

Integración basada en estándares de un laboratorio remoto en una plataforma de gestión de aprendizaje

Lucena, E.^{a,*}, Ruano, I.^b, Estévez, E.^a Gómez, J.^a Gámez, J.^a

^a *Departamento de Ingeniería de Sistemas y Automática, Universidad de Jaén, Paraje Las Lagunillas s.n., 23070, Jaén, España.*

^b *Departamento de Ingeniería de Telecomunicación, Universidad de Jaén, Paraje Las Lagunillas s.n., 23070, Jaén, España.*

To cite this article: Lucena, E., *, Ruano, I., Estévez, E. Gómez, J. Gámez, J. 2023. Standards-based integration of a remote lab into a learning management platform. XLIV Jornadas de Automática, 231-236. <https://doi.org/10.17979/spudc.9788497498609.231>

Resumen

Los laboratorios de tipo online tienen cada vez más aceptación dentro de la educación universitaria relacionada con las ciencias, tecnologías, ingenierías y matemáticas (CTIM o STEM en inglés), donde el trabajo práctico es de vital importancia para los estudiantes. El presente trabajo trata un ejemplo de integración de un laboratorio online en un sistema de gestión de aprendizaje (LMS-Learning Management System). Los estándares SCORM (Shareable Content Object Reference Model) y LTI (Learning Tool Interoperability) ofrecen la posibilidad de integrar laboratorios online en un LMS. En este trabajo, se ha tratado un sistema barra-bola convencional para adaptarlo y mostrar un ejemplo de integración de un laboratorio online remoto al LMS de la Universidad de Jaén usando el estándar SCORM y el protocolo RIP (Remote Interoperability Protocol) que ofrece una práctica con gamificación a los alumnos.

Palabras clave: e-learning, plataforma de aprendizaje, integración de sistemas, laboratorio online, LMS, SCORM, LTI, estándar, RIP.

Standards-based integration of a remote lab into a learning management platform

Abstract

Online labs are increasingly accepted within science, technology, engineering and mathematics (STEM) college education, where practical work is vitally important to students. This paper deals with an example of integrating an online laboratory into a Learning Management System (LMS). For the integration of these, the use of standards is essential. The SCORM (Shareable Content Object Reference Model) and LTI (Learning Tool Interoperability) standards offer the possibility of integrating online laboratories in an LMS (Learning Management System). In this work, a conventional bar-bola system has been treated to adapt it and show an example of integration of a remote online laboratory to the LMS of the University of Jaén using the SCORM standard and the RIP (Remote Interoperability Protocol) protocol that offers a practical with gamification to students.

Keywords: e-learning, learning platform, systems integration, online laboratory, Learning Tool Interoperability (LTI), Learning Management Systems (LMS), standard, Remote Interoperability Protocol (RIP).

1. Introducción

El uso de laboratorios docentes es fundamental para la realización de trabajos prácticos en todas las áreas de

enseñanza. En los estudios universitarios STEM (Science, Technology, Engineering and Mathematics), este uso es mucho más habitual y relevante (Clough, 2002). Sin embargo, la puesta en marcha y el mantenimiento de laboratorios lleva

consigo un coste elevado, además de ser imprescindible el uso de un espacio físico cuando se trata de un laboratorio presencial. Este último condicionante, se ha visto agravado por la situación de pandemia mundial vivida durante los últimos años recientes, que ha imposibilitado el uso de los espacios universitarios por parte de los alumnos. Sin embargo, gracias a ello se ha podido demostrar y ratificar las amplias ventajas de los laboratorios online dentro de la enseñanza (Corter et al., 2007), (Ma & Nickerson, 2006).

Los laboratorios online pueden considerarse un claro ejemplo del uso eficaz y beneficioso de las Tecnologías de la Información y las Comunicaciones (TIC) en la enseñanza. Actualmente, el elemento TIC más usado y más determinante en la docencia son las plataformas de docencia o LMS (Learning Management System), también conocidas como entornos de aprendizaje virtual o VLE (Virtual Learning Environment).

Los LMS se encuentran plenamente instaurados en las instituciones universitarias, constituyen el elemento TIC principal de la docencia actual. Su uso es extensivo no solo en la docencia a distancia, ya que constituye un soporte esencial para la docencia presencial actual. Por estos motivos los estudiantes se encuentran totalmente habituados a su uso. Es lógico pensar que cuando los alumnos deban trabajar en los laboratorios online, sea beneficioso que se presenten integrados con el LMS institucional que dominan y usan a diario.

