

## Seguimiento de trayectorias estudiantiles en Moodle para el programa de la MTIE

### Rastreo do percurso do estudante no Moodle para o programa MTIE

DOI:10.34117/bjdv9n2-014

Recebimento dos originais: 02/01/2023

Aceitação para publicação: 02/02/2023

#### **Raúl Armando Valadez Estrada**

Doctorado en Ciencias de la Educación  
Institución: Universidad Autónoma de Zacatecas  
Correo electrónico: raul.valadez@hotmail

#### **Susana Cordero Dávila**

Doctorado en Ciencias de la Educación  
Institución: Universidad Autónoma de Zacatecas  
Correo electrónico: susy@uaz.edu.mx

#### **José de Jesús Hernández Berumen**

Doctorado en Administración  
Institución: Universidad Autónoma de Zacatecas  
Correo electrónico: jherber@uaz.edu.mx

#### **Alejandro Rodolfo García Villalobos**

Doctorado en Ciencias de la Educación  
Institución: Universidad Autónoma de Zacatecas  
Correo electrónico: martha.hernandez@uaz.edu.mx

#### **Martha Susana Hernández Larios**

Doctorado en Ciencias de la Educación,  
Institución: Universidad Autónoma de Zacatecas  
Correo electrónico: martha.hernandez@uaz.edu.mx

### **RESUMEN**

La presente investigación pretende conocer las necesidades del programa de la Maestría en Tecnología Informática Educativa de la Universidad Autónoma de Zacatecas en lo que respecta al seguimiento de trayectorias estudiantiles de sus alumnos de la modalidad en línea, así como, los requisitos técnicos para desarrollar e implementar un programa conector (plugin) que permita realizar dicho seguimiento en tiempo real en la plataforma LMS Moodle en la que opera dicho programa.

**Palabras clave:** trayectoria, estudiantil, moodle, plugin, lms.

### **RESUMO**

Esta investigação visa compreender as necessidades do programa de Mestrado em Tecnologias da Informação Educativa da Universidade Autónoma de Zacatecas no que respeita ao acompanhamento das trajetórias dos seus alunos em linha, bem como os requisitos técnicos para o desenvolvimento e implementação de um programa plug-in que

permita este acompañamiento en tiempo real na plataforma Moodle LMS em que o programa opera.

**Palavras-chave:** trajetória, estudante, moodle, plugin, lms.

## 1 INTRODUCCIÓN

No cabe duda que las modalidades de educación a distancia llegaron para quedarse, cada vez más se insertan en las funciones medulares operativas de las instituciones educativas, para ello, dichas instituciones se apoyan en plataformas LMS por sus siglas en inglés Learning Management System (Sistema de Gestión de Aprendizaje), sin embargo, estos sistemas se han especializado en dar seguimiento en cuanto al uso de recursos y actividades por parte de los facilitadores y sus alumnos, por otro lado, el aspecto de seguimiento de trayectoria de los estudiantes en estas plataformas es muy básico o nulo en muchas de ellas, siendo otra función medular de las instituciones académicas, y el programa de la Maestría en Tecnología Informática Educativa (MTIE) de la Universidad Autónoma de Zacatecas (UAZ) no es la excepción, de esta problemática surge la necesidad de desarrollar un programa conector (plugin) que se adapte al LMS de la MTIE, el cual, dé solución al seguimiento de trayectorias estudiantiles en tiempo real, para ello, se establecerán las necesidades del programa educativo en lo que respecta a la trayectoria estudiantil en línea y las necesidades técnicas para desarrollar e implementar el módulo de trayectorias estudiantiles que funcionará dentro del LMS.

## 2 DESCRIPCIÓN DEL MÉTODO

Se llevó a cabo una investigación de tipo descriptiva documental, por medio de la cual, se establecen las necesidades procedimentales y de reportes que el programa de la Maestría en Tecnología Informática Educativa requiere para poder llevar a cabo y ofrecer el seguimiento de trayectoria estudiantil en tiempo real y de una manera automática para lograr dicha automatización. La presente investigación debe arrojar los requerimientos técnicos tanto de hardware y software para desarrollar e implementar un programa conector (plugin) de acuerdo al sitio oficial de documentación de Moodle (2020), es llamado así por su nombre en inglés y se define como un complemento que permite añadir características y funcionalidades al Moodle, se desarrolla utilizando el lenguaje PHP y la

API (Interfaz de Programación de Aplicaciones) de dicho LMS y este se pondrá a disposición de la comunidad de Moodle de manera libre bajo una licencia GNU.

