



A6. Pp2.2.- Implantar, Administrar i Adapta Gestors de Contingut. WordPress.

Nom alumne: Nil Massó

## OBJECTIU

- Valora l'ús i utilitat dels gestors de continguts.
- Classifica segons la funcionalitat principal del lloc web que permeten gestionar.
- Instal·la diferents tipus de gestors de continguts.
- Diferencia les seves característiques (ús, llicència, entre d'altres).
- Personalitza i configura els gestors de continguts.
- Activa i configura els mecanismes de seguretat proporcionats pels mateixos gestors de continguts.
- Realitza proves de funcionament.
- Publica els gestors de continguts.
- Adapta i configura els mòduls del gestor de continguts.
- Crea i gestiona usuaris amb diferents perfils.
- Integra mòduls atenent requeriments de funcionalitat.
- Fa còpies de seguretat dels continguts.
- Importa i exporta continguts en diferents formats.
- Gestiona plantilles.
- Integra funcionalitats de sindicació.
- Realitza actualitzacions.
- Obté informes d'accés.



## INSTRUCCIONS

- Llegeix amb calma que s'ha de fer abans de començar. I recorda de posar el teu nom en el lloc del document que ho sol·licita.
- Els caràcters XX corresponen al valor donat a l'inici de la UF pel professor
- Enllaços que et poden servir d'ajuda per realitzar l'activitat:

## EXERCICIS

### Exercici:

En aquesta activitat el **que volem és afegir una adaptació al nostre gestor de continguts, es a dir volem afegir-hi plugins.**

Farem servir el **Wordpress** que vas implementar a l'anterior exercici on hi expliques el teu projecte i li afegirem **dos plugins propis, que farem amb PHP.**

<http://www.nmcagb.cat/wp/4waegfsioulncacms/>

### **Primer plugin:** Valoració màxima(5.0p)

En aquesta extensió es demana que aquesta ens mostri per pantalla les tasques principals que ens falten per acabar el projecte (poden ser inventades o simplement enumeracions). **A més a més es valorarà que des del plugin es puguin anar eliminant i per tant d'aquesta manera, en restaran menys.** Per fer-ho més senzill pots afegir les tasques principals, al principi, directament a la base de dades.

No se si he acabat de entendre lo dels hooks pero tal i com ho he fet funciona. En el mateix php està inclosa la creació i emplenament de la base de dades, per tant no cal cap script sql.



```
<?php

/*
Plugin Name: Task List
Description: A plugin to display and manage project tasks
Version: 1.0
Author: Nil Massó
*/

function task_list_shortcode() {
global $wpdb;
// Check for task deletions
if (isset($_POST['task'])) {
$wpdb->delete($wpdb->prefix . 'tasks', array('task' => $_POST['task']));
}
// Get remaining tasks
$tasks = $wpdb->get_results("SELECT task FROM " . $wpdb->prefix . "tasks");
// Display task list
$output = '<h2>Task List</h2>';
$output .= '<ul>';
foreach ($tasks as $task) {
$output .= '<li> . $task->task . ' <form method="post"><input type="hidden"
name="task" value="' . $task->task . '"><input type="submit"
value="Delete"></form></li>';
}
$output .= '</ul>';
return $output;
}
add_shortcode('task-list', 'task_list_shortcode');

// Create tasks table on plugin activation
function task_list_activate() {
global $wpdb;
$table_name = $wpdb->prefix . 'tasks';
$charset_collate = $wpdb->get_charset_collate();
$sql = "CREATE TABLE if not exists $table_name (
id mediumint(9) NOT NULL AUTO_INCREMENT,
task text NOT NULL,
end_date date DEFAULT '0000-00-00' NOT NULL,
PRIMARY KEY (id)
) $charset_collate;";
require_once(ABSPATH . 'wp-admin/includes/upgrade.php');
```



```
dbDelta($sql);
```

```
// Add initial tasks to database if table is empty
if ($wpdb->get_var("SELECT COUNT(*) FROM " . $wpdb->prefix . "tasks") == 0) {
    $wpdb->insert($wpdb->prefix . 'tasks', array('task' => 'Crear la base de dades',
    'end_date' => '2023-12-01'));
    $wpdb->insert($wpdb->prefix . 'tasks', array('task' => 'Crear el plugin', 'end_date' =>
    '2023-12-02'));
    $wpdb->insert($wpdb->prefix . 'tasks', array('task' => 'Crear la web', 'end_date' =>
    '2023-12-03'));
}
}
register_activation_hook(__FILE__, 'task_list_activate');
?>
```

### Segon plugin: Valoració màxima (5.0p)

En aquesta extensió es demana **que ens calculi el nombre d'hores que estem destinant al projecte.**

D'aquesta manera, a la nova pestanya clicarem en un botó (o de la manera que considereu) per iniciar el comptatge i de nou en un botó (o com estimeu) per finalitzar-ho. Amb això sabrem el temps de treball actual i juntament amb el que tenim acumulat (BDD) també sabem el total de temps destinat al projecte fins ara. La manera de mostrar-ho també la deixo al teu criteri

Per a la web he utilitzat counter.php, l'altre ha sigut una prova de que el contador fos en temps real. En el mateix php està inclosa la creació i emplenament de la base de dades, per tant no cal cap script sql.

