

Un sistema para la docencia a distancia en asignaturas con hardware real

Vladimir Mateev, Esteban Stafford y Pablo Fuentes ¹

Resumen— La docencia práctica en laboratorio de asignaturas centradas en el *hardware* como las del área de Estructura y Organización de Computadores se ha visto severamente afectada por el COVID-19. En este artículo se introduce un nuevo sistema de laboratorio remoto para la realización de sesiones prácticas basadas en Raspberry Pi ejecutando el sistema operativo RISC OS.

El sistema gestiona tanto la alimentación de los equipos como la entrada/salida realizada a través de dispositivos periféricos, y permite al alumno visualizar e interactuar con el escritorio del equipo remoto y con los dispositivos hardware conectados al mismo. Asimismo, el sistema facilita que un alumno y un profesor puedan visualizar el equipo remoto de forma simultánea en tiempo real, lo que facilita la resolución de dudas y la realización de pruebas de evaluación.

El sistema combina una lógica de control basada en módulos Arduino y conexiones Ethernet con una interfaz web programada en lenguaje PHP. Con estas especificaciones, se ha desarrollado con éxito una prueba de concepto dotada de dos equipos remotos y dos interfaces de entrada.

Palabras clave— Docencia remota, laboratorio, Estructura y Organización de Computadores, Raspberry Pi, RISC OS.

I. INTRODUCCIÓN

LA docencia de asignaturas del área de Estructura y Organización de Computadores ha tenido tradicionalmente un fuerte componente práctico, con sesiones de laboratorio sobre un hardware de referencia o un simulador de la arquitectura empleada como referente. El uso de una arquitectura de referencia ayuda a ilustrar los conceptos explicados, y las sesiones prácticas permiten que el alumno asiente el aprendizaje y lo relacione con el funcionamiento de un computador real. Además, se observa que un enfoque práctico de las asignaturas [12] y el manejo de equipamiento físico de forma directa [5], [10] resulta más atractivo para el alumno, particularmente en los primeros cursos de grado, animándole a continuar el seguimiento de las materias.

En la Universidad de Cantabria se ha utilizado desde 2017 la arquitectura ARM, particularmente en sus versiones v6 y v7, como referencia para dicha docencia. Para la realización de sesiones prácticas de laboratorio que complementen a la enseñanza teórica se escogió la familia de dispositivos Raspberry Pi [16], en particular el modelo 1B+. Las ventajas de este dispositivo son numerosas: emplea una arquitectura relevante en el mercado actual, con un conjunto de instrucciones RISC que reduce la complejidad de la arquitectura del dispositivo y facilita la interacción

con la misma que el alumno ha de realizar durante las asignaturas; se trata de un equipo de muy bajo coste disponible en canales de venta minorista; dispone de una documentación detallada y del soporte de una comunidad de desarrollo muy activa, así como de un ecosistema de dispositivos hardware periféricos muy rico; por último, se trata de un equipo que puede funcionar de forma autónoma, ejecutando un sistema operativo completo.

Para la realización de dichas prácticas se seleccionó el sistema operativo RISC OS por su equilibrio entre usabilidad y prestaciones: se trata de un sistema operativo básico pero que dispone de un entorno gráfico, y que permite al alumno manejar directamente los dispositivos hardware sin imponer capas de abstracción. Además, el sistema cuenta con herramientas de edición de código [17] y generación de ejecutables, como la suite GCC [9]. Para completar las necesidades del alumno para la realización de las sesiones prácticas, y para paliar el déficit encontrado en herramientas de depuración de código, se desarrolló un depurador de código libre llamado UCDebug [11], [7]. Esta herramienta permite el control de la ejecución de códigos de alumnos de forma sencilla, pudiendo detenerlos en cualquier punto. Además, cuenta con una interfaz que muestra el estado de la máquina al ejecutar el código del usuario. Es importante recalcar que no se trata de un simulador sino de un depurador, correspondiendo en todo momento los resultados mostrados al alumno con la ejecución real del código desarrollado.

