

PLUGIN RPL QUE PERMITE LA REALIZACIÓN DE LABORATORIOS REMOTOS

DISEÑO Y DESARROLLO DE UN PLUGIN PARA LA PLATAFORMA “MOODLE” QUE  
PERMITE LA REALIZACIÓN DE LABORATORIOS REMOTOS.

HENRY CORTES ROMERO  
CÓDIGO: 2009103013

UNIVERSIDAD PEDAGÓGICA NACIONAL  
FACULTAD DE CIENCIA Y TECNOLOGÍA  
DEPARTAMENTO DE TECNOLOGÍA  
LICENCIATURA EN ELECTRÓNICA

BOGOTA D.C  
2016

PLUGIN RPL QUE PERMITE LA REALIZACIÓN DE LABORATORIOS REMOTOS

DISEÑO Y DESARROLLO DE UN PLUGIN PARA LA PLATAFORMA “MOODLE” QUE  
PERMITA LA REALIZACIÓN DE LABORATORIOS REMOTOS.

PRESENTADO POR:

HENRY CORTES ROMERO  
CÓDIGO: 2009103013

Trabajo de grado presentado para optar al título de licenciado en electrónica

Director:

Diego Fernando Quiroga

Lic. Electrónica

UNIVERSIDAD PEDAGÓGICA NACIONAL  
FACULTAD DE CIENCIA Y TECNOLOGÍA  
DEPARTAMENTO DE TECNOLOGÍA  
LICENCIATURA EN ELECTRÓNICA

BOGOTA D.C  
2016

PLUGIN RPL QUE PERMITE LA REALIZACIÓN DE LABORATORIOS REMOTOS

Nota de Aceptación:

---

---

---

---

---

---

---

---

Firma del Director

---

Firma del Jurado 1

---

Firma del Jurado2

Bogotá Mayo de 2016

## **AGRADECIMIENTOS**

*Agradezco a Dios por darme la fortaleza para culminar un logro más en mi vida personal. A mi madre por sus consejos y sus palabras que han aportado para crecer cada día más como persona, a mi familia, amigos y personas cercanas por su apoyo en el transcurso de mi formación profesional.*

*Agradezco a los docentes que con cada uno de sus aportes y dedicación me han guiado para hacer mejor mi labor y poder desarrollar este trabajo.*

***DEDICATORIAS***

*Dedico este logro a todas las personas que han estado siempre apoyándome en mi carrera universitaria. A los que con su voz de aliento siempre me hicieron creer y nunca desistir.*

*A los que en las dificultades siempre me brindaron su ayuda y tiempo.*

*A mi familia por esperar con paciencia y siempre creer en mí.*

*Henry Cortes Romero.*

## PLUGIN RPL QUE PERMITE LA REALIZACIÓN DE LABORATORIOS REMOTOS

<b>1. Información General</b>	
<b>Tipo de documento</b>	TRABAJO DE GRADO.
<b>Acceso al documento</b>	UNIVERSIDAD PEDAGÓGICA NACIONAL. BIBLIOTECA CENTRAL.
<b>Título del documento</b>	DISEÑO Y DESARROLLO DE UN PLUGIN PARA LA PLATAFORMA "MOODLE" QUE PERMITE LA REALIZACIÓN DE LABORATORIOS REMOTOS.
<b>Autor(es)</b>	CORTES ROMERO, HENRY.
<b>Director</b>	QUIROGA, DIEGO FERNANDO.
<b>Publicación</b>	Bogotá. Universidad Pedagógica Nacional. 2016, 74p.
<b>Unidad Patrocinante</b>	Universidad Pedagógica Nacional.
<b>Palabras Claves</b>	PLUGIN, LABORATORIO, REMOTO, COMUNICACIÓN (USB), TRASMISION, RECEPCION, TICS, MODULO, HERRAMIENTA, MOODLE.

<b>2. Descripción</b>
<p>En el proyecto de grado se trabajó el diseño y construcción de un plugin tipo actividad para la plataforma educativa Moodle, que permita la realización de laboratorios de forma remota. Este plugin llamado RPL (Remote Practices Laboratory) es una herramienta para el desarrollo, construcción y manipulación de laboratorios remotos, cuyo objetivo es generar un ambiente de aprendizaje E-learning.</p> <p>Esta herramienta fue desarrollada por medio de un lenguaje de programación PHP, HTML y NODE JS que se incorpora a la plataforma educativa Moodle. Tiene como propósito facilitar los espacios de los laboratorios remotos en la Universidad, promoviendo el avance de las tics en la educación y de esta forma crear en la comunidad académica espacios de intercambio de conocimientos.</p>

<b>3. Fuentes</b>
<p>Almenara C, Barroso Osuna J, Romero Tena R, Llorente Cejudo M, Román Gravan P. (2007). Definición de Nuevas Tecnologías en línea OCW de la Universidad de Sevilla, Facultad de Ciencias de la Educación. España.</p> <p>Cobo, Romaní JC. (2009). El concepto de tecnologías de la información. Benchmarking sobre las definiciones de las TIC en la sociedad del conocimiento.</p>

Development: lib/formslib.php. (2015). Form Definition [http://docs.moodle.org/en/Development:lib/formslib.php\\_Form\\_Definition](http://docs.moodle.org/en/Development:lib/formslib.php_Form_Definition).

Development: XMLDB. (2015). Defining an XML structure, [http://docs.moodle.org/en/Development:XMLDB\\_defining\\_an\\_XML\\_structure](http://docs.moodle.org/en/Development:XMLDB_defining_an_XML_structure).

Herrera, N. (2008). Incorporando las TICS en el Aula ¿Por dónde empezar? Revista eeducador. Pag.11-14.

Herrera, N. (2008). Incorporando las TICS en el Aula ¿Por dónde empezar? Revista eeducador. Pag.11-14.

Laboratorio de automática vía internet que llamaron (LAVI). (2005). Revista iberoamericana de automática e informática industrial, ISSN: 1697-7912. Vol. 2, Numero 2, Abril 2005. Pág. 30-35.

Manuales de Moodle. (2015). [http://docs.moodle.org/es/Manual\\_de\\_Estilo\\_de\\_C%C3%B3digo](http://docs.moodle.org/es/Manual_de_Estilo_de_C%C3%B3digo).

Herrera, N. (2008). Incorporando las TICS en el Aula ¿Por dónde empezar? Revista eeducador. Pag.11-14.

Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura. UNESCO. (2004). Las tecnologías de la información y la comunicación en la formación docente.

Reyes, M. (2002). MOODLE, UNA PLATAFORMA FORMATIVA CON GRAN PROYECCIÓN EN LOS NUEVOS MODELOS DE ENSEÑANZA.

Sánchez, J. (2005). Plataformas tecnológicas para el entorno educativo. Acción pedagógica(Nº14).recuperado de <http://dialnet.unirioja.es/>.

#### 4. Contenidos

A continuación se observa los contenidos del presente trabajo de grado, enunciados por capítulos.

CAPITULO I: PLANTEAMIENTO DEL PROBLEMA

Se expone el problema y se inicia la investigación para encontrar las posibles soluciones.

CAPITULO II: MARCO TEORICO

Fundamento de la investigación, que conlleva al desarrollo del plugin.

#### CAPITULO III: ANALISIS DE REQUERIMIENTOS

Se analizan los requerimientos que se necesitan para la elaboración del plugin, donde se observan los requerimientos previos de la creación de un módulo para la plataforma Moodle; y así ir evaluando y analizando la información obtenida para su adecuada construcción.

#### CAPITULO IV: DISEÑO Y CONSTRUCCION

Se observan las necesidades del usuario frente al plugin para desarrollar casos de uso que definen los requerimientos que contendrá el módulo, definiendo una interfaz acorde para el desarrollo y manipulación de los laboratorios por parte de los usuarios, con una adecuada trasmisión y recepción de video en tiempo real. Después se realizaran las tareas propuestas en las etapas anteriores, donde se mostrará de manera detallada paso a paso la construcción de este módulo.

#### CAPITULO V: PRUEBAS

Se realizarán unas pruebas piloto, que verifiquen el funcionamiento correcto del mismo; donde se analiza que cumpla con las normas requeridas desde el comienzo del plugin y así satisfacer las necesidades del usuario que lo va a usar.

#### CAPITULO VI: CONCLUSIONES

Por último se concluye sobre los obstáculos y fortalezas al desarrollo, adicionando las recomendaciones de uso del plugin y las posibles adaptaciones que se pueda realizar a futuro.

## 5. Metodología

### METODOLOGIA EN CASCADA:

El siguiente proyecto se basa en la metodología en cascada siendo fundamental en el desarrollo de software, donde se evidencia un proceso consecutivo el cual proporciona una coherencia y orden en la etapa del desarrollo del plugin.

La primera etapa es la recopilación de información, donde se recolectan y establecen las necesidades del usuario. La segunda etapa es el diseño y desarrollo del plugin, en la que se realizan casos de usos para determinar los requerimientos del usuario y así realizar las tareas correspondientes para su desarrollo. Por último las pruebas pilotos para verificar los funcionalidad del producto.

## 6. Conclusiones

## PLUGIN RPL QUE PERMITE LA REALIZACIÓN DE LABORATORIOS REMOTOS

- El plugin RPL proporciona a docentes y estudiantes una herramienta para la interacción y elaboración de laboratorios remotos en tiempo real. Esta herramienta permite a la licenciatura en electrónica innovar en la metodología de práctica para realizar los laboratorios de diferentes asignaturas.
- Esta herramienta permite explorar otros niveles educativos, a través de la plataforma educativa Moodle ya que cuenta con una variedad de plugin prácticos, que aumenta la motivación e iniciativa de investigación por parte de los estudiantes.
- Esta herramienta se convierte en un ambiente de aprendizaje virtual permitiendo a los estudiantes aprender y acentuar sus conocimientos desde cualquier punto con acceso a internet.
- Node.js fue un elemento fundamental para la definición del protocolo de comunicación a través de la librería SerialPort para la recepción y trasmisión de datos en tiempo real.
- El modo desarrollador de Moodle permite identificar las líneas de error que se presentaron durante el desarrollo del plugin facilitando la modificación y culminación de este.

<b>Elaborado por:</b>	HENRY CORTES ROMERO.
<b>Revisado por:</b>	DIEGO FERNANDO QUIROGA.

<b>Fecha de elaboración del Resumen:</b>	09	JUN	2016
--	----	-----	------

## TABLA DE CONTENIDIO

INTRODUCCIÓN .....	17
<b>CAPÍTULO I</b> .....	19
1.1 Planteamiento del problema.....	19
1.2 Objetivos .....	21
1.2.1 Objetivo general.....	21
1.2.2 Objetivos específicos.....	21
1.3 Justificación.....	22
MARCO TEÓRICO.....	24
1.4 Antecedentes .....	24
NIVEL NACIONAL.....	24
1.4.1 Universidad pedagógica nacional:.....	24
1.4.2 Universidad pontificia Bolivariana (Medellín): .....	24
1.4.3 Universidad EAFIT .....	25
1.4.4 Universidad Nacional de Colombia (Medellín): .....	25
NIVEL INTERNACIONAL .....	26
1.4.5 Universidad de León (España): .....	26
1.4.6 Universidad católica del Perú: .....	26
1.4.7 Universidad Europea de Madrid:.....	26
1.5 Metodología.....	28
<b>CAPÍTULO II</b> .....	31
2 Referencias conceptuales.....	31
2.1 Moodle.....	31
2.2 Ambiente virtual de aprendizaje (AVA).....	32

2.3 Las tics en la educación.....	33
2.4 Caso de uso.....	34
2.5 Plugin. ....	34
2.6 Laboratorios remotos.....	35
2.7 Node.JS .....	36
2.7.1 SocketIO .....	36
2.7.2 WebSocket.....	37
2.8 Programación orientada a objetos .....	38
2.9 Modelo Cliente – Servidor .....	38
2.9.1 Servidores HTTP (Hypertext Transfer Protocol) .....	39
2.9.2. Servidores FTP (Protocolo de transferencia de archivos) .....	39
2.9.3 Servidores DNS (Domain Name Server).....	40
<b>CAPÍTULO III.....</b>	<b>41</b>
3. Análisis de requerimientos .....	41
<b>3.1 ETAPA PLUGIN RPL .....</b>	<b>41</b>
3.1.1 mod_form.php:.....	41
3.1.2 version.php: .....	41
3.1.2.1 \$plugin->component.....	42
3.1.2.2 \$plugin->versión.....	42
3.1.2.3. \$plugin->requires: .....	42
3.1.2.4 \$plugin->cron: .....	42
3.1.2.5. \$plugin->maturity:.....	42
3.1.2.6. \$plugin->release: .....	43
3.1.3 Lang.php:.....	43

## PLUGIN RPL QUE PERMITE LA REALIZACIÓN DE LABORATORIOS REMOTOS

3.1.4 Carpeta db: .....	43
3.1.4.1. access.php: .....	43
3.1.4.2. install.xml: .....	44
3.1.4.3. upgrade.php: .....	44
3.1.4.4 events.php: .....	44
3.1.4.5 uninstall.php: .....	44
3.1.5. index.php: .....	44
3.1.6. view.php: .....	44
3.1.7. lib.php:.....	45
3.1.8. settings.php (opcional) .....	45
3.1.9. Carpeta pix .....	46
3.2 TRANSMISION Y RECEPCION DE VIDEO EN TIEMPO REAL .....	46
3.2.1 Carpeta public.....	47
3.2.2 app.js:.....	47
3.3 COMUNICACIÓN CON ARDUINO .....	48
3.3.1 Server.js .....	48
3.3.2 Route.js.....	48
3.3.3 requestHandlers.js.....	48
3.3.4 index.js:.....	48
<b>CAPÍTULO IV</b> .....	<b>50</b>
4. Diseño y construcción .....	50
<b>4.1 ETAPA PLUGIN</b> .....	<b>50</b>
4.1.1 Diseño estructural del Plugin RPL (Prácticas de laboratorios remotos). .....	50
4.1.2. Lista de requerimientos: .....	50

## PLUGIN RPL QUE PERMITE LA REALIZACIÓN DE LABORATORIOS REMOTOS

4.1.3 Descripción caso de uso. ....	51
4.1.4 Diagrama de caso de uso .....	56
4.1.5 Ventana de bienvenida a Moodle: .....	56
4.1.6 Instalación del plugin .....	57
4.1.6.1 Carpeta db:.....	57
4.1.6.1.1 acces.php: .....	58
4.1.6.1.2 install.php: .....	58
4.1.6.1.3 upgrade.php: .....	58
4.1.6.1.4 Version.php: .....	58
4.1.7 Añadir un plugin actividad RPL: .....	59
4.1.7.1 lib.php – Locallib.php:.....	60
4.4.7.2 lang: .....	60
4.1.8 Editando una actividad en RPL: .....	61
4.1.9 Actividad creada:.....	62
<b>4.2 TRANSMISION Y RECEPCION DE VIDEO EN TIEMPO REAL.....</b>	<b>62</b>
4.2.1 package.json: .....	63
4.2.2 app.js:.....	63
4.2.3 index.html: .....	63
4.2.3.1 emitir.html: .....	64
4.2.3.2 visualizar.html: .....	65
<b>4.3 COMUNICACIÓN CON EL ARDUINO .....</b>	<b>65</b>
4.3.1 Diseño del laboratorio: .....	66
4.3.2 Código de arduino: .....	67
4.3.3. Comunicación Node js con arduino: .....	67