El programa de la MTIE adoptó como plataforma de educación en línea al LMS Moodle, el cual se puede definir como el acrónimo de Modular Object Oriented Dynamic Learning Environment (Entorno Modular de Aprendizaje Dinámico Orientado a Objetos), sin embargo, en otras fuentes lo definen como una variante del verbo en inglés *moodle* que se refiere al proceso de deambular perezosamente a través en algo y realizar las cosas cuando se antoje hacerlas, (Martínez, 2008).

Cabe resaltar que Moodle ha sido ampliamente adoptado por diferentes instituciones por ser un software libre de fácil implementación y administración, sin dejar a un lado el aspecto importante de ser muy seguro y adaptable a las necesidades que dichas instituciones requieran, y es aquí donde este proyecto de investigación pretende impactar aprovechando el desarrollo de los plugins que permiten que el LMS Moodle se adecúe a las necesidades de una institución.

### **3 REQUERIMIENTOS TÉCNICOS DE EQUIPO Y DE DISEÑO PARA CREAR EL PLUGIN DE MOODLE**

Para desarrollar el plugin de seguimiento de trayectorias estudiantiles dentro del LMS Moodle del programa académico de la MTIE es necesario conocer previamente a su desarrollo los requerimientos técnicos de hardware y software y de diseño respecto a la API y funcionalidades de Moodle para poder adecuar el plugin a las necesidades de operatividad administrativa de dicho programa, para conocer las necesidades de información y qué proceso es necesario llevar a cabo para obtener la información deseada en tiempo real y que esté acorde a las necesidades del programa, a estos requerimientos de operación les llamaremos los requerimientos procedimentales, y por otro lado están, los requisitos técnicos donde se engloban los de diseño, software y hardware, sobre los cuales se diseñará, implementará y ejecutará el plugin que dará la solución al seguimiento estudiantil dentro del LMS Moodle.

Es recomendado tener una instalación personal de Moodle donde realizar las pruebas, para evitar fallas si estas se realizaran en un servidor de producción, una vez que se finaliza el desarrollo del plugin en dicho ambiente de pruebas y que ha pasado por una serie de test de funcionalidad y seguridad, se puede instalar en los servidores de producción. Para este ambiente de pruebas es necesario contar con un servidor web, que es un programa que implementan el protocolo HTTP por sus siglas en inglés de Protocolo

de Transferencia de HiperTexto, que se encarga de transferir las páginas web realizadas en un Lenguaje de Marcado de HiperTexto (Niño, 2011), un Sistema Gestor de Base de Datos Relacionales los cuales Sánchez (2004) los define como un software que tiene la capacidad de producir, manipular y gestionar bases de datos de tipo relacional, y que puede ser comprendido como un tipo de sistema operativo diseñado para controlar los accesos a los datos y como requisito final es contar con el PHP el cual es un lenguaje de interpretado, libre, para el desarrollo de aplicaciones que se ejecutan del lado del servidor, capaz de generar contenido dinámico para la web (Arias, 2013).

#### 4 REQUERIMIENTOS TÉCNICOS DE SOFTWARE

Existen los requisitos técnicos de software y hardware sobre los que se ejecuta tanto el LMS como el plugin a desarrollar, en lo que corresponde al software que el LMS necesita para funcionar de acuerdo al sitio oficial de Moodle y sus requerimientos para la versión 3.10, que es la más nueva al momento de realizar esta investigación, es necesario que se tengan instalados algunos de los siguientes requerimientos:

Requerimientos de Servidor.

- Servicio Web que se integre con PHP.
- PHP Versión 7.1.0 o superior.
- Extensión PHP Intl.

Requerimientos de bases de datos.

