

Laboratorio remoto de Automática. Una solución de bajo coste basada en Raspberry PI y Arduino

Arévalo, Vicente ⁽¹⁾; García-Morales, Isabel ⁽¹⁾; Cano, Juan

(1) Dpto. de Ingeniería de Sistemas y Automática, Universidad de Málaga, {varevalo,isabelgm}@uma.es

Resumen

Este trabajo presenta un prototipo de laboratorio remoto construido a partir de componentes hardware de bajo coste y software *open-source*. El sistema propuesto tiene como objeto rebajar los costes necesarios para llevar a cabo prácticas de laboratorio en diversas asignaturas impartidas en el Dpto. de Ingeniería de Sistemas y Automática de la Universidad de Málaga. El prototipo se ha utilizado con éxito en una experiencia piloto realizada en la asignatura de Automática del Grado de Ingeniería en Tecnologías Industriales.

Palabras clave: Laboratorio remoto; bajo coste; código abierto; control automático; automática.

I. Introducción

La crisis socio-económica que se está sufriendo en estos últimos años ha traído consigo importantes recortes, entre otras muchas áreas, en Educación. La falta de recursos ha impulsado a los docentes a suplir la falta de medios a través de la innovación en la metodología docente. Esto se manifiesta especialmente en los estudios de grado medio y superior con alto grado de experimentalidad como son las ingenierías (grados y másteres), ciclos formativos tecnológicos, etc. donde el coste de los medios materiales utilizados en las sesiones prácticas son más elevados.

En el ámbito universitario, donde los autores desarrollan su labor docente, y más concretamente, el Departamento de Ingeniería de Sistemas y Automática de Universidad de Málaga (el Dpto. en adelante), el coste de actualización y sustitución de equipos de laboratorio es muy importante. Dicho coste puede oscilar desde unos pocos cientos de euros (ordenadores, multímetros, osciloscopios, etc.) a varias decenas de miles de euros (robots manipuladores, autómatas programables, máquinas CNC, etc.).

Esta problemática relacionada con el coste de adquisición y renovación de los equipos, se acentúa por el elevado número de alumnos matriculados en carreras técnicas. Como consecuencia, en las sesiones de prácticas de los alumnos se evidencia la falta de equipos para todos ellos y, por tanto, una disminución de la calidad de la enseñanza.

Este es el caso, por ejemplo, de *Automática*, una asignatura que forma parte de los planes de estudio de diversas titulaciones de grado impartidas en la Escuela de Ingenierías Industriales de la Universidad de Málaga y cuyo número de alumnos matriculados ha superado los 700 en el presente curso académico. El objetivo principal de dicha asignatura es lograr que el alumno conozca y aprenda los conceptos básicos de los sistemas de control automático. El estudio de este tipo de sistemas tiene su punto de partida en el conocimiento de un modelo matemático del proceso que se pretende controlar, continúa con el análisis de sus características temporales y frecuenciales, y finaliza con el aprendizaje de las técnicas clásicas de diseño de sistemas de control. Las prácticas de laboratorio que se realizan en esta asignatura involucran sistemas que, pese a su sencillez (circuitos RC o RLC, motores de corriente continua, etc.), son de vital importancia para que el alumnado comprenda la dinámica de los sistemas de primer y segundo orden ante la aplicación de diversas señales de prueba.

En este contexto, este trabajo presenta una solución hardware-software de bajo coste y *open-source* que tiene por objetivo rebajar los costes de inversión necesarios para llevar a cabo prácticas de laboratorio en diversas asignaturas impartidas en el Dpto. y maximizar la utilización de los recursos disponibles. La solución que se

propone eliminar los aparatos de medida y generación de señales mediante el uso de dispositivos de bajo coste y ofrece, además, un servicio web que permite la realización de las prácticas y el análisis de los resultados tanto en el laboratorio, como desde otras ubicaciones mediante acceso remoto.

La Fig. 1 muestra un diagrama funcional de la propuesta en la que se dispone de un conjunto de sistemas a estudiar cuya información es registrada por el bloque de *Adquisición de Datos*. Esta información es procesada por el *Interfaz de Datos* de manera que, a través de un *Servidor Web*, es accesible por los usuarios a través de dispositivos que pueden ser tanto locales como remotos.

