

PROYECTO DE PÁGINA WORDPRESS CON UTILIZACIÓN DE UNA API DE FÚTBOL.

MIRIAM TIRADO Y MOUAD TAIEBI

FEBRERO 2019

2 PÁGINA WORDPRESS CON UTILIZACIÓN DE UNA API DE FÚTBOL

2.1 ACERCA DE

Hemos querido crear una página cuya finalidad sea la de informar a todos los amantes del fútbol español y extranjero de los resultados y noticias relacionadas con este mundo.

2.2 WORDPRESS Y API

WordPress es un sistema de gestión de contenidos o CMS (Content Management System) enfocado a la creación de cualquier tipo de página web.

La **interfaz de programación de aplicaciones, (API)**, es un conjunto de subrutinas, funciones y procedimientos en la programación orientada a objetos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción

2.3 REQUISITOS

- Descarga de [Wordpress](#)
- Alojar Wordpress en un sitio (local en nuestro caso)
 - [Instalar Wordpress](#)
- Tener acceso a una API
 - Nos registramos en la página [FootballData](#) y obtenemos el token necesario para que funcione nuestra página.
- Archivos de la API
 - Nos descargamos el zip que encontramos en el siguiente [enlace GitHub](#)
- XAMPP for Linux

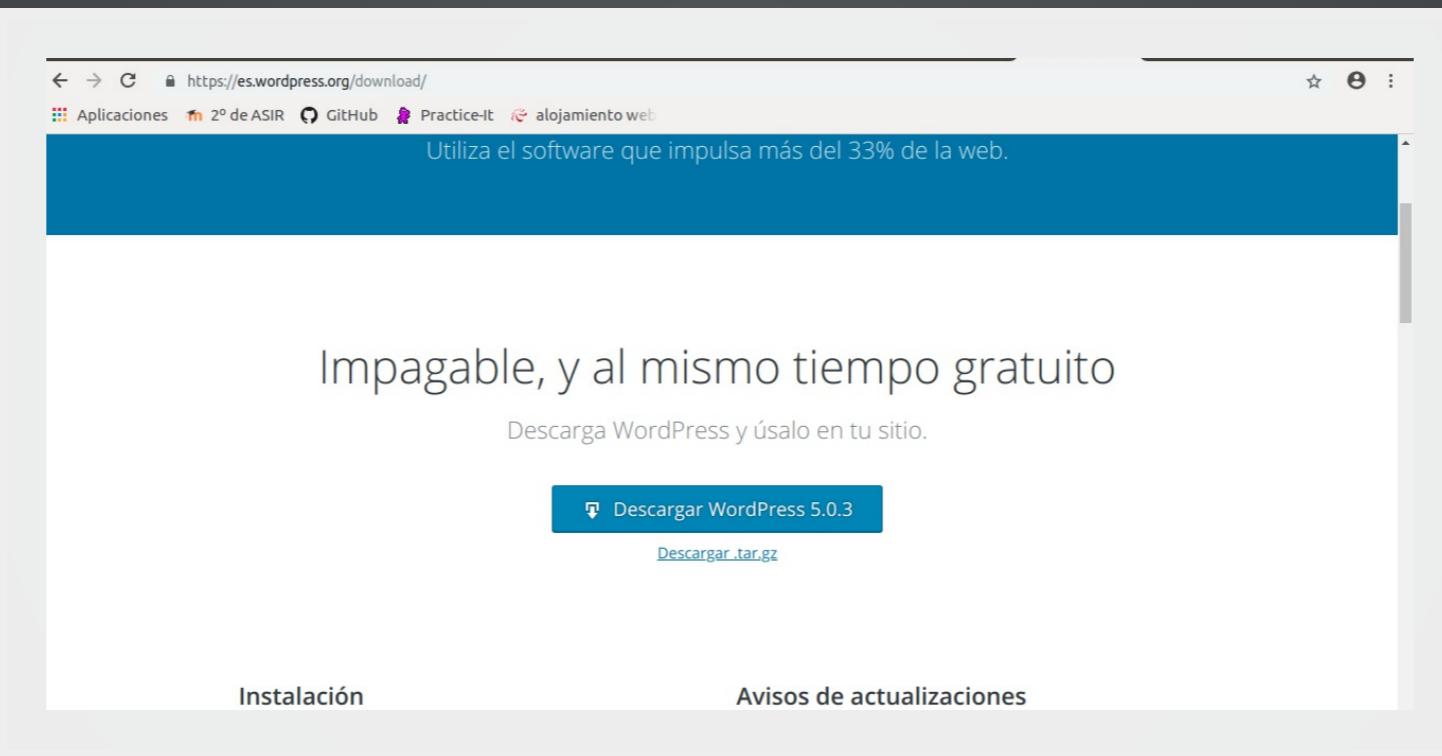
2.4 REQUISITOS (II)

- Creación de un proyecto (para ir tanteando el terreno, hacer pruebas y consultas)
- Utilización del token recibido en el archivo config.ini

3 INSTALACIÓN

3.1 INSTALACIÓN Y CONFIGURACIÓN DE WORDPRESS

Una vez descargada la ultima versión del enlace anterior hay que descomprimirlo:



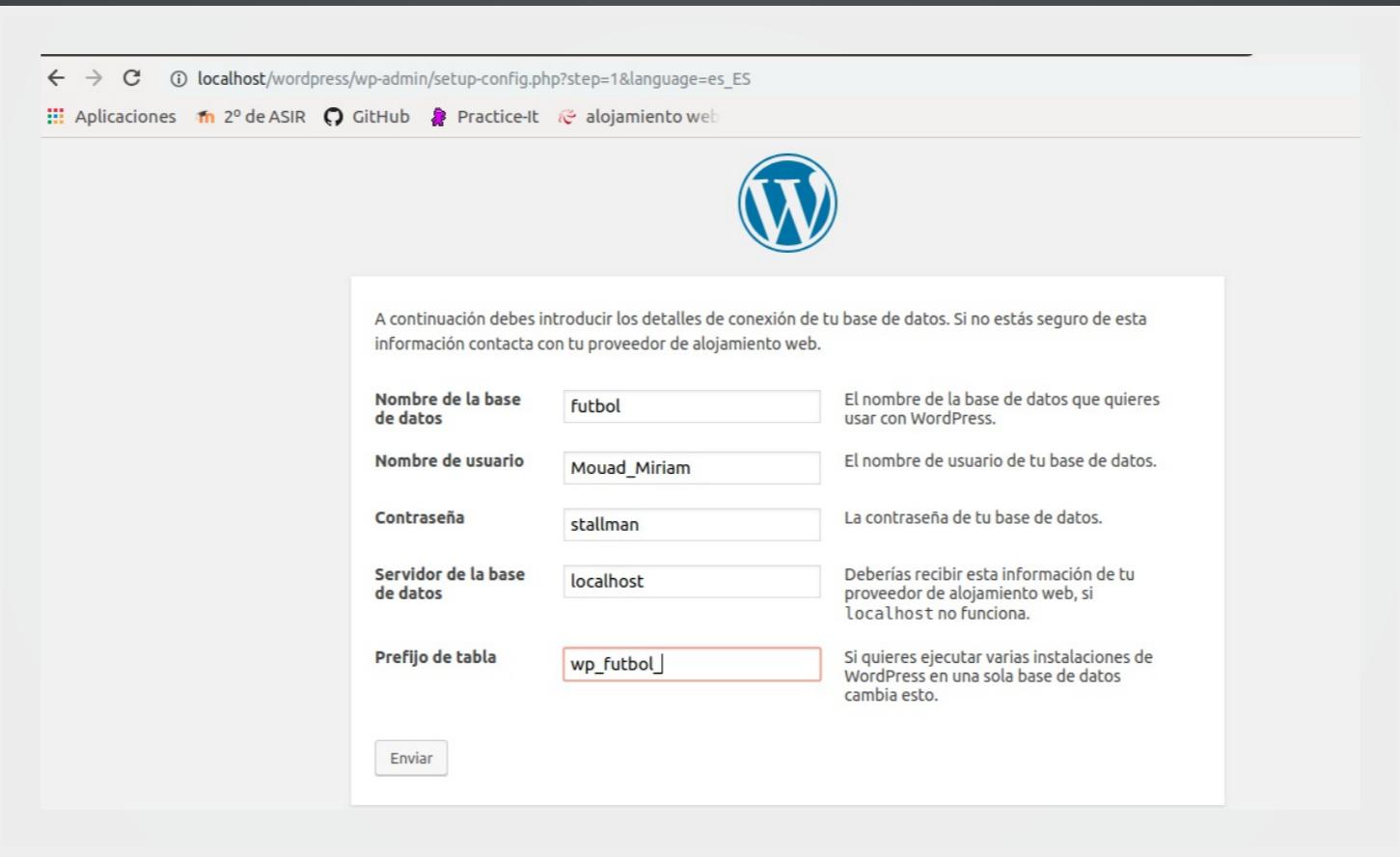
3.2 INSTALACIÓN Y CONFIGURACIÓN DE WORDPRESS

(III)

- sudo mv wordpress-5.0.3-es_ES.zip /opt/lampp/htdocs
- sudo unzip wordpress-5.0.3-es_ES.zip Ésto genera una carpeta dentro de /opt/lampp/htdocs llamada wordpress

Una vez hecho esto entramos al siguiente enlace ->
<http://localhost/wordpress/wp-admin/setup-config.php> para realizar la configuración inicial de nuestra página de Wordpress.

3.3 INSTALACIÓN Y CONFIGURACIÓN DE WORDPRESS (III)



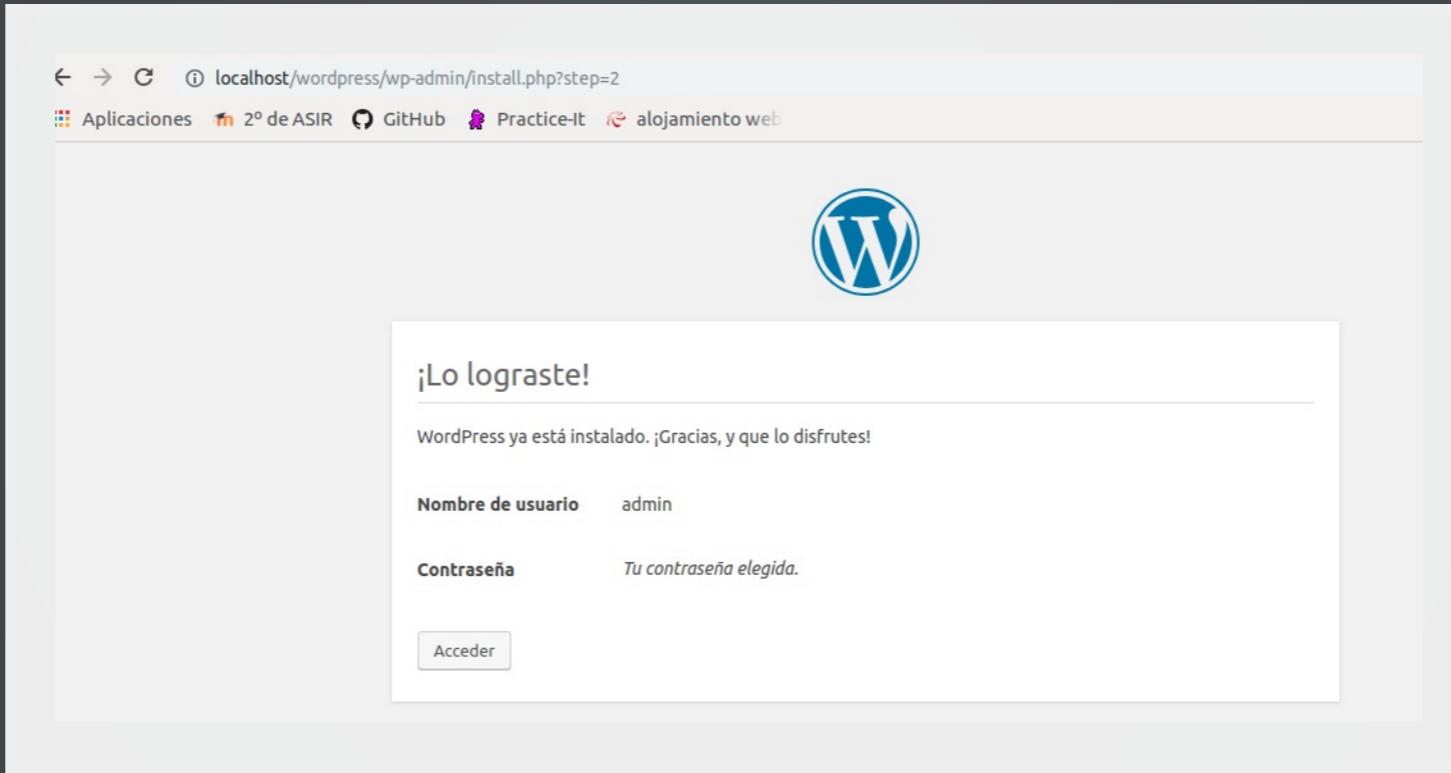
The screenshot shows the 'Setup Configuration' step 1 of the WordPress installation process. The URL in the browser is `localhost/wordpress/wp-admin/setup-config.php?step=1&language=es_ES`. The page features a large blue 'W' logo at the top. Below it, there's a message: 'A continuación debes introducir los detalles de conexión de tu base de datos. Si no estás seguro de esta información contacta con tu proveedor de alojamiento web.' The form contains five input fields with their respective descriptions:

Nombre de la base de datos	futbol	El nombre de la base de datos que quieres usar con WordPress.
Nombre de usuario	Mouad_Miriam	El nombre de usuario de tu base de datos.
Contraseña	stallman	La contraseña de tu base de datos.
Servidor de la base de datos	localhost	Deberías recibir esta información de tu proveedor de alojamiento web, si localhost no funciona.
Prefijo de tabla	wp_futbol_	Si quieres ejecutar varias instalaciones de WordPress en una sola base de datos cambia esto.

