada con una franja de horas seleccionadas, para ir subiendo todas las publicaciones cada 5 minutos y evitar que el algoritmo de dicha red social oculte publicaciones si se suben todas de golpe. Esta es una ampliación del proyecto que no estaba en el diseño principal de objetivos del plugin pero que fue añadida posteriormente como factor distintivo.

1.2 Justificación del proyecto

El TFG en cuestión consiste en el desarrollo de un plugin de WordPress[1] para la web PeriodismoULL [15]. El proyecto surge por la necesidad de alcanzar la mayor cantidad de audiencia posible en dicho sitio, y para ello en el siglo XXI es esencial el uso de redes sociales. Para ello este plugin facilita el proceso y automáticamente realiza las distintas publicaciones, así se consigue evitar problemas de despiste, retraso en una publicación, o publicar una noticia fuera de las horas recomendadas evitando alcanzar un mayor público y tener mayor visibilidad.

Por esa cuestión y para facilitar todo este proceso, se propuso realizar un plugin que facilitara las tareas del sitio web y se pudiera programar un mensaje bien formateado adaptado a la red social. Además de la implementación de un planificador de publicaciones, en el cual se podrá indicar las horas permitidas para publicar en cada red social consiguiendo la visibilidad buscada. Esto mejorará la acogida del público, y los números del sitio web.

1.3 Antecedentes

El tema de las redes sociales cada día marca más importancia, y es algo que no puede pasar desapercibido, ya que todos los negocios buscan su hueco en estas plataformas. Por ello la mayoría de negocios que quieren repercusión o algo de crecimiento hacen uso de esta magnífica herramienta de comunicación. Muchos negocios hacen uso de diferentes plugins para automatizar toda su participación en estas redes sociales, es por esto que PeriodismoULL [15] no es una excepción.

En la actualidad existen diversos plugins que se dedican a la automatización de redes sociales pero ninguno de estos tiene un uso open source sin acceso premium a todas estas funciones con las distintas redes sociales. Además de incluir la novedad de la programación de publicaciones, que en ninguno de estos encontramos algo semejante con estas características. Podemos incluir los dos más completos del mercado:

- NextScripts [13] es uno de los más populares en la actualidad con más de cien mil instalaciones, permite el acceso a diferentes redes sociales como Facebook, Instagram, Twitter, etc, y la automatización de publicaciones. Con la desventaja que para Instagram y muchas más de estas redes sociales se requiere tener el servicio premium del plugin.
- Social Media Auto Publish [14] es otro antecedente que con más de diez mil instalaciones lidera los plugins de automatización en redes sociales, lo que no incluye instagram, ni telegram y tampoco un planificador.

Estos dos plugins citados fueron una buena guía inicial para empezar el desarrollo del proyecto y tener un ejemplo a seguir. Pero se introdujeron novedades como las que ya hemos hablado antes de la programación de publicaciones en el tiempo y la inclusión de nuevas redes sociales fuera de un servicio premium.

Capítulo 2 Herramientas y Tecnología

2.1 WordPress

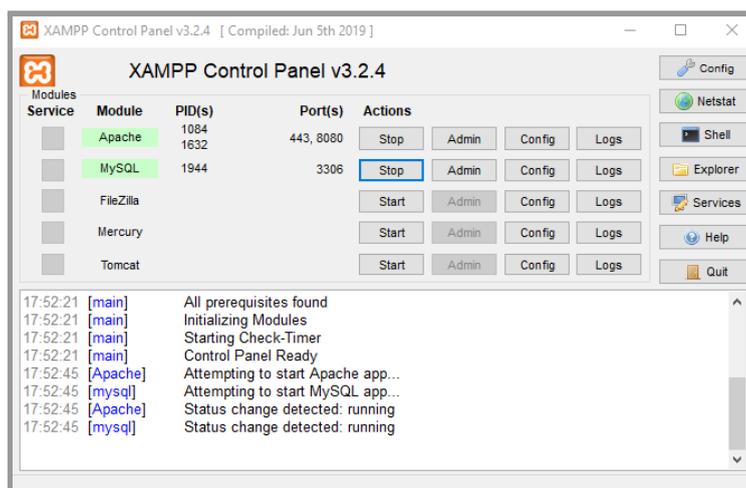
Se utilizará Wordpress [1] como framework, siendo este actualmente en el mercado el CMS (Sistema de gestión de contenido) más utilizado, y por ende, el que mejor soporte técnico ofrece, mayor rapidez de actualización, corrección de errores, además que es intuitivo y dinámico. Dispone de otras ventajas como ser un software libre lo cual nos facilita mucho la accesibilidad. Otra de las razones de su uso, es que cada vez hay más plugins en el mercado que manejan diferentes funcionalidades en el ambiente de trabajo de Wordpress, enriqueciendo la experiencia del usuario, y aumentando la escalabilidad de nuestro proyecto.

Un dato muy importante para el uso de este framework es su popularidad, y es que según estadísticas “WordPress se utiliza en más de un 40% de las páginas web a nivel mundial” [11](W3Techs. WordPress, 2021). Una de estas es PeriodismoULL, así que la creación del plugin será más que abarcada por muchas páginas. Además WordPress [1] permite el uso de muchas funciones para el desarrollo de sus plugins lo que facilita su implementación, le podemos agregar a esto mucha cantidad de hooks , clases y métodos.

2.2 XAMPP

XAMPP [8] es un servidor de plataforma libre, es un software que integra en una sola aplicación, un servidor web Apache, interprete de lenguaje PHP [3], un servidor de base de datos MySQL y un servidor FTP FileZilla, PHPMyAdmin, etc. En nuestro proyecto XAMPP nos ayudará a instalar WordPress en él y a realizar múltiples pruebas de creación de plugins. Se encuentran múltiples ventajas en el uso de este servidor:

- La facilidad y rapidez de toda la instalación.
- Configuraciones mínimas, instalar y empezar a trabajar.



2.2 Panel de Control de XAMPP.

Por estas y más razones nos sirvió para muchas de las pruebas en un servidor local. Pero ya al tener el plugin medianamente desarrollado, el cotutor Eduardo Nacimiento nos proporcionó un sitio web en su servidor en línea (<https://alu0100888423.fg.eduardonacimiento.com/>) para poder comprobar el uso de enlaces, imágenes, etc, en un entorno real, con WordPress instalado y preparado para el uso de plugins.

2.3 PHP

PHP [3], es un lenguaje de código abierto muy popular en el mundo, adecuado para el desarrollo de webs y que puede ser incrustado de manera muy flexible con HTML [4]. Por lo tanto, si vemos en internet un gran número de páginas y portales web están usando PHP de manera abierta. “Más del 70% de todos los sitios webs del mundo cuyo lenguaje de programación del lado del servidor conocemos, utilizan actualmente una versión de PHP” (W3Techs. PHP, 2021). Unas de las ventajas del uso de este lenguaje son:

- PHP es un lenguaje libre y abierto.
- Su sintaxis y arquitectura es simple y cumple estándares básicos de la programación orientada a objetos (OOP).
- No requiere complejos entornos para su desarrollo.
- La instalación es realmente sencilla gracias a servidores locales como XAMPP [8].
- El motor de WordPress es PHP por lo tanto para el desarrollo de un plugin en esta plataforma se debe usar este lenguaje

2.4 Codebird

No tan conocido como los anteriores, Codebird [5] es una herramienta de código abierto que permite conectarse a la API REST de Twitter, streaming API, Collections API, TON (Object Nest) API y Twitter Ads API a tu código en PHP [3]. Todo esto con una única biblioteca, Codebird admite Twitter OAuth completo de 3 vías, así como la autenticación de aplicaciones.

Codebird [5] nos ha dado la posibilidad de automatizar la publicación en Twitter directamente de una manera simplificada con el uso de una sola biblioteca. Y además de permitir publicar en Twitter texto simple, nos da la posibilidad de subir imágenes para posteriormente enviarlas en un tweet. Por supuesto antes de poder hacer esto debes haber creado una aplicación para desarrolladores en tu cuenta de Twitter, y acceder con las credenciales que te ofrecen, clave de la API, clave secreta de la API, token de acceso, y token secreto de acceso. Codebird te concede la posibilidad de acceder fácilmente al control de tu cuenta con esas cuatro credenciales.