4.3.3.1 index.js:.....	67
4.3.3.2 server.js:.....	68
4.3.3.3 route.js: .....	68
4.3.3.4 requestHandler.js: .....	68
<b>CAPITULO V</b> .....	70
5. Pruebas y resultados .....	70
<b>CAPITULO VI</b> .....	75
6.1 CONCLUSIONES .....	75
6.2 RECOMENDACIONES Y TRABAJOS A FUTURO .....	76
6.3 REFERENCIAS BIBLIOGRAFICAS .....	77

## INDICE DE FIGURAS

Figura 1. LMS (Learning Managment System).....	22
Figura 2. Metodología en cascada. ....	28
Figura 3. Plataforma educativa Moodle.....	32
Figura 4. Estructura de laboratorio remoto. ....	36
Figura 5. Conexión desde el navegador al programa nodo.....	37
Figura 6. Modelo cliente servidor.....	39
Figura 7. Definición de DNS .....	40
Figura 8. Icono del plugin RPL.....	46
Figura 9. Esquema entre Servidor, Cliente y Laboratorio. ....	47
Figura 10. Diagrama de caso de uso .....	56

## PLUGIN RPL QUE PERMITE LA REALIZACIÓN DE LABORATORIOS REMOTOS

Figura 11. Ventana de Bienvenida de Moodle.....	57
Figura 12. Validación de requisitos plugin RPL.....	59
Figura 13. Añadir una actividad o recurso.....	60
Figura 14. Agregar una actividad RPL. ....	61
Figura 15. Actividad creada dentro del plugin RPL .....	62
Figura 16. Página web index.html .....	63
Figura 17. Emisión de video en tiempo real .....	64
Figura 18. Visualización en tiempo real. ....	65
Figura 19. Laboratorio leds.....	66
Figura 20. Laboratorio físico. ....	67
Figura 21. Interfaz de laboratorio. ....	68
Figura 22. Unión de las tres fases. ....	69
Figura 23. Agregando la actividad RPL .....	70
Figura 24. Registrar laboratorio en RPL.....	71
Figura 25. Laboratorio registrado en el curso.....	71
Figura 26. Lado del servidor.....	72
Figura 27. Página de inicio de Estudiante.....	72
Figura 28. Actividades creadas por el docente. ....	73
Figura 29 Interfaz de laboratorio remoto .....	73
Figura 30. Ordenador parte Cliente. ....	74

**INDICE DE TABLAS**

Tabla 1. Lista de requerimientos para RPL. ....	51
Tabla 2. Caso de uso 1 .....	52
Tabla 3. Caso de uso 2. ....	55

## INTRODUCCIÓN

La licenciatura en electrónica de la Universidad Pedagógica Nacional en sus áreas disciplinares, utiliza métodos de trabajo enfocados en la teoría que es complementada con la práctica, donde los estudiantes a través de la implementación y manipulación de elementos de programación, logran comprobar los conceptos teóricos. Este proyecto RPL sus siglas en inglés (Prácticas de laboratorio remoto) es una herramienta para el desarrollo, construcción y manipulación de laboratorios remotos, cuyo objetivo es generar un ambiente de aprendizaje E-learning<sup>1</sup> como lo expone keegan (2012) “el arte de la educación a distancia apoyadas por las Tics”.

Esta herramienta fue desarrollada por medio de un lenguaje de programación PHP, HTML y NODE JS que se incorpora a Moodle. La construcción del plugin<sup>2</sup> tiene como propósito facilitar los espacios de los laboratorios remotos en la Universidad, promoviendo el avance de las tics en la educación y de esta forma crear en la comunidad académica espacios de intercambio de conocimientos.

Para el desarrollo y construcción se inició con el diseño de casos de uso, donde se muestra los requerimientos más importantes al momento de manipular el laboratorio por parte del estudiante. Este plugin le permitirá al estudiante interactuar con elementos reales de forma remota, a partir de un laboratorio que el docente haya construido en RPL de la plataforma educativa Moodle.

---

<sup>1</sup> Sarmiento (2010) “E-learning consiste en la educación y capacitación a través de Internet. Este tipo de enseñanza online permite la interacción del usuario con el material mediante la utilización de diversas herramientas informáticas”.

<sup>2</sup>Según la compañía la página web Definicion.De (2016) “Un plugin es aquella aplicación que, en un programa informático, añade una funcionalidad adicional o una nueva característica al software”.

Este documento presenta cada una de las etapas que se realizaron para el desarrollo e implementación del plugin.

A continuación se expone cada uno de los 6 capítulos que componen este proyecto:

Capítulo I: “Planteamiento del problema”, se expone el problema y se inicia la investigación para encontrar las posibles soluciones.

Capítulo II: “Marco teórico”, fundamento de la investigación, que conlleva al desarrollo del plugin.

Capítulo III: “Análisis de requerimientos”, donde se referencia los requisitos que debe tener el plugin para su construcción.

Capítulo IV: “Diseño y construcción”, se describe como fue el proceso de desarrollo del plugin.

Capítulo V: “Pruebas”, En esta etapa se realizan unas pruebas piloto para comprobar la funcionalidad y factibilidad del plugin.

Capítulo VI: “Conclusiones” Por último se concluye sobre los obstáculos y fortalezas al desarrollo, adicionando las recomendaciones de uso del plugin y las posibles adaptaciones que se pueda realizar a futuro.

## CAPÍTULO I

### *1.1 Planteamiento del problema*

La licenciatura en electrónica presenta en sus áreas disciplinares, métodos de trabajo dentro del aula de clase, enfocado en la parte teórica que se complementa con una parte práctica; donde estudiantes y docentes comprueban los conceptos teóricos a partir de la implementación y manipulación de elementos de programación. Es por esto que se propone un proyecto que involucra la forma de realizar los laboratorios en el aula de clase, siendo una herramienta de apoyo para desarrollar de forma remota las prácticas, partiendo de un plugin de uso libre que permita la elaboración y manipulación de un hardware.

Este proyecto se elaborará a partir de un lenguaje estructurado, que busca establecer un punto en común para el desarrollo de prácticas educativas dentro y fuera del aula de clase. Al postular este proyecto primero se define el término de laboratorio remoto, tomando la definición de Villada (2013) quien afirma que “un laboratorio remoto es la manipulación de recursos de forma remota a través de Internet, haciendo uso de Webcams, micrófonos, hardware de adquisición de datos, etc. Permitiendo que estos recursos puedan adquirir información que puede ser enviada a través de redes informáticas de tipo Internet y reproducida de forma remota”.

El desarrollo de esta herramienta de apoyo para el docente y el estudiante, es un recurso útil que tiene su base y estructura en el diseño del módulo, integrando elementos de programación como Arduino, Raspberry pi, etc. Con base en esto el diseño de la estructura del módulo debe responder a las necesidades que requiera el usuario, asegurando una interfaz que se adapte a lo que pretenda desarrollar el laboratorio y así el docente tenga una herramienta que pueda evaluar un proceso con

## PLUGIN RPL QUE PERMITE LA REALIZACIÓN DE LABORATORIOS REMOTOS

los estudiantes de manera virtual. De esta manera se debe determinar el protocolo de comunicación que permita interactuar la capa física con el plugin que se va a desarrollar.

Su desarrollo se hará a partir de la plataforma educativa “Moodle”, que está diseñada por un sistema a base de “plugins” y un lenguaje de programación en PHP el cual proveen una funcionalidad específica para el adecuado desarrollo del módulo e integra tecnologías de información y la comunicación (Tic), como una herramienta que permite el mejoramiento de las actividades propuestas dentro del aula, por su fácil acceso desde cualquier punto con conexión a internet para ejecutar los laboratorios remotos.

## **1.2 Objetivos**

### ***1.2.1 Objetivo general.***

- Diseñar y desarrollar un plugin para la plataforma “Moodle” que permita la interacción y realización de laboratorios remotos.

### ***1.2.2 Objetivos específicos.***

- Diseño del Módulo a partir de la plataforma educativa Moodle.
- Diseñar y determinar el protocolo de comunicación, para interactuar con la capa física del laboratorio.
- Realizar prueba piloto con algún prototipo, para comprobando la funcionalidad y efectividad del Plugin.



PLUGIN RPL QUE PERMITE LA REALIZACIÓN DE LABORATORIOS REMOTOS

*<http://www.smdwebtech.com.my/lms/>*

En la universidad Pedagógica Nacional no existen plugins que estén direccionados a la práctica de las asignaturas del programa de Licenciatura en Electrónica, es por esto que se generará un gran aporte creando uno propio para la plataforma Moodle.

## MARCO TEÓRICO

### 1.4 Antecedentes

A continuación se expone el análisis realizado para los aportes teóricos de algunos proyectos a nivel nacional e internacional sobre laboratorios remotos, donde se evidencia las experiencias al momento de desarrollar nuevas tecnologías y aplicaciones en beneficio de la sociedad.

#### NIVEL NACIONAL

##### 1.4.1 Universidad pedagógica nacional:

*Diseño e implementación de un laboratorio remoto para la asignatura de control 1.*

Acosta en el año 2013 uso dos plantas de temperatura y barra bola, donde implemento un laboratorio remoto enfocado para la asignatura de Sistemas de Control 1. Se apoya en microcontroladores PSoC que integran tecnología digital y analógica. Este laboratorio fue un gran aporte para la licenciatura ya que fomento el uso de microcontroladores manejado en varias asignaturas, el cual plantean múltiples laboratorios dentro del aula para constatar su parte teórica.

##### 1.4.2 Universidad pontificia Bolivariana (Medellín):

*LAVI (Laboratorio de automática vía internet)*

Zuluaga, Sánchez y Rodríguez en el año 2013 elaboraron un laboratorio de automática vía internet el cual llamaron (LAVI), donde consta de una plataforma de software que permite realizar pruebas remotamente sobre una planta de control lineal. El proceso de petición y respuesta de usuario es

ayudado por programas escritos en Perl, Cuenta con una base de datos MySQL para controlar el acceso de estudiantes e investigadores.

### **1.4.3 Universidad EAFIT**

*Plataforma web para acceso remoto a instrumentación física avanzada.*

Montoya y Hernández en el año 2010 enfocaron un estudio a determinar las tendencias a futuro de los laboratorios remotos a partir del concepto que se tiene en la actualidad de los laboratorios remotos, mostrando las ventajas y desventajas que posee una herramienta educativa. El autor afirmar que los laboratorio remotos a futuro se dirigen hacia la integración de los mismos con Moodle (Plataforma educativa de aprendizaje).

### **1.4.4 Universidad Nacional de Colombia (Medellín):**

*Implementación de la plataforma Moodle en la institución Educativa Luis López de Mesa*

Grisales en el año 2013, En la sede de la Universidad Nacional de Colombia sede Medellín desarrollo un proyecto de grado donde se implementa la plataforma educativa Moodle, con el fin de promover| cursos virtuales como apoyo en instituciones educativas. En esta misma sede se cuenta con un laboratorio remoto de física general, donde los equipos que se utilizan para este laboratorio Consta de un equipo completo de PASCO<sup>3</sup> de física general; en la que se desarrolla una programación basada en JAVA, PHP y LabView.

---

<sup>3</sup> PASCO es una empresa especializada en productos para enseñar ciencias, su página web es: <http://www.pasco.com>

## **NIVEL INTERNACIONAL:**

### **1.4.5 Universidad de León (España):**

*Laboratorio remoto de automática.*

El laboratorio remoto de automática de la universidad de león de España cubre principalmente las necesidades de las asignaturas impartidas en el área de automática, para esto cuenta con varios módulos como lo son: (Control de nivel y planta piloto industrial). El manejo del hardware se puede hacer en tiempo real o de manera diferida, con la condición de que cada módulo debe ser usado por un único usuario a la vez; pero también cuenta con la opción de activación para varios usuarios lo que es útil en sesiones teóricas.

### **1.4.6 Universidad católica del Perú:**

*Diseño e implementación de una interfaz multimedia interactiva basada en un servidor de video para un laboratorio remoto virtual*

Ramos en el año 2012 presenta una tesis que consiste en el desarrollo de una interfaz multimedia interactiva basada en un servidor de video para implementar un laboratorio remoto virtual, Esta interfaz le permite visualizar los equipos que se encuentren remotamente y la interacción entre usuarios conectados a una sesión de laboratorio.

### **1.4.7 Universidad Europea de Madrid:**

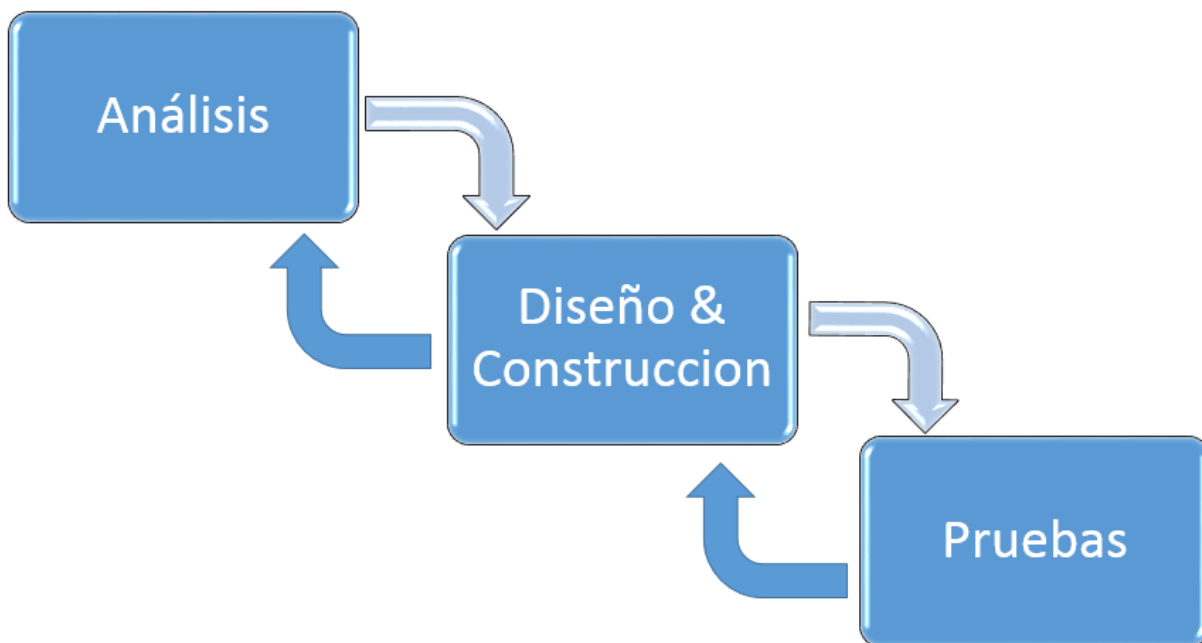
*Implementación de un laboratorio remoto, diseñado como herramienta de apoyo para la enseñanza de ingeniería de control de la Universidad Europea de España.*

## PLUGIN RPL QUE PERMITE LA REALIZACIÓN DE LABORATORIOS REMOTOS

Mejía y Hernández en el año 2011 implementaron un laboratorio remoto diseñado como una herramienta de apoyo para la enseñanza de la ingeniería de control de la Universidad Europea de España. Este sistema realiza pruebas piloto en lazo abierto y cerrado con algunos tipos de control que se encuentran predefinidos. Uno de los beneficios más importantes de este laboratorio es que posibilita un acceso concurrente y que lo puedan utilizar máximo cuatro personas permitiendo generar automáticamente los informes de los experimentos.