At the bottom of the form is a single button labeled 'Enviar'.

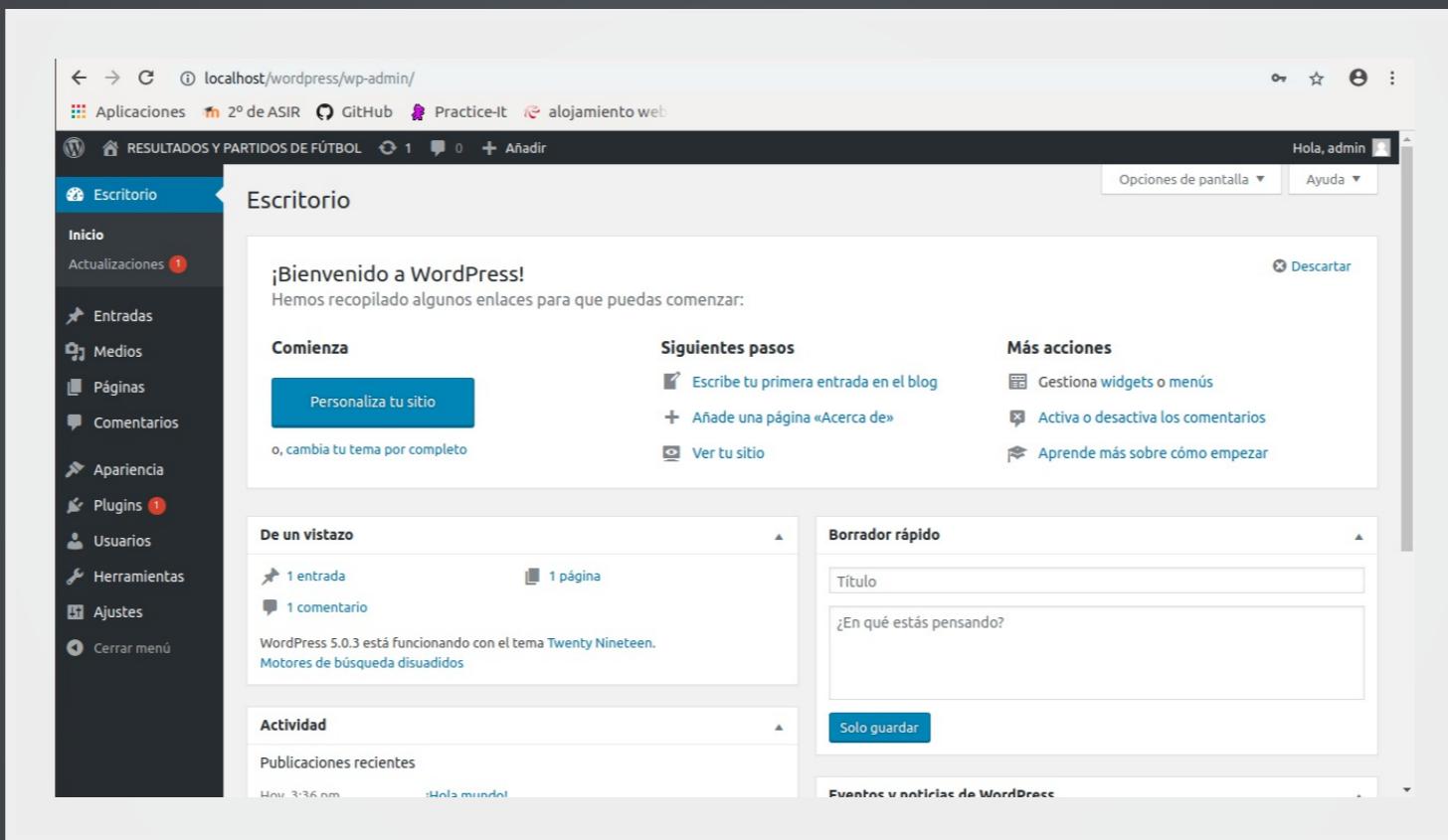
3.4 INSTALACIÓN Y CONFIGURACIÓN DE WORDPRESS

(IV)



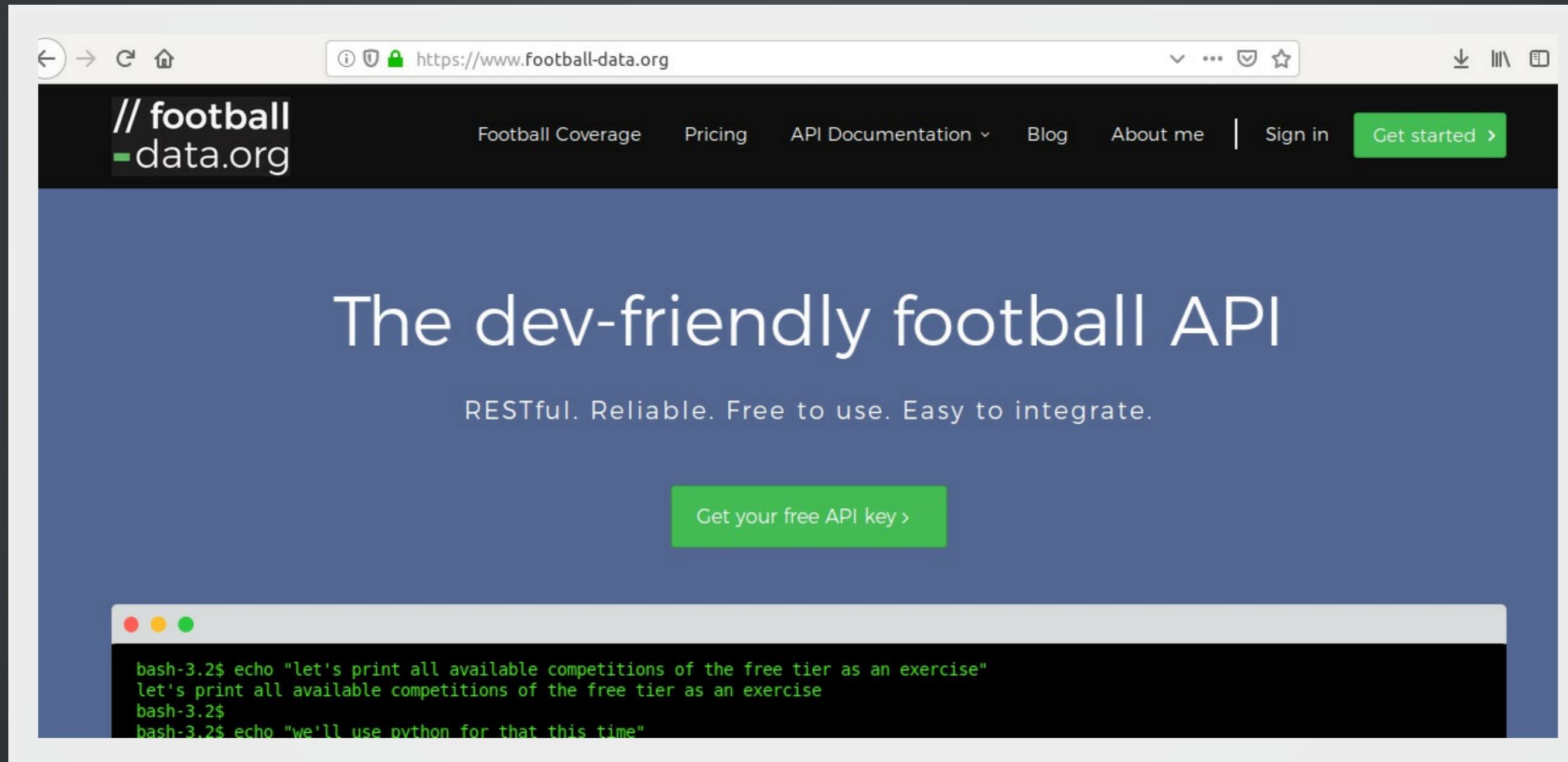
3.5 INSTALACIÓN Y CONFIGURACIÓN DE WORDPRESS

(V)



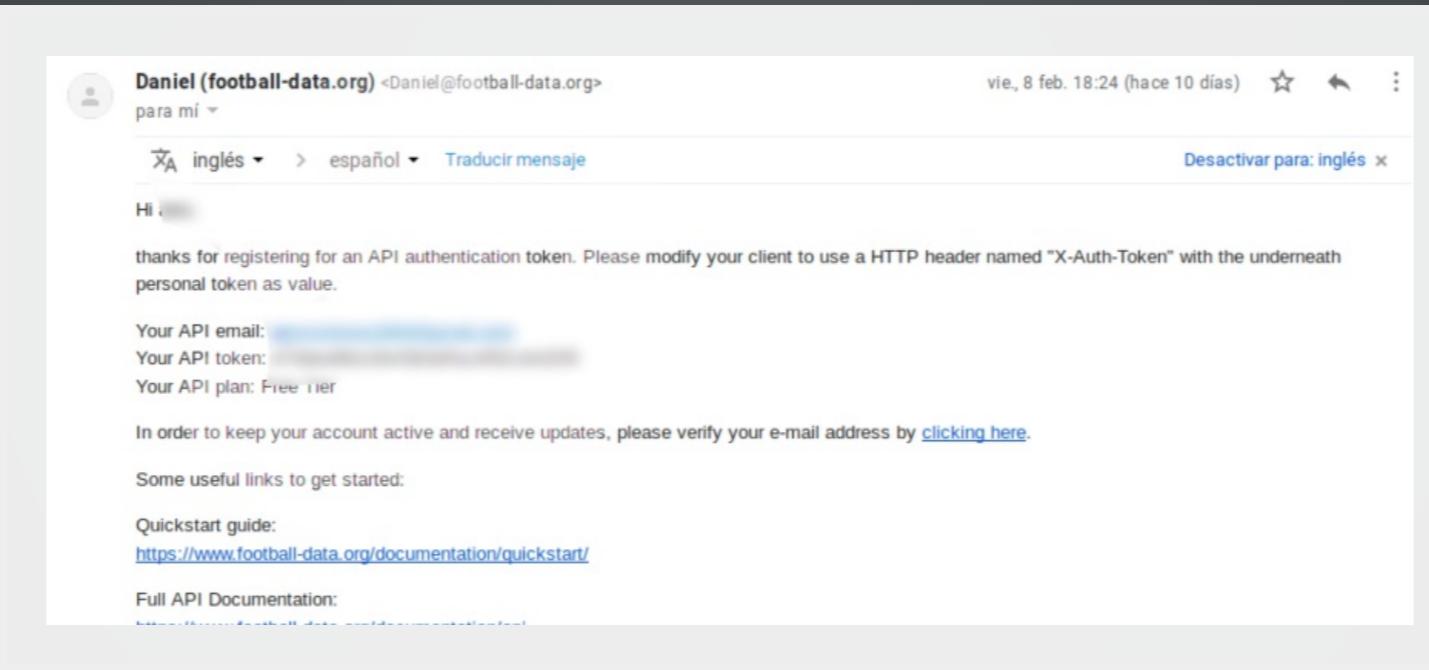
3.6 CONFIGURACIÓN DE LA API

Entramos en la siguiente [página](#)



3.7 CONFIGURACIÓN DE LA API (II)

Una vez nos encontramos registrados en dicha página recibiremos un correo a la cuenta introducida, en el que se nos asigna el token.

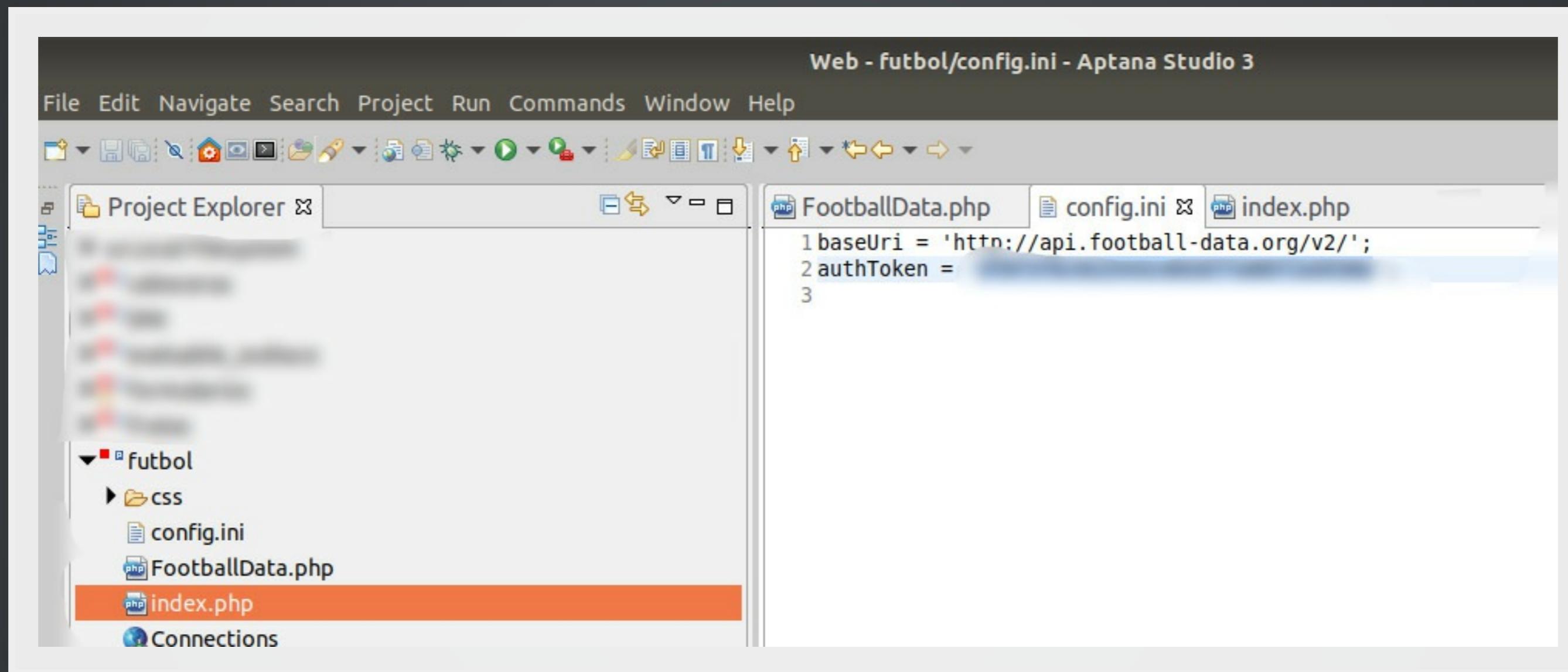


3.8 CONFIGURACIÓN DE LA API (III)

A continuación entramos en Aptana para ver y probar los .php que nos hemos descargado con el token.