En el entorno tecnológico, la finalidad de un estándar es implantar normas que permitan el entendimiento y la colaboración entre los diferentes sistemas. Existen múltiples estándares de e-learning que pueden utilizarse con el fin de integrar los laboratorios online en los LMS. Los estándares de e-learning más extendidos, usados y soportados por los LMS son SCORM (Shareable Content Object Reference Model) y LTI (Learning Tool Interoperability). Estos estándares se han utilizado en diferentes trabajos para la integración de laboratorios online y LMS, y su uso tiene muchas ventajas (Advanced Distributed Learning, 2009).

Debido al uso del LTI los estudiantes pueden acceder a un laboratorio online, que se encuentra alojado en un servidor externo al LMS, como si estuviera en el mismo LMS. Por otro lado, mediante el uso de SCORM los estudiantes pueden acceder a un laboratorio online que se encuentra alojado en el mismo LMS, embebido en el interior de un paquete SCORM. En ambos casos los estudiantes deben identificarse en el LMS, con LTI acceden al laboratorio al iniciar un elemento enlazado con la aplicación del laboratorio, mientras que con SCORM acceden a la aplicación del laboratorio al iniciar un paquete SCORM.

El presente trabajo se centra precisamente en facilitar unas pautas para la integración de un laboratorio a un LMS a través de dichos estándares. Por ello, el segundo apartado indica los pasos a seguir para integrar un laboratorio a través de los estándares mencionados. El apartado 3 se centra en caracterizar y adaptar un sistema barra-bola para posteriormente, en el apartado 4, describir cómo se produce su integración en el LMS de la UJA usando LTI y SCORM. Finalmente, se incluye un apartado de Conclusiones.

2. Metodologías y herramientas para integrar laboratorios online en LMS a través de SCORM

A la hora de integrar un laboratorio en un LMS se puede hacer uso de muchos estándares, muchas veces incluso de más de uno. Destacan LTI, SCORM, xAPI y IEEE1876. Este trabajo se ha centrado en SCORM ya que es el soportado por el LMS de la Universidad de Jaén (PLATEA). La Figura 1 presenta el escenario general de las tecnologías y herramientas involucradas. Cualquier otro LMS que soporte SCORM podría seguir el mismo procedimiento de integración que el explicado a continuación.

Para poder hacer llegar los datos necesarios del laboratorio remoto a la aplicación el protocolo RIP (Routing Interchange Protocol) es fundamental. Dicho protocolo sigue el modelo cliente servidor. Los siguientes sub-apartados detallan las pautas y herramientas para: (1) configurar un servidor RIP; (2) Generar el fichero SCORM; y (3) integrar fichero SCORM en el LMS.

2.1. Instalación y Configuración y puesta en marcha del Servidor RIP

En (RIP, 2023) se puede descargar todo lo necesario para poder configurar un servidor RIP (en laview y phyton). Este trabajo detalla a continuación la configuración del servidor RIP en phyton (Chacón et al., 2015)..

Posteriormente, se crea un entorno virtual en el dispositivo que vaya a usar (e.g. Raspberry Pi 4B), a través de los siguientes comandos:

```
pip3 install virtualenv
virtualenv -p python3 venv
venv/bin/pip install cherrypy ujson RPi.GPIO
```

Creado dicho entorno el servidor está listo para ponerse en marcha (*App.py* responsable de inicializarlo). Previamente, es imprescindible adaptar una serie de ficheros que lo componen que son dependientes de la aplicación (laboratorio remoto).