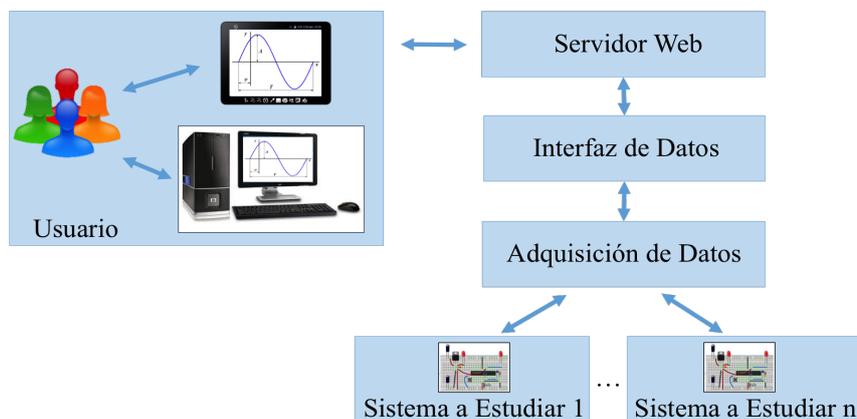


Figura 1. Diagrama funcional del sistema propuesto

Las ventajas de esta solución son evidentes: se abaratan los costes de adquisición y puesta en marcha de los sistemas involucrados en las prácticas; se maximiza la utilización de los recursos de prácticas disponibles, ya que los alumnos podrán acceder a la plataforma siempre que lo deseen (y haya recursos libres); permite la realización de pruebas automatizadas de evaluación en base a los resultados obtenidos por los alumnos; y por último, cada alumno posee un mayor control de su proceso de aprendizaje al poder elegir su propio ritmo en función de su capacidad y circunstancias personales.

De esta manera, el artículo se organiza de la siguiente forma: en la sección 2 se describe la plataforma de laboratorio remoto propuesta en el presente trabajo; en la sección 3 se muestra la implantación realizada en la asignatura *Automática* del Grado en Ingeniería en Tecnologías Industriales de la Universidad de Málaga; la sección 4 muestra algunos resultados preliminares obtenidos a partir de encuestas de satisfacción de los alumnos. Por último, se presentan algunas conclusiones finales y se proponen algunas líneas de mejora.

II. Descripción del laboratorio remoto de prácticas

El diseño del laboratorio remoto de prácticas se centra fundamentalmente en la elección de los dispositivos de bajo coste que van a asumir la función de los bloques que aparecen en la Fig. 1 con los nombres de Servidor Web, Interfaz de Datos y Adquisición de Datos. Los dispositivos que asumen dicha función son *Raspberry PI* para los dos primeros bloques y *Arduino* para el tercero. Ambas plataformas son muy conocidas en la comunidad DIY (acrónimo de los términos en inglés, *Do It Yourself*) y poseen una excelente calidad/precio. Las características técnicas básicas de dichas plataformas son las siguientes:

- Raspberry PI, versión 2, modelo B (RASPBerryPI, 2017). Se trata de un microordenador que dispone de un SoC Broadcom BCM2836 con un chip ARM Cortex A7 a 900Mhz. Posee una salida HDMI de audio y vídeo, 17 pines GPIO, 4 puertos USB y sistema operativo Linux.
- Arduino, modelo MEGA (ARDUINO, 2017). Este dispositivo dispone un microcontrolador ATmega1280 a 16MHz, cuenta con 54 pines GPIO, 16 entradas analógicas y, a diferencia de Raspberry PI, no cuenta con sistema operativo, ni salida de audio/vídeo. Se ha optado por este modelo porque cuenta con un mayor número de pines GPIO y entradas analógicas.