Como se puede ver en la foto, dentro del archivo **config.ini** introducimos nuestro token para poder comenzar.

3.9 CONFIGURACIÓN DE LA API (IV)



3.10 ARCHIVO INDEX.PHP

Dentro de este archivo hay una variable llamada `$api` con un parámetro llamado `findMatchesByCompetitionAndMatchDay`; en este parámetro podemos encontrar dos códigos ID separados por comas.

Nosotros hemos separado dichos códigos ID el primero sigue metido dentro de esa variable y parámetro anteriores; pero, el segundo código lo hemos sacado de ese parametro y metido en una variable para poder utilizarlo con una lista desplegable en nuestra página de WordPress.

3.11 ARCHIVO INDEX.PHP (II)

Para encontrar que numero ID debiamos de poner en esos códigos hemos tenido que buscar informacion en la [página siguiente](#).

El primer código equivale a la liga que quieras elegir, en nuestro caso **La Liga Santander**; en el segundo prámetro encontramos que era para la elección de la jornada en la que querias visualizar los resultados. por este motivo es por el que hemos separado este código ID del otro (para que nuestros usuarios puedan ver los resultados de la jornada que deseen).

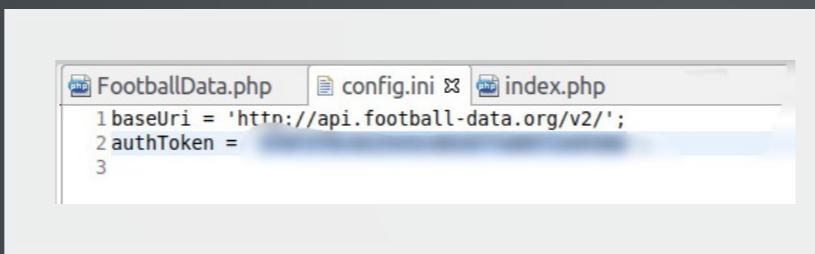
3.12 MODIFICACIÓN DE ARCHIVOS .PHP E .INI

Teníamos tres acrhivos, dos .php y uno .ini

Para que fuera más fácil la introducción de los archivos en la página hemos eliminado el archivo .ini de la siguiente forma:

3.13 MODIFICACIÓN DE ARCHIVOS .PHP E .INI (II)

- En un principio lo teníamos así:



A screenshot of a file explorer window. The window title is 'FootballData.php'. Inside, there are three files: 'FootballData.php', 'config.ini', and 'index.php'. The 'config.ini' file is currently selected and its contents are visible in the main pane:

```
1 baseUri = 'https://api.football-data.org/v2/';
2 authToken =
3
```

3.14 MODIFICACIÓN DE ARCHIVOS .PHP E .INI (III)

- Lo cambiamos a la siguiente manera:

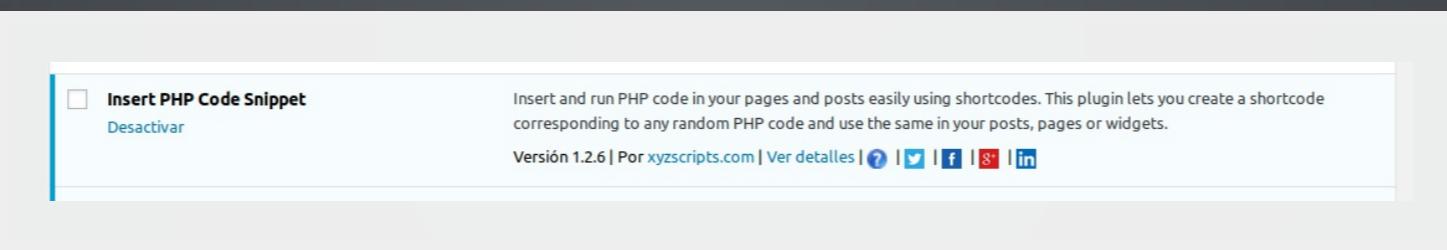
```
class FootballData {  
    public $config;  
    public $baseUri;  
    public $reqPrefs = array();  
  
    public function __construct()  
    {  
        // $this->config = parse_ini_file('config.ini', true);  
        $this->config = array ("baseUrl" => "http://api.football-data.org/v2/",  
                             "authToken" => "");  
        // print_r($this->config);  
    }  
}
```

3.15 INTRODUCCIÓN DE LA API A NUESTRA PÁGINA DE WORDPRESS

Necesitábamos encontrar un plugin que hiciera que nuestro código PHP fuera leído por WordPress.

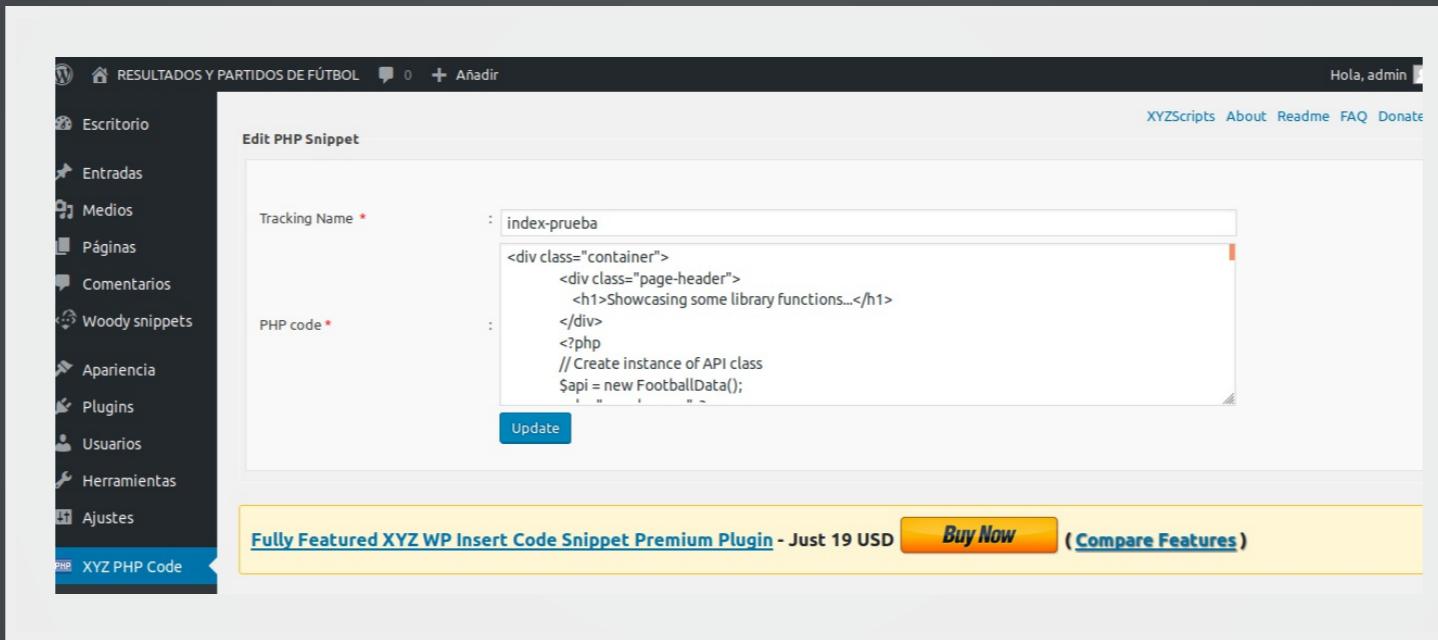
3.16 XYZ PHP CODE

Entramos en **plugins** y buscamos XYZ PHP CODE, lo descargamos, instalamos y activamos.



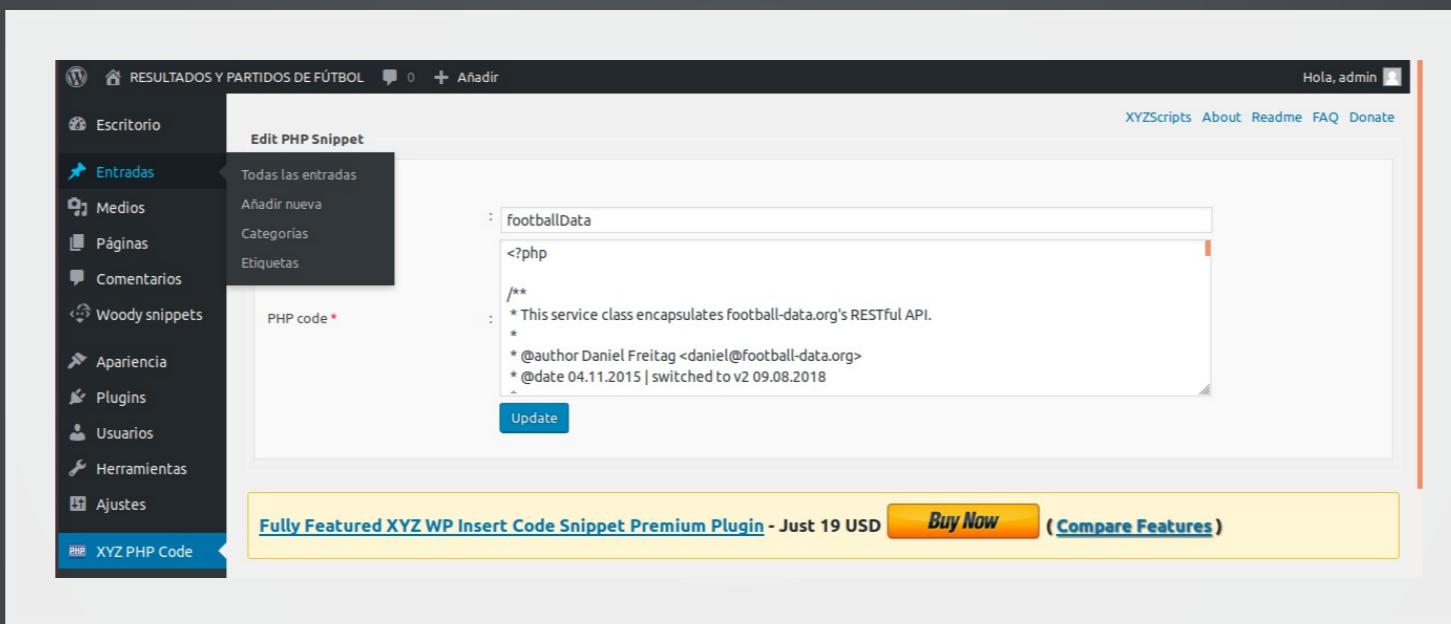
3.17 XYZ PHP CODE (II)

Una vez lo tenemos activado nos vamos a add New PHP Code Snippet e insertamos el código php de index.php.



3.18 XYZ PHP CODE (III)

A continuación insertamos también el código php de football-data.php.



3.19 XYZ PHP CODE (IV)

Ya tenemos nuestros .php insertados para ver como funciona en WordPress.

Vamos a Entradas para añadir una nueva entrada y nos damos cuenta de que no sale la opción de añadir los archivos creados .php.

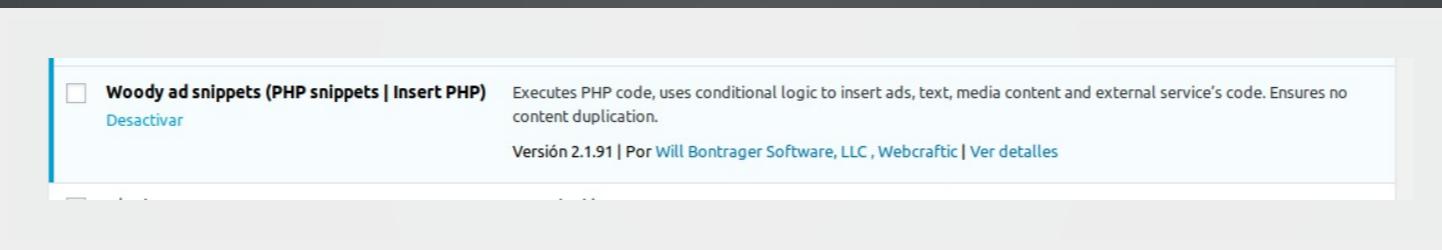
Investigamos por Internet y vemos que puede ser que nuestra versión de WordPress no sea compatible con la del plugin XYZ PHP CODE.