### 1.5 Metodología

El siguiente proyecto se basa en la metodología en cascada siendo fundamental en el desarrollo de software, donde se evidencia un proceso consecutivo el cual proporciona una coherencia y orden en la etapa del desarrollo del plugin. La primera etapa es la recopilación de información, donde se recolectan y establecen las necesidades del usuario. La segunda etapa es el diseño y desarrollo del plugin, en la que se realizan casos de usos para determinar los requerimientos del usuario y así realizar las tareas correspondientes para su desarrollo. Por último las pruebas pilotos para verificar los funcionalidad del producto.



**Figura 2.** Metodología en cascada.

**Nota Fuente:** (2016). Creado por el autor.

A continuación se describen las fases de este proyecto teniendo en cuenta el modelo en cascada:

1. **Análisis:** En esta etapa se analizan los requerimientos que se necesitan para la elaboración del plugin, donde se observan los requerimientos previos de la creación de un módulo para la plataforma Moodle; y así ir evaluando y analizando la información obtenida para su adecuada construcción.
  2. **Diseño y construcción:** En esta etapa se observan las necesidades del usuario frente al plugin para desarrollar casos de uso que definen los requerimientos que contendrá el módulo, definiendo una interfaz acorde para el desarrollo y manipulación de los laboratorios por parte de los usuarios, con una adecuada transmisión y recepción de video en tiempo real. Después se realizarán las tareas propuestas en las etapas anteriores, donde se mostrará de manera detallada paso a paso la construcción de este módulo.
- **Plugin (RPL):** Los usuarios se encontrarán con un plugin actividad llamado RPL, el que contendrá 4 roles diferentes para su uso (Administrador, Docente, Estudiante y Usuario). A partir de cada rol se asignaron permisos que son fundamentales en el módulo como se muestra a continuación. El administrador contendrá todos los permisos activos desde consultar hasta crear nuevos laboratorios. El docente es la persona encargada de diseñar y construir los laboratorios para el curso; autorizando el ingreso de los estudiantes quienes deberán registrarse al curso y manipular los laboratorios de manera remota. Por último el usuario es un invitado que solo tiene los permisos de ver lo que contiene el curso.
  - **Recepción y transmisión de video en tiempo real:** Al ejecutar el laboratorio remoto se encenderá la cámara web, que está ubicada en la parte del servidor enfocando la planta a

manipular. Esto con el fin de mostrar en tiempo real el comportamiento de la planta al prender y apagar un led remotamente cuando el cliente accione algún comando.

- **Comunicación con Arduino:** Cuando el docente ha creado una actividad nueva dentro del curso está incluyendo un laboratorio remoto que el estudiante podrá manipular, puede ser una planta, un circuito, etc. pero que se puede manejar en tiempo real. Para este caso se realizó un circuito sencillo programado en una tarjeta de programación arduino el cual ira variando la intensidad de luz de dos leds a medida que el estudiante manipule este laboratorio detrás de su ordenar. Este laboratorio es opcional ya que solo se realiza para probar el plugin.

3. **Pruebas:** En la fase de evaluación del plugin se realizarán unas pruebas piloto, que verifiquen el funcionamiento correcto del mismo; donde se analiza que cumpla con las normas requeridas desde el comienzo del plugin y así satisfacer las necesidades del usuario que lo va a usar.

## CAPÍTULO II

### 2 Referencias conceptuales.

#### 2.1 Moodle.

Según la página virtual de Moodle.org “Moodle es un Sistema de Gestión de Cursos de Código Abierto (*Open Source Course Management System, CMS*)<sup>4</sup>, conocido también como Sistema de Gestión del Aprendizaje (*Learning Management System, LMS*)<sup>5</sup> o como Entorno de Aprendizaje Virtual (*Virtual Learning Environment, VLE*)<sup>6</sup>. Es una aplicación web gratuita que los educadores pueden utilizar para crear sitios de aprendizaje efectivo en línea”

En resumen Moodle es un completo sistema de administración de cursos, su nombre es el acrónimo de Modular Object – Oriented Dynamic Learning Environment (Entorno de Aprendizaje Dinámico Orientado a Objetos y Modular). Es un Ambiente Educativo Virtual de distribución libre, que ayuda a los profesores y demás interesados en la educación virtual a crear comunidades de aprendizaje en línea, originando tipos de plataformas tecnológicas conocidas como LMS (Learning Management System). Este sistema fue creado por Martin Dougiamas , quien basó su diseño en las ideas pedagógicas que afirman que “el conocimiento se construye en la mente del estudiante a través del aprendizaje colaborativo en vez de la continuar con educación tradicional que transmite literal y sin cambios de los libros”.

4. Cañellas 2014” (Sistema de Gestión de Contenido): Permite la creación y administración de los contenidos de una página Web, principalmente, de forma automática”.

5. Cañellas 2014 “(Sistema de Gestión de Aprendizaje) Automatiza la administración de acciones de formación. Son variadas las funcionalidades de un LMS: registra a todos los actores que intervienen en el acto de aprendizaje (alumnos, profesores, administradores, etc.), organiza los diferentes cursos en un catálogo, almacena datos sobre los usuarios, realiza un seguimiento del aprendizaje”.

6. Salinas. (2011) “(Entorno virtual de aprendizaje) s un espacio educativo alojado en la web, conformado por un conjunto de herramientas informáticas o sistema de software que posibilitan la interacción didáctica”.



**Figura 3.** Plataforma educativa Moodle

**Nota Fuente:** *El nombre Moodle™ es una marca comercial registrada de Moodle Trust.*

*<https://moodle.org/logo/>*

## *2.2 Ambiente virtual de aprendizaje (AVA).*

En los últimos tiempos se ha venido incrementado el uso de las tecnologías y el masivo uso del internet como una herramienta que facilita y mejora los nuevos ambientes de aprendizaje, conocidos como AVA, ambientes virtuales de aprendizaje. Estos sistemas funcionan generalmente con un servidor, que ayuda al ingreso de los estudiantes incluyendo entre sus componentes por lo general plantillas para elaborar contenidos, foros, charlas, cuestionarios y ejercicios tipo opción-múltiple, verdadero/falso y respuestas de una palabra, las cuales son completadas por los profesores para su posterior publicación, de manera que puedan ser utilizados por los estudiantes.

Por ello se puede decir entonces que en la educación actual se debe fomentar este tipo de ambientes de aprendizaje de forma interactiva, donde el docente tiene un papel como mediador y facilitador, comprometido realmente con el aprendizaje de sus estudiantes, fomentando el desarrollo de habilidades y modos de trabajo innovadores, en los cuales puedan utilizar tecnologías de vanguardia, materiales didácticos, recursos de información y contenidos en un ambiente de aprendizaje ameno y motivador en beneficio de la educación.

En la actualidad la educación a distancia, asimilada como educación basada en Web o e-learning, es todo un mundo de muchísimos centros de enseñanza a distancia tanto públicos como privados,

orientados a todos los niveles y tipos de educación: primaria, secundaria, superior. como lo expone Romero (2005) *“Cada vez son más los sistemas basados en Web, la tecnología más utilizada para la educación a distancia, debido a la facilidad de utilización y disponibilidad de las herramientas para navegar por el Web y la facilidad del desarrollo y mantenimiento de los recursos Web”*.

Es una alternativa que permite pensar en un sin fin de posibilidades que se puede hacer en beneficio de la educación en nuestro país en donde todo gira en torno a los estudiantes que deben involucrarse con este tipo de tecnologías educativas e innovadoras que cada vez más hacen parte de la cotidianidad y avanzan a una velocidad impresionante.

Estos sistemas de enseñanza virtual se apoyan en una plataforma basada en web que pueden ser múltiples sistemas comerciales existentes como Web- CT, Virtual-U, Top Class, etc. O de libre distribución como: ATutor, ILIAS, Moodle etc. Estos proporcionan servicios útiles para la enseñanza a distancia como son herramientas para la comunicación sincrónica y asincrónica, para la gestión de materiales de aprendizaje y para la gestión, seguimiento y evaluación de los estudiantes.

### *2.3 Las tics en la educación*

La UNESCO en el 2016 en su documento *“Estándares de competencias en Tecnologías de Información en Educación, para docentes”*, hace referencia sobre la utilización de los recursos tecnológicos en la labor docente, permitiendo a los estudiantes llegar a ser:

- Competentes para utilizar tecnologías de la información.
- Buscadores, analizadores y evaluadores de información.

- Solucionadores de problemas y tomadores de decisiones.
- Usuarios creativos y eficaces de herramientas de productividad.
- Comunicadores, colaboradores, publicadores y productores.
- Ciudadanos informados, responsables y capaces de contribuir a la sociedad.

De esta forma se logra desarrollar en los estudiantes sus potencialidades en el manejo de las TIC, esta labor debe ser desempeñada por el docente, para generar estrategias, establecer actividades, crear oportunidades y un entorno propicio que posibilite el aprendizaje aprovechando el uso de los diferentes recursos tecnológicos, tanto en el aula como por fuera de ella.

#### *2.4 Caso de uso*

Un caso de uso describe los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso. Los personajes o entidades que participarán en un caso de uso se denominan actores. El caso de uso para un proyecto de desarrollo de software es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema. Los diagramas de casos de uso se utilizan para ilustrar los requerimientos del sistema al mostrar cómo reacciona a eventos que se producen en su ámbito o en él mismo (Ceria, 2014).

#### *2.5 Plugin.*

Los plugins son extensiones creadas por usuarios, en el que se complementan funcionalidades y características a un programa base. Los plugin son aplicaciones adicionales que se ejecutan por una aplicación principal que en este caso es la plataforma de Moodle.

Moodle cuenta con más de 400 plugin gratuitos y de código abierto, los cuales pueden ser instalados y/o modificados y los caracteriza de la siguiente manera:

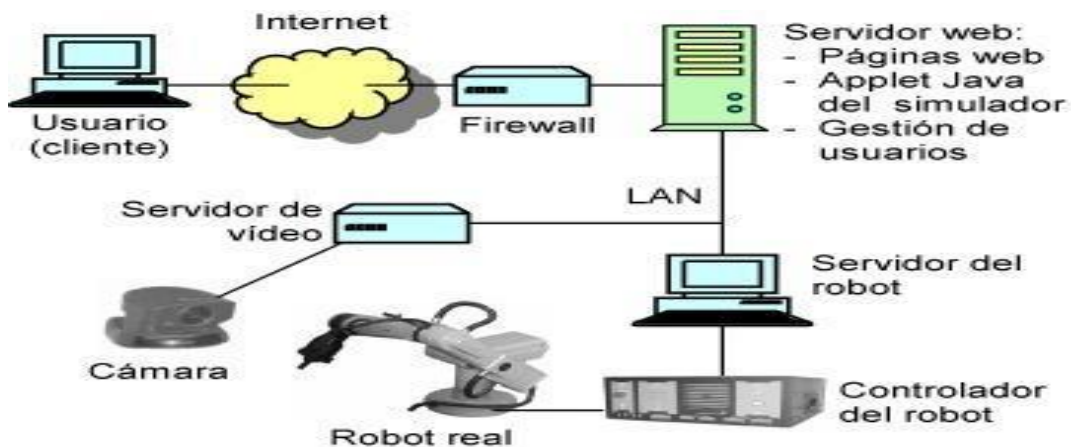
- Actividades
- Recursos
- Temas
- Bloques
- Filtros

### *2.6 Laboratorios remotos.*

La principal característica que diferencia a un laboratorio remoto de uno virtual es que detrás del laboratorio remoto hay hardware real. La persona que hace uso de ese laboratorio durante una sesión tiene el control sobre los recursos hardware involucrados en el laboratorio que está utilizando. Un laboratorio virtual, en cambio, emula el comportamiento del experimento mediante software. Utilizar un laboratorio remoto es por tanto una experiencia mucho más cercana a un uso real en un laboratorio presencial, por lo que es capaz de sustituir a éste sin afectar negativamente a la labor del usuario. En su contra tiene el coste, puesto que los recursos utilizados deben existir físicamente. Sin embargo, esta desventaja frente a los laboratorios virtuales es al contrario una ventaja en comparación con los laboratorios presenciales, y una de las grandes virtudes que hacen que la experimentación remota tenga sentido. El ahorro de costes se refleja en varias ventajas:

## PLUGIN RPL QUE PERMITE LA REALIZACIÓN DE LABORATORIOS REMOTOS

- Disponibilidad plena del experimento.
- Eficiencia máxima en el tiempo de uso.
- Mantenimiento necesario notablemente menor



**Figura 4.** Estructura de laboratorio remoto.

**Nota fuente:** Aurova (2008). Estructura de laboratorio remoto (Robolab).

<http://www.aurova.ua.es/robolab/principal.htm>

### 2.7 Node.JS

Node.Js es un intérprete JavaScript del lado del servidor la cual cambia la noción de cómo debería trabajar un servidor. Su fin es permitir a un programador construir aplicaciones altamente escalables y escribir líneas de código que maneje miles de conexiones simultáneas en una máquina física.

#### 2.7.1 SocketIO

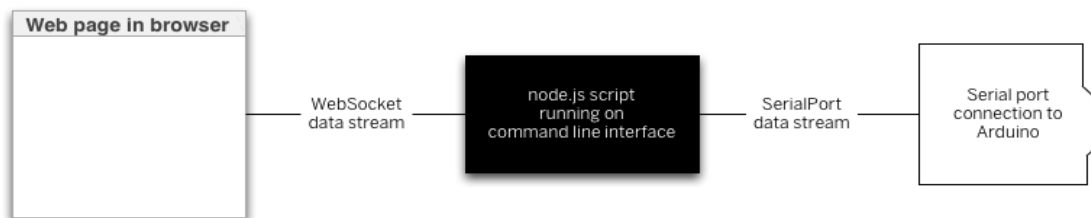
Es una aplicación web de tiempo real de la biblioteca de JavaScript que permite la comunicación bidireccional entre los clientes y los servidores web. Contiene dos partes una del lado del

cliente biblioteca que se ejecuta en el navegador y una del lado del servidor biblioteca para Node.js . Ambos componentes tienen una casi idéntica API . Al igual que Node.js, es dirigida por eventos .