2.5 Facebook SDK

Primero explicar lo que significa el acrónimo SDK, “Software Development Kit” (Kit de desarrollo de Software), en Facebook SDK [7] se reúne un grupo de herramientas que permiten la programación de aplicaciones con la API de Facebook e Instagram en consecuencia, ya que son la misma empresa. Esta biblioteca nos proporciona ciertos beneficios:

- **Graph API [7]:** la más usada, es una característica fundamental en los negocios ya que recoge estadísticas de conexiones entre las personas, páginas, eventos, mensajes, fotos, comentarios y likes. Además incluye una función de búsqueda que te permite fácilmente buscar y analizar estos datos mientras los datos del usuario sean públicos. Además de esto permite manejar todos los post de las páginas, también de un cuenta comercial de Instagram vinculada con la página en cuestión, desde su publicación hasta estadísticas. Esta es la que vamos a dar uso en nuestro plugin.
- **Chat API:** permite integrar la aplicación del chat en una aplicación basada en Web o de escritorio. Esta API permite enviar y recibir a los usuarios mensajes de texto, además de recuperar la foto de perfil de los distintos usuarios que interactúan.
- **Créditos API:** Sistema interno de pago de la plataforma para comprar artículos de aplicaciones, servicios y más. Básicamente se basa en poder integrar tu negocio con la opción de pagar con Facebook.

- **Anuncios API:** Ofrece a los desarrolladores la capacidad de crear sus propias aplicaciones para satisfacer sus necesidades y preocupaciones de la publicidad.

Para el uso de esta biblioteca, debemos acceder con las credenciales de nuestra cuenta, que nos hace entrega Facebook una vez nos creamos una aplicación para desarrolladores con nuestra cuenta de Facebook. Además de esto necesitaremos el ID de la página de Facebook la cual publicará todo lo que se envíe.

2.6 CURL

Curl [6] es una librería que permite realizar peticiones HTTP, con el objetivo de transferir información con sintaxis de URL. En el contexto de PHP [3], permite crear un script que se comporte como un navegador el cual interpreta llamadas HTTP y así realizar una petición a otro servidor remoto. Su utilidad es similar al objeto XMLHttpRequest de JavaScript, en el sentido de comunicarse con un servidor distinto para la obtención de recursos, intercambio de información, etc. En PHP [3] se implementa como un conjunto de funciones, las principales y esenciales son:

- **curl_init:** inicia una nueva sesión CURL. Esencial para cualquier comunicación.
- **curl_setopt:** Define opciones para nuestra sesión cURL.
- **curl_exec:** Ejecuta la petición HTTP.
- **curl_close:** Cierra la sesión CURL

Esta biblioteca se hará uso para conectar Telegram [9] con nuestro plugin, y mediante llamadas HTTP interactuar como el bot Administrador de nuestro canal que debemos haber creado con anterioridad y darle ese rol. Con llamadas a la api con el token de nuestro bot administrador del canal podemos enviar mensajes a este mismo, solo texto o con imágenes.

Capítulo 3 Desarrollo del proyecto

3.1 Funcionamiento de un plugin para WordPress

Antes que nada, hay que explicar qué es un plugin, y es que WordPress [1] dota de dos principales posibilidades a los desarrolladores para modificar un sitio en WordPress, un tema y un plugin. Este último se puede describir como un paquete de código que extiende la funcionalidad del core de WordPress, creado mediante PHP y en ocasiones algo de HTML y CSS.

Para empezar a desarrollar un plugin en esta plataforma lo primero es crear un nuevo directorio en la carpeta de los plugins *wp-content/plugins*. El archivo principal del plugin debe tener el mismo nombre que el directorio con la extensión php. El primer paso es colocar la cabecera del plugin con su información en este archivo, algo tal que así:

```
<?php
/*
Plugin name: Auto Poster ULL
Plugin URI: http://autoposterull.com
Description: A plugin to auto post your wordpress posts to your favourite social media.
Version: 1.0
Author: Juan Eduardo Flores
Author URI: http://miurlpersonal.com
Text Domain: autoposterull
Domain Path: /languages
License: GPL2

Miplugin es un software libre y está distribuido...
*/
```

3.1 Cabecera de plugin

En esta imagen se puede observar datos relevantes a la hora de visualizar este en el apartado de plugins en el panel de administración de WordPress, como puede ser el título, la descripción, la versión, el autor, etc, que son datos necesarios para la creación de cualquier mínimo plugin. Otro de los aspectos necesarios para esta elaboración son funciones propias de WordPress [1] que se activan cuando un plugin ha sido activado o desactivado. Estas funciones son las siguientes:

- **register_activation_hook(__FILE__, 'mpInstall')**.
- **register_deactivation_hook(__FILE__, 'mpDeactivation')**.

Estas son dos funciones muy importantes a la hora de confeccionar un plugin, para su activación y su correcta desactivación. '___FILE___' define la ruta completa y el archivo que se está ejecutando, mientras 'mpInstall' y 'mpDeactivation' son las funciones que se ejecutan en ese momento, en el cual se active o se desactive correspondientemente. Hay muchas funciones preestablecidas en WordPress [1] que nos facilitan todo este desarrollo, y serán de muy importante uso en nuestra creación.

3.2 Hooks en WordPress

Para conocer correctamente el desarrollo de plugins es importante hablar de los hooks. Hook es un término genérico que se usa como referencia, en lugares estratégicos dentro del core, para inyectar código propio y agregar comportamientos o cambiar el funcionamiento por defecto de WordPress [1]. Existen dos tipos de hook en esta plataforma:

- Action Hooks: Te permite ejecutar tus propias funciones en un lugar específico dentro del tiempo de ejecución de WordPress, por ejemplo:

- **`add_action('wp_head', 'my_function');`**

En este ejemplo especificamos que cuando se cargue el header del tema de la página de WordPress, con la función preestablecida 'wp_head' se ejecute mi función 'my_function'. Como wp_head, podemos encontrar un montón de ellas como 'init', 'publish_post', 'save_post', etc. Estos ejemplos anteriormente nombrados son los que se usan en nuestro plugin para activar nuestra función cada vez que se publique un post se active automáticamente, aquí un ejemplo:

- **`add_action('init', 'Twitter::init')`.**
- **`add_action('publish_post', 'Twitter::tweet')`.**

Este ejemplo funciona con el método init de la clase Twitter se activa después de que WordPress haya terminado de cargarse pero antes de los headers, y luego cada vez que se publique un post, se activa el método tweet de la clase Twitter. Además de estas funciones establecidas, se pueden crear personalizados con dos funciones propias.

- Filter Hooks: permiten manipular la información de texto y/o contenidos antes de que sean utilizados, como modificar el título de una entrada antes de terminar de cargarse. Por ejemplo para realizar este caso que se acaba de comentar sería de la siguiente forma

- `add_filter('wp_title', 'change_title');`

La diferencia con los action hook es que ahora se recibe un parámetro y es el que se cambia con la función `'change_title'`, que recibe `'wp_title'` como argumento y retorna lo que desea, así cambiando este título.