3.20 WOODY SNIPPETS

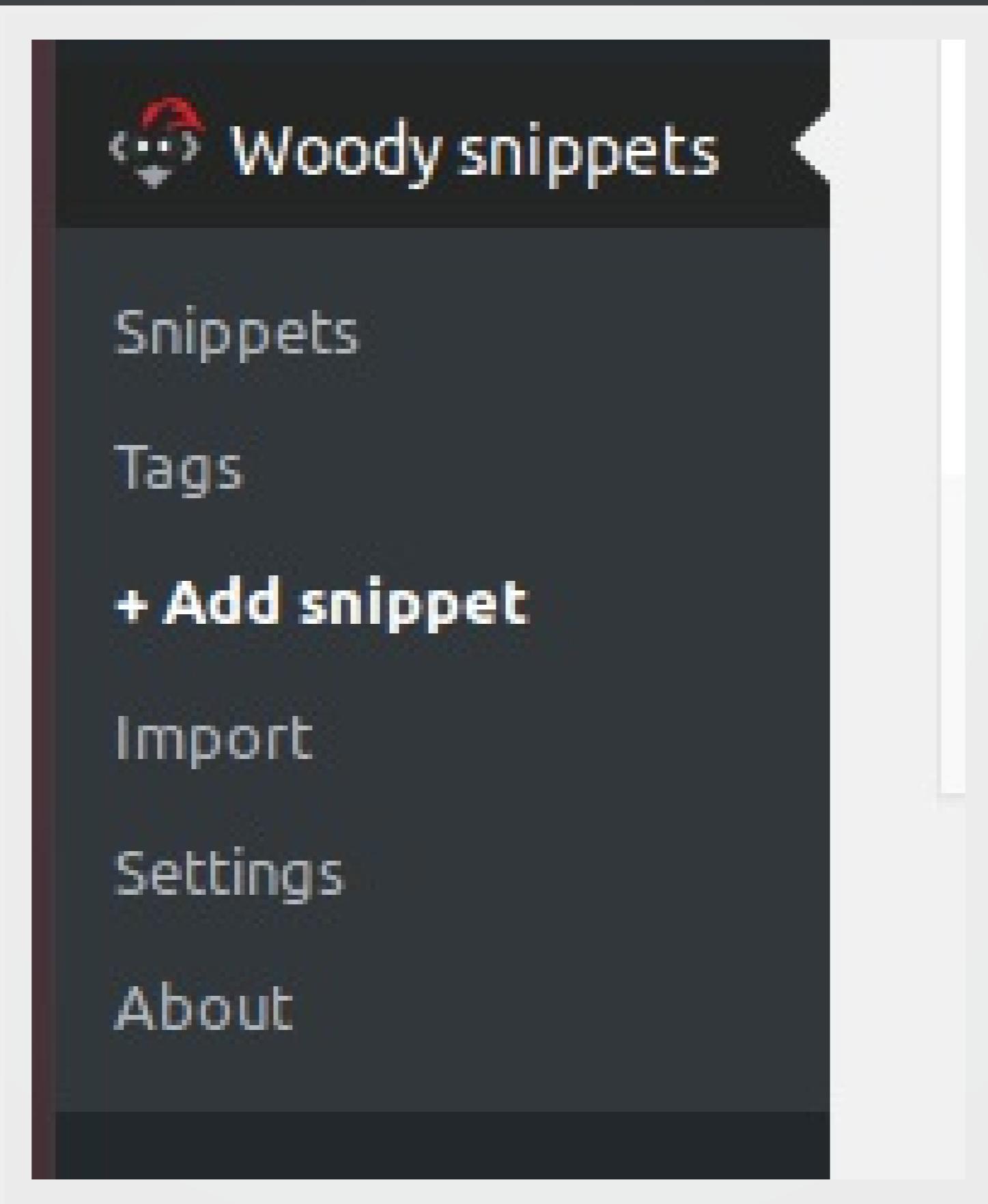
Buscando en Internet encontramos **WOODY SNIPPETS** y nos disponemos a probarlo, para ver si con este si podemos realizar todo el proceso para introducir los .php en nuestra página.

3.21 WOODY SNIPPETS (II)

Dentro de plugin lo descargamos, instalamos y activamos.

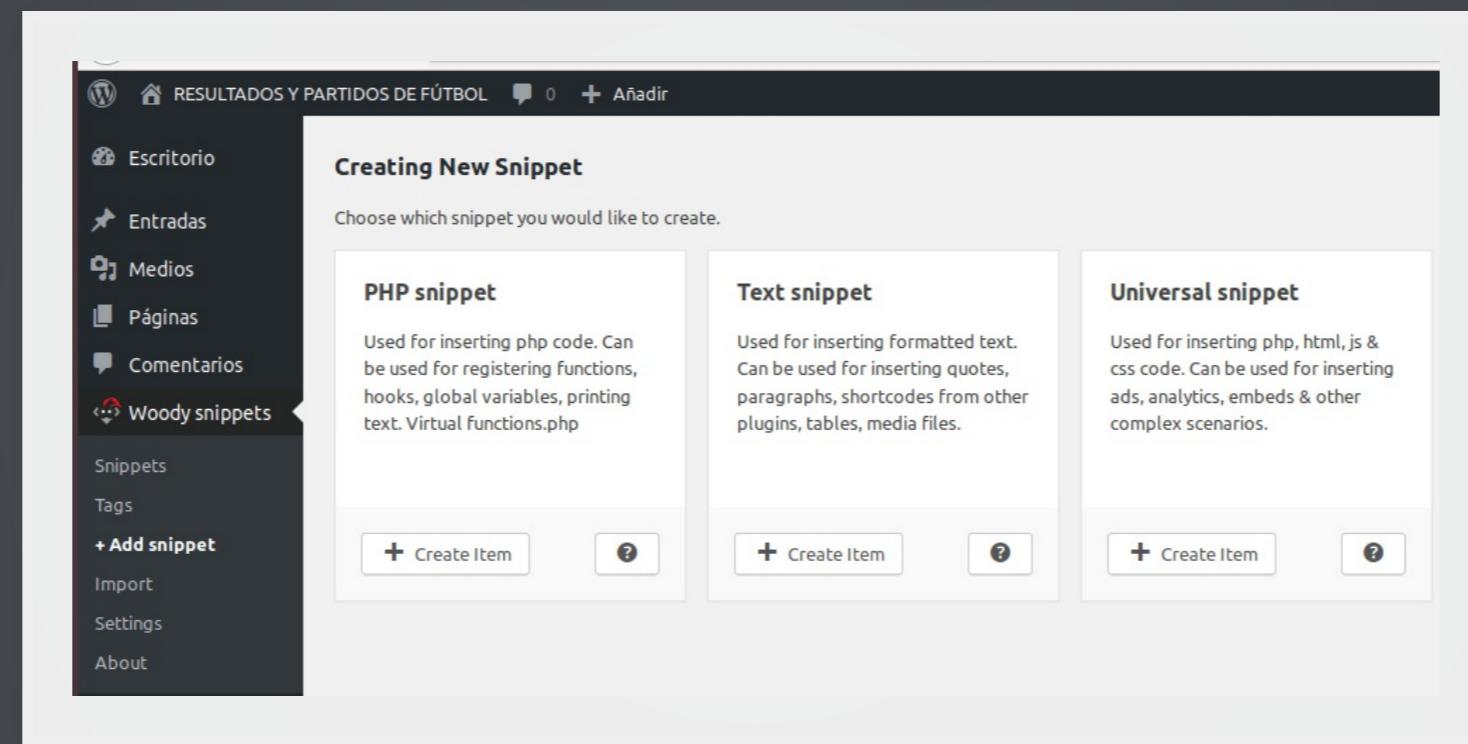


3.22 WOODY SNIPPETS (III)



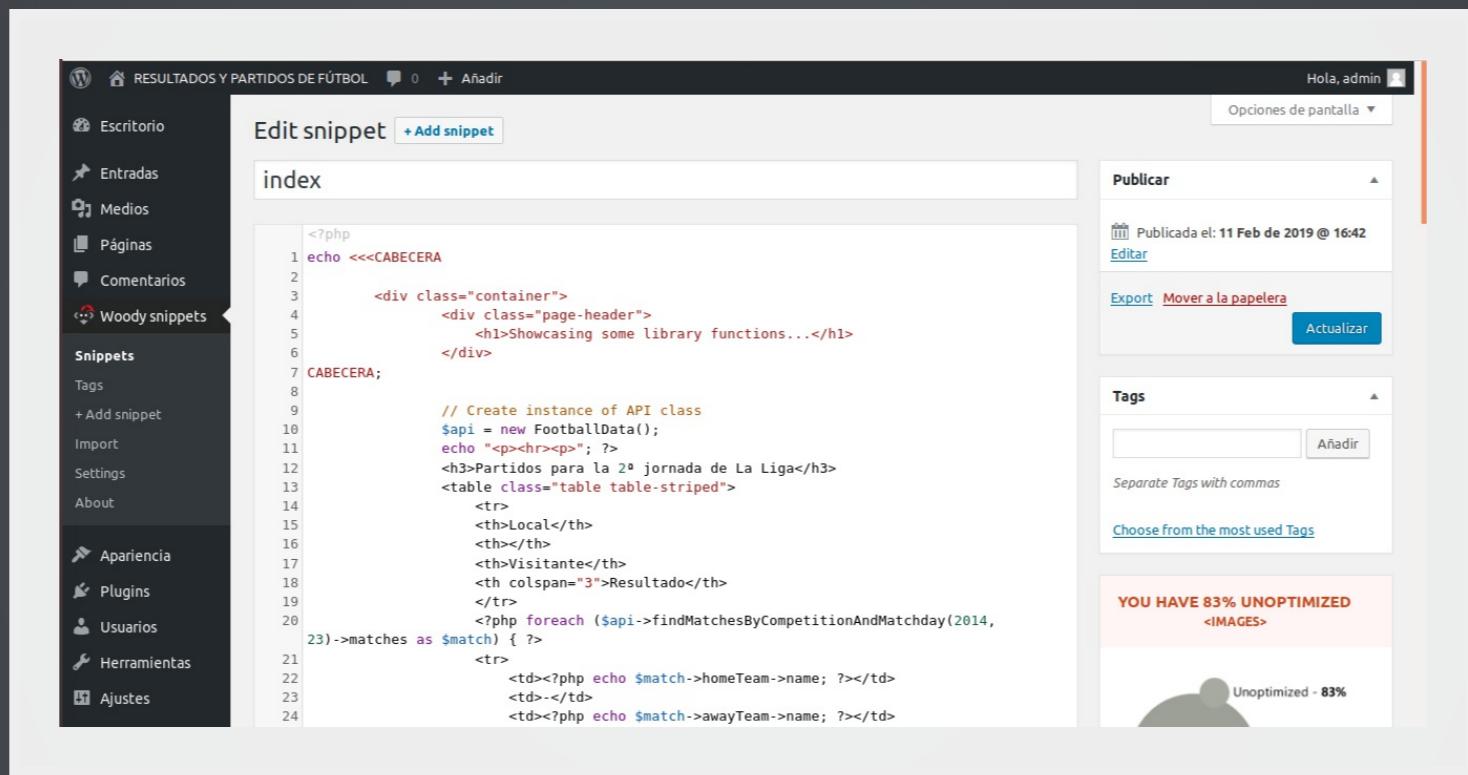
3.23 WOODY SNIPPETS (IV)

Nos disponemos a introducir los códigos, dentro de Woody, en **PHP Snippet** y **Create Item**.



3.24 WOODY SNIPPETS (V)

Insertamos primero index.php



The screenshot shows the 'Edit snippet' page for the 'index' snippet in the Woody Snippets plugin. The left sidebar contains links like Escritorio, Entradas, Medios, Páginas, Comentarios, Woody snippets (selected), Snippets, Apariencia, Plugins, Usuarios, Herramientas, and Ajustes. The main area displays the following PHP code:

```
<?php
echo <<<CABECERA
<div class="container">
<div class="page-header">
<h1>Showcasing some library functions...</h1>
</div>
<h2>CABECERA;

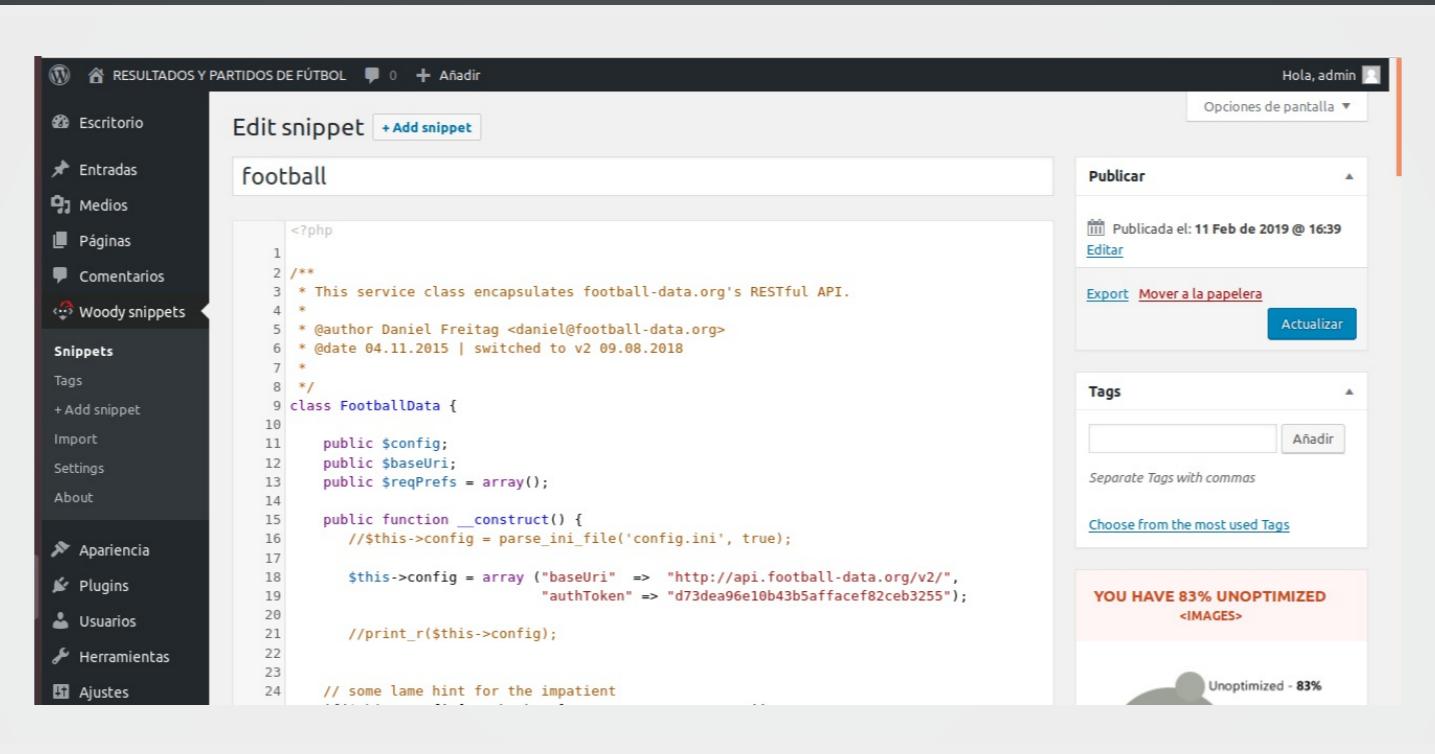
// Create instance of API class
$api = new FootballData();
echo "<p><hr><p>"; ?>
<h3>Partidos para la 2ª jornada de La Liga</h3>
<table class="table table-striped">
<tr>
<th>Local</th>
<th>Visitante</th>
<th>Resultado</th>
</tr>
<php foreach ($api->findMatchesByCompetitionAndMatchday(2014,
23)->matches as $match) { ?>
<tr>
<td><?php echo $match->homeTeam->name; ?></td>
<td><?php echo $match->awayTeam->name; ?></td>
<td><?php echo $match->score; ?></td>
</tr>
</php>

```

The right side of the interface includes sections for 'Publicar' (Publish), 'Tags', and a warning about unoptimized code: 'YOU HAVE 83% UNOPTIMIZED <IMAGES>'.