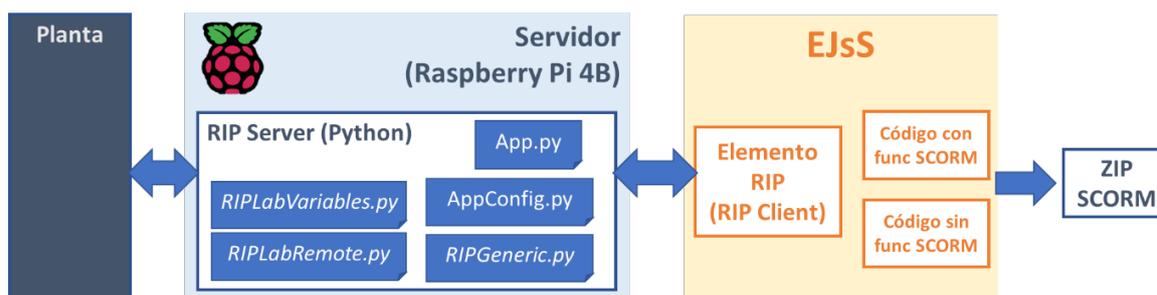


Figura 1. Escenario General de la plataforma para generar fichero SCORM

En concreto:

- *AppConfig.py*. encargado de la parametrización del servidor. Se encuentra en la carpeta principal junto con *App.py*. En dicho fichero se fijan parámetros tan importantes como la IP y puerto y si se desea seguridad. En caso de requerir seguridad el fichero *httpserver* indica las instrucciones necesarias a seguir para crear los certificados autofirmados. En este fichero también se indican las variables indicando su nombre, descripción, valor mínimo y máximo, precisión y si se clasifican como de lectura y/o escritura.

Dentro de la carpeta RIP hay una serie de ficheros entre los que destaca *RIPGeneric.py* que da soporte a la funcionalidad común del protocolo RIP. Este trabajo propone la creación de dos nuevos ficheros que serán totalmente dependientes del caso de uso:

- *RIPLabVariables.py*. Contiene las definiciones variables del laboratorio
- *RIPLabRemote.py*. recoge las funciones del laboratorio remoto. También se intercambia información con el servidor, por ello, al igual que en *AppConfig.py* se deben agregar las variables de tipo lectura y/o escritura.

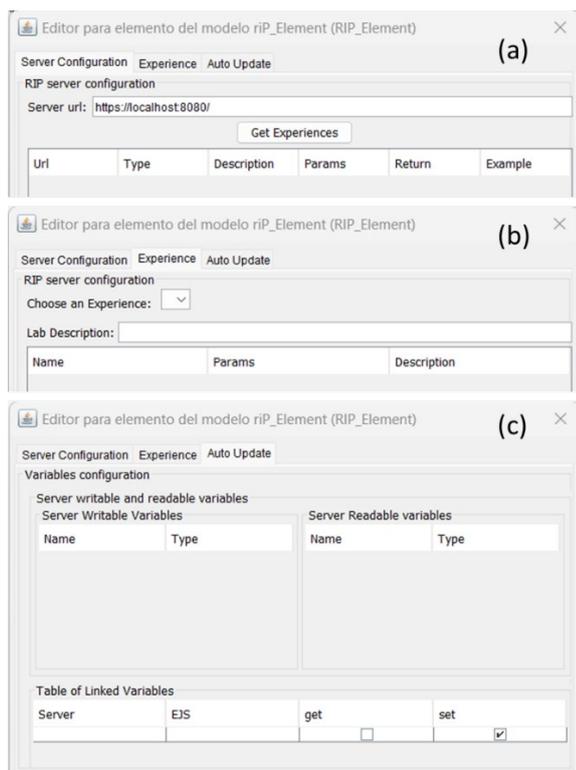


Figura 2. Configuración elemento modelo RIP

2.2. Generación del Modelo SCORM

Easy Java/JavaScript Simulations es una herramienta de software diseñada para la creación de simulaciones informáticas discretas. Existen muchos programas que ayudan a crear otros programas. EJS se diferencia del resto porque está concebido para personas que están más interesadas en el contenido de la simulación, en el fenómeno simulado en sí, y

no tanto en los aspectos técnicos necesarios para llegar a ella. EJS ofrece soporte de diferentes elementos entre ellos, permite añadir elementos entre los cuales un elemento RIP (“**Modelo->Elementos**”). De esta manera permite al usuario configurar a través de ventanas un cliente RIP. Para ello, hay tres partes: (1) Servidor RIP al que se conecta; (2) variables que se enlazan y (3) enlazar dichas variables con las que existen en el modelo de simulación indicando si son de lectura y/o escritura (Véase Figura 2).

En Figura 2.a ilustra dónde se fija la dirección URL del servidor. Pulsando en *Get Experiences*, se obtiene la información del servidor. En esta ventana, tras introducir la dirección URL del servidor y obtener las experiencias, los campos inferiores se completarán con la misma información que se obtiene al abrir el enlace en el buscador.