3.3 Creación de menús y submenús

Para el manejo de plugins se frecuenta el uso de menús y submenús, y en nuestro plugin no va a ser una excepción y es que es la mejor manera de representar distintos apartados en un mismo plugin. Para esto se crea un menú principal y dentro de este mismo, varios submenús con cada una de las redes sociales tratadas para tenerlas bien diferenciadas y que sea accesible e intuitivo al usuario. Para la elaboración de un menú personalizado a la barra lateral de administración se puede usar el hook que nos proporciona WordPress [1], `'admin_menu'`, de manera fácil y rápida podremos personalizar este menú, y agregar submenús, como se puede observar en la siguiente imagen:

```
if(!function_exists('mp_options_page')){
    add_action('admin_menu', 'mp_options_page');
    function mp_options_page() {

        $menus = [];
        $submenus =[];

        $menus[] = [
            'pageTitle' => 'Auto Poster Ull',
            'menuTitle' => 'Auto Poster Ull',
            'capability' => 'manage_options',
            'menuSlug' => 'auto_poster_ull',
            'functionName' => 'auto_poster_ull_display',
            'iconUrl' => plugin_dir_url(__FILE__) . 'lib/images/icono.svg',
            'position' => 15,
        ];

        addMenuPage( $menus );
    }
}
```

3.3 Creación de menú padre

Se puede apreciar distintos campos en la creación de un menú:

- **PageTitle**: el primer parámetro es el título de la página. La etiqueta del título de la página cuando se selecciona el menú.
- **MenuTitle**: es el título del menú, el texto que se visualizará para el título del menú.
- **Capability**: el permiso requerido para que este menú se muestre al usuario.
- **MenuSlug**: es la barra de menú, que se utiliza para crear la URL de la página, se tiene que mantener único para cada uno de los menús.
- **FunctionName**: la función de devolución de llamada de esta página. La función que se llamará para generar el contenido de la página en cuestión.
- **IconUrl**: proporciona el URL para el icono del menú en la barra lateral.
- **Position**: posición del menú en la barra lateral.

La creación de Submenús requiere todos estos parámetros, menos el icono y la posición. Y un parámetro más que será el `parentSlug`, que será la referencia del menú padre al que pertenece el submenú en cuestión, en nuestro caso sería `'auto_poster_ull'`.

3.4 Creación de Settings

Después de tener creado el menú con todos los submenús para las diferentes redes sociales. Toca encargarse de rellenar cada uno de estos submenús con las diferentes opciones y configuraciones para cada uno y así conseguir que funcione correctamente. Para esto tenemos que hacer uso de la función `'register_setting'`, que proporciona WordPress [1] para todos sus desarrolladores, además de la función `'add_settings_section()'` y `'add_settings_field()'`. Como se puede apreciar en la siguiente imagen:

```

function twitter_settings_init(){
    //Registrando una nueva sección en el submenu twitter
    register_setting('twitterSettings', 'twitter_config');

    add_settings_section(
        'twitter_config_section',
        'Configuración de twitter',
        'title_section',
        'twitterSettings'
    );

    add_settings_field(
        'twitter_config_campo1',
        'API Key',
        'config_field_api',
        'twitterSettings',
        'twitter_config_section',
        [
            'label' => 'api_key',
        ]
    );
}

```

3.4.1 Creación Settings Twitter

Primero se registra el nombre de la configuración que en este caso siendo el ejemplo de Twitter, será *'twitterSettings'* que servirá para referenciar cuando se añaden secciones, y *'twitter_config'*, es el nombre de la opción que se guardará en la base de datos y que podremos usar en todo el código, mediante la función *'get_option('twitter_config')*. Posteriormente se crea la sección de configuración con la función *'add_settings_section()'*, la cual contiene cuatro parámetros, son estos con el orden correspondiente:

- **Id** obligatorio para identificar la sección, se utiliza en el atributo id de las etiquetas.
- **Título formateado de la sección.** Se muestra como el título de la sección.
- **Devolución de la llamada.** Función que hace eco de cualquier contenido en la parte superior de la sección.
- **Nombre de la página de configuración,** anteriormente añadida cuando se registra la configuración. En este caso sería *'twitterSettings'*.

Ahora solo hay que añadir los distintos campos que tendrá la configuración para ello hacemos uso de la función *'add_settings_field()'* con los siguientes parámetros obligatorios correspondientemente:

- **Id** para identificar el campo en cuestión. Se utiliza en el atributo id de las etiquetas.
- **Título formateado del campo.** Se muestra como la etiqueta del campo durante la salida.
- **Devolución de llamada.** Función que llena el campo con las entradas de formulario deseadas. La función debería hacer un eco de su salida.
- **Identificador de la página** de configuración en la que se muestra la sección, en este ejemplo es *'twitterSettings'*.
- **Identificador de la sección** de la página de configuración en la que se muestra el cuadro.
- **Etiqueta** que se usará para guardar la configuración y poder acceder a esta mediante la función *'get_option('twitter_config')[api_key]'*.

Dentro de la devolución de llamada, encontramos una función la cual realiza un *'echo'* de un campo de formulario, en el que su resultado se guardará en nuestra base de datos y se mostrará una vez guardado. Y por último para que todo esto funcione se debe incluir en la página principal del plugin la etiqueta del formulario, mostrar los diferentes campos llamando a la funciones *'settings_fields('twitterSettings')*, y *do_settings_sections('twitterSettings')*, y para terminar el formulario agregar el botón de submit. Todo esto como se muestra a continuación:

```

if(!function_exists('twitter_auto_poster_ull')){
    function twitter_auto_poster_ull(){
        if(current_user_can('manage_options')){
            echo "<form action='options.php' method='post'>";
            settings_fields('twitterSettings');
            do_settings_sections('twitterSettings');
            submit_button('Guardar cambios');
        }
    }
}

```

3.4.2 Formulario settings Twitter

Se han de incluir todas las secciones necesarias de parámetro configurables para los ajustes de Twitter, y se hace así con las demás redes sociales. Hasta tener completado todos los submenús con los distintos campos de configuración que requiera el plugin.

3.5 Clase de Twitter

A la hora de explicar las funcionalidades del plugin debemos separarlas por red social, ya que cada una tiene su método único. Empezamos por Twitter, y con la ayuda de la biblioteca de Codebird [5] que ayuda a conectar las credenciales de una cuenta de Twitter para poder publicar libremente.

```
protected static $codebird;

public static function init() {
    $settings = get_option('twitter_config');

    $apiKey = $settings['api_key'];
    $apiSecret = $settings['api_key_secret'];
    $accessToken = $settings['access_token'];
    $accessTokenSecret = $settings['access_token_secret'];

    \Codebird\Codebird::setConsumerKey($apiKey, $apiSecret);
    self::$codebird = \Codebird\Codebird::getInstance();
    self::$codebird->setToken($accessToken, $accessTokenSecret);
}
```

3.5.1 Método init de Twitter

El método init, conecta con las credenciales de nuestra cuenta, clave de la API, clave secreta de la API, token de acceso, y token secreto de acceso. Anteriormente ya debe haber sido rellena cada una de estas credenciales en el submenú de settings de Twitter. Con la propiedad protegida '\$codebird' creamos una instancia, con la cual podremos tener acceso a nuestra cuenta y poder publicar tranquilamente en el nuevo método 'tweet(\$postID)'. De la siguiente manera:

```
if($attachImage == 'si' && $thumbID != ""){
    $reply = self::$codebird->media_upload(array(
        'media' => $imgLink
    ));
    $image_id = $reply->media_id_string;
    $params = array(
        'status' => $message,
        'media_ids' => $image_id
    );
}
else if(($attachImage == 'si' && $thumbID == "") || ($attachImage == 'no')){
    $params = array(
        'status' => $message
    );
}
if(!empty($post_link) && $post->post_status == 'publish') {
    self::$codebird->statuses_update($params);
}
```

3.5.2 Método post de Twitter

Cuando en las opciones de incluir imagen la tienes marcada como sí, se sube primero la imagen con el método `'media_upload'` a Twitter y luego ya se incluye en la variable `$params`, para poder publicar, si no tiene marcada la opción de incluir imagen o no tiene ninguna imagen la entrada, se publica sólo el texto seleccionado. Para acceder al enlace de la imagen destacada de una entrada se accede con las siguientes funciones:

- **get_post_thumbnail_id():** devuelve el identificador de la imagen destacada. Para ello se debe pasar por argumento el identificador del post.
- **wp_get_attachment_image_url():** devuelve el enlace de la imagen destacada. Se debe pasar por argumento el identificador de la imagen, y el tamaño de la imagen en nuestro caso mantenemos el `'original'`.

Se guardará el enlace a la imagen destacada del post en cuestión en tamaño original, si no hay ninguna imagen en el post en cambio la función devolverá una cadena vacía. Estos dos métodos de la clase que se han visto son los principales para realizar la acción de publicar en Twitter con imagen o sin ella.