- PostgreSQL versión mínima 9.4
- MySQL versión mínima 5.6
- MariaDB versión mínima 5.5.31
- Microsoft SQL Server versión mínima 2008
- Oracle Database versión mínima 11.2

#### 5 REQUERIMIENTOS TÉCNICOS DE HARDWARE

De acuerdo a la documentación de LMS Moodle (2011) el número exacto de usuarios depende de la combinación hardware/software/red. Lo usual es que la cantidad de memoria instalada (RAM) sea el factor determinante, pero un procesador lo bastante rápido reducirá los tiempos de espera de carga de las páginas web de la plataforma LMS.

La regla es que para un servidor su *aproximado máximo de conexiones concurrentes* =  $RAM (GB) * 50$  y *el aproximado de usuarios que navegan la plataforma* = *aproximado máximo de conexiones concurrentes* \* 5.

Como ejemplo, si una institución educativa tiene un total de 500 computadoras en su campus y 100 conexiones concurrentes en cualquier momento, requerirá aproximadamente 2 GB de memoria RAM en un servidor para soportar ese número de usuarios concurrentes.

## 6 REQUERIMIENTOS PROCEDIMENTALES DEL PROGRAMA DE LA MTIE

El programa de la MTIE, recientemente ingresó al Padrón Nacional de Posgrados de Calidad (PNPC) de CONACyT en 2019, con la categoría de nueva creación, por lo que se han planeado una serie de mejoras en el programa y surge la idea de que los estudiantes puedan visualizar su trayectoria estudiantil de calificaciones en tiempo real, esto sin la necesidad de solicitar un kardex a departamento escolar, ya que una queja recurrente de los estudiantes es que no conocen con certeza sus calificaciones finales semestrales, mismas que quedan inaccesibles una vez que se reinician los cursos para dar cabida a los nuevos estudiantes, por lo que, al perder el acceso a dichas materias y no contar con un kardex inmediato, genera angustia y desconocimiento de cuáles materias se aprobaron o reprobaron para solicitarlas en el nuevo semestre, se requiere que el estudiante tenga acceso a sus calificaciones finales (ya sean de semestres pasados o del actual) en tiempo real en la misma plataforma Moodle.

Una vez que se tiene acceso al ambiente de pruebas del LMS Moodle, se puede comenzar a codificar el plugin, iniciando con una consulta SQL de prueba de lo que se desea extraer de la base de datos, en este caso para conocer las calificaciones de los estudiantes la consulta de prueba quedaría de la siguiente manera:

```
SELECT u.firstname AS "First" , u.lastname AS "Last" , u.firstname, u.lastname  
AS "Display_Name",  
cc.name AS "Category",  
CASE  
WHEN gi.itemtype = 'course'  
THEN c.fullname  
ELSE gi.itemname  
END AS "Item_Name",  
ROUND(gg.finalgrade,2) AS Grade,  
gi.timemodified AS TIME  
FROM mdl_course AS c  
JOIN mdl_context AS ctx ON c.id = ctx.instanceid
```

```
JOIN mdl_role_assignments AS ra ON ra.contextid = ctx.id
JOIN mdl_user AS u ON u.id = ra.userid
JOIN mdl_grade_grades AS gg ON gg.userid = u.id
JOIN mdl_grade_items AS gi ON gi.id = gg.itemid
JOIN mdl_course_categories AS cc ON cc.id = c.category
WHERE gi.courseid = c.id AND gi.itemtype = 'course'
ORDER BY lastname;
```

Lo que arrojará resultados como lo muestra la figura 1.