3.25 WOODY SNIPPETS (IV)

Una vez insertado nuestro index.php insertamos también footballData.php.



Nos damos cuenta de que esta manera para insertar código php en WordPress tampoco nos sirve.

3.26 PROBLEMA PARA INSERTAR CÓDIGO PHP EN WORDPRESS VERSIÓN 5.0.3

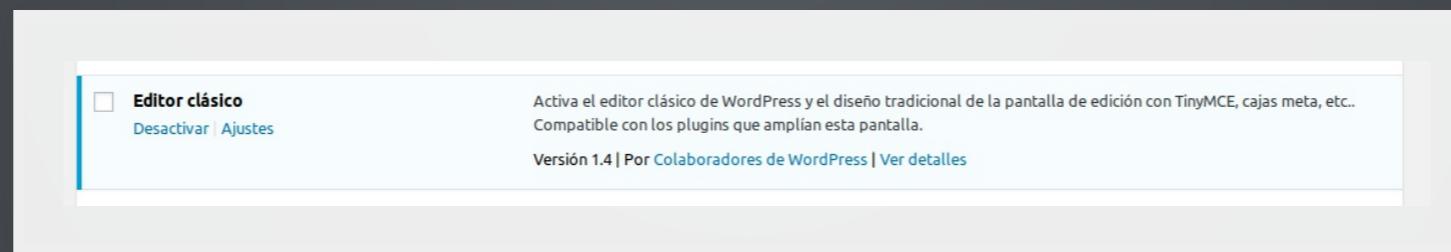
El problema para ambos insertadores de código es que a partir de la versión 5 de WordPress el editor que tenemos por defecto es Gutenberg.

3.27 PROBLEMA PARA INSERTAR CÓDIGO PHP EN WORDPRESS VERSIÓN 5.0.3 (II)

Para la utilización de los plugin que hemos probado tenemos que utilizar un [Editor Clasico](#).

3.28 PROBLEMA PARA INSERTAR CÓDIGO PHP EN WORDPRESS VERSIÓN 5.0.3 (III)

Hay que ir a Plugin, instalarlo y activarlo.



3.29 PROBLEMA PARA INSERTAR CÓDIGO PHP EN WORDPRESS VERSIÓN 5.0.3 (IV)

A continuación, en Ajustes de escritura habilitamos la utilización
del Editor Clasico.

3.30

The screenshot shows the 'Ajustes de escritura' (Writing settings) page in the WordPress admin interface. The left sidebar lists various menu items: Escritorio, Entradas, Medios, Páginas, Comentarios, Woody snippets, Apariencia, Plugins, Usuarios, and Herramientas. The 'Herramientas' item is currently selected, indicated by a blue bar at the bottom of the sidebar.

Categoría predeterminada para las entradas: Sin categoría

Formato de entrada por defecto: Estándar

Editor predeterminado para todos los usuarios: Editor clásico Editor de bloques

Permite a los usuarios cambiar de editor: Sí No

3.31 PROBLEMA PARA INSERTAR CÓDIGO PHP EN WORDPRESS VERSIÓN 5.0.3 (V)

Una vez hecho ésto ya podemos insertar código php tanto con XYZ PHP CODE, como con WOODY SNIPPETS. Hemos optado por acabar utilizando WOODY SNIPPETS.

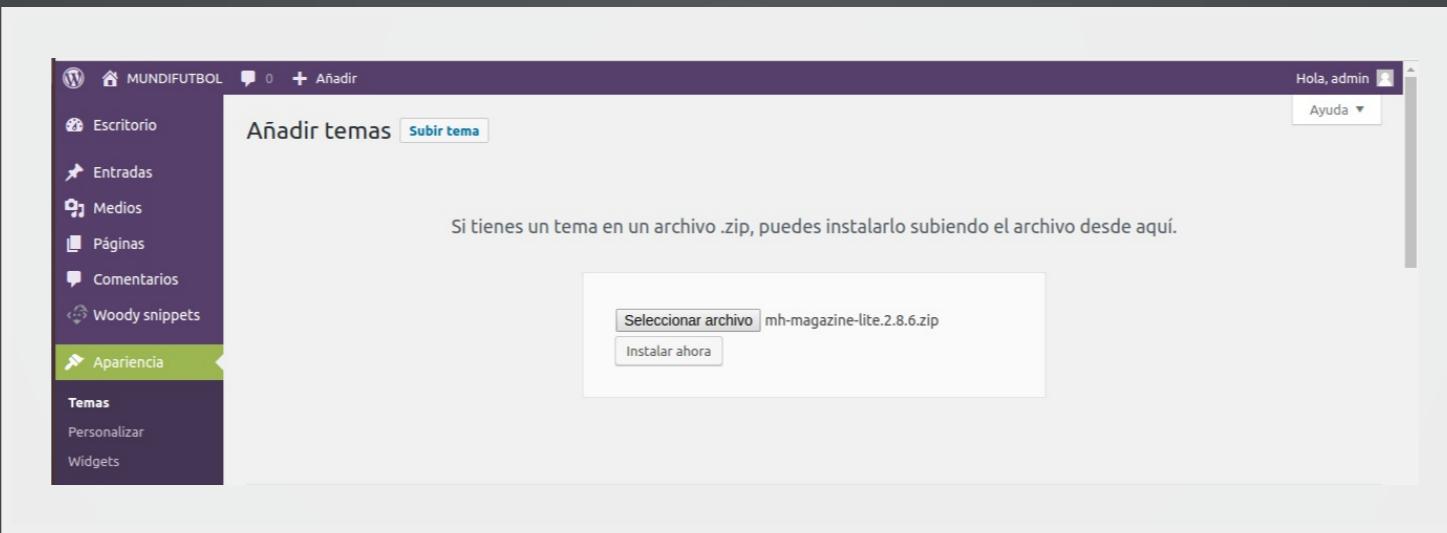
4 INSTALACIÓN Y PERSONALIZACIÓN

4.1 INSTALACIÓN

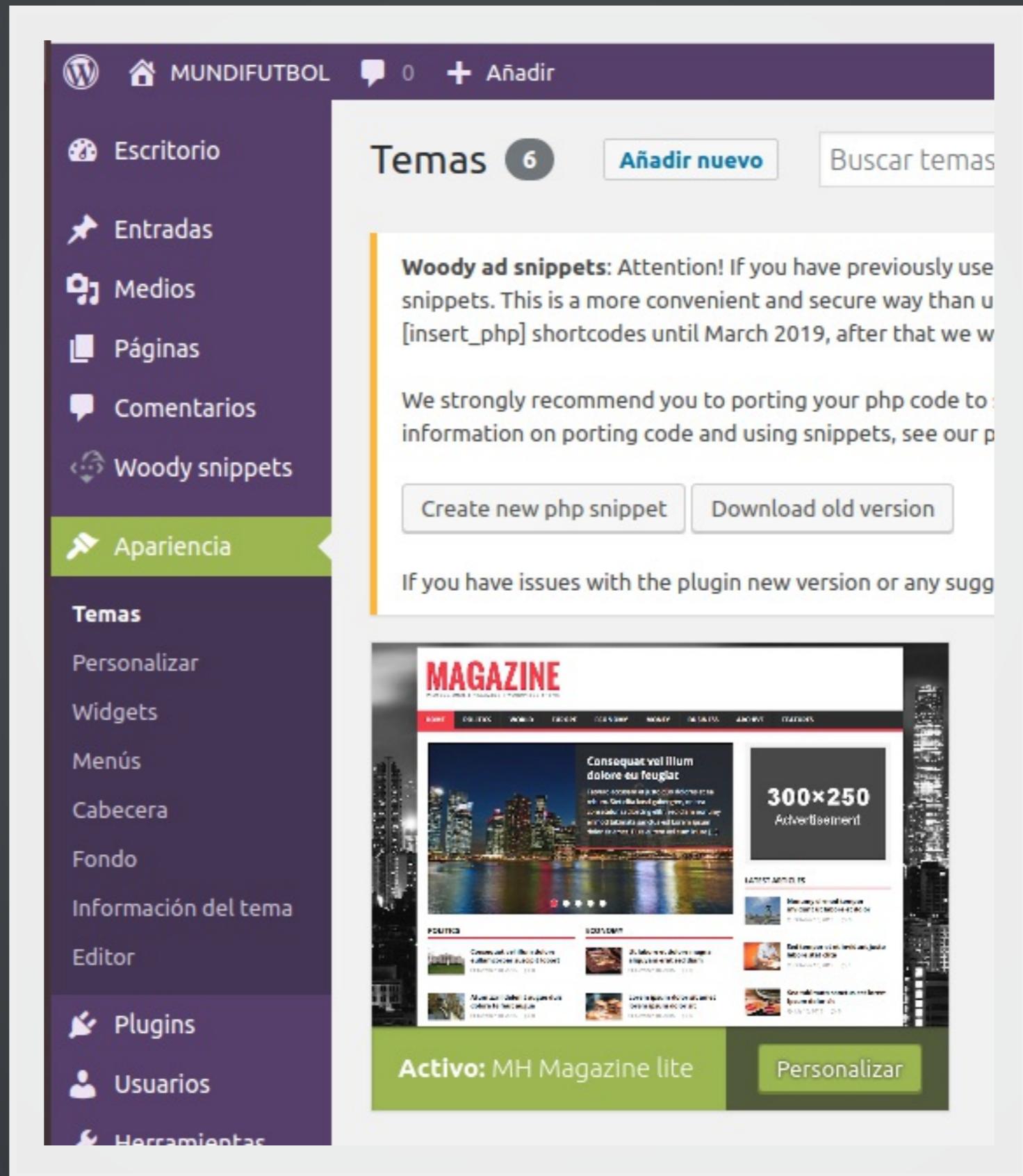
Descargamos [una plantilla gratuita](#) para su posterior utilización.

4.2 INSTALACIÓN (II)

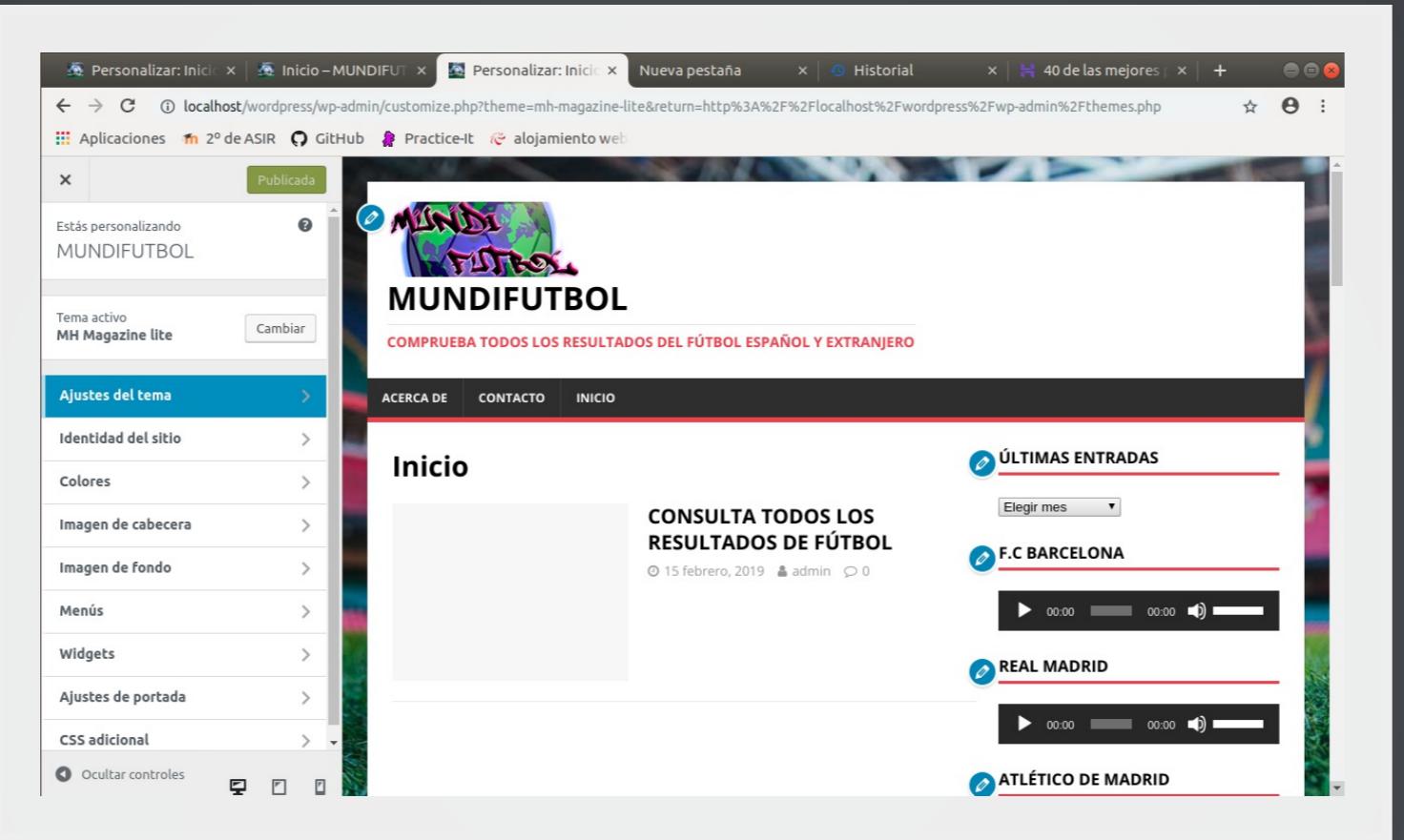
Dentro de WordPress nos vamos a **apariencia, temas, añadir uno nuevo, subir tema, seleccionar archivo**; cuando está seleccionado lo instalamos.