3.6 Clase de Facebook

Después continuamos con Facebook y su respectiva clase, como Twitter tendrá un método `init` público y estático. Este método recoge las credenciales guardadas en los settings de Facebook, y se conecta a la cuenta de Facebook correspondiente con el identificador de la app y la identificación secreta de la app.

```
protected static $fb;

public static function init() {
    $settings = get_option('facebook_config');

    $appId = $settings['app_id'];
    $appSecret = $settings['app_secret'];

    self::$fb = new Facebook\Facebook(array(
        'app_id' => $appId,
        'app_secret' => $appSecret,
    ));
}
```

3.6.1 Método init de Facebook

Una vez realizada la llamada a una nueva instancia de *Facebook\Facebook* mediante la biblioteca de FacebookSDK [7], ya tendremos la variable protegida y estática *\$fb*, con la cual se podrá acceder a la cuenta asociada a estas credenciales. Posteriormente toca encargarse de enviar una publicación al mensaje y para eso creamos un nuevo método llamado *'post_fc(\$postID)'* que se encargará de enviar un link asociado a una publicación.

```
if($link == false){
    $linkData = [
        'message' => $message,
        'link' => $post_link,
    ];
}else{
    $linkData = [
        'message' => $message,
        'link' => $link,
    ];
}

$page_id = get_option('facebook_config')['id_page'];
$pageAccessToken = get_option('facebook_config')['access_token'];
if(!empty($post_link) && $post->post_status == 'publish') {
    $response = self::$fb->post('/'.$page_id.'/feed/', $linkData, $pageAccessToken);
    $graphNode = $response->getGraphNode();
}
}
```

3.6.2 Método post de Facebook

En la última imagen se puede apreciar que primero se valida si se ha enviado algún enlace en específico para asociar la publicación o directamente asocia el enlace de la entrada creada que se puede obtener mediante la función *get_permalink()*, con el *\$postID* como argumento. Se crea un array con el mensaje creado para la publicación y el enlace en cuestión, listo para enviar. Recogidas las credenciales guardadas del identificador de la página de Facebook y el token de acceso a esa misma, podemos realizar el post con la instancia de la cuenta de Facebook conectada. Para hacer esta llamada hacemos uso de la graph API [7] que proporciona Facebook a sus desarrolladores.

3.7 Clase de Instagram

Con Instagram no se requiere de otra biblioteca adicional, ya que usa la misma que Facebook, además funciona con una cuenta comercial de Instagram asociada a una página de Facebook. Para conectarse a la cuenta se necesita el identificador de la página de Facebook, el token de acceso a esta página, y con estos datos podemos adquirir el identificador de la cuenta de Instagram mediante una solicitud HTTP. Para realizar esta solicitud se hace uso de la función de WordPress, `wp_safe_remote_get()` que retorna la respuesta de una solicitud HTTP segura mediante el método GET.

```
protected static $instagram_id;
protected static $api_url;

public static function init() {
    $page_id = get_option('instagram_config')['page_id'];
    $pageAccessToken = get_option('instagram_config')['access_token'];
    self::$api_url = 'https://graph.facebook.com/v11.0/';
    $response = wp_safe_remote_get( self::$api_url . $page_id .
    '?fields=instagram_business_account&access_token=' . $pageAccessToken );

    if ( is_array( $response ) && ! is_wp_error( $response ) ) {
        $body = $response['body'];
        error_log('INSTAGRAM BODY: ' . $body);

        $json_resp = json_decode($response['body']);
        self::$instagram_id = $json_resp->instagram_business_account->id;
        error_log('INSTAGRAM ID: ' . self::$instagram_id);
    }
}
```

3.7.1. Método init de la clase Instagram

Una vez haber obtenido el identificador de la cuenta de Instagram, se dispone a realizar la publicación mediante otro método llamado `post_ig($postID)`. En este mismo se retorna la respuesta sin procesar de una solicitud HTTP segura mediante el método POST con `wp_safe_remote_post()`, cuyos argumentos son las credenciales.

```
if(!empty($post_link) && $post->post_status == 'publish') {
    $response = wp_safe_remote_post( self::$api_url . self::$instagram_id . '/media?image_url=' .
    $image_url . '&caption=' . $message . '&access_token=' . $pageAccessToken , array(
        'method' => 'POST',
        'timeout' => 45,
        'redirection' => 5,
        'httpversion' => '1.0',
        'blocking' => true,
        'headers' => array(),
        'cookies' => array()
    )
);
```

3.7.2. Método post de la clase Instagram, contenedor.

El retorno de esa llamada post es un JSON el cual se puede descifrar y sacar el identificador del contenedor que hace falta para realizar la publicación en instagram. Primero se realiza un `'json_decode($response['body'])'`, que decodifica un JSON en una variable PHP y luego con `'$container = $json_resp->id'`, sacamos el identificador del container para realizar la siguiente petición:

```
$response = wp_safe_remote_post( self::$api_url . self::$instagram_id . '/media_publish?creation_id=' .
$container . '&access_token=' . $pageAccessToken , array(
    'method' => 'POST',
    'timeout' => 45,
    'redirection' => 5,
    'httpversion' => '1.0',
    'blocking' => true,
    'headers' => array(),
    'cookies' => array()
)
```

3.7.3. Método post de la clase Instagram, publicar.

Por último se realiza la petición de publicación con el container que contiene los datos del mensaje y la foto de la publicación en cuestión. Para que se publique en la cuenta indicada, con la siguiente función: `'wp_safe_remote_post()'`.

3.8 Clase de Telegram

Telegram es un caso particular y es que no hace falta una biblioteca en concreto, ni unas credenciales de una cuenta, solo necesitamos crear un bot. Esto se puede realizar mediante el asistente suministrado por Telegram [9] BotFather buscándolo simplemente en la aplicación.



3.8.1 BotFather

Una vez dentro de su chat se debe enviar el comando `/newbot` y posteriormente pedirá el nombre del bot y luego el nombre de usuario del bot. Una vez se hayan hecho estos pasos, se mostrará el token el cual podremos conectar con nuestro código para realizar las distintas funciones. Además de esto se requiere de un canal, agregar al bot en cuestión al canal y atribuirle el rol de administrador para poder publicar en este.

Primero se realiza el caso en el cual solo se envía un mensaje sin incluir la imagen destacada en el mensaje, para esta se hará uso de cURL [6]. Como se ha hablado con anterioridad cURL permite realizar peticiones HTTP, como un navegador el cual interpreta llamadas HTTP y así consigue realizar una petición a otro servidor remoto.

```
if ( ($post->post_status == 'publish') && ($attachImage == 'no')) {
    $urlMsg = "https://api.telegram.org/bot{$token}/sendMessage";
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $urlMsg);
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_POSTFIELDS, "chat_id={$id}&parse_mode=HTML&text=$message");
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    $server_output = curl_exec($ch);
    curl_close($ch);
}
```

3.8.2 Método post de Telegram

Se puede observar en la imagen como primero se define la URL de la petición HTTP con el token del bot, para enviar un mensaje por la API de Telegram Web. Luego se pide una petición POST, y se colocan los campos del id del canal en cuestión que se puede encontrar en la URL en Telegram Web, o el mismo nombre de canal, también se coloca el mensaje que se publicará. Y por último la otra condición se incluye la imagen destacada en la publicación, en la cual solo se cambian los campos que se pasan en la petición POST en la cual se incluye el id del canal que se mantiene, la foto en cuestión y el texto que acompañará a la foto.

3.9 Funcionamiento del cronómetro

A la hora de tratar el cron de WordPress se tiene que explicar qué es esta herramienta unix/Linux que nos ofrece WordPress. Es básicamente un archivo de programación de tareas, que se encarga de ejecutar las tareas programadas de tu sitio WordPress en segundo plano, siempre está revisando nuevas actualizaciones, temas y plugins. Este cron funciona con el tiempo de Unix, segundos que han pasado desde el 1 de Enero de 1970, lo que se tendrá que aplicar a sus distintas funciones este estándar.