En el sistema diseñado, el dispositivo Arduino, que como se ha mencionado, actúa como un sistema de adquisición de datos a través del envío y recepción de señales del sistema que se está analizando, se conecta a Raspberry PI mediante un puerto USB. La comunicación entre ambos dispositivos se realiza a través de una

conexión serie UART capaz de ofrecer, en el mejor de los casos, una velocidad máxima de 250 KBaudios. Este dato condiciona la forma en la que se realiza la presentación de datos al usuario que rara vez podrá analizar sistemas en tiempo real, salvo que se esté trabajando en modo local (no de forma remota) y el tiempo de respuesta del sistema que se esté analizando sea lo suficientemente lento.

En la plataforma Raspberry PI se llevan a cabo las funciones de Servidor Web y de Interfaz de Datos, de manera que este dispositivo es el encargado tanto de gestionar la información desde/hacia Arduino como de ofrecer al usuario un interfaz en el que se recoja toda la información relacionada con los recursos disponibles. Para ello, además, la Raspberry PI debe albergar la gestión de usuarios de la plataforma, la puesta en marcha de prácticas, la presentación de resultados (ya sea en tiempo real o en lotes) así como, opcionalmente, la gestión de pruebas de conocimiento relacionadas con las prácticas.

Como se ha mencionado anteriormente, Arduino no cuenta con las especificaciones técnicas necesarias para disponer de un sistema operativo propio. Esta característica tiene como ventaja que la ejecución del código es muy eficiente, pero como contrapartida dificulta su programación. El lenguaje que se utiliza para programar de forma nativa Arduino es C++, aunque cada vez hay más iniciativas destinadas a facilitar este proceso con lenguajes de más alto nivel, como es el caso, por ejemplo, de *Processing* (PROCESSING, 2017) o *Wiring* (WIRING, 2017). Las ventajas de estos lenguajes, no sólo radica en su facilidad de uso, sino que además disponen de librerías que amplían la capacidad de interconexión de Arduino con otras plataformas.

En este trabajo se aprovecha la capacidad de *Processing*, un lenguaje de programación y entorno de desarrollo integrado de código abierto basado en Java, para comunicarse, entre otras plataformas, con *Spacebrew* (SPACEBREW, 2017). *Spacebrew* es un servicio web que permite el intercambio dinámico de información entre soluciones Web generadoras/consumidoras de información. La aplicación *Processing*, apoyándose en las librerías proporcionadas por *Spacebrew*, permite comunicar de forma transparente Arduino con la aplicación Web que se ejecuta en el bloque Servidor Web.

El funcionamiento de la dupla *Processing-Spacebrew* es muy sencillo, *Spacebrew* implementa un mecanismo de “pizarra virtual” en el que se recogen “suscriptores” y “publicadores” de información. El programa que ejecuta Arduino, implementado en *Processing*, crea una serie de objetos “publicadores” que serán los encargados de recibir la información leída a través de las entradas digitales o analógicas de Arduino y transferirla a *Spacebrew*. De forma análoga, el programa crea una serie de objetos “suscriptores” que serán los encargados de recibir la información (señales y datos de configuración) de *Spacebrew* y escribirla en las salidas digitales de Arduino. La transferencia de información se puede realizar tanto en lotes como en tiempo real.

La ventaja más importante que ofrece de *Spacebrew* con respecto a otras alternativas es que el intercambio de datos a través de la “pizarra virtual” es reconfigurable de forma dinámica, es decir, en tiempo de ejecución. De este modo, el usuario puede acceder a todo tipo de montajes tan sólo seleccionando los “suscriptores” (entradas de Arduino disponibles) y “publicadores” (salidas de Arduino) asociados a la práctica que desea estudiar, o realizar sus propios montajes utilizando los “suscriptores”/“publicadores” como entradas/salidas virtuales de Arduino. En este proyecto no se explota, aún, esta interesante característica, por lo que el rol que desempeña *Spacebrew* en la actualidad es el de mero adaptador/intercambiador de información entre *Processing* y la aplicación Web.

La aplicación Web es la encargada de gestionar/presentar la información generada por los montajes de prácticas a los usuarios. El acceso a la misma no es libre y requiere de un usuario y clave que es suministrada por el profesor de prácticas. El usuario sólo tiene acceso a las prácticas disponibles según su perfil, también asignado por el profesor. Cada práctica cuenta con una página Web diseñada que incluye el enunciado de la misma, los parámetros necesarios para la ejecución, las preguntas que debe responder el alumno (si procede) y los resultados de la práctica (imagen de vídeo, gráficas, valores numéricos, etc.). La aplicación gestiona los recursos disponibles (el conjunto de montajes de prácticas) y los asigna de forma dinámica y transparente para el usuario.