4.3 INSTALACIÓN (III)



4.4 PERSONALIZACIÓN DEL TEMA



4.5 PERSONALIZACIÓN DEL TEMA (II)

- Identidad del sitio
- Añadir logo, Título, Descripción e Icono
- Color
- Color cabecera
- Imagen fondo
- Fondo página
- Menú
- Menú superior
 - Inicio, Contacto y Acerca de...

4.6 PERSONALIZACIÓN DEL TEMA (III)

- Widgets
- Barra lateral
 - Últimas entradas, Audio (5 primeros clubs clasificados en La Liga), Comentarios recientes, Galería (mejores goles)
- Pie Página 1 - anchura 1/4 > [RSS \(EL MUNDO DEPORTIVO\)](#)
- Pie pagina 2 - anchura 1/4 > [RSS \(MARCA\)](#)
- Pie pagina 3 - anchura 1/4 > [RSS \(SPORT\)](#)
- Pie pagina 4 - anchura 1/4 > [RSS \(DIARIO GOL\)](#)

5 TABLAS DE RESULTADOS DEPORTIVOS

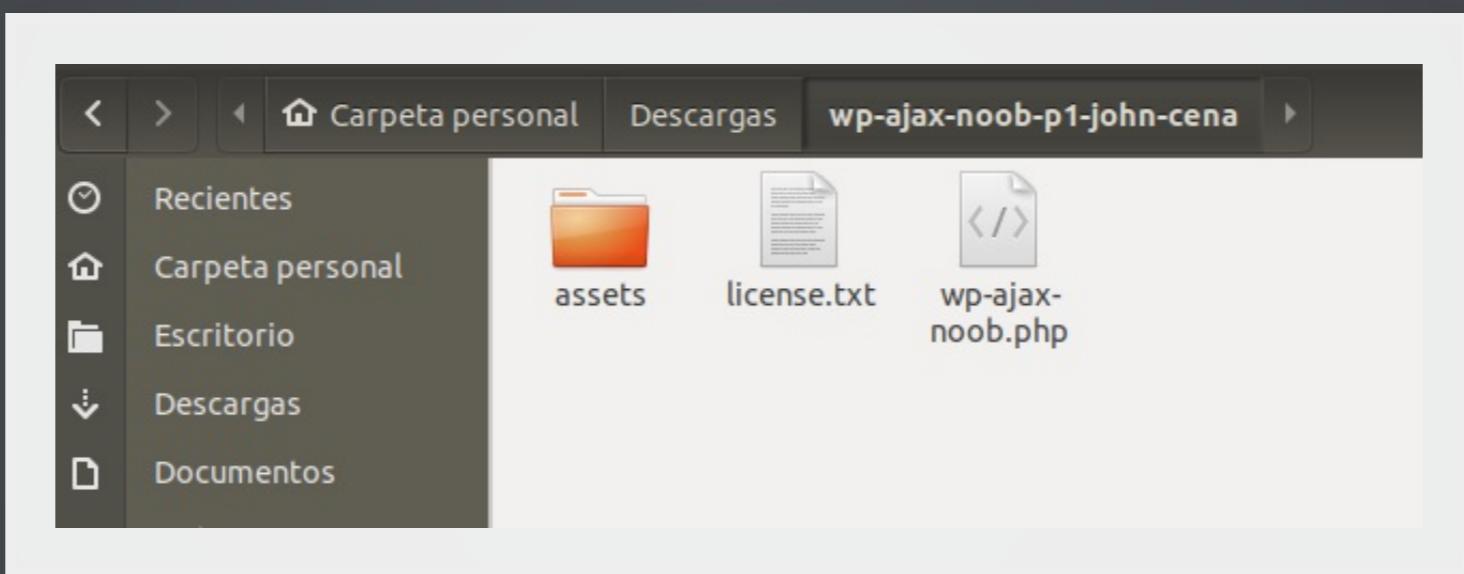
5.1 CREACIÓN PRIMERAS TABLAS

Su creación ha sido la más complicada.

En esta página descagamos el .zip "john-cena", es un plugin que utilizaremos a continuación.

5.2 CREACIÓN PRIMERAS TABLAS (II)

En el .zip encontramos



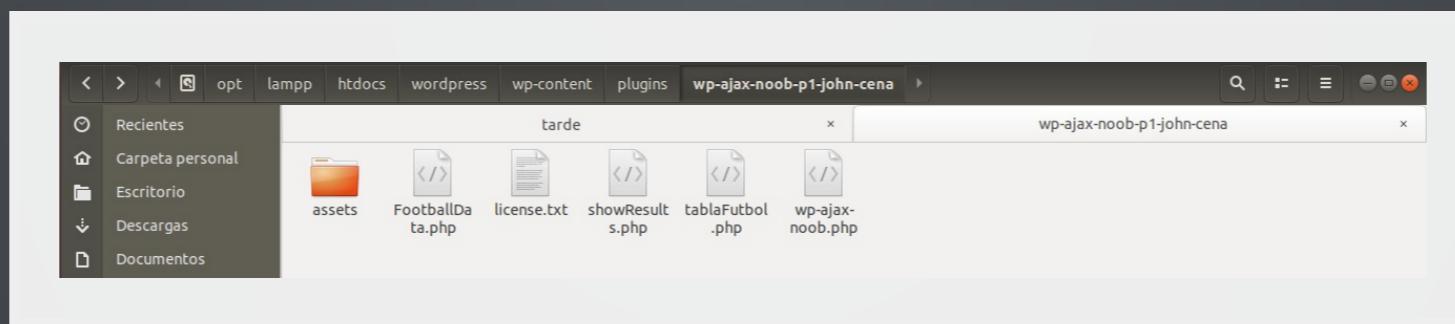
5.3 CREACIÓN PRIMERAS TABLAS (III)

Dentro de WordPress, plugin, añadir nuevo, subir plugin, lo seleccionamos, subimos, instalamos y activamos.



5.4 CREACIÓN PRIMERAS TABLAS (IV)

Añadimos los .php a la dirección
`/opt/lampp/htdocs/wordpress/wp-content/plugins/wp-ajax-noob-p1-john-cena`



5.5 MODIFICACIONES

- Modificación wp-ajax-noob.php

```
function my_wp_ajax_noob_john_cena_shortcode_callback(){
    /* Enqueue JS only if this shortcode loaded. */
    wp_enqueue_script( 'my-wp-ajax-noob-john-cena-script' );

    /* Output empty div. */

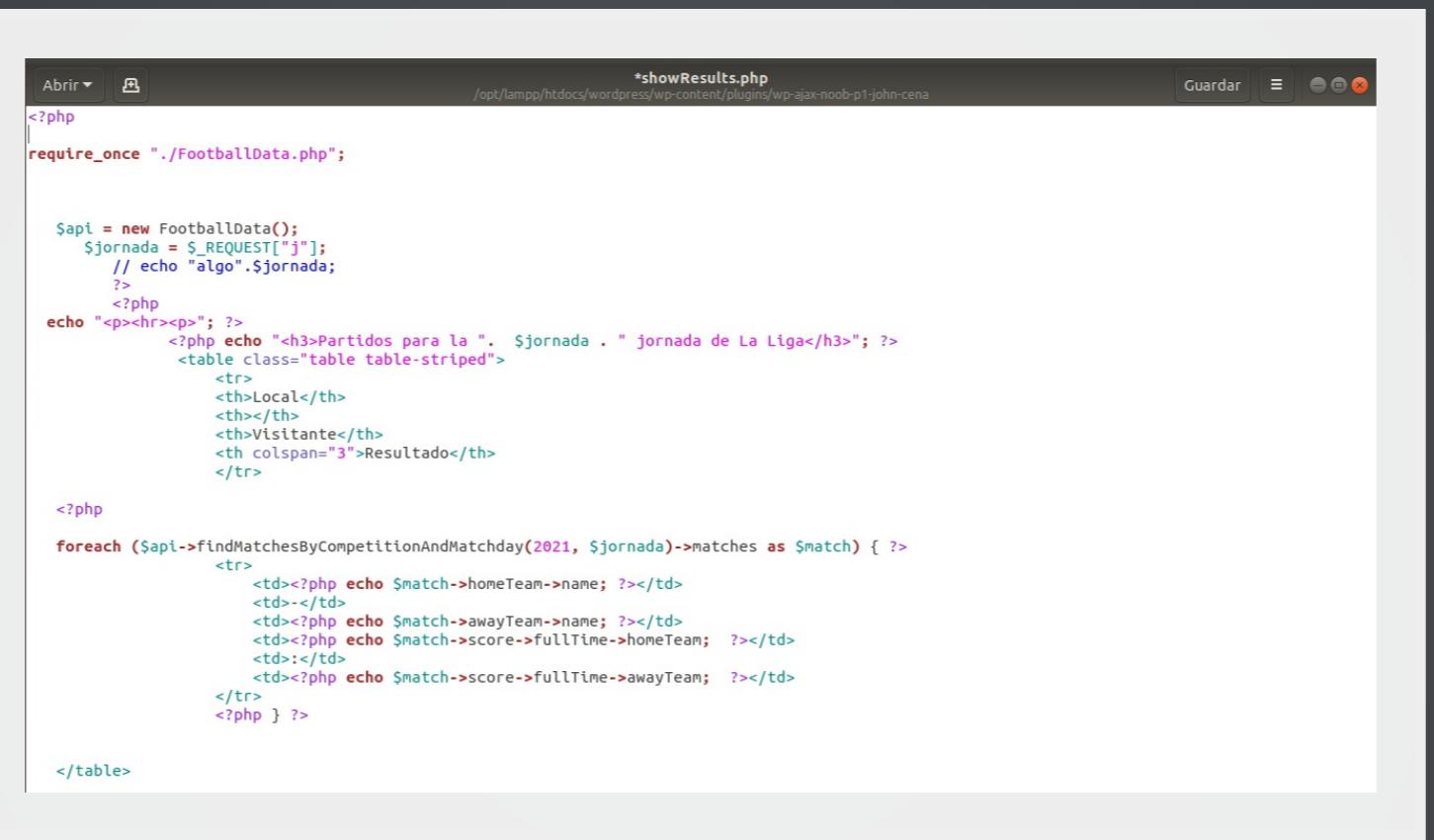
    $formulario = "<form action=''";
    $formulario .= "<p>Jornada:</p>";
    $formulario .= "<select id='jornada' name='jornada' onchange='showHint()'>";
    $formulario .= "<option value='1'>Jornada 1</option>";

    for ($j=1; $j <= 34; $j++)
    {
        $formulario .= "<option value='".$j."'>Jornada ".$j."</option>";
    }
    $formulario .= "</select></form><span id='txtHint'></span></p> ";

    return '<div id="john-cena"></div>' . $formulario;
}
```

5.6 MODIFICACIONES (II)

- Modificación showResult



The screenshot shows a code editor window with the file 'showResults.php' open. The code is a PHP script that retrieves football match results from a database and displays them in a table. The script includes imports, variable declarations, and a loop that iterates through matches, printing their details to the screen.

```
<?php
| require_once "./FootballData.php";

$api = new FootballData();
$jornada = $_REQUEST["j"];
// echo "algo".$jornada;
?>
<?php
echo "<p><hr><p>"; ?>
<?php echo "<h3>Partidos para la ". $jornada . " jornada de La Liga</h3>"; ?>
<table class="table table-striped">
<tr>
<th>Local</th>
<th></th>
<th>Visitante</th>
<th colspan="3">Resultado</th>
</tr>
<?php
foreach ($api->findMatchesByCompetitionAndMatchday(2021, $jornada)->matches as $match) { ?>
<tr>
<td><?php echo $match->homeTeam->name; ?></td>
<td></td>
<td><?php echo $match->awayTeam->name; ?></td>
<td><?php echo $match->score->fullTime->homeTeam; ?></td>
<td></td>
<td><?php echo $match->score->fullTime->awayTeam; ?></td>
</tr>
<?php } ?>

</table>
```

5.7 MODIFICACIONES (III)

Existen dos maneras de hacerlo

5.8 MODIFICACIONES (IV)

Dentro del archivo redireccionar para que coja el archivo
FootballData.php

```
<?php  
|  
require_once "./FootballData.php";  
  
$apt = new FootballData();  
$jornada = $REQUEST["j"];
```

5.9 MODIFICACIONES (V)

- Copiar archivo FootballData.php en el directorio:
 - /opt/lampp/htdocs/wordpress/wp-includes/
 - y darle permisos
 - sudo chown daemon:daemon
/opt/lampp/htdocs/wordpress/wp
includes/FootballData.php