Socket.IO utiliza principalmente el WebSocket que es un protocolo con el sondeo como una opción de reserva, mientras que proporciona la misma interfaz. Se puede instalar con la herramienta NPM.

### 2.7.2 WebSocket

WebSockets son conexiones entre clientes y servidores web que funcionan con el puerto serie. Ambas conexiones en serie y conexiones WebSocket son flujos de datos. Los flujos de datos son estructuras comunes de programación que se utilizan en muchos lugares. Se conectan los programas a los archivos de su ordenador, o programas de cliente a servidor, programas de escritorio a los puertos serie o de red.



**Figura 5.** Conexión desde el navegador al programa nodo

**Nota Fuente:** *Diagrama muestra cómo el navegador se conecta a su programa de Node.js y cómo el programa de Node.js se conecta al puerto serie.*

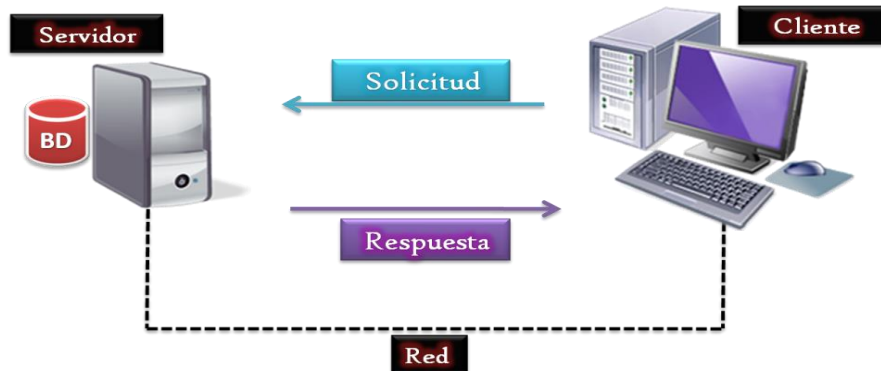
*<https://itp.nyu.edu/physcomp/labs/labs-serial-communication/lab-serial-communication-with-node-js/>*

### *2.8 Programación orientada a objetos*

La programación orientada a objetos se basa en programas llamados objetos que se definen por clases que pueden interactuar entre ellos. Así un objeto puede instanciar (crear un objeto de una clase) otro objeto dentro de él mismo dando origen a lo que se denomina herencia. Lo anterior permite a la programación versatilidad adaptando el programa a diferentes requerimientos del usuario al instanciar objetos que implementen nuevas funciones. (Joyanes Aguilar, 2003).

### *2.9 Modelo Cliente – Servidor*

La arquitectura Cliente – Servidor se puede definir como una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información. En este modelo, el cliente envía un mensaje solicitando un determinado servicio a un servidor (hace una petición), y este envía uno o varios mensajes con la respuesta (provee el servicio). En un sistema distribuido cada máquina puede cumplir el rol de servidor para algunas tareas y el rol de cliente para otras. El servidor se identifica con una dirección IP fija (*Internet Protocol*), y el cliente generalmente posee una dirección IP dinámica. De esta forma, un programa que permita ofrecer servicios web, como por ejemplo Apache, debe ejecutarse en un servidor. Por otro lado, el cliente debe ejecutar programas de tipo cliente como por ejemplo un navegador web o un cliente FTP (*File Transfer Protocol*) (Gonzales, 2005).



**Figura 6.** Modelo cliente servidor

**Nota fuente.** *Desarrollo de aplicaciones.*

<https://mind42.com/public/19bcae1c-8be9-4c00-9ef3-951d304fe900>

### 2.9.1 Servidores HTTP (*Hypertext Transfer Protocol*)

El Protocolo de Transferencia de Hipertexto (**HTTP**) es un sencillo protocolo cliente-servidor que articula los intercambios de información entre los clientes Web y los servidores HTTP. Todas las páginas web están escritas en lenguaje de hipertexto, por lo que es el contenido de las páginas web.

Es un protocolo de transferencia es el sistema mediante el cual se transfiere información entre los servidores y los clientes (por ejemplo los navegadores). Utiliza por defecto el puerto 80 y no posee memoria por lo que se han desarrollado sistemas como los cookies para guardar datos temporales.

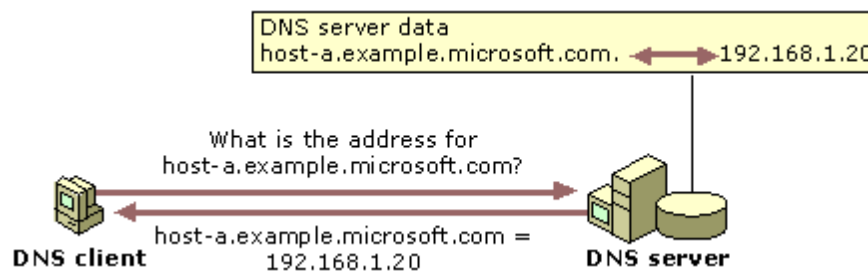
### 2.9.2. Servidores FTP (*Protocolo de transferencia de archivos*)

El protocolo FTP es un protocolo que permite transferir archivos, carga y descarga de archivos desde y hacia un servidor sin necesidad de una página web, como en el caso de HTTP.

Generalmente un servidor FTP es un ordenador que permite manipular archivos que se encuentran en su base de datos. Cargar, descargar, editar, listar, crear y borrar son algunas de las funciones que un cliente puede realizar con los archivos de un servidor FTP. Sin embargo, es necesario disponer de un nombre de usuario y una contraseña para poder llevar a cabo dichas operaciones. Además, el administrador del servidor FTP puede prohibir a los usuarios algunas de las acciones que pueden realizar con los archivos, como por ejemplo borrar un archivo (Cañón, 2016).

### 2.9.3 Servidores DNS (Domain Name Server)

DNS es un sistema para asignar nombres a equipos y servicios de red que se organizan en una jerarquía de dominios. Los servidores y clientes se identifican por medio de una IP la cual identifica al ordenador dentro de la red, permitiendo comunicarse entre servidores y clientes. A pesar de ello, la dirección IP está compuesta de 32 bits organizados en 4 grupos de 8 bits, un ejemplo de dirección IP es 204.34.56.21. Lo anterior pone en evidencia la complejidad de interconectar muchos ordenadores pues se hace necesario conocer todas las direcciones IP de los equipos. Cuando un usuario escriba un nombre DNS en una aplicación, los servicios DNS podrán traducir el nombre a otra información asociada con el mismo, como una dirección IP.



**Figura 7.** Definición de DNS

*Nota Fuentes: Definición de DNS*

[https://msdn.microsoft.com/es-es/library/cc787920\(v=ws.10\).aspx](https://msdn.microsoft.com/es-es/library/cc787920(v=ws.10).aspx)

## CAPÍTULO III

### 3. Análisis de requerimientos

En este capítulo se explicara con más detalle los requerimientos del plugin; donde cada sección contiene el propósito del desarrollo y los aportes que este traerá a los estudiantes de la licenciatura en electrónica. Se explicara sobre lo que debe contener el plugin RPL, la comunicación con arduino y la transmisión y recepción de video en tiempo real.

#### 3.1 Plugin RPL

Al hablar de RPL es importante decir que es un módulo de tipo actividad donde le permite al usuario manipular, diseñar y eliminar laboratorios remotos, A continuación se explicará el esquema de este módulo, donde se analizarán los requerimientos al momento de instalar, manipular, crear y eliminar el plugin en Moodle. Estos archivos que se presentan a continuación son prerequisites de la página Moodle para crear un plugin tipo actividad:

##### ***3.1.1 mod\_form.php:***

Este archivo debe contener los elementos que se mostraran al momento de crear/installar una instancia del módulo. Esto va dirigido al docente o administrador del plugin que al añadir RPL contenga un formulario para crear un laboratorio nuevo, en el que podrá agregar y guardar información, como el nombre del laboratorio, encabezado, tipo de laboratorio a desarrollar, etc.

##### ***3.1.2 version.php:***

El archivo contiene una lista de atributos que es de gran importancia dentro del directorio raíz del plugin, el cual contiene las propiedades que se utilizan durante la instalación y proceso de

actualización. Permite asegurarse de que el plugin es compatible con el sitio de moodle y la detección de una actualización.

Las siguientes propiedades que contiene el fichero versión son:

#### *3.1.2.1 \$plugin->component*

Este componente se utiliza durante el proceso de instalación y actualización, el cual diagnostica los propósitos de validación para asegurarse que el código del plugin sea correcto dentro del árbol del código Moodle.

#### *3.1.2.2 \$plugin->versión*

La versión del plugin RPL corresponde al año mes y día que se desarrolló (YY-MM-DD-XX), y los dos últimos dígitos corresponden a un contador incremental.

#### *3.1.2.3. \$plugin->requires:*

Especifica el número de la versión mínima de Moodle que este plugin requiere.

#### *3.1.2.4 \$plugin->cron:*

Este valor representa una brecha mínima requerida en segundos entre dos funciones cron del plugin. Si es 0 quiere decir que la función cron está deshabilitada.

#### *3.1.2.5. \$plugin->maturity:*

El nivel de madurez declara que tan estable o no es el modulo dentro de Moodle, afectando las notificaciones a medida que se actualice el módulo. En los estados de madurez se encuentran: Maturity\_beta, maturity\_rc, maturity\_stable o maturity\_alpha que es el estado de RPL actualmente que declara que aún no es un módulo estable para su producción.

### *3.1.2.6. \$plugin->release:*

Nombre de la versión que ayuda a identificar el plugin.

### **3.1.3 Lang.php:**

En esta carpeta se encuentra todos los ficheros de los idiomas, estructurado de la misma manera dentro de esta carpeta. Según el código de estilo se debe definir los textos en inglés siendo un prerequisite al momento de su instalación, pero se pueden agregar los idiomas que el administrador o docente deseen. Como el modulo se desarrolló en Colombia y está dirigida a una población que habla el idioma español se crear un archivo “es” que contiene los ficheros en el lenguaje español internacional es\_utf8.

### **3.1.4 Carpeta db:**

En esta carpeta se almacenan los ficheros y las tablas de bases de datos necesarias para la instalación del plugin RPL en Moodle y crear un módulo actividad:

#### *3.1.4.1. access.php:*

Este archivo define los permisos del módulo, permisos que no son obligatorios pero sí muy recomendables para garantizar que los usuarios puedan acceder a las distintas partes de él. En este fichero utilizamos 3 permisos: view, addinstance, y accesslaboratorioremoto, a los que se conceden permisos que restringen o permiten realizar una acción a sus usuarios. Los administrativos y docentes se les habilitaran todos los permisos, ya que son los usuarios que más van a estar manipulando RPL, mientras que los estudiantes e invitados deben primero registrarse al curso para manipular la planta se les restringen la mayoría de permisos.

#### *3.1.4.2. install.xml:*

Este fichero se definirá las tablas que necesita el plugin, en el que se provee mecanismos para la creación de tablas y así acceder y modificar los datos. Para el módulo se necesitan dos tablas, la primera con el mismo nombre del plugin y que es obligatoria para Moodle, así como los campos *id*, *course* y *name*, unos campos adicionales que se requieran y las tablas que el plugin amerite.

#### *3.1.4.3. upgrade.php:*

En este fichero se encuentran las acciones para actualizar a una versión anterior del plugin, en las que se deben hacer algunas modificaciones de las tablas de acuerdo a la versión de Moodle que se tenga instalado.

#### *3.1.4.4 events.php:*

Este fichero se crear para observar los eventos de supervisión que se evidencian en el curso para informar a Moodle sobre ellos.

#### *3.1.4.5 uninstall.php:*

Este fichero se utiliza para desinstalar el plugin de la plataforma educativa Moodle.

#### ***3.1.5. index.php:***

Este fichero se utiliza por Moodle para enumerar todas las instancias o eventos del plugin RPL que se encuentran en un curso particular con la ID.

#### ***3.1.6. view.php:***

Este es uno de los ficheros más importantes en el que se hace un diseño previo donde se crearan las actividades propuestas a partir de unos enlaces. Es de gran importancia para el estudiante que

al ingresar a la actividad se cargue una cámara web en tiempo real que muestre la planta y una interfaz adecuada para manipular la planta.

### ***3.1.7. lib.php:***

En este fichero se deben implementar todas las funciones y procedimientos del plugin, las cuales son llamadas por otros archivos dentro de rpl. A continuación se muestran las funciones que son obligatorias para crear el plugin de actividad:

- `rpl_add_intance ()`: Función para añadir una nueva instancia.
- `rpl_update_instance ()`: Función para actualizar una instancia existente.
- `rpl_delete_instance ()`: Función para borrar una instancia.
- `rpl_user_outline ()`: Da un resumen concreto de la actividad desarrollada por el usuario.
- `rpl_user_complete ()`: Función que arroja un informe detallado de la contribución del usuario.
- `rpl_get_view_actions ()`: Clasifica las acciones para el log. Usualmente se usa en el informe de participación,

Todas las funciones, procedimientos y constantes creados en la carpeta `lib.php`, comienzan con el nombre del plugin a crear. Este fichero viene conectado con `locallib.php` que es una carpeta complementaria para liberar espacio al momento de cargarlas.

### ***3.1.8. settings.php (opcional)***

Este fichero es opcional, utilizado para configurar el módulo y sus valores que se establecen para todos los módulos para no crear tablas o campos nuevos.

### ***3.1.9. Carpeta pix***

Esta carpeta almacena el icono del plugin RPL para identificarlo de los otros módulos, el cual debe ser guardado en un formato .gif y con una resolución de 16x16.



**Figura 8.** Icono del plugin RPL.

**Nota Fuente:** *Logo Universidad Pedagógica Nacional.*

*<http://www.pedagogica.edu.co/>*

## **3.2 TRANSMISION Y RECEPCION DE VIDEO EN TIEMPO REAL**

Esta sección permite identificar el adicional del plugin para laboratorios remotos, ya que a partir de la transmisión y recepción del video en tiempo real el estudiante podrá recrear lo que este manipulando como si estuviera en un espacio de laboratorio de la universidad. Para poder realizar esta acción se requiere utilizar primero una cámara web de mínimo 2 MPX que debe estar conectado en el pc del servidor y a través de unas líneas de código y librerías de Node js permite transmitir eventos en tiempo real para que llegue al cliente. De esta manera se pretende proporcionar al estudiante información de lo que sucede con el laboratorio en vivo. A continuación se mostraran requerimientos y análisis que debe contener esta etapa para el video en tiempo real:

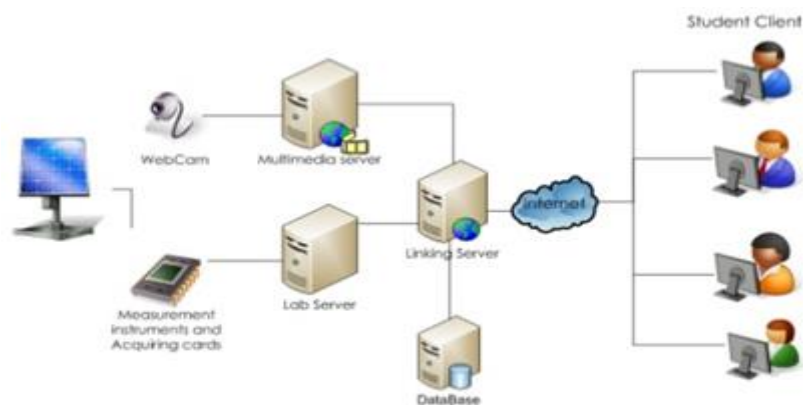
Para poder realizar esta acción es necesario usar un intérprete de JavaScript del lado del servidor “Node.js” que cambia la forma de trabajar de un servidor y el cual postula la siguiente arquitectura:

### 3.2.1 Carpeta public

Esta carpeta debe contener los archivos HTML y así crear primero una página web “index.html”, para enlazar dos archivos: emitir “Servidor” y visualizar “Cliente” en las que se verificaran a través del navegador el funcionamiento adecuado de la transmisión y recepción del video.