Por último, mencionar que tanto *Processing* como *Spacebrew* se ejecutan en Raspberry PI, pero requieren de software adicional para su funcionamiento. En particular, el *runtime* de Java y el servidor de aplicaciones

Javascript Node.js (KIESSLING, 2017). Para desplegar la aplicación Web en Raspberry PI se requiere la instalación de LAMP (acrónimo de Linux, Apache, MySQL y PHP) (PARKER, 2015), una pila de servicios web ampliamente utilizada en el desarrollo web: el servidor web (Apache), el gestor de bases de datos relacionales (MySQL) y el lenguaje de programación web (PHP). Es importante mencionar que todo el software utilizado en el desarrollo de esta plataforma es gratuito.

La Fig. 2 muestra un esquema simplificado del laboratorio de prácticas remoto diseñado en el que se hace patente la posibilidad de escalar el sistema, ya que bastaría con disponer de distintos sistemas de Adquisición de Datos a través de la conexión de más dispositivos Arduino a la Raspberry PI a través de un concentrador USB. Cada uno de los dispositivos Arduino precisaría de una aplicación *Processing* de manera que los “publicadores”/“suscriptores” de cada Arduino estén disponibles en *Spacebrew* para su uso en la aplicación Web. Además cada Arduino sería capaz de gestionar más de un Sistema a Estudiar, en función de la complejidad de los mismos.