La primera implementación de esta herramienta en el plugin fue el uso de un cronómetro que cuando pasara el tiempo estimado publique dicha tarea. Así poder programar una publicación para dentro de dos horas, o las horas y minutos que desee. Para esto se creó un nuevo método en todas las clases llamado `'schedule($postID)'`, como el siguiente ejemplo en la clase de Twitter.

```
public static function schedule($postID) {
    $minutes = get_option('twitter_config')['auto_time'][3];
    $minutes1 = get_option('twitter_config')['auto_time'][4];
    $hours = get_option('twitter_config')['auto_time'][0];
    $hours1 = get_option('twitter_config')['auto_time'][1];

    if( !wp_next_scheduled( 'share_the_post', $postID) ){
        wp_clear_scheduled_hook( 'share_the_post' );
        wp_schedule_single_event( time() + $minutes1*60+ $minutes*600+$hours*3600+$hours1*3600
        , 'share_the_post', array($postID));
    }
}
```

3.9.1 Método schedule del cronómetro.

En el nuevo método se recogen las horas y minutos para la cual se programó la publicación, con estos datos recogidos en la configuración de dicha red social y con la ayuda de la función `time()` de PHP [3] que devuelve la fecha Unix actual, calculamos en fecha Unix cuando se activará el hook `'share the post'`. La función `'wp_schedule_single_event()'`, programa un hook (`'share_the_post'`) que será activado en el tiempo que se pase en el primer parámetro, en formato Unix siempre para que esto funcione. El tercer parámetro son los argumentos que se pasan al hook en este caso el identificador de la entrada. Por último para activar este nuevo método creado debe usarse distintos hooks de acción como:

- **add_action('init', 'Twitter::init').**
- **add_action('publish_post', 'Twitter::schedule').**
- **add_action('share_the_post', 'Twitter::tweet', 10, 1).**

En el último hook de acción, tiene dos últimos parámetros que pertenecen a la prioridad y a los argumentos aceptados correspondientemente, son opcionales.

3.10 Funcionamiento del planificador

Se quiso ampliar la funcionalidad del plugin para crear una huella de distinción entre todo el mercado de plugins de automatización. Por ello se propuso un planificador que ayudará a evitar los problemas de retraso del cron de WordPress debido al posible poco flujo de usuarios en el sitio web y para adaptarse a las mejores franjas horarias de publicación según red social y público existente. Esta nueva funcionalidad hará uso de una cola de planificación que una vez se llegue a la franja de hora seleccionada se vayan enviando todas las publicaciones a sus distintas redes sociales. Además esto se producirá en un tiempo espaciado de 5 minutos cada publicación, para que el algoritmo de dicha red social no perjudique subir todas las publicaciones a la vez.

Lo primero que se tuvo que integrar es un cron corriendo cada 5 minutos y para ello se tiene que crear una función con un cron modificado que se ejecute cada 5 minutos.

```
function my_cron_schedules($schedules){
    $schedules["5min"] = array(
        'interval' => 5*60,
        'display' => __('Once every 5 minutes'));
    return $schedules;
}

add_filter('cron_schedules','my_cron_schedules');

$array = [];
add_option('recop_tw', $array);

if( !wp_next_scheduled( 'programm_tw' ) ){
    wp_schedule_event(time(), '5min', 'programm_tw');
}
```

3.10.1. Función my_cron_schedules del planificador

Con `'add_filter('cron_schedules', 'my_cron_schedules')` se añade la nueva opción de cada 5 minutos al cron por defecto de WordPress [1]. Luego con `'wp_schedule_event()`, crea un cron cada 5 minutos desde ese momento, y una nueva opción con `'add_option'` que se usará como cola del programador . Con la herramienta *"cron manager"* [10], se puede visualizar los distintos cron y cuánto tiempo queda para activarse. Así se pudo comprobar cómo se generaban cada 5 minutos.

<input type="checkbox"/>	programm_tw	Once every 5 minutes	En 4 mins agosto 30, 2021 4:40 pm
--------------------------	-------------	----------------------	---

3.10.2. Comprobación del planificador

A continuación se creó otro método en las distintas clases de las diferentes redes sociales para recopilar todas las publicaciones pendientes del programador. Para ello se elaboró el siguiente método:

```
public static function recop($postID) {
    $options_program = get_option('recop_tw');
    if(end($options_program) !== $postID){
        $elemento = array_push($options_program, $postID);
        update_option('recop_tw', $options_program);
    }
}
```

3.10.3. Función recop del planificador

Cada vez que se tenga activado el planificador, de todas las entradas que se publiquen se guardará su identificador en la cola llamada '*\$options_program*'. Esta cola se guarda en la base de datos, y se actualiza con la función '*update_option*', unido como primer parámetro al nombre de la opción y la variable que se actualizaba en esa opción. Pero todavía falta un método muy importante para poder terminar este planificador llamado *programm(\$postID)*, el cual se activará cada 5 minutos. Cuando haya algo en la cola de recopilación y se esté en la franja de tiempo marcada en la configuración de esa red social se realizará la publicación en cuestión, eliminando el primero de la cola. Y así cada 5 minutos hasta que no queda ninguna entrada en la cola, o ya no se está en la franja de horas seleccionada.

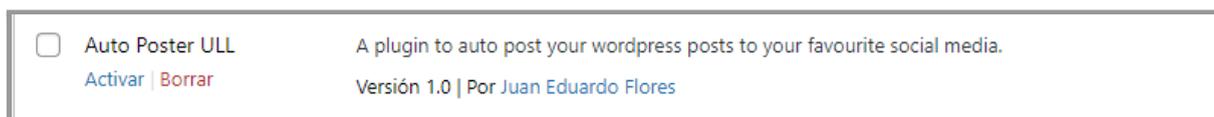
Con el método '*array_shift()*' quitamos el primer elemento de un array, y luego de nuevo con el '*update_option()*' volvemos a actualizar el array sin ese elemento. Esto debe estar dentro de la condición, y ya estaría completado la función de programador.

Capítulo 4 Guía de Uso

En este capítulo se puede encontrar una guía de uso del plugin para intentar resolver la mayor parte de problemas o dudas que le surjan al usuario a la hora de usarla.

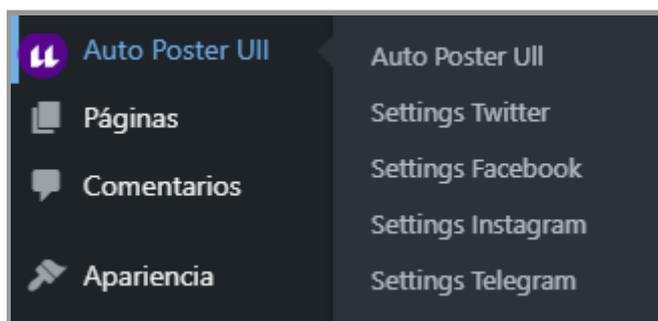
4.1 Instalación y activación del plugin

Para instalar el plugin es tan fácil como entrar en el panel de control de WordPress en la sección de plugins, añadir nuevo y luego subir plugin, seleccionar la carpeta del plugin en cuestión y luego ya en la sección de plugins se muestra así:



4.1.1 Casilla del plugin

Y se pulsa el botón de activar, lo que genera en la barra lateral la sección del plugin con el icono, y los distintos submenús de las redes sociales.