counter.php:

```
<?php
```

```
/*
Plugin Name: Hour Counter
Description: A plugin to count hours worked
Version: 1.0
Author: Nil Massó
*/
```



```
function hour_counter_shortcode() {
global $wpdb;
// Get total hours from database
$total_hours = $wpdb->get_var("SELECT total_hours FROM " . $wpdb->prefix . "hours" .
" WHERE id = 1");
// Check for timer start and stop
if (isset($_POST['start'])) {
update_option('hour_counter_start', time());
} elseif (isset($_POST['stop'])) {
$start_time = get_option('hour_counter_start');
if ($start_time) {
$end_time = time();
$hours = ($end_time - $start_time) / 3600;
$total_hours += $hours;
$wpdb->replace($wpdb->prefix . 'hours', array('total_hours' => $total_hours) +
array('id' => 1));
delete_option('hour_counter_start');
}
}
// Display hour counter
$output = '<h2>Hour Counter</h2>';
$output .= '<p>Total hours: ' . number_format($total_hours, 2) . '</p>';
$output .= '<form method="post">';
if (get_option('hour_counter_start')) {
$output .= '<input type="submit" name="stop" value="Stop">';
} else {
$output .= '<input type="submit" name="start" value="Start"
onclick="setInterval(updateTime, 1000)">';
}
$output .= '</form>';

return $output;
}

add_shortcode('hour-counter', 'hour_counter_shortcode');

// Create hours table on plugin activation
function hour_counter_activate() {
global $wpdb;
$table_name = $wpdb->prefix . 'hours';
$charset_collate = $wpdb->get_charset_collate();
$sql = "CREATE TABLE if not exists $table_name (
id mediumint(9) NOT NULL AUTO_INCREMENT,
```



```
total_hours float(10) NOT NULL,  
PRIMARY KEY (id)  
) $charset_collate;";  
require_once(ABSPATH . 'wp-admin/includes/upgrade.php');  
dbDelta($sql);  
$wpdb->insert($table_name, array('total_hours' => 0));  
}  
register_activation_hook(__FILE__, 'hour_counter_activate');  
  
?>
```

counter-prova.php:

```
<script>  
  
var start_time = 0;  
function updateTime() {  
if (start_time == 0) {  
start_time = ' . time() . ';  
} else {  
start_time = start_time;  
}  
var end_time = ' . time() . ';  
var total_seconds = end_time - start_time;  
var hours = Math.floor(total_seconds / 3600);  
total_seconds %= 3600;  
var minutes = Math.floor(total_seconds / 60);  
var seconds = total_seconds % 60;  
document.getElementById("elapsed-time").innerHTML = hours + "h " + minutes + "m "  
+ seconds + "s";  
}  
</script>  
<?php  
/*  
Plugin Name: Hour Counter  
Description: A plugin to count hours worked  
Version: 1.0  
Author: Nil Massó  
*/  
  
function hour_counter_shortcode() {
```



```
global $wpdb;
// Get total hours from database
$total_hours = $wpdb->get_var("SELECT total_hours FROM " . $wpdb->prefix . "hours" .
" WHERE id = 1");
// Check for timer start and stop
if (isset($_POST['start'])) {
    update_option('hour_counter_start', time());
} elseif (isset($_POST['stop'])) {
    $start_time = get_option('hour_counter_start');
    if ($start_time) {
        $end_time = time();
        $hours = ($end_time - $start_time) / 3600;
        $total_hours += $hours;
        $wpdb->replace($wpdb->prefix . 'hours', array('total_hours' => $total_hours) +
        array('id' => 1));
        delete_option('hour_counter_start');
    }
}

// Display hour counter
$output = '<h2>Hour Counter</h2>';
$output .= '<p>Total hours: ' . number_format($total_hours, 2) . '</p>';
//Elapsed time in real time using JavaScript
$output .= '<p>Elapsed time: <span id="elapsed-time">0</span> seconds</p>';
$output .= '<form method="post">';
if (get_option('hour_counter_start')) {
    $output .= '<input type="submit" name="stop" value="Stop">';
} else {
    $output .= '<input type="submit" name="start" value="Start"
    onclick="setInterval(updateTime, 1000)">';
}
$output .= '</form>';

return $output;
}

add_shortcode('hour-counter', 'hour_counter_shortcode');

// Create hours table on plugin activation
function hour_counter_activate() {
    global $wpdb;
    $table_name = $wpdb->prefix . 'hours';
    $charset_collate = $wpdb->get_charset_collate();
    $sql = "CREATE TABLE if not exists $table_name (
```



```
id mediumint(9) NOT NULL AUTO_INCREMENT,  
total_hours float(10) NOT NULL,  
PRIMARY KEY (id)  
) $charset_collate;";  
require_once(ABSPATH . 'wp-admin/includes/upgrade.php');  
dbDelta($sql);  
$wpdb->insert($table_name, array('total_hours' => 0));  
}  
register_activation_hook(__FILE__, 'hour_counter_activate');
```

??

Per obtenir la màxima valoració aquests **addon** s'han de realitzar primer a la màquina on tinguem el Wordpress en local (i posteriorment provar-los en el hosting. Pensa que en aquest cas la base de dades pot no ser la mateixa que tens en local).

### Entrega:

S'ha de posar **dins aquest document els codis php i scrips de BDD que hagi realitzat per tal de donar resposta a la proposta de l'activitat.** I entregar-lo en el temps i forma habituals.

Així mateix, i per tal de simplificar la correcció es farà **una demostració en viu** a classe.

## AVALUACIÓ

Que es valorarà en cadascun dels plugin:

- Si compleix el què és demana (també al hosting)(3.0p)
- Utilització de funcions, estructures i altres ítems de “bon” usuari del PHP (1.0p)
- L'usabilitat, disseny i presència del pluggin creat. (1.0p)