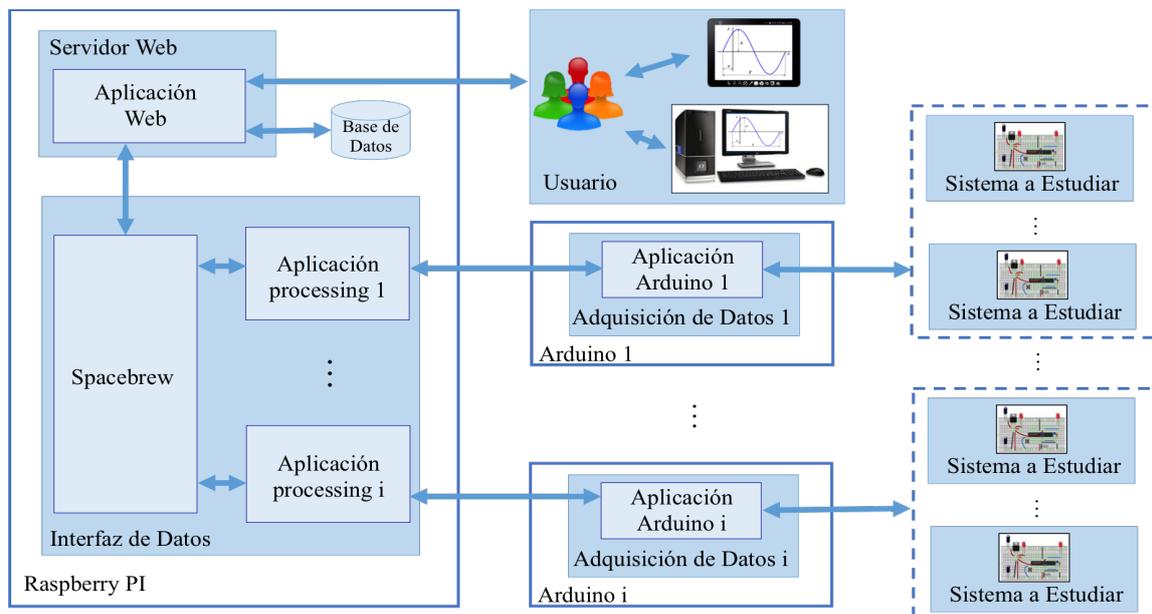


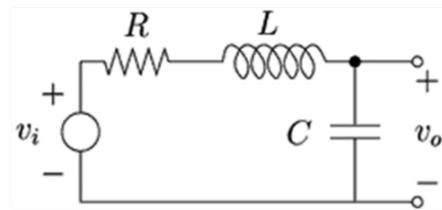
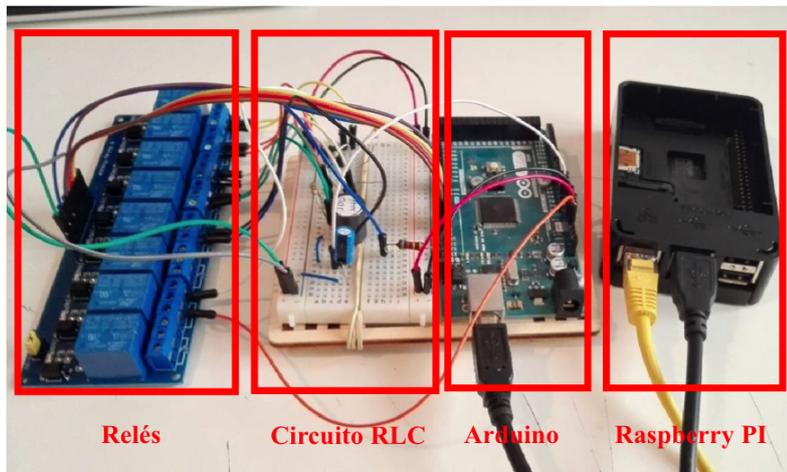
Figura 2. Esquema simplificado del sistema de prácticas remoto.

III. Estudio de la respuesta temporal de un circuito RLC

En esta sección se detalla una experiencia desarrollada durante el curso 2016-2017 en el que se ha empleado el prototipo de laboratorio de prácticas descrito en el apartado anterior con el objeto de llevar a cabo una práctica que permita el estudio de la respuesta temporal de un circuito RLC. El estudio de dicho sistema forma parte del bloque práctico de la asignatura *Automática*, que pertenece al Grado de Ingeniería en Tecnologías Industriales, impartido en la Escuela de Ingenierías Industriales de la Universidad de Málaga. El montaje de la práctica se muestra en la Fig. 3. En dicha figura se muestra como el circuito RLC se ha implantado sobre una placa de prototipado, en el cual se permite establecer diferentes configuraciones en función de la activación y desactivación de distintas resistencias a través de relés. Arduino se conecta a este montaje, de manera que es capaz de suministrar la alimentación necesaria al circuito (v_i), activar y desactivar las distintas resistencias a través de los relés así como registrar la tensión de salida (v_o) que se pretende analizar. Finalmente, la plataforma Raspberry PI se conecta al Arduino de manera que le proporciona tanto la alimentación necesaria como los datos de configuración de la práctica y, por otro lado, obtiene los datos adquiridos del circuito.

El desarrollo de la práctica es muy sencillo: se genera una tensión de entrada v_i (un escalón de amplitud conocida), y se mide la tensión de salida en los terminales del condensador C , v_o . La constante de tiempo del sistema y, por tanto, la respuesta temporal depende del valor de la resistencia del circuito R . El valor de R se calcula a través de cuatro resistencias conectadas en serie que se activan o desactivan en función de la respuesta que se desea analizar y cuya selección se realiza a través de las opciones mostradas a la izquierda de la Fig. 4.a. La activación de las resistencias se realiza mediante un paquete de configuración que se envía a Arduino a través de un “suscriptor” (ver fig. 4.b). La salida del experimento se transfiere (en forma de array

de datos) a la aplicación web a través de un “publicador”. La información recibida es mostrada en una gráfica para que el alumno la analice, tal y como se muestra en la parte derecha de la Fig. 4.a.