Figura 1. Resultados de la consulta SQL

```

MariaDB [bitnami_moodle]> SELECT u.id as 'id', u.firstname AS 'First', u.lastname AS 'Last', u.firstname, u.
lastname AS 'Display_Name', cc.name AS 'Category', CASE WHEN gi.itemtype = 'course' THEN c.fullnam
e ELSE gi.itemname END AS 'Item_Name', ROUND(gg.finalgrade,2) AS Grade, gi.timemodified AS TIME, FR
OM mdl_course AS c JOIN mdl_context AS ctx ON c.id = ctx.instanceid JOIN mdl_role_assignments AS ra ON ra.c
ontextid = ctx.id JOIN mdl_user AS u ON u.id = ra.userid JOIN mdl_grade_grades AS gg ON gg.userid = u.id J
OIN mdl_grade_items AS gi ON gi.id = gg.itemid JOIN mdl_course_categories AS cc ON cc.id = c.category WHE
RE gi.courseid = c.id AND gi.itemtype = 'course' ORDER BY lastname;
+----+-----+-----+-----+-----+-----+-----+-----+
| id | First | Last | firstname | Display_Name | Category | Item_Name | Grade | TIME |
+----+-----+-----+-----+-----+-----+-----+-----+
| 3 | alumno de | prueba | alumno de | prueba | Miscellaneous | curso de prueba | 8.50 | 1620311682 |
+----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.001 sec)
MariaDB [bitnami_moodle]>

```

## 7 ESTRUCTURA DE ARCHIVOS DEL PLUGIN

El plugin requiere de una serie de archivos y contenidos específicos para que funcione correctamente, y estos deben residir dentro de un folder que represente el contenido del plugin, para caso práctico y por tratarse de un prototipo inicial el folder que contendrá el contenido del plugin se llama “traye” por Trayectoria Estudiantil, una vez finalizado se pueden adecuar correctamente los nombres y debe residir en el folder grade/report de Moodle, por ejemplo: /var/www/html/moodle/grade/report/traye, esta ruta se debe adecuar a donde se tienen instalado el Moodle en los servidores, o bien, el equipo del ambiente de pruebas.

Dentro de este folder deben residir al menos los siguientes archivos y carpetas:

**traye** (folder)

**db** (folder)

access.php (archivo)

upgrade.php (archivo)

**lang** (folder)

**es** (folder)

gradereport\_traye.php (archivo)

**index.php** (archivo)

**lib.php** (archivo)

**version.php** (*archivo*)Contenido de **version.php**:

```
<?php
    $plugin->version = 2012122100;
    $plugin->requires = 2012122100;
    $plugin->component = 'gradereport_traye';
?>
```

Dentro de db existe otro archivo llamado **upgrade.php**, el cual por el momento esta vacío, pero se incluye por convenciones del funcionamiento de Moodle.

Contenido de **access.php**:

```
<?php
$capabilities = array('gradereport/traye:view' => array('riskbitmask' => ",
'captype' => 'read','contextlevel' => CONTEXT_COURSE,
'legacy' => array('student' => CAP_ALLOW,'teacher' => CAP_ALLOW,
'editingteacher' => CAP_ALLOW,'manager' => CAP_ALLOW)),
);
?>
```

Contenido de **gradereport\_traye.php**:

```
<?php
$string['modulename'] = 'Reporte de usuario';
$string['user:view'] = 'Ver reporte de calificaciones de trayectoria';
$string['pluginname'] = 'Trayectoria Estudiantil';
?>
```

Contenido de **index.php**:

```
<?php
include('.././../config.php');
require($CFG->libdir './gradelib.php');
require($CFG->dirroot.'/grade/lib.php');
require($CFG->dirroot.'/grade/report/traye/lib.php');
$courseid = required_param('id', PARAM_INT);// course id
$userid = optional_param('userid', $USER->id, PARAM_INT);
if (!$course = $DB->get_record('course', array('id' => $courseid))) {
```

```
print_error('invalidcourseid');}
$context = get_context_instance(CONTEXT_COURSE, $course->id);
$usercontext = get_context_instance(CONTEXT_USER, $user->id);
require_capability('gradereport/traye:view', $context);
print_grade_page_head($course->id, 'report', 'traye', get_string('pluginname',
'gradereport_traye') . $OUTPUT->help_icon('pluginname',
'gradereport_traye'));
$gpr = new grade_plugin_return(array('type'=>'report', 'plugin'=>'traye',
'courseid'=>$course->id, 'userid'=>$userid));
$report = new grade_report_traye($courseid, $gpr, $context);// Initialise the
grader report object
$report->show($userid); echo $OUTPUT->footer();
?>
```