En la pestaña ilustrada en Figura 2.b se visualiza la descripción del laboratorio que se haya modificado en los archivos *AppConfig.py* y *RipLabRemote.py*. En el campo inferior se pueden visualizar 3 métodos distintos.

En la pestaña de Auto Update (véase Figura 2.c) se obtienen todas las variables de tipo Writable y Readable que previamente se han definido en *RipLabRemote.py* y *AppConfig.py*. Desde esa misma pestaña en ‘Table of Linked Variables’ se han de poner las variables, con el nombre que tienen en el servidor y con el que tendrán en ejs, por comodidad se recomienda utilizar los mismos. Además, se debe fijar si se hará un *get* o un *set*.

Por otro lado, en “**Modelo->Variables**” tienen que estar agregadas dichas variables para poder trabajar con ellas. Así:

Se tienen que agregar todas las variables que se vayan a enviar y recibir tanto en la tabla de variables como en la tabla de variables vinculadas. También se han de añadir en la tabla de variables aquellas que son necesarias para implementar la lógica del laboratorio remoto, que serán usadas en la ventana de Evolución. Otra instrucción importante a llevar a cabo para poder realizar la conexión siempre que accedamos a la interfaz de manera automática se lleva a cabo en el apartado de inicialización, bastará con indicar *rip.connect()*.

Posteriormente, se ha de acceder a la ventana de HtmlView para crear la interfaz gráfica del laboratorio.

Finalmente, si se desea una integración avanzada (entendiendo como tal a que se produzcan intercambios de información entre el software de laboratorio y el LMS), se puede hacer uso de las funciones JavaScript de la sub-especificación RTE de SCORM, de este modo se puede obtener información como por ejemplo nombre de usuario y su ID para personalizar el funcionamiento del laboratorio. La Figura 3 muestra un ejemplo.



Figura 3. Uso de variables SCORM.

2.3. Integración del LMS con SCORM

Como se ha comentado en el apartado anterior, el modo de integrar un laboratorio online con un LMS es incluyendo el software del laboratorio online dentro de uno de los elementos de aprendizaje del módulo SCORM. De este modo, cuando un usuario acceda al módulo SCORM podrá navegar por sus elementos hasta llegar al elemento de aprendizaje que contiene el software del laboratorio y ejecutarlo. Para ello el software de laboratorio debe estar realizado en una tecnología compatible con los elementos de aprendizaje que componen las páginas SCORM, que en realidad son páginas Web. La tecnología recomendada es JavaScript.

Si se ha usado JavaScript y se desea realizar una integración avanzada se tiene la posibilidad de utilizar la librería JavaScript de SCORM o adaptaciones de la misma trabajo (Ruano et al., 2016) que permitan intercambiar datos entre el laboratorio online y el LMS usando el modelo de datos RTE. De este modo se pueden programar software de laboratorios que tengan en cuenta al usuario para crear prácticas adaptativas, que tengan en cuentas los problemas de accesibilidad y gustos del usuario y sus resultados y progresos previos.

El mismo software de EJS ofrece la posibilidad de exportarlo como paquete SCORM. Este paquete ya podría ser importado en cualquier LMS compatible. Desde el LMS de la UJA debe importarse el fichero ZIP para crear un paquete SCORM en una ubicación accesible por el alumno. Este módulo SCORM incluye en su interior la aplicación del laboratorio (definida en el sub-apartado anterior).

3. Sistema Barra-Bola

Se ha trabajado en un sistema barra-bola que se ha caracterizado y adaptado para ofrecerse como laboratorio online remoto. Este tipo de laboratorio es de gran utilidad para los Grados de Ingeniería Electrónica Industrial, en aquellas asignaturas que tienen relación con el control automático.

El sistema barra-bola está compuesto por tres módulos principales: el motor, la barra y la bola tal y como ilustra la Figura 4. En un sistema barra bola tradicional, la bola se encuentra situada encima de la barra.



Figura 4. Maqueta del Sistema Barra-bola

La función de este sistema es mover la bola sobre la barra. Para ello, el motor rota un ángulo, Θ , y, por consecuencia, la barra se inclina un ángulo, α , que termina generando el movimiento de la bola sobre la barra.