### 3.2.2 app.js:

Este fichero es el que se corre en el command Window, donde se crea un servidor http que se le asigna un puerto para ingresar desde el navegador; app requiere de socket.io que permite la comunicación bidireccional entre los clientes y los servidores web en tiempo real. Esta comunicación bidireccional son las imágenes que se emiten desde el servidor y en cuestión de milisegundos se transmiten al cliente y se muestran como video.



**Figura 9.** Esquema entre Servidor, Cliente y Laboratorio.

**Nota Fuente:** Componentes de un laboratorio remoto.

<http://es.slideshare.net/emadridnet/emadrid-2015-20-02-uned-rafael-pastor-vargas-desarrolloexplotacin-de-entornos-experimentales-basados-en-laboratorios-remotos-y-virtuales-related-44979219>

### **3.3 COMUNICACIÓN CON ARDUINO**

Esta etapa de programación se realizará entre un Arduino y Node.js en el cual se realiza una comunicación entre dos aplicaciones, una se ejecuta en el arduino (C++) y la otra corre en el computador con Node.js, la cual se comunican a través del puerto serial (USB). Esta comunicación se realiza por medio de un protocolo en http en el que se transfiere información entre los servidores http y los clientes web. El objetivo principal es crear un servidor http en Node.js el cual recibirá una petición por parte del cliente envía datos a través de una página web e ir vareando la intensidad de los leds.

#### ***3.3.1 Server.js***

En este fichero se crea el servidor http y se realiza la conexión del arduino con el servidor a través de la librería de SerialPort, donde se configura el puerto COM del arduino, la velocidad de transmisión, tamaño de los caracteres, la paridad, el tipo de control de flujo y los bits de parada, esto con el fin de leer el arduino.

#### ***3.3.2 Route.js***

Este fichero en ruta las solicitudes, clasificando el tipo de datos que se están enviando y recibiendo.

#### ***3.3.3 requestHandlers.js***

Este fichero contiene los atributos, el cual llama un archivo HTML que contiene la interfaz del laboratorio de prueba

#### ***3.3.4 index.js:***

## PLUGIN RPL QUE PERMITE LA REALIZACIÓN DE LABORATORIOS REMOTOS

Este fichero contiene todos los archivos anteriores, es el encargado de correr en el command Windows para realizar la comunicación con el arduino.

## CAPÍTULO IV

### 4. Diseño y construcción

En este capítulo se mostrarán las diferentes etapas del diseño y desarrollo del plugin, donde se evidenciara el material de apoyo de cada etapa; y de esta manera cumplir con los objetivos planteados en el inicio de este documento.

#### 4.1 ETAPA PLUGIN

##### *4.1.1 Diseño estructural del Plugin RPL (Prácticas de laboratorios remotos).*

Para comenzar el desarrollo del plugin RPL de tipo actividad se determinó elaborar casos de usos, los cuales muestran los requerimientos funcionales. Estos requerimientos son aquellos que muestran el comportamiento de entradas y salidas del sistema, y van dirigidos a un sujeto en específico. A partir de un buen diseño de casos de uso se puede desarrollar un sistema de acuerdo a las necesidades del sistema.

##### *4.1.2. Lista de requerimientos:*

Esta lista de requerimientos funcionales se realizó de acuerdo a las necesidades del usuario al momento de ir a manipular el plugin. A continuación se mostrara la lista de requerimientos para posteriormente pasar a la realización de casos de uso.

<b>SISTEMA PARA PLUGIN RPL</b>			
<b>LISTA DE REQUERIMIENTOS FUNCIONALES</b>			
<b>CODIGO</b>	<b>NOMBRE DEL REQUERIMIENTO</b>	<b>CASO DE USO</b>	<b>ACTOR</b>

PLUGIN RPL QUE PERMITE LA REALIZACIÓN DE LABORATORIOS REMOTOS

RF1	El sistema debe permitir al docente mostrar la información de los estudiantes y usuarios registrados en el aula	Consultar registro de estudiantes y usuarios en el aula	PROFESOR
RF2	El sistema debe permitir al docente ingresar las notas de acuerdo a los resultados enviados por los estudiantes.	Registrar notas de curso	PROFESOR
RF3	El sistema debe permitir al docente autorizar el registro de usuarios al plugin	Registrar usuarios autorizados que correspondan al curso	PROFESOR
RF4	El sistema debe permitir al docente crear nuevos cursos en el aula	Registrar Cursos dentro del aula	PROFESOR
RF5	El sistema debe permitir añadir, modificar o eliminar los laboratorios remotos al aula	Mantener laboratorios remotos (Añadir, modificar, eliminar).	PROFESOR
RF6	El sistema debe permitir actualizar información del curso y actividades propuestas en el aula	Mantener información del curso (Modificar, añadir o eliminar).	PROFESOR

**Tabla 1.** Lista de requerimientos para RPL.

**Nota de Fuente:** (2016). Elaborado por el autor.

**4.1.3 Descripción caso de uso.**

La descripción de casos de uso muestra la secuencia de acciones e interacciones entre el actor y el sistema, a continuación se mostrara dos descripciones de casos de uso que se realizó para el plugin RPL.

### CASO DE USO 1

**Caso de uso:**<sup>1</sup> Acceso a los laboratorios remotos.

**Actor:**<sup>2</sup> Estudiante.

**Precondición:**<sup>3</sup> El usuario ha sido admitido en la plataforma y en el curso con el rol de estudiante.

**Poscondición:**<sup>4</sup> Se ha ingresado al laboratorio remoto y se ha desarrollado la actividad propuesta por el docente.

**Flujo Básico:**<sup>5</sup>

1. El caso de uso comienza cuando el usuario ingresa a Moodle.
2. El sistema muestra los cursos creados por el docente o administrador.
3. El usuario ingresa al curso correspondiente al cual se encuentre inscrito.
4. El sistema muestra las actividades propuestas en este curso a trabajar.
5. El usuario elige la actividad a desarrollar.
6. El sistema muestra el laboratorio en tiempo real.
7. El usuario accede y trabaja con la planta de forma remota durante un tiempo.
8. El caso de uso finaliza.

**Flujo alternativo:**<sup>6</sup>

**Datos incorrectos:**

En el paso 1, si el sistema determina algún error de datos al ingresar, muestra mensaje de error y vuelve a solicitar los datos.

**Verificación de espacio:**

En el paso 6, si el sistema determina que el docente no ha creado un laboratorio, mostrara un mensaje que le avisa al estudiante que aún no hay un laboratorio cargado.

**Tabla 2.** Caso de uso 1

**Nota de Fuente:** (2016). Elaborado por el autor.

---

1. Caso de uso: Nombre del caso de uso.

2. Actor: Conjunto de roles de usuarios del sistema (Profesor, Estudiante, Administrador, Usuario invitado).

3. Precondición: Condición que debe ser verdadero para iniciar el caso de uso

4. Poscondición: Condición que debe cumplirse para indicar que el caso de uso ha terminado con éxito.

5. Flujo básico: Pasos sin ningún inconveniente

6. Flujo alternativo: Pasos cuando sucede algo extraordinario.

## CASO DE USO 2

**Caso de uso:** Mantener laboratorios remotos (Añadir, Modificar, Eliminar).

**Actor:** Profesor

**Precondición:** El usuario ha sido admitido en el sistema con el rol de Profesor.

**Poscondición:** Se ha añadido, modificado, eliminado un laboratorio remoto en la plataforma Moodle.

**Flujo Básico:**

### AÑADIR

1. El caso comienza cuando el profesor ingresa a Moodle.
2. El sistema muestra las opciones de inicio.
3. El profesor ingresa al curso que va a añadir el laboratorio.
4. El sistema muestra lo que contiene el curso.
5. El profesor selecciona la opción habilitar edición.
6. El sistema le muestra la opción de añadir actividad.
7. El profesor va a la opción de añadir actividad o un recurso.
8. El sistema muestra el listado de plugin actividad o recursos.
9. El profesor selecciona la actividad RPL.
10. El sistema muestra información sobre la actividad recordando para qué funciona el plugin.
11. El profesor le da añadir.
12. El sistema muestra un formulario con unos campos para editar.
13. El usuario edita la información del laboratorio.
14. El sistema muestra la opción de guardar o cancelar.
15. El profesor le da guardar.
16. El caso de uso finaliza.

### MODIFICAR

1. El caso comienza cuando el profesor ingresa a Moodle.

2. El sistema muestra las opciones de inicio.
3. El profesor ingresa al curso que va añadir el laboratorio.
4. El sistema muestra lo que contiene el curso.
5. El profesor selecciona la opción habilitar edición.
6. El sistema habilita la opción de editar las actividades creadas en el curso.
7. El profesor selecciona la actividad a modificar.
8. El sistema muestra un menú de opciones de edición.
9. El profesor selecciona editar.
10. El sistema el formulario de la actividad con los campos que se han editado.
11. El profesor modifica el campo que va a remplazar.
12. El sistema muestra la opción de guarda o cancelar.
13. El profesor le da guardar.
14. El caso de uso finaliza.

#### **ELIMINAR**

1. El caso comienza cuando el profesor ingresa a Moodle.
2. El sistema muestra las opciones de inicio.
3. El profesor ingresa al curso que va añadir el laboratorio.
4. El sistema muestra lo que contiene el curso.
5. El profesor selecciona la opción habilitar edición.
6. El sistema habilita la opción de editar las actividades creadas en el curso.
7. El profesor selecciona la actividad a eliminar.
8. El sistema muestra un menú de opciones.
9. El profesor selecciona borrar.
10. El sistema elimina la actividad del curso.
11. El caso de uso finaliza.

#### **Flujo alternativo:**

#### **AÑADIR**

En el paso 1, si el sistema determina algún error de datos al ingresar, muestra mensaje de error y vuelve a solicitar los datos.

En el paso 3, si el profesor ingresa a un curso el cual no corresponde añadir el laboratorio el profesor puede volver al inicio e ingresar al curso adecuado.

En el paso 8, si el sistema muestra el listado de plugin actividad o recursos y no aparece RPL el profesor o administrador deberán revisar si está instalada el plugin adecuadamente.

En el paso 15, si el profesor le da guardar y no ha llenado lo campos obligatorios el sistema no dejara guardar el laboratorio y lo llevara a los campos obligatorios que faltan por editar.

### **MODIFICAR**

En el paso 1, si el sistema determina algún error de datos al ingresar, muestra mensaje de error y vuelve a solicitar los datos.

En el paso 3, si el profesor ingresa a un curso el cual no corresponde añadir el laboratorio el profesor puede volver al inicio e ingresar al curso adecuado.

En el paso 13, si el profesor le da guardar y no ha llenado lo campos obligatorios el sistema no dejara guardar el laboratorio y lo llevara a los campos obligatorios que faltan por editar.

### **ELIMINAR**

En el paso 1, si el sistema determina algún error de datos al ingresar, muestra mensaje de error y vuelve a solicitar los datos.

En el paso 3, si el profesor ingresa a un curso el cual no corresponde añadir el laboratorio el profesor puede volver al inicio e ingresar al curso adecuado.

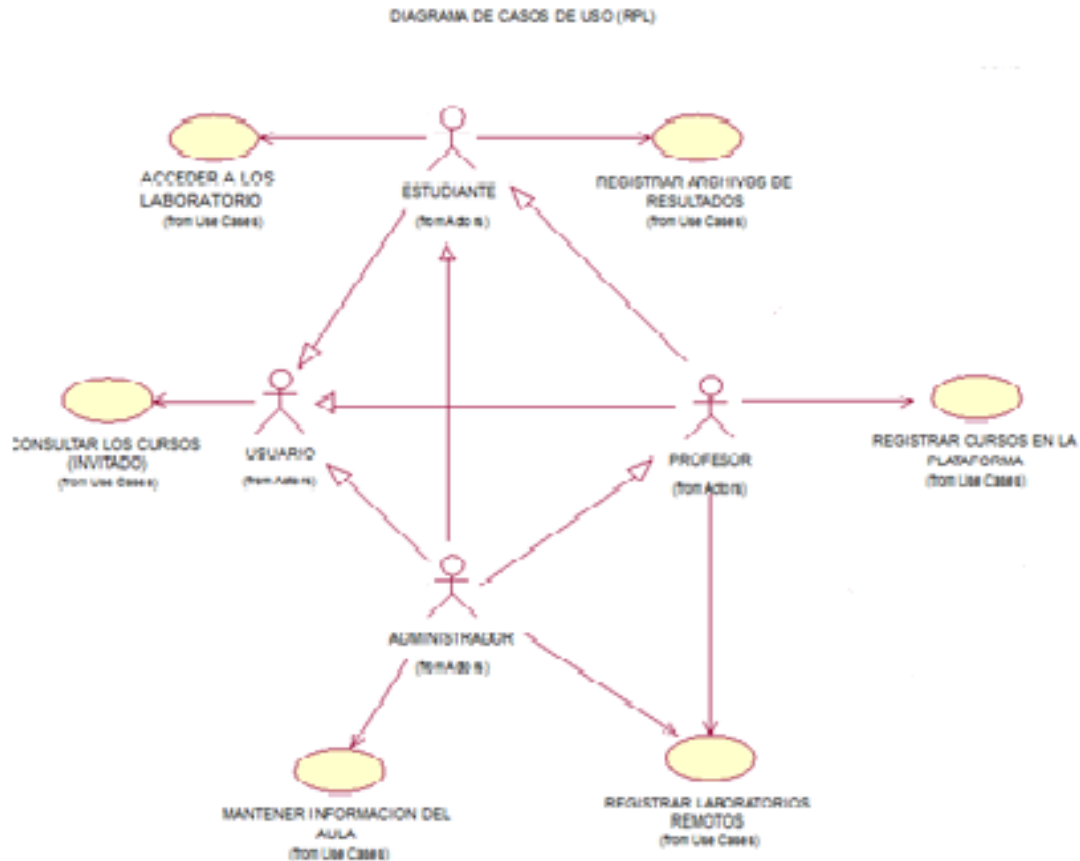
En el paso 7, Si el profesor selecciona una actividad que no quiere eliminar se puede ir a la página de inicio y volver a seleccionar la actividad a eliminar.