5.10 MODIFICACIONES (VI)

```
<!DOCTYPE html>
<html>
<body>

<h3>Start typing a name in the input field below:</h3>

<form action="">

<p>Jornada:<br/>
<select id="jornada" name="jornada" onchange="showHint()">
<?php

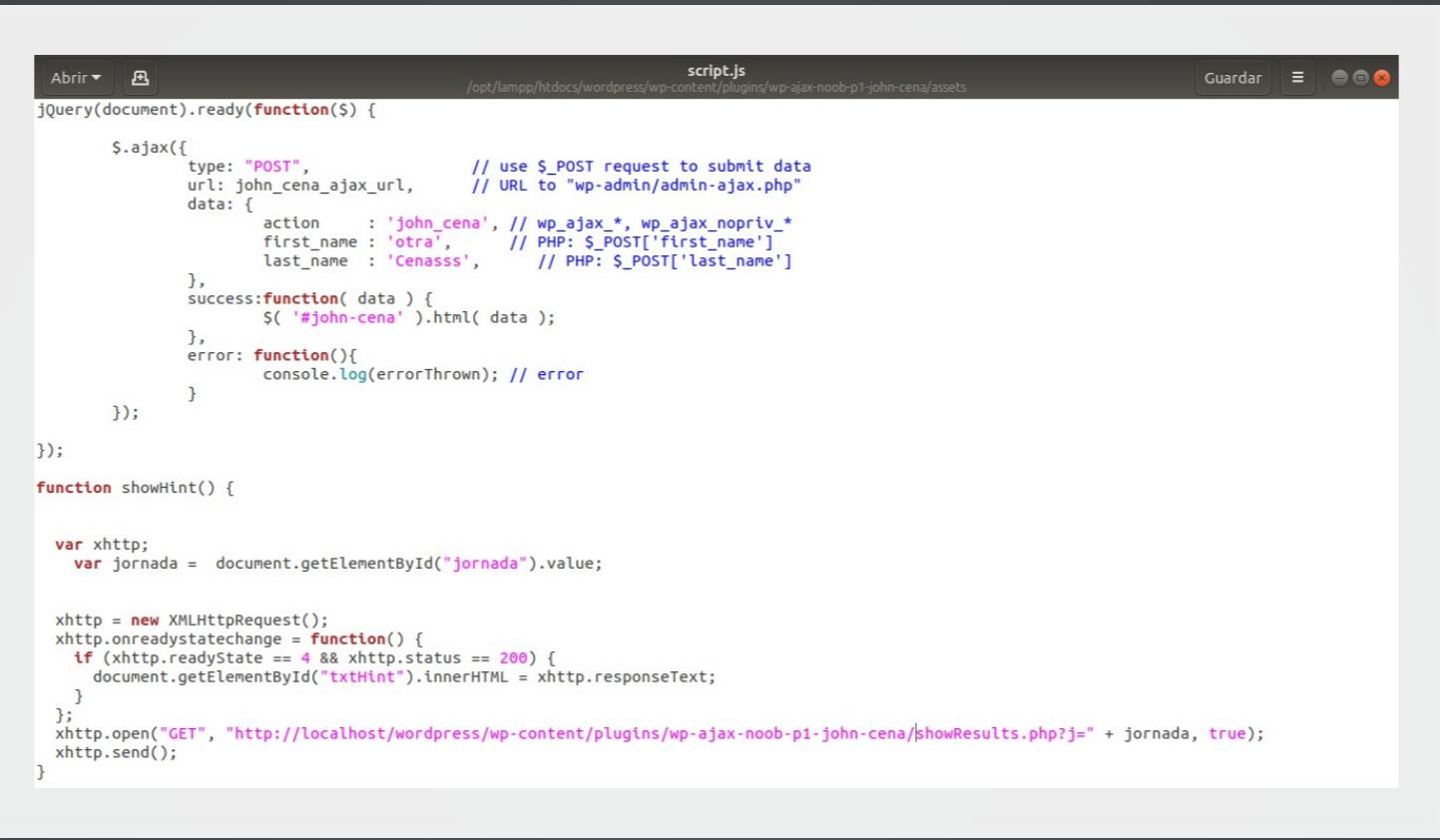
    for ($j=1; $j <= 34; $j++)
    {
        echo "<option value='".$j.">Jornada ".$j."</option>";
    }
    ?>
</select>

</form>

<p>Suggestions: <span id="txtHint"></span></p>

<script>
function showHint() {
    var xhttp;
    var jornada = document.getElementById("jornada").value;
    xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (xhttp.readyState == 4 && xhttp.status == 200) {
            document.getElementById("txtHint").innerHTML = xhttp.responseText;
        }
    }
}
```

5.11 MODIFICACIONES (VII)



A screenshot of a code editor window titled "script.js". The file path is "/opt/lampp/htdocs/wordpress/wp-content/plugins/wp-ajax-noob-p1-john-cena/assets". The code is written in JavaScript and contains two main functions: an anonymous function for document ready and a "showHint" function.

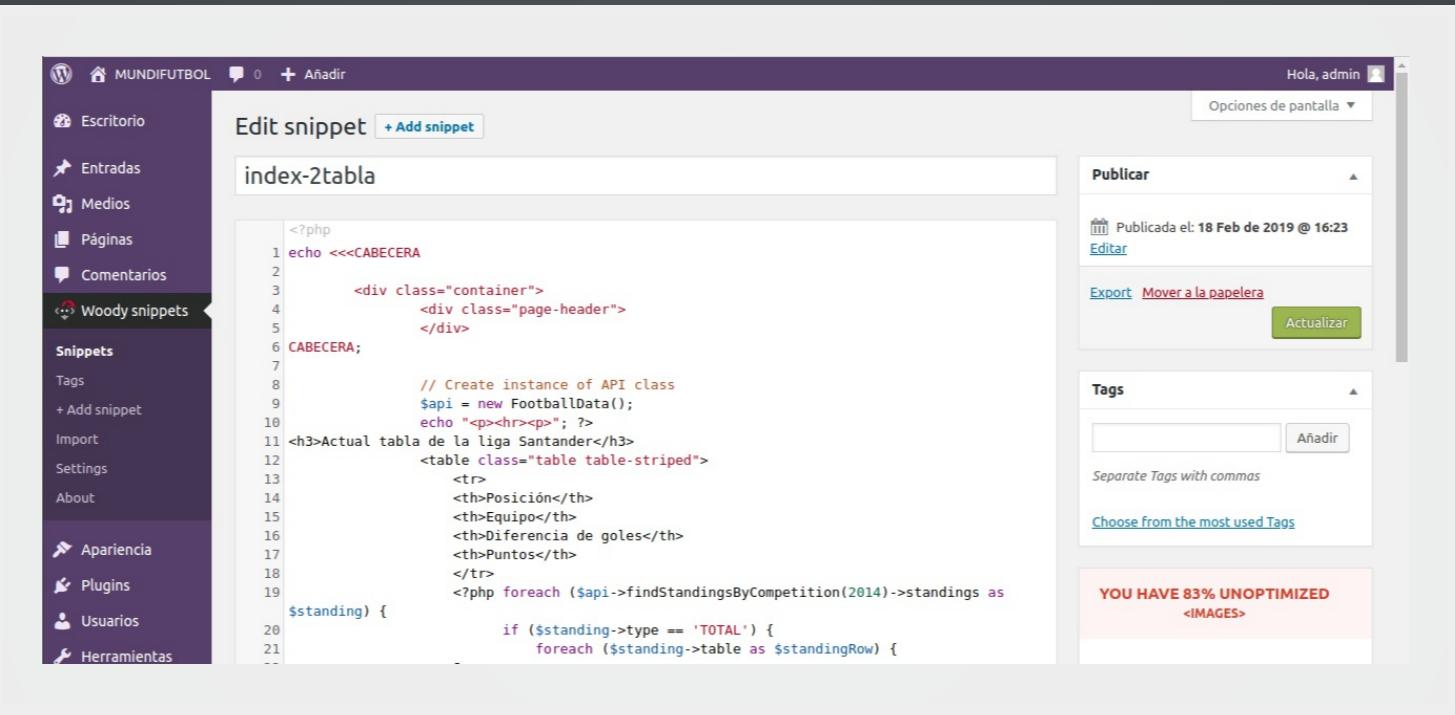
```
jQuery(document).ready(function($) {
    $.ajax({
        type: "POST", // use $_POST request to submit data
        url: john_cena_ajax_url, // URL to "wp-admin/admin-ajax.php"
        data: {
            action : 'john_cena', // wp_ajax_*, wp_ajax_nopriv_*
            first_name : 'otra', // PHP: $_POST['first_name']
            last_name : 'Cenasss', // PHP: $_POST['last_name']
        },
        success:function( data ) {
            $( '#john-cena' ).html( data );
        },
        error: function(){
            console.log(errorThrown); // error
        }
    });
});

function showHint() {
    var xhttp;
    var jornada = document.getElementById("jornada").value;

    xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (xhttp.readyState == 4 && xhttp.status == 200) {
            document.getElementById("txtHint").innerHTML = xhttp.responseText;
        }
    };
    xhttp.open("GET", "http://localhost/wordpress/wp-content/plugins/wp-ajax-noob-p1-john-cena/showResults.php?j=" + jornada, true);
    xhttp.send();
}
```