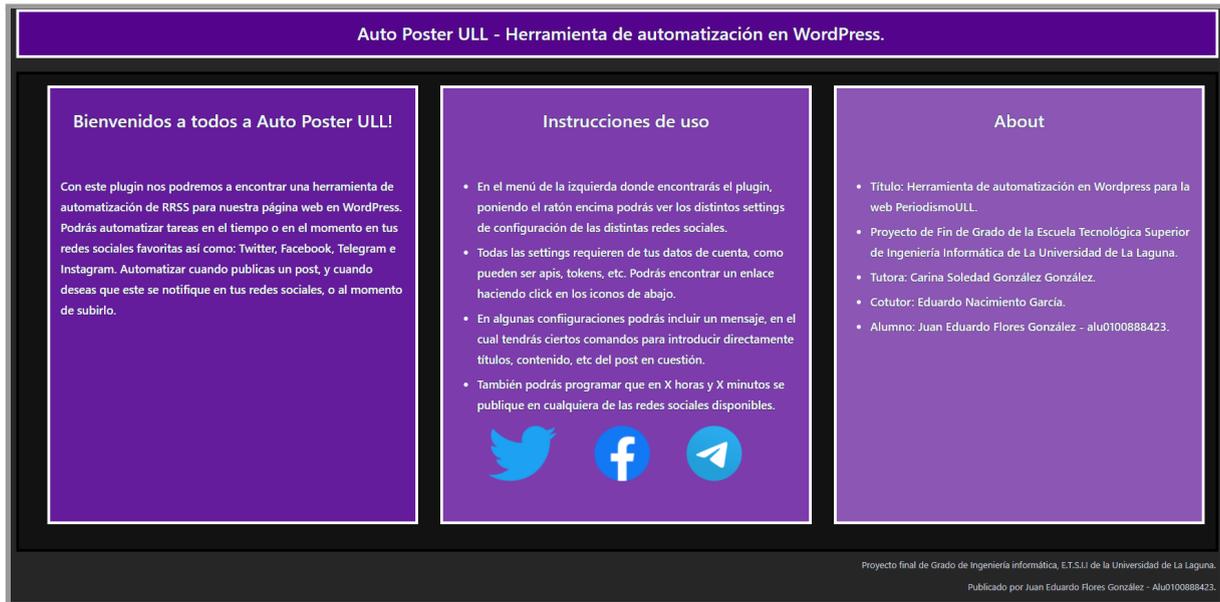


4.1.2 Menús y submenús.

Ya se puede empezar a usar el plugin y rellenar las diferentes configuraciones de las distintas redes sociales que se deseen usar para automatizar todo este proceso, y facilitar sus publicaciones.

4.2 Página principal del plugin

En la página principal se encuentra una bienvenida con una introducción al plugin con un breve resumen, luego unas rápidas instrucciones de uso con tres imágenes con los logos de Twitter, Facebook y Telegram con un enlace en cada uno para acceder a crear las distintas aplicaciones y así obtener las credenciales en el caso de Twitter y Facebook. Y luego Telegram con un enlace de ayuda para crear el bot administrador necesario.



4.2 Página principal

4.3 Configuración de Twitter

Dentro de la pestaña de 'Settings Twitter' se encuentra un formulario en la que guardaremos las distintas credenciales de nuestra cuenta de Twitter, los cuáles tendremos que generar si no tenemos una app para desarrolladores, en el siguiente enlace: <https://developer.twitter.com/en/apps>.

Configuración de twitter

API Key

API Secret Key

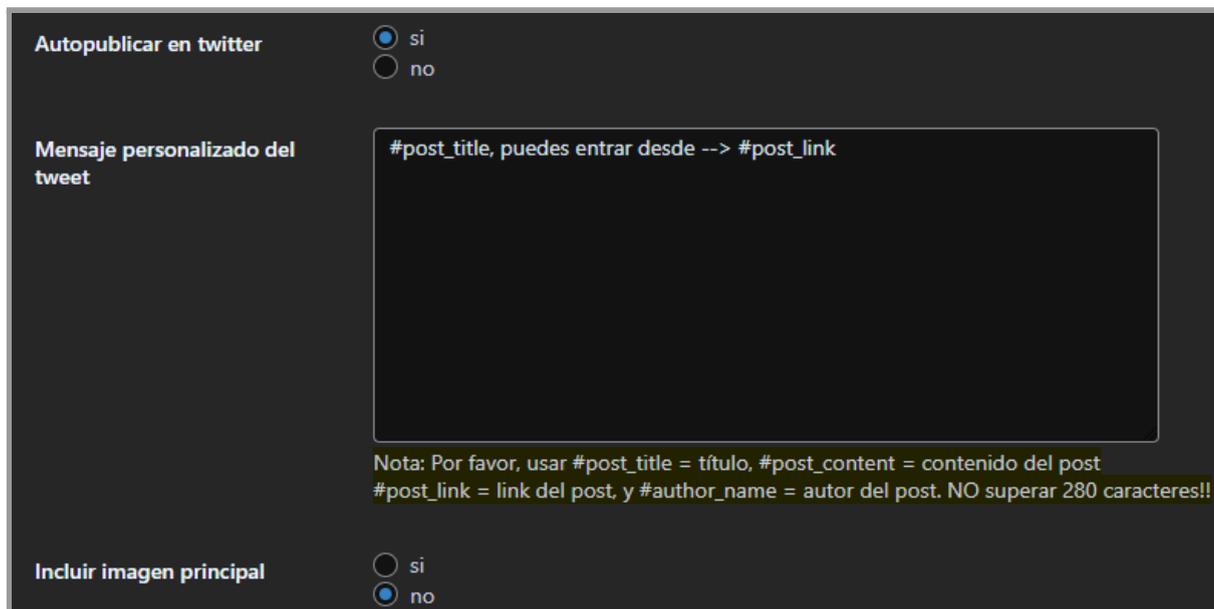
Access Token

Access Secret Token

Nota: No olvide acceder al [enlace](#) para crear una nueva app de Twitter, con sus respectivas API Keys, y Access Tokens.

4.3.1 Credenciales de Twitter

En la nota también se puede encontrar el enlace para generar una nueva app de Twitter, si el usuario en cuestión no tiene unas credenciales para que funcione la API y el plugin correctamente. Luego seleccionamos distintos parámetros configurables para la configuración de Twitter y la publicación de entradas, como son los siguientes:



Autopublicar en twitter si
 no

Mensaje personalizado del tweet

#post_title, puedes entrar desde --> #post_link

Nota: Por favor, usar #post_title = título, #post_content = contenido del post
#post_link = link del post, y #author_name = autor del post. NO superar 280 caracteres!!

Incluir imagen principal si
 no

4.3.2 Parámetros configuración Twitter

La primera opción tiene que estar activada para funcionar y publicar en Twitter de cualquiera de las maneras. Luego se encuentra un campo de texto en el cuál se debe introducir el mensaje de la publicación, con posibilidad de agregar partes de la entrada en el mensaje, como son:

- **#post_title:** El título de la entrada publicada en el sitio web.
- **#post_content:** El contenido de la entrada publicada en el sitio web. Hay que tener cuidado porque el contenido suele superar el límite de 280 caracteres de un tweet y habría problemas con la publicación en la red social.
- **#post_link:** El enlace de la entrada publicada en el sitio web. Para que los usuarios puedan acceder directamente.
- **#author_name:** El nombre del autor que publicó la entrada en el sitio web con su correspondiente usuario en WordPress.

Y por último se encuentra la opción de incluir imagen principal, que sólo se producirá cuando esté activada y en la entrada creada haya adjunta una imagen destacada. A continuación se introduce un ejemplo del resultado de un tweet automatizado con el plugin:



4.3.3 Prueba final de Twitter

4.4 Configuración de Facebook

Dentro de la pestaña de Settings Facebook se accede a la configuración de Facebook, para ello también se tendrá que tener las credenciales de una aplicación para desarrolladores, si no se tiene de antes, se puede acceder en este enlace: <https://developers.facebook.com/>, y ahí generar una nueva aplicación con sus credenciales.

4.4.1. Credenciales de Facebook

En la nota se puede acceder directamente al enlace, y después de haber introducido el identificador de la aplicación, la clave secreta y el token de acceso se tiene que introducir el id de la página de Facebook en la cual se publicará todo. Para acceder a este ID se debe acceder a la página de Facebook en cuestión y en el apartado de información, buscar Más Información ahí sale bien indicado el ID.

Autopublicar en facebook si no

Link Enlazado

Nota: Si dejas vacío el campo anterior del link enlazado, se te pondrá automáticamente el link del post en cuestión.

Mensaje personalizado de la publicación

Nota: Por favor, usar #post_title = título, #post_content = contenido del post
#post_link = link del post, y #author_name = autor del post.