**Tabla 3.** Caso de uso 2.

**Nota Fuente:** (2016). Elaborado por el autor.

#### 4.1.4 Diagrama de caso de uso

El diagrama de caso de uso muestra los actores, los casos y la relación que hay entre ellos al momento de diseñar el plugin RPL.

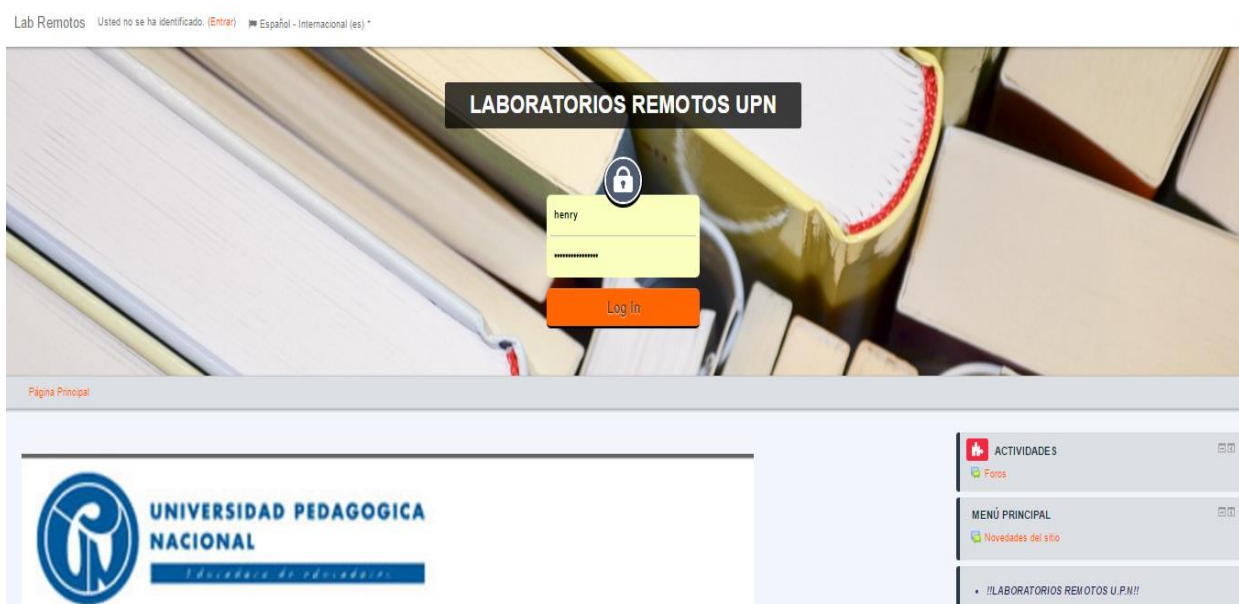


**Figura 10.** Diagrama de caso de uso

Nota Fuente: (2016). Elaborado por el autor

#### 4.1.5 Ventana de bienvenida a Moodle:

En la Figura 9 se muestra la ventana de bienvenida a Moodle en la que los usuarios podrán ingresar o registrarse en la plataforma. El usuario debe contener obligatoriamente un nombre de usuario y una contraseña para poder acceder.



**Figura 11.** Ventana de Bienvenida de Moodle.

**Nota Fuente:** (2016) Elaborado por el autor.

Cada archivo que contiene la carpeta del plugin RPL se describirá a continuación desde la plataforma Moodle, en la que se mostrara en detalle cada una de sus funcionalidades descritas en un lenguaje de código en PHP.

#### ***4.1.6 Instalación del plugin***

Cuando se va a realizar la instalación del plugin en la plataforma, pide como requisitos tener dentro de nuestro archivo .zip los siguientes archivos:

##### ***4.1.6.1 Carpeta db:***

Esta carpeta contiene los ficheros que se mencionaron en el capítulo 3 de este documento, que se retoman pues son requisitos para instalar y desarrollar el RPL en Moodle.

#### 4.1.6.1.1 acces.php:

Los permisos que se le asignan a cada usuario dependen del rol que este desempeñando dentro del módulo los cuales se explicaron anteriormente en el análisis de requerimiento. Los usuarios creados dentro en este archivo son:

- Administrador.
- Profesor.
- Estudiante.
- Usuario.

#### 4.1.6.1.2 install.php:

En este fichero se crearon dos tipos de tabla una con el nombre del módulo y la otra con el nombre de rpl\_registros las cuales contienen unos campos designados para el plugin, como name, course, id, etc.

#### 4.1.6.1.3 upgrade.php:

Este fichero se encarga de actualizar el módulo para que coincida con la versión más reciente del plugin y se realiza con el fin de no alterar las tablas.

#### 4.1.6.1.4 Version.php:

Este fichero para el plugin RPL tiene la versión “2016051000” que fue la última modificación que se le hizo al plugin, siendo un módulo en estado “Maturity\_alpha” el cual no es estable para su producción.

## PLUGIN RPL QUE PERMITE LA REALIZACIÓN DE LABORATORIOS REMOTOS

Para realizar la instalación de RPL dentro de la plataforma Moodle se tiene que ingresar como administrador, y luego ir a la opción de extensión. La cual desplegara unas opciones en la que dejara instalar un módulo externo. Cuando se carga el archivo .zip en Moodle el valida los requisitos antes mencionados como se muestra en la figura 12.

Estado	Mensaje	Info
OK	Nombre del módulo externo que se debe instalar	rp1
OK	Versión del módulo externo	2016032000
OK	Versión de Moodle requerida	2013111800
OK	Nombre completo del componente	mod_rp1
Advertencia	Nivel de madurez	MATURITY_ALPHA
OK	Versión del módulo externo	v2.7-r1
Depuración	Buscar fichero de idioma	rp1
OK	Chequeo de permisos de escritura	C:\xampp\htdocs\moodle/mod

**Figura 12.** Validación de requisitos plugin RPL.

**Nota Fuente:** *Validación del paquete de módulo externo.*

<http://localhost/moodle/admin/tool/installaddon/validate.php?sesskey=RB7IoBrOXN&jobid=5a0742d6d2d18c53f66ef43e93bfa53f&zip=rpl.zip&type=mod&rootdir>

### 4.1.7 Añadir un plugin actividad RPL:

Después de la instalación se continúa con añadir el plugin en un curso específico, para esto el docente debe habilitar la edición dentro de Moodle y dar clic en añadir una actividad y nos mostrara el siguiente recuadro como se ve en la figura 12.

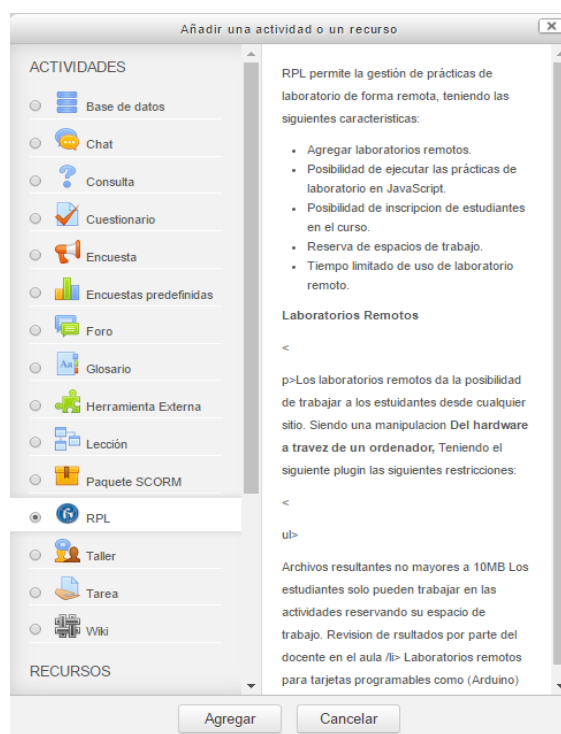
Los archivos requeridos para añadir una actividad al curso son:

#### 4.1.7.1 lib.php – Locallib.php:

En este fichero se crearon las funciones básicas del módulo las cuales son obligatorias para poder ejecutar la actividad y otras que son propias del módulo. Estas librerías se ejecutan al momento que realizamos una acción dentro del módulo.

#### 4.4.7.2 lang:

Esta carpeta contiene dos idiomas internacionales, español por el lugar de origen donde se realizó el proyecto e inglés porque es un requisito de Moodle para su instalación. En ella se encuentran todas las sentencias utilizadas en archivos de RPL y las cuales se traducirán al idioma que este configurado Moodle.



**Figura 13.** Añadir una actividad o recurso

**Nota Fuente:** (2016). Creado por el autor.

#### 4.1.8 Editando una actividad en RPL:

Cuando se agrega la actividad esta se direcciona al fichero de mod\_form.php que está compuesto por unas instancia que definirán el contenido del plugin, cuando el docente crea el laboratorio debe editar cada instancia del módulo. Se añade campos al formulario con funciones básicas:

- addElement: Añade un elemento al formulario en este caso el encabezado, nombre del laboratorio, etc.
- setType: Establece el tipo de contenido del elemento, en este caso utilizamos dos tipos (PARAM TEXT, PARAM INT).
- set Default: Establecer el valor por defecto de un elemento que es el mismo nombre que tiempo el addElement.

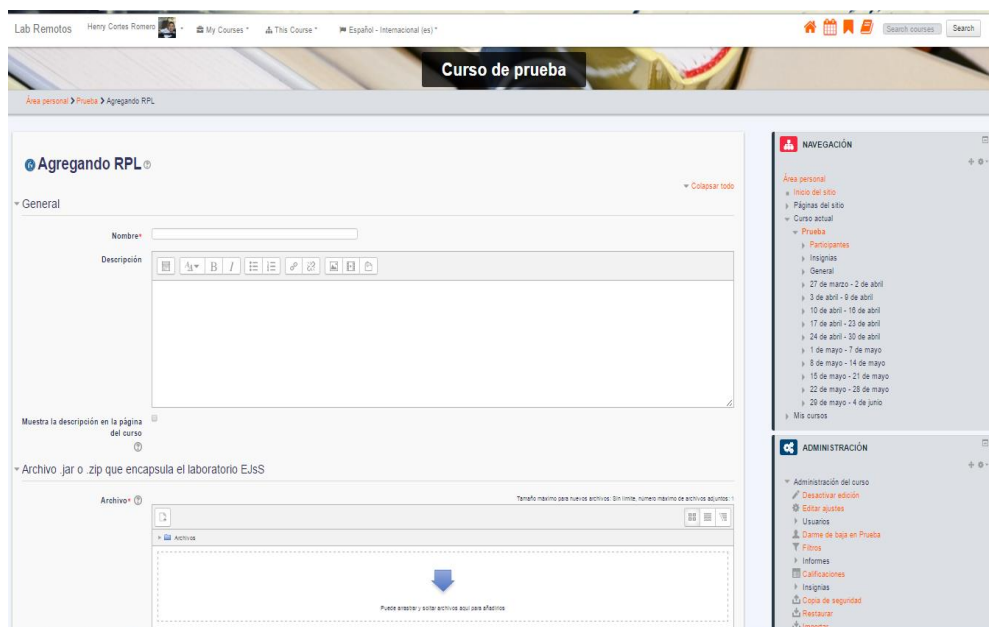


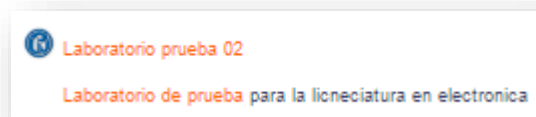
Figura 14. Agregar una actividad RPL.

Nota Fuente: Agregando RPL.

Al ingresar una actividad es importante llenar los campos obligatorios, para que posterior a ellos guarde la actividad; Estos datos se ingresan en la base de datos del plugin.

#### **4.1.9 Actividad creada:**

Cuando el docente y/o administrador han subido un laboratorio, el estudiante podrá ingresar a manipularlo. Pero antes de entrar a manipular deberá estar inscrito y aceptado al curso que registre el docente.



**Figura 15.** Actividad creada dentro del plugin RPL

**Nota Fuente.** (2016). Creada por el autor

Cuando el estudiante ingresa al laboratorio se generan dos procesos independientes dentro del fichero view.php, la trasmisión y recepción de video en tiempo real y la comunicación con el arduino.

#### **4.2 TRASNMISSION Y RECEPCION DE VIDEO EN TIEMPO REAL:**

Para esta parte se necesita una cámara web de 2 Mega pixeles que se utilizara para probar el desarrollo de la trasmisión de video, la cual se realizó en un proceso de manera independiente al plugin, en un lenguaje de programación por node.js como se explica a continuación:

#### 4.2.1 *package.json*:

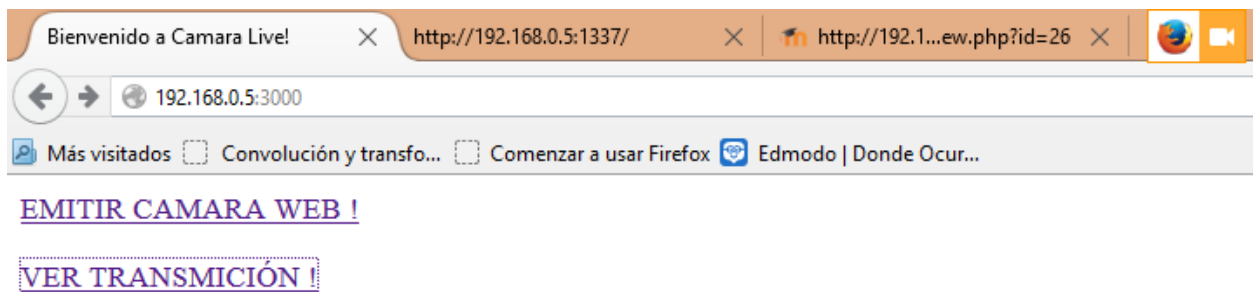
Este archivo contiene los campos de nombre y versión que son importantes al momentos de instarlos, estos campos juntos forman un identificador que es completamente único y unas dependencias como socket.io, express y log donde emiten la fecha, hora y lugar donde se está transmitiendo.

#### 4.2.2 *app.js*:

Este archivo se correrá en command Windows y activa el servidor http local que se haya creado, donde se asigna una dirección 192.168.0.5:3000, y variables que se llaman al descargar los módulos npm.

#### 4.2.3 *index.html*:

Esta carpeta especifica la URL de las páginas “emitir” y “visualizar” que se van a enlazar.



**Figura 16.** Página web index.html

**Nota Fuente:** (2016). Creado por el autor

#### 4.2.3.1 emitir.html:

En este archivo se configura la resolución de la pantalla la cual es 600x800 Mega pixeles configurada con la etiqueta “canvas” de html5. Cuando la cámara no está conectada el command Windows arroja un mensaje advirtiendole que ha fallado la conexión. Este fichero utiliza una función particular en la que se configura cuantas imágenes se envían por segundo, como se muestra en la siguiente ecuación:

$$1000 \text{ fps} = 1 \text{ seg}$$

$$10 \text{ fps} = x$$

Al disminuir a 10fps nos arroja un tiempo de 0.01 seg, lo que mejora la emisión de video ya que se están enviando mayor cantidad de imágenes en un segundo.