$$R_1=100 \Omega, R_2=330 \Omega, R_3=470 \Omega, R_4=690 \Omega$$

$$R=R_1+R_2+R_3+R_4$$

$$L(\text{bobina})=100\text{mH}$$

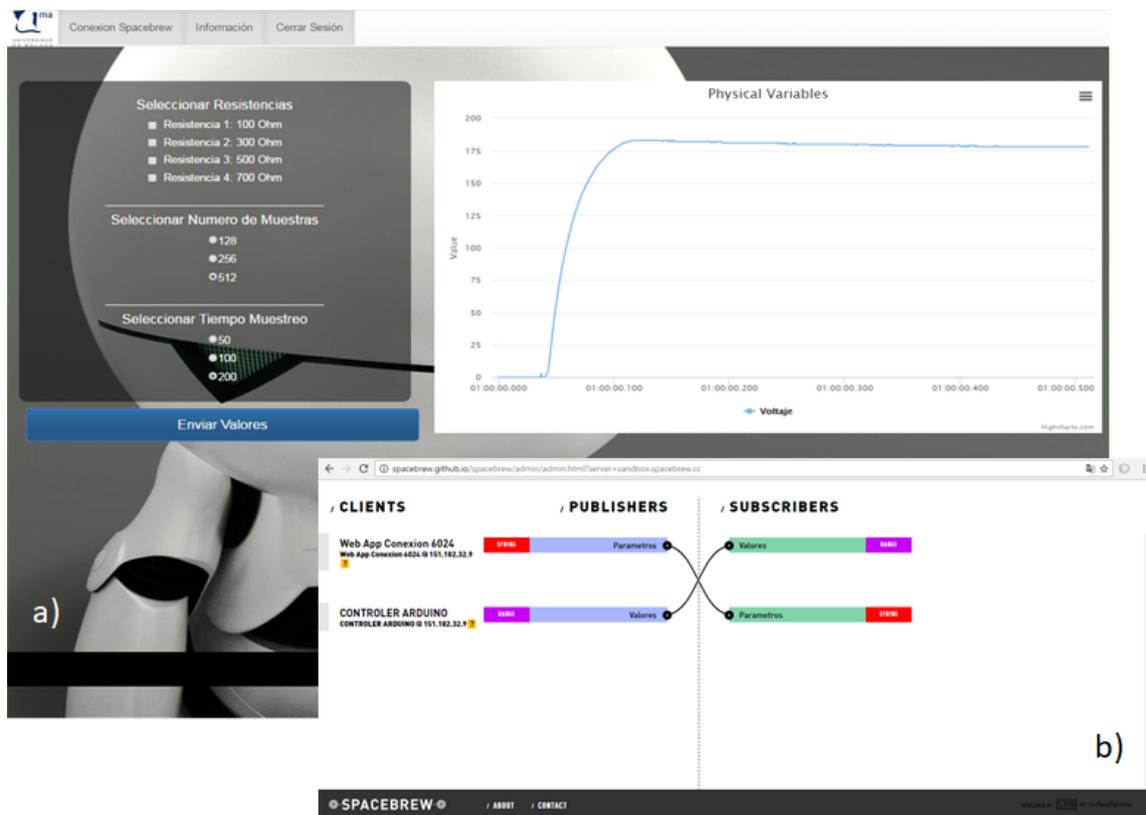
$$C(\text{condensador})=100\mu\text{F}$$

a)

b)

Figura 3. a) Implantación del laboratorio de prácticas remoto para el estudio de un circuito RLC. b) Esquema del circuito RLC

La puesta en marcha del laboratorio se ha realizado con un conjunto inicial de 10 Arduinos conectados a otros tantos montajes de circuitos RLC y se ha puesto a disposición del alumnado durante un periodo de 2 semanas, 24h al día. Durante este periodo, la práctica sólo ha estado disponible mediante acceso remoto, esto es, el alumnado no tuvo acceso a los montajes de prácticas. En la siguiente sección se muestran algunos resultados preliminares obtenidos de esta primera experiencia llevada a cabo con el prototipo.



a)

b)

Figura 4. Capturas de (a) la página desarrollada para la realización de la práctica de análisis de respuesta temporal de un circuito RLC y (b) la interconexión de “suscriptores”/“publicadores” entre la aplicación Web y *Processing* realizada a través de *Spacebrew*.

IV. Resultados preliminares y discusión

El número de prácticas puestas a disposición del alumnado en la prueba del prototipo sólo ha sido una, por lo que es difícil cuantificar, por ahora, el beneficio obtenido por alumno en términos académicos. No obstante, durante el periodo que ha estado disponible el laboratorio remoto se han realizado encuestas para conocer el grado de satisfacción de los alumnos y su opinión sobre este nuevo enfoque.