Contenido de **lib.php**:

```
<?php
require_once($CFG->dirroot.'/grade/report/lib.php');
class grade_report_traye extends grade_report {
function __construct($courseid, $gpr, $context, $page=null) {
parent::__construct($courseid, $gpr, $context, $page);
$this->course_grade_item = grade_item::fetch_course_item($this->courseid);
}
function process_data($data){ }
function process_action($target, $action){ }
public function show($userid) {
global $DB, $CFG;
$sql3 ="SELECT c.fullname Course, u.firstname Firstname, u.lastname
Lastname, round(gh.finalgrade,2) calificacion, gi.itemname Item,
gh.timemodified timemod, cc.name categoria FROM mdl_user u
JOIN mdl_grade_grades_history gh ON u.id=gh.userid JOIN mdl_grade_items
gi ON gh.itemid=gi.id
JOIN mdl_course c ON gi.courseid=c.id JOIN mdl_course_categories AS cc ON
cc.id = c.category
WHERE u.id=".$userid." and gh.finalgrade IS NOT NULL and gi.itemname=\"
\" ORDER BY timemod asc;";
```



```

$resultado = $DB->get_records_sql($sql3, array ());
if(empty($resultado)){
echo "<br><h3>No cuentas con historial de calificaciones en este
momento.</h3><br>";
} else {echo "Categoria - Curso - Calificación Final<br><br>"; foreach
($resultado as $id =>$record){
echo $record->categoria." - ".$record->course." - ".$record-
>calificacion."<br>";}
}
}}
?>

```

En la figura 2 se puede apreciar los resultados arrojados por el plugin al elegir en el menú del curso la opción de calificaciones y una vez ahí se elige el espacio del plugin: **[[pluginname]]**, en ese momento se mostrará en la parte inferior en texto, un renglón con la leyenda: *Categoría – Curso – Calificación Final* y por debajo de el, aparecerá un renglón por cada curso que tenga el estudiante con calificación final, en este caso son dos cursos que pertenecen a la categoría de Miscellaneous y sus respectivos nombres de curso y calificaciones finales, tanto de cursos que se encuentra matriculado (enrolado) el estudiante actualmente, como de aquellos que ya no tiene acceso, lo que permite un seguimiento de calificaciones históricas que almacena Moodle en su tabla `mld_grade_grades_history`.

Figura 2. Calificaciones de cursos finales, en el apartado de calificaciones de Moodle



## **8 COMENTARIOS FINALES**

### **8.1 CONCLUSIONES**

Sin duda uno de los aspectos más importantes para desarrollar una nueva tecnología es el proceso de buscar y reunir documentación para comprender los conceptos teóricos y técnicos respecto al sistema sobre el cual se desea diseñar algo, en este caso, comprender los requisitos técnicos y de diseño del LMS Moodle, cuya interfaz de programación de aplicaciones (API) es muy específica, sin duda, es necesario tener otros conocimientos y habilidades, tales como, el lenguaje SQL, PHP y HTML para poder codificar siguiendo los lineamientos de la API de Moodle, gracias a dicha API, ahora se pueden agregar funcionalidades especiales que se requieren por parte de las instituciones que incorporan este tipo de LMS, y que no están incluidas, ahora los estudiantes del programa de la MTIE, podrán consultar su Kardex en tiempo real, desde su propia cuenta de la plataforma, lo cual, les facilitará el proceso de inscripción a las próximas materias, pues conocerán cuales tienen aprobadas y cuales reprobadas, para realizar la carga de ellas ante departamento escolar del programa.

## REFERENCIAS

Arias, M. A. (2013). Introducción a PHP. IT Campus Academy.

Moodle (2011). User site capacities. Recuperado de [https://docs.moodle.org/20/en/User\\_site\\_capacities](https://docs.moodle.org/20/en/User_site_capacities)

Moodle (2020). Tutorial. Recuperado de <https://docs.moodle.org/dev/Tutorial>

Niño, J. (2011). Servidores de aplicaciones web (Aplicaciones web). Editex.

Martínez, R. I. (2008). Moodle, la plataforma para la enseñanza y organización escolar. Recuperado de <http://hdl.handle.net/10810/6876>

Sánchez, J. (2004). Arquitectura del DBMS Oracle. Guía de iniciación.