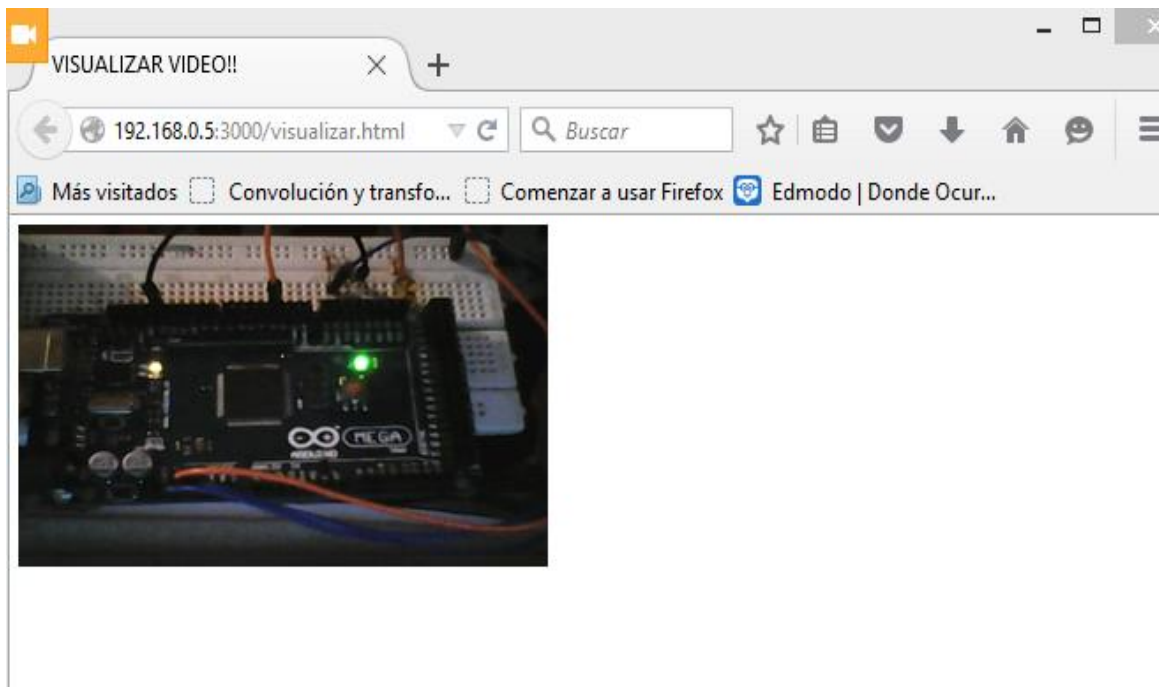


**Figura 17.** Emisión de video en tiempo real

**Nota Fuente:** 192.168.0.5:3000/emitir.html

#### 4.2.3.2 *visualizar.html*:

En este fichero se activara automáticamente la cámara y lo que se esté trasmitiendo del parte del servidor.



**Figura 18.** Visualización en tiempo real.

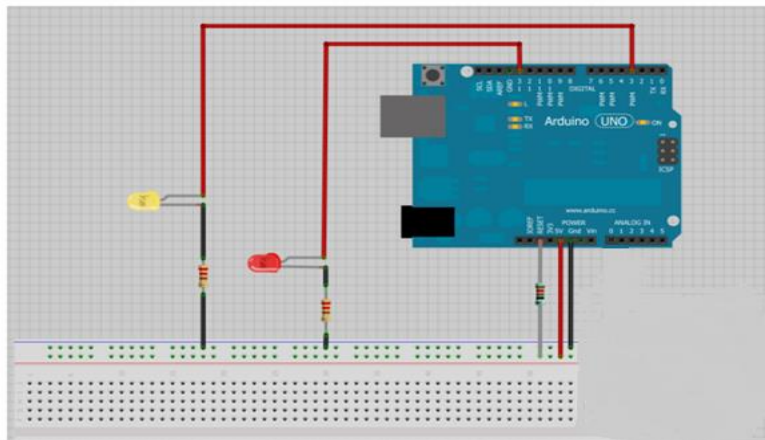
**Nota Fuente:** 192.168.0.5:3000/visualizar.html

### 4.3 COMUNICACIÓN CON EL ARDUINO:

Para el diseño del laboratorio que se cargara en el plugin RPL, se decidió realizar una prueba con arduino, en el que se ira vareando la intensidad de los leds. Primero se diseñó del laboratorio a manipular, seguido el código para programar el arduino y por último la comunicación a partir del puerto serie de arduino.

#### **4.3.1 Diseño del laboratorio:**

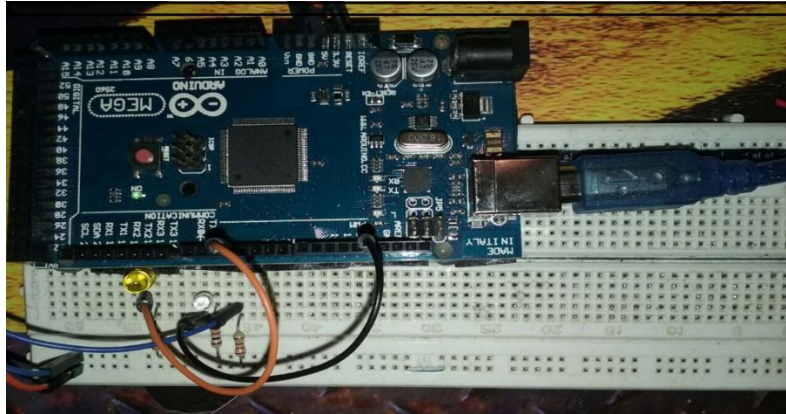
El diseño del laboratorio físicamente son dos leds que estarán conectados a los pines PWM 13 y 3 del arduino Mega y estará del lado del servidor. Pasando al lado del cliente, él se encontrara con una interfaz realizada en HTML que está diseñada esencialmente para el laboratorio propuesto, donde se coloca un botón “on/off” para encender los leds y una barra espaciadora que a medida que se mueva ira vareando la intensidad de los leds. Las siguientes imágenes 19 y 20 muestran el laboratorio en forma de simulación y física.



**Figura 19.** Laboratorio leds.

**Nota Fuente:** (2016). Creado por el autor en el software Fritzing.

## PLUGIN RPL QUE PERMITE LA REALIZACIÓN DE LABORATORIOS REMOTOS



**Figura 20.** Laboratorio físico.

**Nota Fuente:** (2016). Creado por el autor.

### ***4.3.2 Código de arduino:***

Este código que se carga en la tarjeta de programación arduino está diseñado para el laboratorio remoto, esto quiere decir que cuando el estudiante realice una acción desde la interfaz del plugin esta se comunicara con el puerto serial de arduino y se compilara en la planta. Para el desarrollo se utilizaron sentencias como `Serial.available` el cual obtienen el número de caracteres disponibles para su lectura, desde el puerto serie y esos datos que lleguen se almacenen en la memoria intermedia de recepción serie. Después se convierte esos caracteres en enteros para volverlos análogos y variando el voltaje de los leds de 0 a 5 voltios.

### ***4.3.3. Comunicación Node js con arduino:***

Esta comunicación entre dos aplicaciones se hace a través del protocolo http que ya se explicó en el capítulo 3 del presente documento, el cual permite enviar y transmitir datos en tiempo real.

#### ***4.3.3.1 index.js:***

Es el componente de partida que conecta todos los archivos creados para la comunicación de arduino a través del SerialPort y que se corre en el command Windows.

#### 4.3.3.2 *server.js*:

Se crea el servidor `http://localhost:1337` para hacer la conexión del arduino con Node.js a través de la librería SerialPort, donde abre el puerto que se comunicara; y se configuro de la siguiente forma: `varport= COM4` que es el puerto del arduino, los baudios = 9600 velocidad de trasmisión recomendada, y 8 bits para los datos.

#### 4.3.3.3 *route.js*:

Enruta solicitudes o eventos para que se visualicen en la consola cmd al momento de estar corriendo el servidor.

#### 4.3.3.4 *requestHandler.js*:

Controla las solicitudes a la consola y envía la interface al navegador.

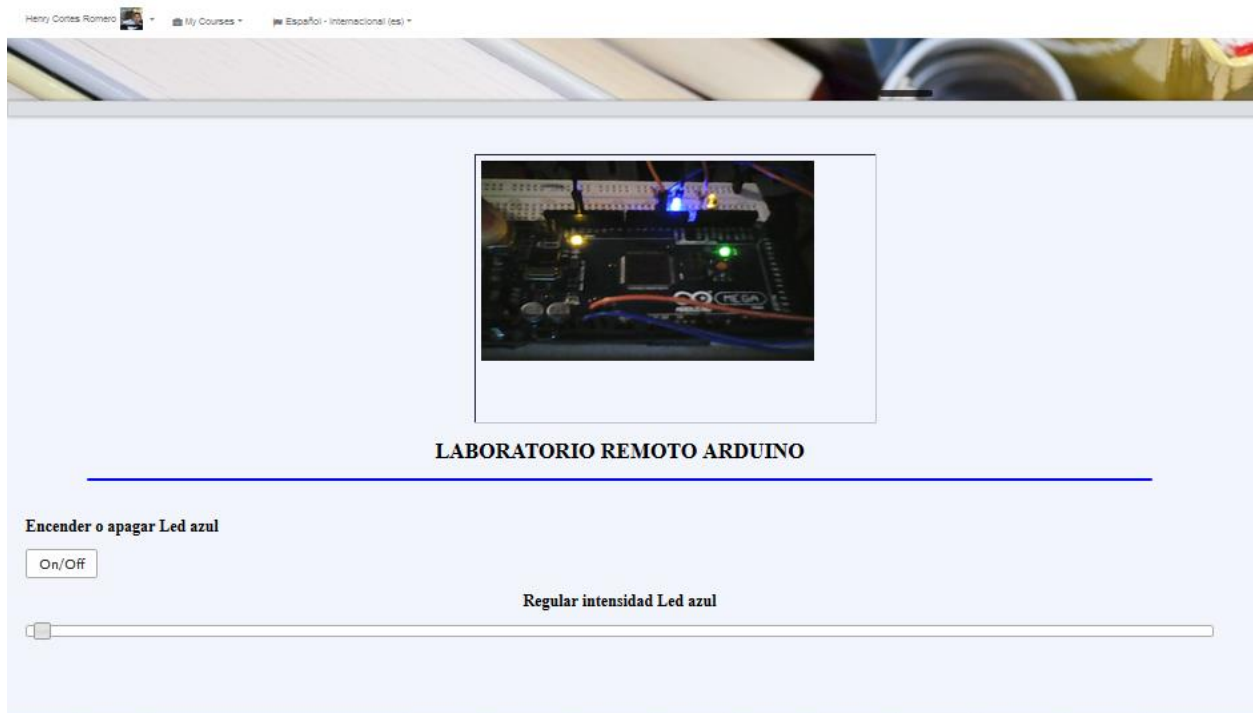


**Figura 21.** Interfaz de laboratorio.

**Nota Fuente:** (2016). Creado por el autor.

## PLUGIN RPL QUE PERMITE LA REALIZACIÓN DE LABORATORIOS REMOTOS

El ultimo paso del desarrollo del plugin RPL es la union de las tres fases mencionadas anteriormente. El archivo que realiza esta fusion es view.php la cual se escriben unas sentencias que corre los dos servidores y de esta manera se vializa el laboratorio y la interfaz para manipularla en tiempo real. Es aquí donde el estudiante ingresar a una red local asociada con el servidor y de esta manera manipular el laboratorio de manera remota desde otro pc.



**Figura 22.** Unión de las tres fases.

**Nota Fuente:** Laboratorio remoto, variación de luz.

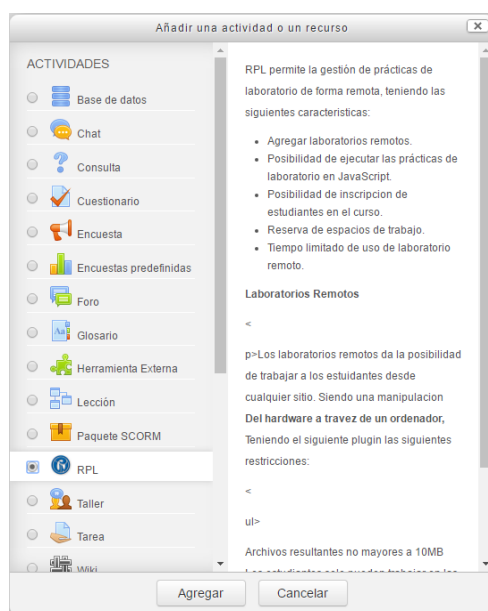
## CAPITULO V

### 5. Pruebas y resultados

Para realizar las pruebas y resultados del plugin se requirió un adicional al proyecto, el cual fue escoger un laboratorio e implementarlo dentro de RPL. De esta manera se prueba que tan eficiente es al momento de realizar pruebas y verificar su funcionamiento, y ver si se cumplieron los objetivos planteados en el capítulo 1.2.

A continuación se mostrara las pruebas pilotos que se realizaron en el plugin, donde se mostrara el resultado de las tres fases del proyecto. Para esta prueba se creara un laboratorio por parte del docente que será de parte del servidor donde estará conectado la cámara la planta; y el estudiante manipulara la planta desde su ordenador visualizando lo que pasa desde el otro lado.