Los resultados obtenidos han sido, en términos generales, muy prometedores. Todos los encuestados han valorado muy positivamente el hecho de que pudiesen realizar la práctica en cualquier momento y lugar; la libertad para poder elegir el dispositivo desde donde conectar (portátil, PC, tablet, etc.); y la facilidad de uso de la aplicación. Sin embargo, han manifestado que el diseño de la aplicación web no es muy atractivo y que no se adapta bien a algunos dispositivos (a los teléfonos móviles, fundamentalmente), esto es, no ofrece un comportamiento *responsive*. Asimismo, se han reportado ciertos problemas de disponibilidad de los montajes en algunos momentos, fundamentalmente en los últimos días del periodo en el que ha estado disponible. Es de valorar el hecho de que se ha solicitado la inclusión nuevos montajes en la aplicación.

En relación a estos comentarios, hay aspectos claramente mejorables en los que habría que trabajar. El diseño de la aplicación y el comportamiento *responsive* de la misma son dos cuestiones prioritarias que habría que abordar a corto plazo. No obstante, los problemas sobre la disponibilidad de montajes en los últimos días son claramente achacables a la costumbre, más que generalizada, de dejar para el último día la entrega de prácticas. No obstante, dado que el número de matriculados en la asignatura de *Automática* en el que se ha implantado la experiencia ronda los 80 alumnos, circunstancia que se repite otras asignaturas que imparte el Dpto, sería conveniente el poder contar con un mayor número de montajes, dado que el costo no es excesivamente importante. Otra estrategia que permitiría solventar esta problemática, sin costes adicionales, consistiría en habilitar la posibilidad de hacer las simulaciones por lotes. De esta forma, si el sistema no cuenta con circuitos libres, el alumno podría solicitar una simulación que es encolada hasta que haya un dispositivo disponible. Los resultados le serían enviados mediante un email que contendría un enlace a la página de resultados.

V. Conclusiones y futuras mejoras

En este artículo se presenta un prototipo de laboratorio de prácticas remoto construido a partir de elementos hardware de bajo coste y gestionado mediante una aplicación desarrollada con software de código abierto. El prototipo ha sido probado satisfactoriamente en la asignatura de *Automática* impartida en el Grado de Ingeniería en Tecnologías Industriales de la Escuela de Ingenierías Industriales de la Universidad de Málaga. La arquitectura del sistema propuesto en este trabajo se caracteriza por flexibilidad y escalabilidad, ya que añadir nuevas placas Arduino es tan fácil como conectarlos a la Raspberry PI mediante un concentrador USB.

En cuanto al trabajo futuro, sería conveniente dotar al sistema de la funcionalidad necesaria para poder responder, con unos tiempos aceptables, a picos de utilización; mejorar la adaptación de la aplicación web a diferentes tamaños de pantalla (comportamiento *responsive*) mediante hojas de estilo y, por último, trabajar en el aspecto estético de la aplicación. Sería interesante, por otro lado, dada la buena acogida del laboratorio remoto adaptar nuevos montajes de prácticas para su utilización el próximo curso académico.

Referencias

- ARDUINO. Arduino Mega Specifications. <https://www.arduino.cc/en/Main/arduinoBoardMega> (consultado 24/07/2017).
- RASPBERRYPI. Raspberry PI 2 Model B Specifications. <https://www.raspberrypi.org/products/raspberrypi-2-model-b/> (consultado 24/07/2017).
- PROCESSING. Processing webpage. <https://processing.org/> (consultado 24/07/2017).
- WIRING. Wiring webpage. <http://wiring.org.co/> (consultado 24/07/2017).
- SPACEBREW. Spacebrew webpage. <http://docs.spacebrew.cc/> (consultado 24/07/2017).
- PARKER, S. (2015). *How to Build a LAMP Server*. USA: CreateSpace Independent Publishing Platform.
- KIESSLING, M. (consultado 24/07/2017). *The Node Beginner Book*. online: <https://www.nodebeginner.org/>.