Para el sistema real (maqueta) se han utilizado los siguientes componentes:

- **Actuador:** Un motor en corriente continua con escobillas capaz de funcionar a 12V que contiene una caja de engranajes con relación 70:1. Esta caja de engranajes está compuesta principalmente de engranajes rectos, pero en la primera etapa cuenta con engranajes helicoidales que reduce el ruido y lo vuelve más eficiente. Al motor se le acopla un puente en H por el cual se realiza el movimiento de giro hacia derecha e izquierda del motor.
- **Planta:** Formada por la barra y la bola. Al tratarse de un laboratorio remoto, se han realizado varias modificaciones con respecto a un sistema barra-bola tradicional: (1) Se ha cambiado la barra por un tubo de metacrilato donde se introduce la bola; (2) se ha añadido dentro del tubo de metacrilato, junto con la bola, una mezcla con un cierto porcentaje de glicerina con agua. Así, se impide la caída de la bola, no se requiere una persona vigilando que el sistema esté bien y al ralentizar el sistema es más visual para el alumnado.
- **Sensor de posición:** Se trata de un sensor virtual creado por software por el cual mediante visión se calcula la distancia desde el centro de la bola al centro de la barra

4. Caso de uso: integración del sistema barra bola con el LMS de la UJA

El desarrollo de los laboratorios online del sistema barra-bola (virtual y remoto) se han realizado dentro del proyecto UniLabs (University Network of Interactive Laboratories, <https://unilabs.dia.uned.es/>). En este proyecto están involucradas 14 universidades españolas, entre las que se encuentra la UJA, y está siendo dirigido por la UNED.

Este laboratorio remoto permite a los alumnos trabajar en una práctica presentada en forma de juego, en el que pueden realizar varios intentos para situar la bola en un lugar determinado.

Las variables usadas en el laboratorio remoto pueden ser clasificadas en 4 grupos:

- **Readables:** variables leídas del cliente RIP situado en el EJS, facilitadas por el servidor RIP (véase esquema Figura 1). En concreto se contemplan las siguientes:
 - *time*: tiempo en segundos del servidor.
 - *dist-bola*: posición de la bola en centímetros respecto al centro de la barra.
 - *velocidad*: velocidad a la que el usuario podrá controlar el motor en tanto por ciento.
 - *pmacabada*: booleano que indicará en EJS que la puesta en marcha ha finalizado
 - *tiemporestante*: tiempo que le queda al usuario para cumplir con el objetivo del juego.

- *estadojuego*: es un entero que permite a EJS conocer el estado del juego (0→ESPERA,1→JUEGOENMARCHA, 2→PIERDES,3→GANAS).
- *tiempovictoria*: tiempo que ha tardado el usuario en ganar la partida.
- Writables: variables cuyo valor se envía desde el cliente RIP situado en EJS al servidor. Concretamente:
 - *led*: booleano que se modifica en EJS para encender y apagar el panel led.
 - *giraizq*: booleano que permite girar el motor en sentido antihorario.
 - *girader*: booleano que permite girar el motor en sentido horario.
 - *subirvelocidad*: booleano que permite aumentar la velocidad del motor.
 - *bajarvelocidad*: booleano que permite disminuir la velocidad del motor.
 - *puestaenmarcha*: variable que activa la puesta en marcha para situar la planta en unas condiciones iniciales prefijadas.
 - *modotrabajo*: booleano que nos permite elegir entre modo automático (futuro desarrollo) y manual.
 - *startjuego*: booleano cuya función es indicar que el juego ha comenzado.
 - *reset*: booleano que nos permitirá hacer un reset global.
- Variables globales usadas en el servidor: están definidas en el archivo LABVariables.py, estas variables serán usadas en las distintas funciones del archivo RIPLabRemote.py. Merece la pena destacar las usadas para referenciar los pines de la Raspberry, el setup de estos pines, la frecuencia de la señal PWM que controla la velocidad del motor, el valor de los pulsos del motor que se obtiene del encoder y la posición de la bola obtenida a través de visión por computador.
- Variables usadas en el cliente: Están definidas en EJS. Todas las variables de los dos primeros grupos tienen su equivalente en el cliente EJS y, adicionalmente, se han empleado otras variables para el correcto funcionamiento del interfaz de usuario y para poder realizar las comunicaciones usando el RTE de SCORM.