DOCENTE: Primero se añade la actividad RPL al curso:

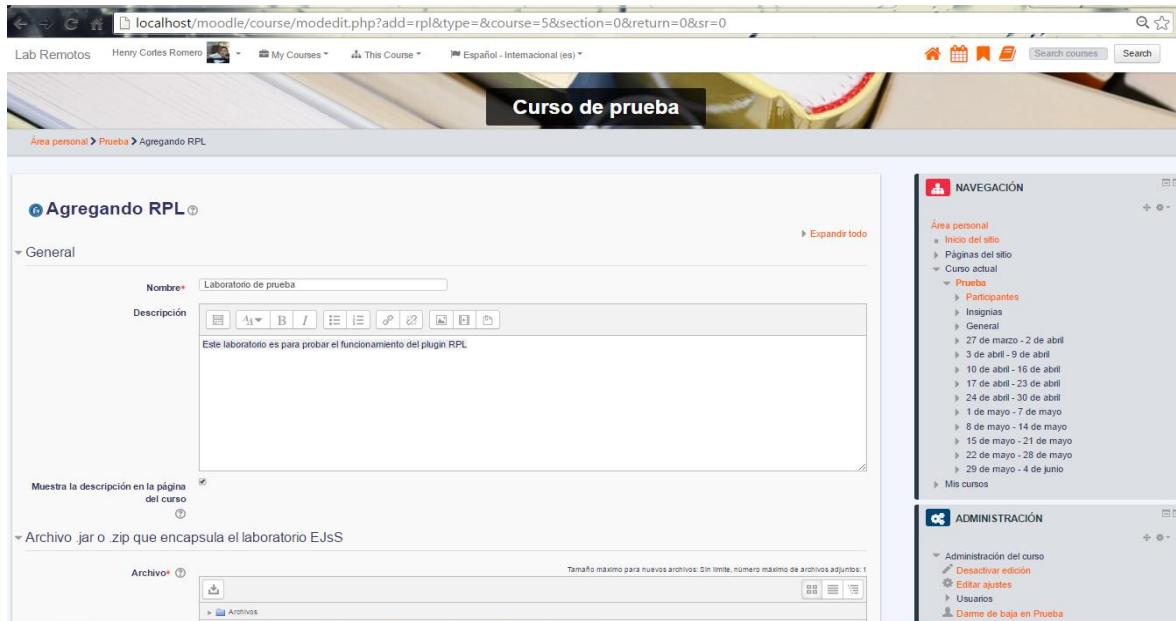


**Figura 23.** Agregando la actividad RPL

**Nota Fuente:** Prueba piloto añadir plugin.

## PLUGIN RPL QUE PERMITE LA REALIZACIÓN DE LABORATORIOS REMOTOS

Seguido nos aparecerá para registrar el laboratorio:



**Figura 24.** Registrar laboratorio en RPL.

**Nota Fuente:** Laboratorio de prueba.

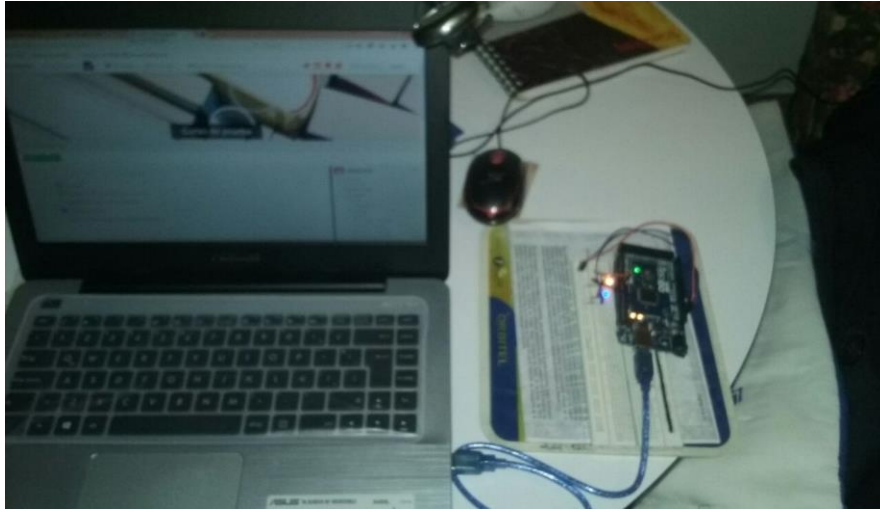
El laboratorio está registrado dentro del curso de prueba:



**Figura 25.** Laboratorio registrado en el curso.

**Nota Fuente:** Laboratorio de prueba.

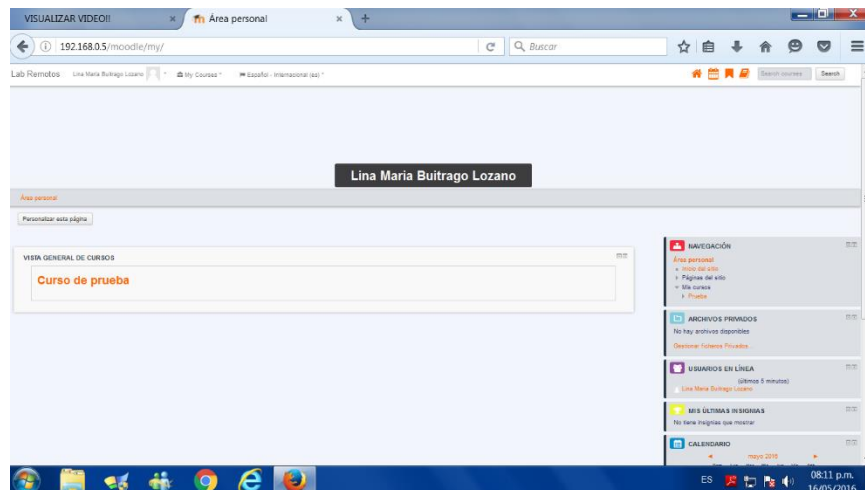
## PLUGIN RPL QUE PERMITE LA REALIZACIÓN DE LABORATORIOS REMOTOS



**Figura 26.** Lado del servidor.

**Nota Fuente:** *Laboratorio, Cámara web, Planta. Creado por el autor.*

ESTUDIANTE: Cuando el estudiante ingresa desde su computador a Moodle le pedirá el nombre de usuario y contraseña; seguido le mostrara los cursos creados por el docente como se muestra a continuación:

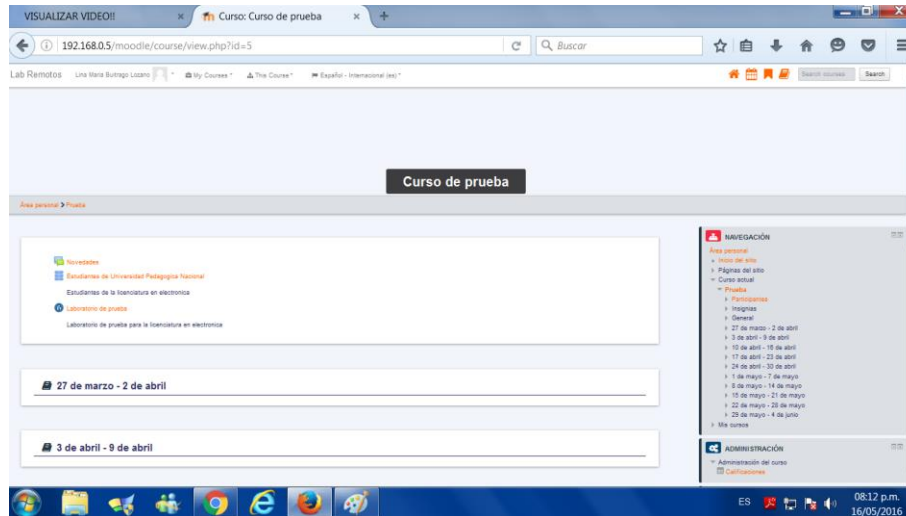


**Figura 27.** Página de inicio de Estudiante.

**Nota Fuente:** (2016). Creado por el autor.

## PLUGIN RPL QUE PERMITE LA REALIZACIÓN DE LABORATORIOS REMOTOS

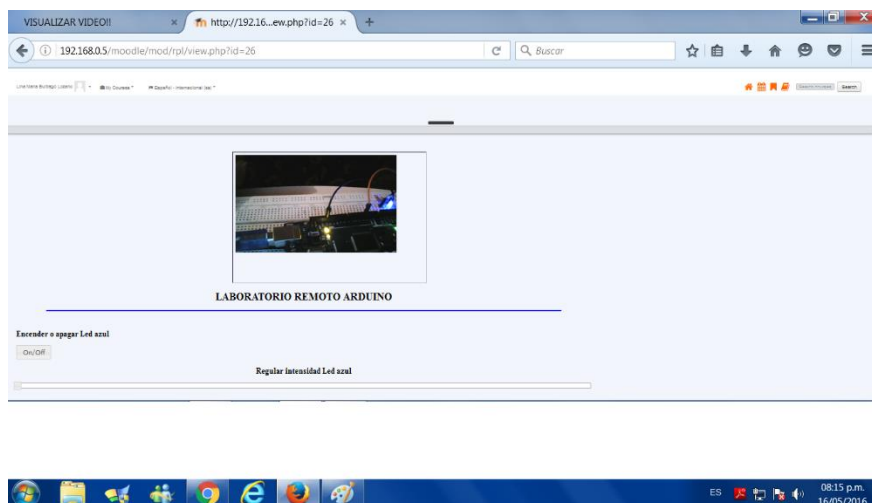
Cuando el estudiante selecciona el curso en que se ha inscrito le mostrara las actividades propuestas por el docente.



**Figura 28.** Actividades creadas por el docente.

**Nota Fuente:** (2016). Creadas por el autor.

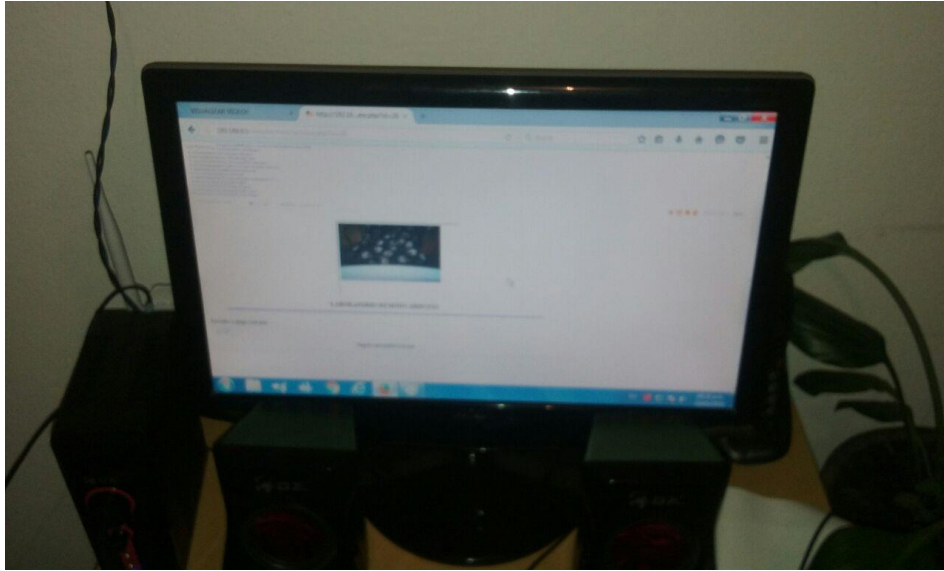
Luego que esta el listado de actividades el estudiante ingresa al laboratorio de prueba y le da clic. Automáticamente se activara la cámara web y se mostrara la interfaz del laboratorio remoto.



**Figura 29** Interfaz de laboratorio remoto

PLUGIN RPL QUE PERMITE LA REALIZACIÓN DE LABORATORIOS REMOTOS

**Nota Fuente:** *Laboratorio a manipular por parte del estudiante*



**Figura 30.** Ordenador parte Cliente.

**Nota Fuente:** *Ordenador donde el estudiante manipula el laboratorio remoto.*

## CAPITULO VI

### 6.1 CONCLUSIONES

Al finalizar este proyecto podemos concluir lo siguiente:

- El plugin RPL proporciona a docentes y estudiantes una herramienta para la interacción y elaboración de laboratorios remotos en tiempo real. Esta herramienta permite a la licenciatura en electrónica innovar en la metodología de práctica para realizar los laboratorios de diferentes asignaturas.
- Esta herramienta permite explorar otros niveles educativos, a través de la plataforma educativa Moodle ya que cuenta con una variedad de plugin prácticos, que aumenta la motivación e iniciativa de investigación por parte de los estudiantes.
- Esta herramienta se convierte en un ambiente de aprendizaje virtual permitiendo a los estudiantes aprender y acentuar sus conocimientos desde cualquier punto con acceso a internet.
- Node.js fue un elemento fundamental para la definición del protocolo http de comunicación a través de la librería SerialPort para la recepción y transmisión de datos en tiempo real.
- El modo desarrollador de Moodle permite identificar las líneas de error que se presentaron durante el desarrollo del plugin facilitando la modificación y culminación de este.

## **6.2 RECOMENDACIONES Y TRABAJOS A FUTURO**

Existen muchas posibilidades para plantear el diseño y construcción del plugin RPL, ya que tienen múltiples necesidades y objetivos por cumplir. Para investigaciones o desarrollos futuros se sugiere realizar un control temporal del uso de los laboratorios remotos, en los que se permitan separar los espacios de trabajo. Poder generar laboratorios remotos con otras tarjetas de programación, la cual integre una simulación a partir de Matlab o Labview que muestre la simulación del comportamiento de la planta. Generar un manual de laboratorio que involucren las especificaciones del plugin para subir y desarrollar los laboratorios remotos.

La intención principal de este proyecto es que el usuario que manipule RPL construya sus propios laboratorios a trabajar y que a medida que surjan nuevas necesidades se vayan integrando al plugin.

### 6.3 REFERENCIAS BIBLIOGRAFICAS

- Almenara C, Barroso Osuna J, Romero Tena R, Llorente Cejudo M, Román Gravan P. (2007). Definición de Nuevas Tecnologías [en línea] OCW de la Universidad de Sevilla, Facultad de Ciencias de la Educación. España. Rev. (Noviembre 2015).
- Cobo Romaní JC. (2009) El concepto de tecnologías de la información. Benchmarking sobre las definiciones de las TIC en la sociedad del conocimiento.
- Development: lib/formslib.php Form Definition, [http://docs.moodle.org/en/Development:lib/formslib.php\\_Form\\_Definition](http://docs.moodle.org/en/Development:lib/formslib.php_Form_Definition) (noviembre 2015).
- Development: XMLDB defining an XML structure, [http://docs.moodle.org/en/Development:XMLDB\\_defining\\_an\\_XML\\_structure](http://docs.moodle.org/en/Development:XMLDB_defining_an_XML_structure) Rev: (noviembre 2015).
- Herrera, N. (2008). Incorporando las TICS en el Aula ¿Por dónde empezar? Revista eleducador. pag.11-14.
- Instalación de Moodle. (2015) [http://docs.moodle.org/es/Instalaci%C3%B3n\\_de\\_moodle](http://docs.moodle.org/es/Instalaci%C3%B3n_de_moodle). Rev: (Noviembre 2015).
- Laboratorio de automática vía internet que llamaron (LAVI), Revista iberoamericana de automática e informática industrial, ISSN: 1697-7912. Vol. 2, Numero 2, Abril 2005. Pág. 30-35.
- Manuales de Moodle. (2015) [http://docs.moodle.org/es/Manual\\_de\\_Estilo\\_de\\_C%C3%B3digo](http://docs.moodle.org/es/Manual_de_Estilo_de_C%C3%B3digo). Rev. (Noviembre 2015).
- Moodle (2015). Disponible en: <http://moodle.org>. Rev. (Marzo 2015).
- Modulos para Moodle <http://www.genbetadev.com/desarrollo-web/escribiendo-modulos-para-moodle-introduccion>. Rev. (Marzo 2015).

- Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura. UNESCO. (2004). Las tecnologías de la información y la comunicación en la formación docente.
- PASCO <http://www.pasco.com> Rev. (Noviembre 2015).
- Reyes, M. (2002). MOODLE, UNA PLATAFORMA FORMATIVA CON GRAN PROYECCIÓN EN LOS NUEVOS MODELOS DE ENSEÑANZA.
- Sánchez, José. (2005) plataformas tecnológicas para el entorno educativo. Acción pedagógica (Nº 14). Recuperado de <http://dialnet.unirioja.es/>. Rev. octubre 2015.