5.12 CREACIÓN SEGUNDAS TABLAS

Añadimos los php index y footballData de la segunda tabla

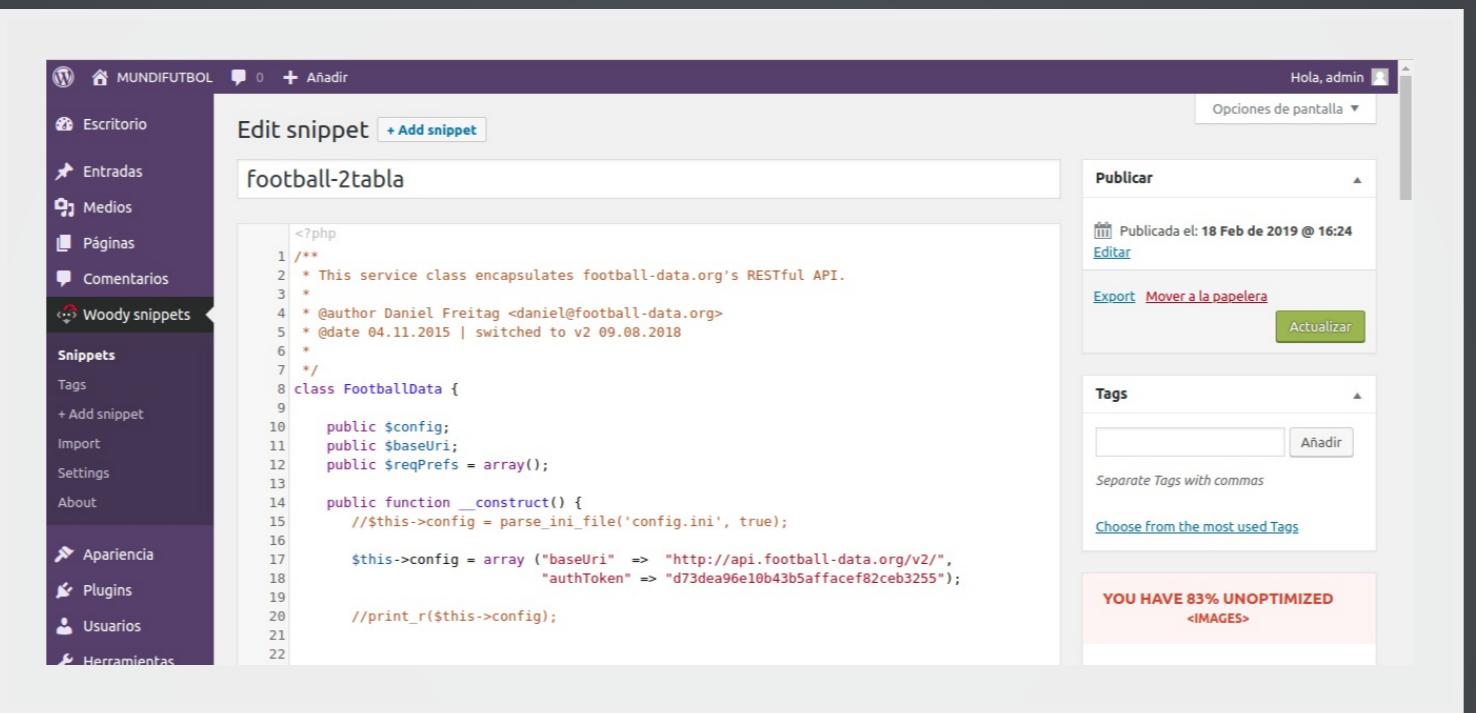


The screenshot shows the WordPress admin interface with the 'Woody snippets' plugin active. A modal window titled 'Edit snippet' is open, displaying the code for the 'index-2tabla' snippet. The code is a PHP script that outputs a header section and a table of football data from a 'Footballdata' API. The modal includes sections for 'Publicar' (Publish), 'Tags' (Tags), and a note about unoptimized code.

```
<?php
echo <<<CABECERA
3      <div class="container">
4          <div class="page-header">
5      </div>
6 CABECERA;
7
8      // Create instance of API class
9      $api = new Footballdata();
10     echo "<p><br><br>"; ?>
11 <h3>Actual tabla de la liga Santander</h3>
12     <table class="table table-striped">
13         <tr>
14             <th>Posición</th>
15             <th>Equipo</th>
16             <th>Diferencia de goles</th>
17             <th>Puntos</th>
18         </tr>
19         <?php foreach ($api->findStandingsByCompetition(2014)->standings as
$standing) {
20             if ($standing->type == 'TOTAL') {
21                 foreach ($standing->table as $standingRow) {

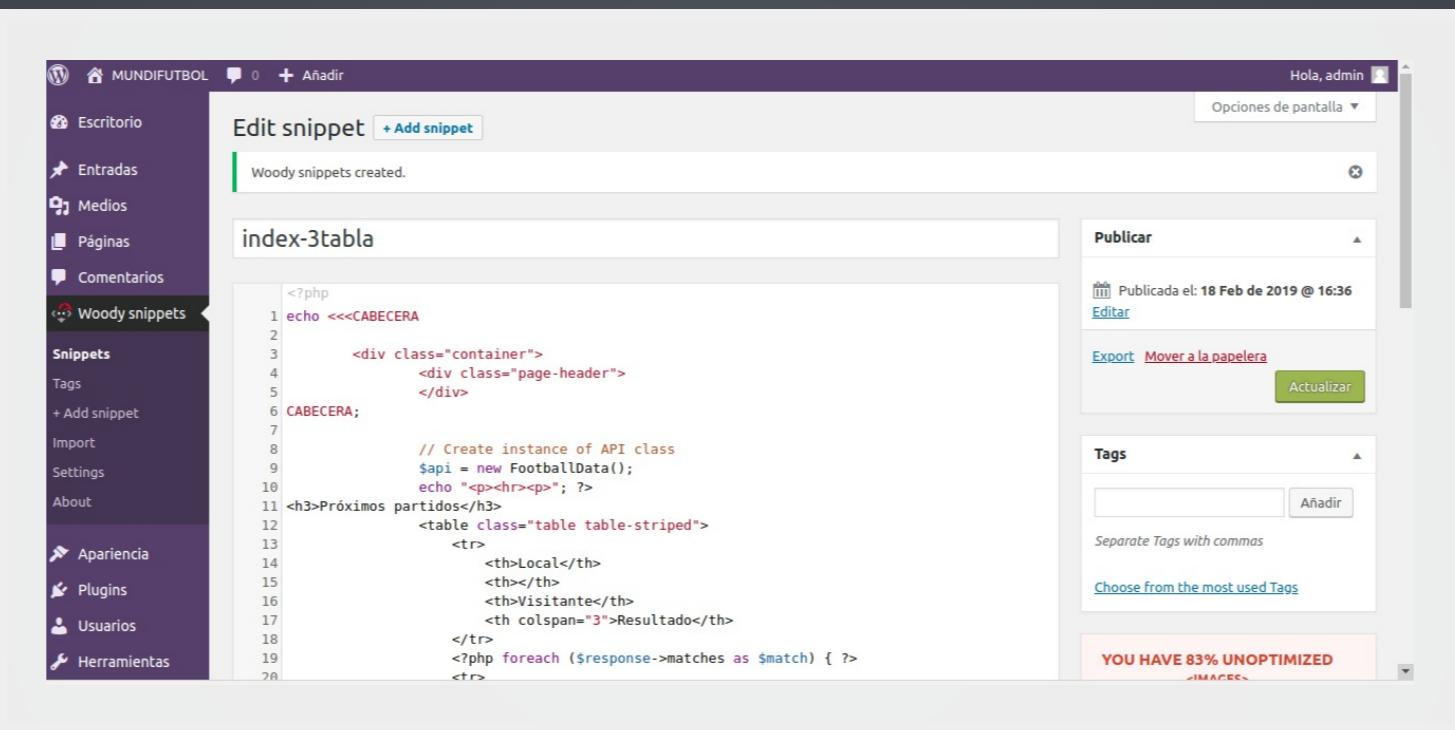
```

5.13 CREACIÓN SEGUNDAS TABLAS (II)



5.14 CREACIÓN TERCERAS TABLAS

Añadimos los php index y footballData de la tercera tabla

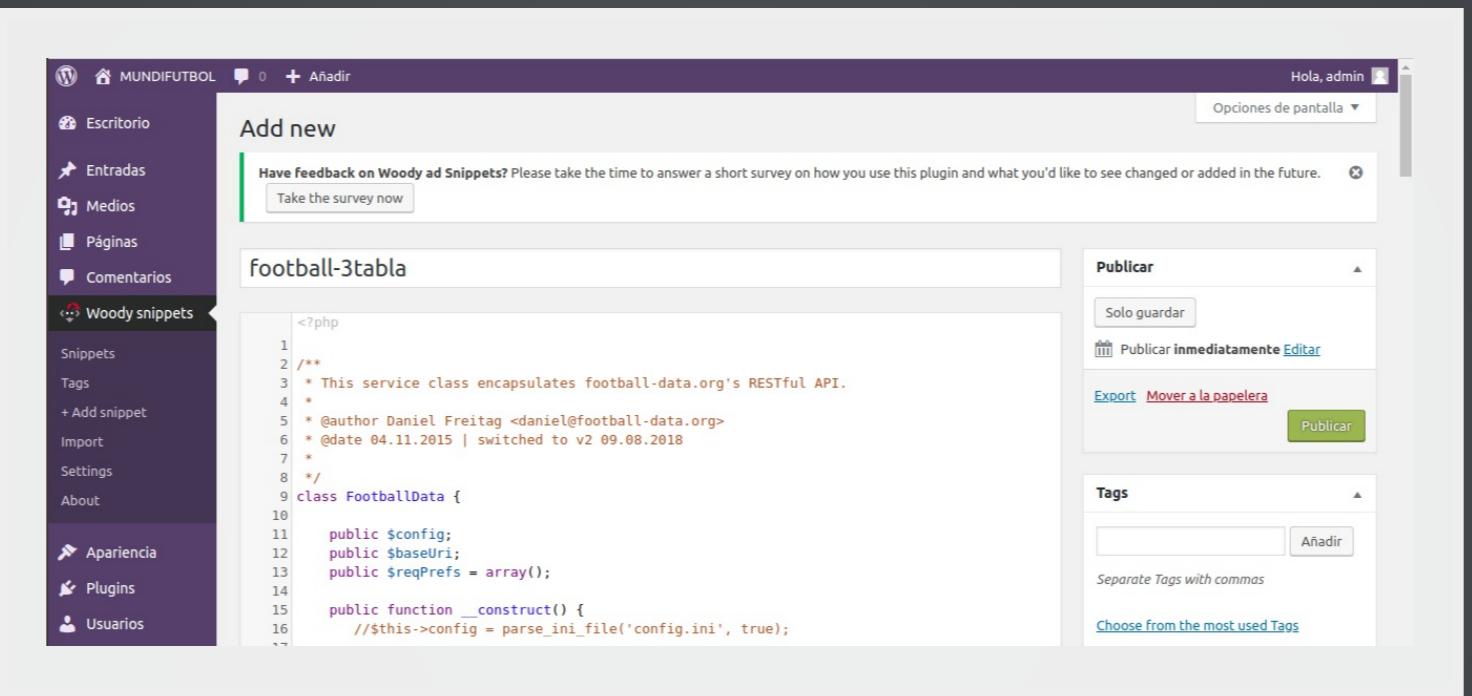


The screenshot shows the 'Edit snippet' screen for the 'index-3tabla' snippet. The left sidebar has 'Woody snippets' selected. The main area contains the following PHP code:

```
<?php
echo <<<CABECERA
<div class="container">
<div class="page-header">
</div>
CABECERA;
// Create instance of API class
$api = new FootballData();
echo "<p><hr><p>"; ?
<h3>Próximos partidos</h3>
<table class="table table-striped">
<tr>
<th>Local</th>
<th></th>
<th>Visitante</th>
<th colspan="3">Resultado</th>
</tr>
<?php foreach ($response->matches as $match) { ?>
<tr>
```

On the right, there are sections for 'Publicar' (Published on: 18 Feb de 2019 @ 16:36, Editar), 'Tags' (Tags field, Añadir), and a note 'YOU HAVE 83% UNOPTIMIZED'.

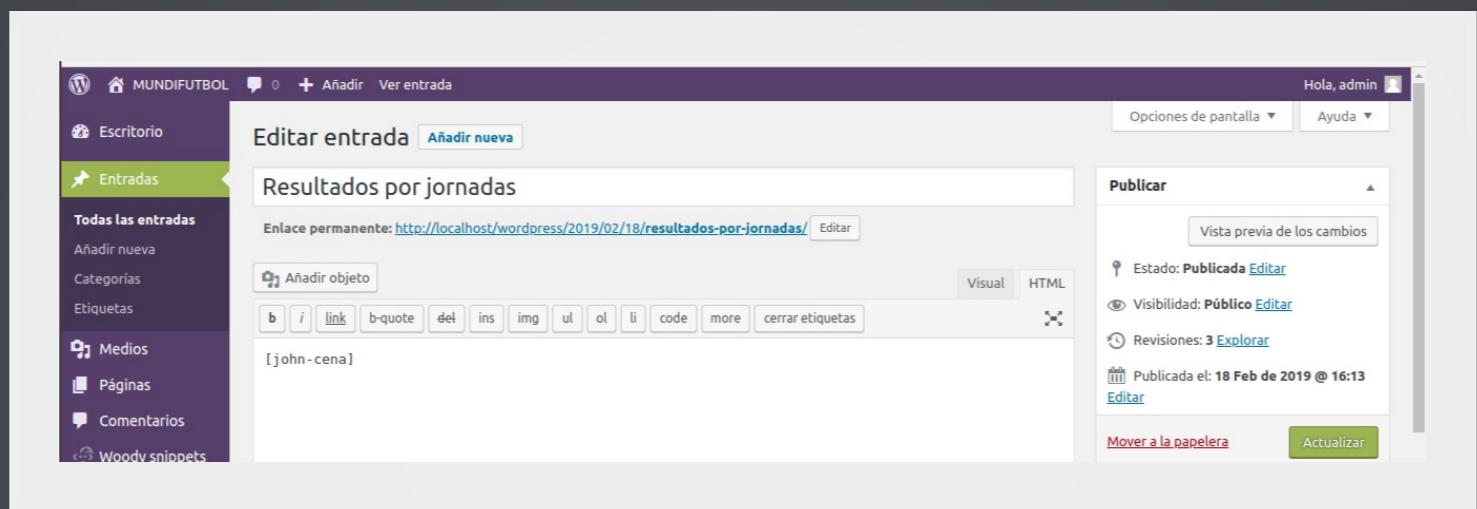
5.15 CREACIÓN TERCERAS TABLAS (II)



5.16 CREACIÓN PRIMERA PÁGINA (RESULTADOS POR JORNADAS)

- Entradas, añadir nueva
- Título
- Insertar [john-cena]
- Imagen destacada

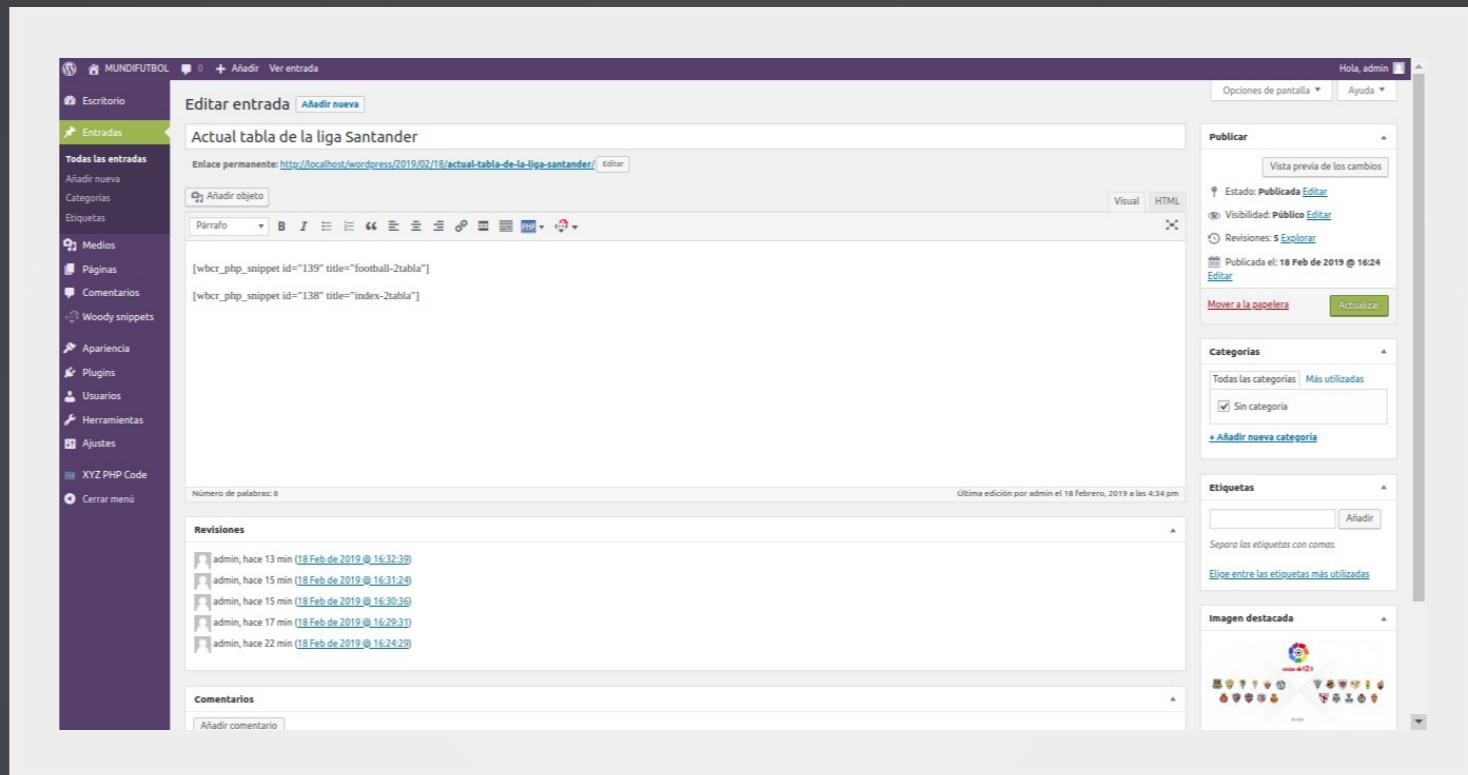
5.17 RESULTADOS POR JORNADAS



5.18 CREACIÓN SEGUNDA PÁGINA (ACTUAL TABLA DE LA LIGA SANTANDER)

- Entradas, añadir nueva
- Título
- Insertar código php
- Imagen destacada

5.19 ACTUAL TABLA DE LA LIGA SANTANDER



5.20 CREACIÓN TERCERA PÁGINA (PRÓXIMOS PARTIDOS)

- Entradas, añadir nueva
- Título
- Insertar código php
- Imagen destacada

5.21 PRÓXIMOS PARTIDOS