4.4.2. Parámetros configurables de Facebook

Por último se encuentra la casilla de autopublicación y un campo del link, el cual si se deja vacío se colocará automáticamente el link del post en la publicación. Se te incluirá directamente la imagen del post, por eso no hay un campo de imagen, ya que esto se hace automático en Facebook, el campo mensaje se realizará como el de Twitter. A continuación se introduce una prueba del funcionamiento del plugin con una publicación en Facebook:



4.4.3. Prueba final de Facebook

4.5 Configuración de Instagram

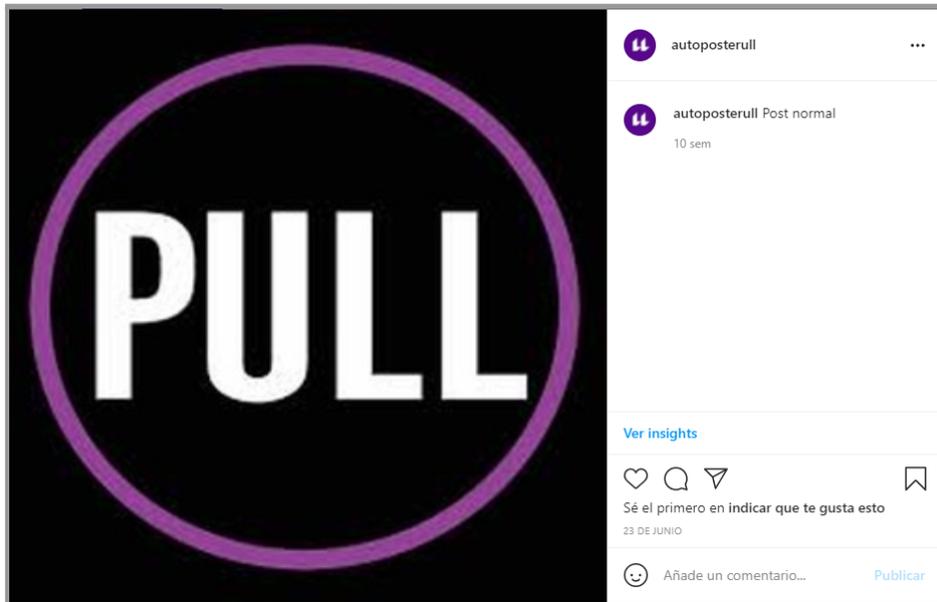
Accedemos a la configuración de Instagram, la cual está muy asociada a la cuenta de Facebook, ya que es necesaria esa vinculación con una página de esa plataforma. Además de la necesidad de una cuenta comercial de Instagram.

4.5.1 Credenciales de Instagram

Las dos credenciales necesarias para realizar esta configuración son el token de acceso creado en la app de Facebook anterior y el identificador de la página de Facebook que se comentó en la sección anterior. Ninguna de las credenciales es nueva, se pueden reciclar de la configuración de Facebook, pero hay que tener en cuenta que se necesita el permiso para poder publicar en Facebook en la graph API [7]. Para ello hay que certificar los papeles de un negocio en rigor y demás datos acerca de tu negocio.

4.5.2 Parámetros configurables de Instagram

En los parámetros configurables encontramos la casilla de publicación automática que nos permite realizar la acción. También encontramos el campo de incluir la imagen con una nota que nos advierte que si no seleccionamos esta casilla o no incluimos imagen en el post, por defecto se subirá el logo de PeriodismoULL [15] . Y por último se encuentra el campo de texto del mensaje personalizado con los mismos comandos que ya hemos visto anteriormente. A continuación se introduce una prueba del funcionamiento del plugin con una publicación en Instagram:



4.5.3 Prueba final de Instagram

4.6 Configuración de Telegram

La última de las configuraciones es la referida a Telegram [9], para ello se debe introducir las distintas credenciales esta vez relacionadas con el bot y nuestro canal de comunicación en esta aplicación. Para ello introducimos una guía proporcionado por Telegram de la elaboración simple de un bot que requerimos para esta tarea y se podrá acceder, en el siguiente enlace <https://core.telegram.org/bots#3-how-do-i-create-a-bot>. Tan fácil como acceder al BotFather buscando en Telegram y con el comando /newBot realizar la creación que nos proporcionará el token de este bot, luego solo se debe dar el rol de administrador a este bot en el canal y ya estaría preparado.

Configuración de telegram

Token del Bot de Telegram

Nota: Tienes que generar un bot con BotFather y el comando /newbot luego elegir el nombre y el username,y nos ofrecerá el token de este.

Id del Canal/Usuario

4.6.1 Credencial del bot de Telegram

En la nota te explica resumidamente lo que se debe hacer para adquirir el token del Bot de Telegram y como explicamos antes. Adicionalmente también nos solicitará el ID del canal, o Usuario con restricciones para evitar spam por parte de bots, los usuarios deben escribir antes al bot en concreto para que esto se pueda realizar con usuarios individuales, esto no ocurre con canales.

Autopublicar en telegram si
 no

Mensaje personalizado a enviar

Nota: Por favor, usar #post_title = título, #post_content = contenido del post
#post_link = link del post, y #author_name = autor del post.

Incluir imagen principal si
 no

4.6.2 Parámetros configurables de Telegram

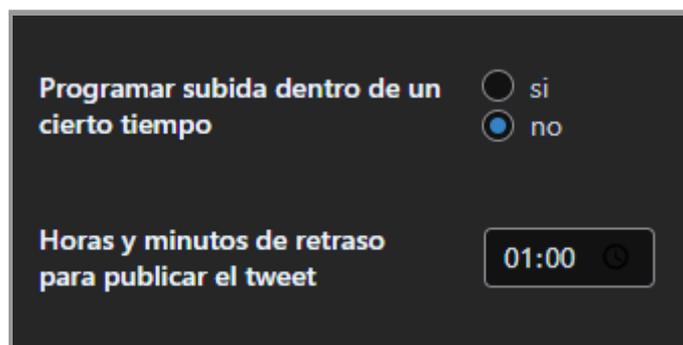
No encontramos características diferenciales en comparación con el resto de configuraciones, un campo de autopublicar, mensaje personalizado e inclusión de imagen. A continuación se introduce una prueba del funcionamiento del plugin con un mensaje en Telegram:



4.6.3 Prueba final de Telegram

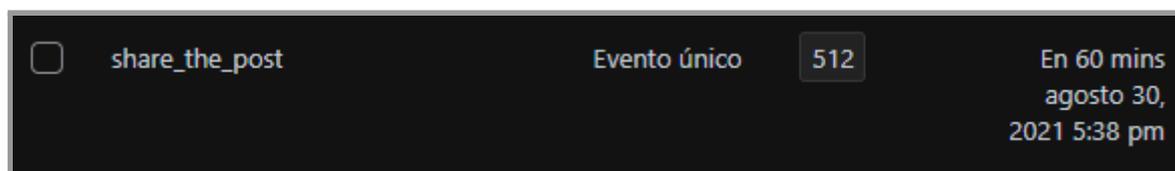
4.7 Configuración del cronómetro

Toca comentar cómo configurar el cronómetro que está disponible en todas las redes sociales, y va a permitir programar una publicación en una hora y minutos determinados.



4.7.1 Parámetros del cronómetro

La configuración es tan simple como un campo de confirmar que deseas programar la publicación, y un campo en el cual puedes introducir una hora desde el 0 hasta el 23 y unos minutos del 0 al 59. Una vez configurado una hora de cronómetro y publicado una entrada con esta configuración, en el manager de cron [10] podremos observar lo siguiente:



4.7.2 Prueba de cron del cronómetro

Un evento único que activará el método de publicación en tu red social elegida después del transcurso de esos 60 minutos. Depende del flujo de usuarios de tu sitio web en WordPress la exactitud de este cronómetro ya que el cron de WordPress no es tan riguroso. Para páginas con menos flujo de usuarios es más aconsejable el uso del planificador.

4.8 Configuración del planificador

La última de las características de nuestro plugin es el planificador de publicaciones que permite usar las estadísticas de nuestro sitio web como la hora de mayor audiencia en diferentes redes sociales a nuestro favor. Además una ventaja considerable para aquellos sitios con poco flujo de usuarios que pierde rigurosidad el uso del cron. Para hacer uso de este funcionamiento es tan sencillo como:

Programar a cierta franja de horas
 si
 no

Desde esta hora se puede publicar

Hasta esta hora se puede publicar

4.8.1 Parámetros configurables del planificador

Muy parecido al cronómetro, pero contiene dos rangos de hora en los cuáles se puede publicar, con un intervalo de 5 minutos cada publicación. Para no acumular muchas publicaciones al mismo tiempo. Observando el manager del cron [10], se puede observar como el cron de las distintas redes sociales están funcionando cada 5 minutos para saber si hay alguna publicación dentro de la cola, además de comprobando si está en la franja de tiempo.