Por otro lado, la práctica desarrollada en el laboratorio, tiene como finalidad llevar a cabo el control manual de la posición de la bola en la barra, siendo el objetivo final posicionar la bola en el centro de la barra durante un tiempo determinado. Además de la conexión con el servidor RIP, existe una segunda conexión que se establece entre el cliente JS y una cámara IP, que manda imágenes del sistema barra bola en tiempo real para que el alumno pueda observar el sistema y actuar para posicionar la bola correctamente. Esta práctica se presenta al alumnado en forma de juego, el alumno puede obtener una

puntuación que depende del tiempo tardado en posicionar la bola, disponiendo para ello de varios intentos.

La sesión práctica tipo que deben realizar los alumnos es como sigue:

1. En un dispositivo con conexión a Internet debe abrir un navegador y acceder a la web de PLATEA (LMS de la UJA).
2. Identificarse como alumno en PLATEA, navegar por PLATEA para localizar el módulo SCORM en el espacio virtual de la asignatura correspondiente.
3. Abrir el módulo SCORM del Laboratorio remoto, al hacer esto el cliente JS del laboratorio creado en EJS que está incrustado en el módulo SCORM muestra el interfaz gráfico resumido al alumno después de abrir una sesión de comunicación RTE de SCORM con el LMS para obtener el nombre del alumno, su identificación y la versión de SCORM usada.
4. El alumno, en la interfaz gráfica (Figura 5), puede ver la versión de SCORM usada, su nombre y su Identificador en PLATEA y, a continuación, se le presentan botones con los que puede interactuar para realizar las siguientes acciones: 4.1. Acceder al modo manual (botón “MANUAL”, 4.2. Encender el panel led, botón “ENCENDER”, 4.3. Poner en marcha en sistema, botón “Puesta en Marcha”. Tras estas acciones el motor se pone en marcha para girar la barra en contra de las agujas del reloj hasta alcanzar su límite y situar la bola en el extremo izquierdo de la barra (condiciones iniciales).

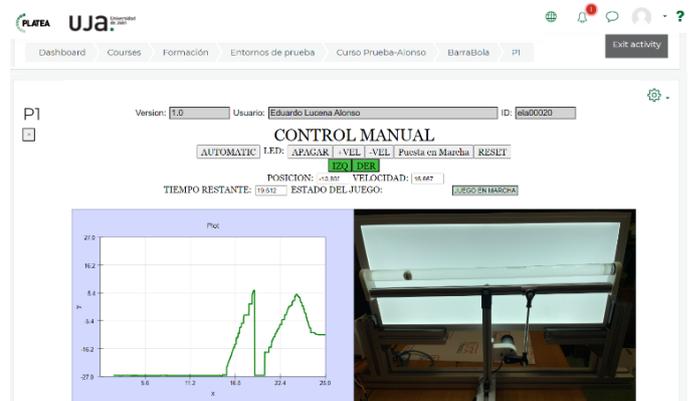


Figura 5. Interfaz Gráfica de Usuario cuando el juego está en marcha.

5. El alumno debe pulsar el botón “START GAME” para iniciar un intento de experimento, en ese momento podrá mover la barra mediante dos botones verdes (“IZQ” y “DER”), aumentar y disminuir la velocidad del movimiento del motor (“+VEL” y “-VEL”), y visualizar en una gráfica la posición de la bola. Una vez terminado el proceso, ya sea porque el usuario lo ha conseguido o porque haya agotado su tiempo queda grabada su puntuación, esto se lleva a cabo con SCORM, cuanto menos tarde en completar el objetivo más elevada será esta.

Tras lo cual el cliente tendrá dos opciones:

- Volver a repetir el proceso si no está conforme con su resultado, para ello deberá de pulsar “RESET” y volver a hacer el proceso exactamente igual.
- Salir de la interfaz gráfica y del módulo SCORM, pulsando el botón de PLATEA “Exit Activity”.

Las calificaciones obtenidas por cada alumno podrán ser revisadas posteriormente por el profesorado en el apartado de informes del módulo SCORM en el que se encuentra el laboratorio.