<input type="checkbox"/>	Evento	Programar	Argumentos	Siguiente ejecución ▲
<input type="checkbox"/>	programm_tw	Once every 5 minutes		En 4 mins agosto 30, 2021 4:40 pm
<input type="checkbox"/>	programm_fc	Once every 5 minutes		En 4 mins agosto 30, 2021 4:40 pm
<input type="checkbox"/>	programm_ig	Once every 5 minutes		En 4 mins agosto 30, 2021 4:40 pm
<input type="checkbox"/>	programm_tg	Once every 5 minutes		En 4 mins agosto 30, 2021 4:40 pm

4.8.2 Pruebas cron del planificador

Capítulo 5 Conclusiones y líneas futuras

En este proyecto se creó un plugin de automatización de redes sociales para la web PeriodismoULL [15] y sus diferentes objetivos han sido cumplidos. Se ha conseguido todo esto de una manera sencilla e intuitiva para el usuario. Así este podrá configurar su sitio web de manera que siempre que cree una entrada se automatice la publicación en sus diferentes redes sociales. Además de esto puede retrasar la publicación cierto tiempo si así le interesa, configurar el mensaje de la publicación o incluir alguna imagen.

A pesar de existir otros plugins que permitan realizar operaciones semejantes a las que se realizan en este plugin, se pudo crear una vertiente de distinción entre todo este mercado. Abarcando además de la automatización, el uso de un planificador para maximizar la popularidad y visibilidad en nuestras redes sociales. Planificador que funciona de manera que solo se suban estas publicaciones en una franja de horas, configurada a la elección. Estas publicaciones programadas no se subirán a la vez, sino espaciadas en el tiempo cada 5 minutos, para que así el algoritmo de dicha red social no oculte alguna de estas por subirse todas al mismo tiempo.

A lo largo del proyecto se dieron ciertas problemáticas, un factor importante de esto fue la falta de experiencia en la creación de plugins en WordPress que supuso algún que otro retraso en las entregas. Además podemos hablar de las planificaciones no ajustadas del todo a la realidad, ya que es muy complicado estimar algo con lo que nunca has trabajado. Por eso es que se dieron estimaciones erróneas de tiempo o imprevistos. Y por último el problema que supuso más retraso fue que dejara de funcionar el token de Instagram, y es que se había revocado el permiso a publicar en esta red social. Todo esto debido a que solicitaba veracidad de un negocio real y en regla, ya que la cuenta de instagram debe ser comercial para acceder a este permiso. Se perdió el permiso a publicar en instagram con nuestro token, pero por suerte ya se había comprobado su funcionamiento a la perfección, lo que con un token con ese permiso no habría ningún problema en realizar las publicaciones.

Las líneas futuras de la aplicación están marcadas por unos frentes claros, para agregar nuevas mejoras al plugin:

- **Ampliar audiencia con nuevas redes sociales.** El siguiente objetivo es LinkedIn, red social cada vez con mayor importancia y usuarios.
- **Agregar nuevos parámetros configurables.** Cambios sugeridos y demandados por los usuarios mediante su feedback.
- **Nuevo método cron.** Se puede valorar el uso de otro método más preciso para realizar los distintos planificadores.

Capítulo 6 Summary and Conclusions

In this project, a social media automation plugin was created for the PeriodismoULL [15] website, its different objectives have been completed. All this has been achieved in a simple and intuitive way for the user. This way, they can configure their website so that whenever they create an entry, the publication on their different social networks is automated. In addition to this, you can delay the publication for a certain time if you are interested, configure the message of the publication or include an image.

Although there are other plugins that allow operations similar to those developed in this plugin, it was possible to create an aspect of distinction between this entire market. In addition to automation, the use of a planner to maximize popularity and visibility on our social networks. Scheduler that works in such a way that these publications are only uploaded in a slot of hours, configured to the choice. These scheduled publications will not be uploaded at the same time, but spaced in time every 5 minutes, so that the algorithm of said social network does not hide any of these by uploading all of them at the same time.

Throughout the project there were certain problems, an important factor in this was the lack of experience in creating plugins in WordPress that led to the occasional delay in deliveries. We can also talk about the plans not fully adjusted to reality, since it is very difficult to estimate something with which you have never worked. That is why erroneous estimates of time or unforeseen events were given. And finally, the problem that caused more delays was that the Instagram token stopped working, and that is that the permission to publish on this social network had been revoked. All this because it requested the veracity of a real business and in order, since the instagram account must be commercial to access this permission. The permission to publish on instagram with our token was lost, but luckily its operation had already been verified perfectly, so with a token with that permission there would be no problem in making the publications.

The future lines of the application are marked by clear fronts, to add new improvements to the plugin:

- **Expand the audience with new social networks.** The next objective is LinkedIn, a social network with increasing importance and users.
- **Add new configurable parameters.** Changes suggested and demanded by users through their feedback.
- **New cron method.** The use of another more precise method to develop the different planners can be valued.

Capítulo 7 Presupuesto

7.1 Tabla de presupuesto

A continuación se indica una baremación del coste que supone todo este trabajo, de las tareas con su duración, todo esto sin contar los inconvenientes y problemas encontrados en mitad de su desarrollo. Para llevarlo a cabo se han tenido en cuenta las tarifas de un programador junior. A continuación se presenta una tabla con el presupuesto:

Tipo de Actividad	Cantidad	Coste/unidad	Coste Total
Estudio Preliminar	30 horas	10€/h	300€
Búsqueda e Investigación de plugins semejantes	24 horas	12€/h	288€
Análisis de Redes sociales	22 horas	12€/h	264€
Primeros prototipos y pruebas con plugins	40 horas	15€/h	600€
Investigación de bibliotecas y uso de las diferentes API's	60 horas	12€/h	720€
Desarrollo de clases y métodos de automatización por red social	120 horas	16€/h	1920€
Desarrollo de cronómetro y planificador de publicaciones	80 horas	16€/h	1280€
TOTAL	376 horas		5372€

Cuadro 8.1. Tabla de presupuesto

7.2 Obtención de ingresos con el plugin

Después de tener toda la herramienta terminada hay varias posibilidades de obtener algún tipo de ingreso por el uso de la misma.

- La primera posibilidad es una opción premium que contenga más opciones configurables además de más redes sociales. También se podría valorar la opción de multicuenta para poder subir publicaciones a más de una cuenta ya que algunas empresas usan más de una misma red social.
- La segunda opción sería poder vender estos servicios aunque el plugin sea software libre, pero esta opción es menos viable ya que hay algunas opciones gratuitas que contienen menos ventajas y características pero aún así el valor nulo de precio llama mucho al usuario.

Capítulo 8 Apéndice del repositorio

8.1 [Repositorio del plugin en Github](#)

<https://github.com/alu0100888423/AutoPosterUII> [2]

BIBLIOGRAFÍA

- [1] WordPress. <https://developer.wordpress.org/reference/>
- [2] Github. <https://github.com/>
- [3] PHP. <https://www.php.net/docs.php>
- [4] HTML. <https://developer.mozilla.org/es/docs/Web/HTML>
- [5] Codebird. <https://github.com/jublo/codebird-php>
- [6] CURL. <https://www.php.net/manual/es/ref.curl.php>
- [7] Facebook SDK. <https://developers.facebook.com/docs/graph-api/overview>
- [8] XAMPP. <https://www.apachefriends.org/es/index.html>
- [9] Telegram. <https://core.telegram.org/>
- [10] Manager del cron. <https://es.wordpress.org/plugins/advanced-cron-manager/>
- [11] W3Techs. WordPress, 2021. <https://w3techs.com/technologies/details/cm-wordpress>
- [12] W3Techs. PHP, 2021. <https://w3techs.com/technologies/details/pl-php>
- [13] NextScripts. <https://www.nextscripts.com/social-networks-auto-poster-for-wordpress>
- [14] Social Media Auto Publish. <https://es.wordpress.org/plugins/social-media-auto-publish>
- [15] PeriodismoULL. <http://periodismo.ull.es/>