5. Conclusiones

En las titulaciones STEM (Matemáticas, Ciencias Tecnológicas e Ingenierías) el trabajo práctico que realizan los alumnos en su formación es esencial para que comprendan y asimilen correctamente los conceptos teóricos. Este trabajo práctico se ha desarrollado tradicionalmente en laboratorios presenciales, aunque desde hace ya bastante tiempo, los laboratorios online han resultado ser una herramienta complementaria que puede aportar múltiples ventajas. La situación vivida en los últimos cursos debido a la pandemia del COVID-19 ha obligado a desarrollar los laboratorios online convirtiéndolos, en muchos casos, en la única posibilidad que tenían los alumnos de realizar los trabajos prácticos.

Este trabajo incluye recomendaciones que ayudan a lograr la integración de laboratorios online en un LMS (Sistemas de gestión de aprendizaje) utilizando LTI y SCORM, los dos estándares de e-learning más extendidos y ampliamente soportados por la mayoría de los LMS existentes. En concreto, se ha presentado unas pautas y herramientas para integrar un laboratorio remoto a través de SCORM en el LMS institucional de la Universidad de Jaén, el cual es un sistema Moodle LMS de la UJA (PLATEA). Adicionalmente se ha incluido como ejemplo de integración un laboratorio remoto de un sistema barra-bola. La planta física de este laboratorio se ha obtenido como adaptación de un sistema barra bola tradicional para tener en cuenta las características especiales que debe presentar un laboratorio online. Se ha usado un servidor RIP Python configurado en un microcontrolador Raspberry Pi 4B donde se han incluido los programas que permiten monitorizar y controlar la planta y una cámara IP para visualizar la planta del sistema barra-bola. El programa cliente JavaScript ha sido desarrollado con EJS, incluye un elemento de tipo cliente RIP para intercambiar con el servidor RIP las variables que permiten monitorizar la planta y controlarla remotamente y muestra las imágenes de la planta captadas por la cámara IP. El programa cliente se ha incrustado en un paquete SCORM para lograr su integración en PLATEA, este programa hace uso de las funciones de la sub-especificación RTE de SCORM para identificar a los alumnos y monitorizar el trabajo realizado.

El laboratorio incluye una práctica experimental que se ofrece a los usuarios en forma de juego y almacena en

PLATEA la calificación correspondiente al desempeño logrado por el alumno en la misma, permitiendo que el profesorado pueda posteriormente visualizar los resultados obtenidos por cada alumno.

Agradecimientos

Los autores quieren agradecer la subvención parcial de este trabajo a través de los proyectos PIMED01_201921, PID2019-110291RB-I00.

Referencias

- Advanced Distributed Learning. (2009). *SCORM 2004 4th Ed. (Shared Content Object reference model)*. <https://www.adlnet.gov/adl-research/scorm/scorm-2004-4th-edition/>
- Chacón, J., Farias, G., Vargas, H., Visioli, A., & Dormido, S. (2015). Remote Interoperability Protocol: A bridge between interactive interfaces and engineering systems. *IFAC-PapersOnLine*, 48(29), 247-252. <https://doi.org/10.1016/j.ifacol.2015.11.244>
- Clough, M. P. (2002). Using the laboratory to enhance student learning. *Learning Science and the Science of Learning*, JANUARY 2002, 85-94.
- Corter, J. E., Nickerson, J. V., Esche, S. K., Chassapis, C., Im, S., & Ma, J. (2007). Constructing reality: A Study of Remote, Hands-On, and Simulated Laboratories. *ACM Transactions on Computer-Human Interaction*, 14(2), 7-es. <https://doi.org/10.1145/1275511.1275513>
- Ma, J., & Nickerson, J. V. (2006). Hands-on, simulated, and remote laboratories: A Comparative Literature Review. *ACM Computing Surveys*, 38(3), 7-es. <https://doi.org/10.1145/1132960.1132961>
- RIP Server, (2023). Disponible en: <https://github.com/UNEDLabs/rip-python-server> último acceso: junio de 2023.
- Ruano, I., Cano, P., Gamez, J., & Gomez, J. (2016). Advanced LMS Integration of SCORM Web Laboratories. *IEEE Access*, 4, 6352-6363. <https://doi.org/10.1109/ACCESS.2016.2587805>
- Ruano-Ruano, I., Gamez, J., Dormido, S., & Gomez, J. (2016). A Methodology to Obtain Learning Effective Laboratories with Learning Management System Integration. *IEEE Transactions on Learning Technologies*, 9(4), 391-399. <https://doi.org/10.1109/TLT.2016.2594771>