Una de las características de RISC OS es que se trata de un sistema operativo cooperativo. Esta característica permite que el alumno pueda analizar el comportamiento de su código con interferencias mínimas por parte del sistema; sin embargo, también supone que un código con errores pueda fácilmente acaparar el uso de la máquina, dejando el sistema en un estado no responsivo. La herramienta de depuración supone un gran avance en este sentido, pues permite al alumno detener la ejecución de su código en todo momento, pero no impide por completo que el sistema se pueda quedar bloqueado, por lo que es frecuente que el alumno necesite forzar el reinicio de la máquina desconectando y volviendo a conectar la alimentación.

El enfoque de las sesiones prácticas está fuertemente orientado a fomentar el trabajo autónomo del alumno. Para ello, todos los elementos necesarios para la realización de las prácticas pueden ser fácilmente adquiridos por los alumnos al comienzo de la asignatura, facilitando así la flexibilidad horaria. No obstante, se sigue ofreciendo un horario de sesiones pre-

¹Dpto. de Ingeniería Informática y Electrónica, Universidad de Cantabria, e-mail: vladimir-kirilov.mateev@alumnos.unican.es, {esteban.stafford, pablo.fuentes}@unican.es

senciales en el laboratorio, con equipamiento puesto a disposición por la Universidad, para aquellos alumnos que así lo prefieran.

En dichas sesiones se trabajan dos grandes bloques de conocimiento: el aprendizaje de los principios de funcionamiento de un computador a través de la programación de códigos en lenguaje ensamblador ARM, y el manejo de la gestión de Entrada/Salida mediante la programación de *drivers*. Para ello, se trabaja con un editor de textos con resaltado de sintaxis para la generación de códigos en lenguaje ensamblador. Posteriormente se ensamblan y enlazan dichos códigos mediante la suite de compilación GCC [9], para obtener un fichero ejecutable con el que poder realizar el proceso de depuración del código generado.

El planteamiento inicial del puesto de trabajo de laboratorio es emplear el equipo Raspberry Pi conectando el equipo directamente a los periféricos necesarios (monitor, teclado y ratón), sin necesidad de un PC convencional. Sin embargo, pronto se observa que muchos de los alumnos que deciden adquirir el equipamiento para completar la formación en casa no disponen de un puesto completo para manejar la Raspberry Pi, ya que utilizan como ordenador de trabajo un portátil. Para mitigar este obstáculo se decide ofrecer la opción de manejar el equipo de forma remota a través de un PC convencional mediante una conexión de computación virtual en red (*Virtual Network Computing*, VNC). Se ofrece a los alumnos una ISO del sistema RISC OS en la que se encuentra arrancado por defecto un servidor VNC [15]. Conectando el PC a la Raspberry Pi mediante un cable Ethernet, y utilizando un programa cliente VNC en el PC, es posible visualizar e interactuar desde el PC en tiempo real con el escritorio del sistema RISC OS corriendo en la Raspberry Pi.

A. Impacto de la pandemia y adaptación de las sesiones prácticas

Con la erupción de la pandemia causada por el SARS-CoV-2 y el comienzo del estado de alarma y las medidas de confinamiento domiciliario adoptadas en marzo de 2020, se hace necesario adaptar la organización de las asignaturas en curso por la imposibilidad de los alumnos para acceder a las instalaciones de la Universidad. En el caso de la asignatura Introducción a los Computadores, del primer curso del Grado en Ingeniería Informática, esto supone entre otros cambios una modificación del formato de trabajo de las sesiones de laboratorio, que tienen un fuerte peso en la asignatura. En previsión de dicho escenario, aunque sin ser capaces de predecir la severidad y duración de las medidas, se adopta un sistema de préstamo de equipos la semana previa al comienzo del estado de alarma. Mediante dicho sistema, aquellos alumnos que no dispongan de su propio equipo Raspberry Pi pueden solicitar el préstamo de uno propiedad de la Universidad para llevárselo a casa.

Esta actuación, unido al cambio sobrevenido a un modelo de docencia exclusivamente online, permite a

los alumnos el seguimiento de la asignatura durante el resto del cuatrimestre, con unos resultados positivos dadas las circunstancias. Sin embargo, se encuentran varios elementos de fricción en el cambio: en primer lugar, algunos de los alumnos que carecen de equipo propio no llegan a solicitar el préstamo, no pudiendo desarrollar las sesiones prácticas (que en total representarán el 50% del total de sesiones de la asignatura). En segundo lugar, la no presencialidad dificulta la aclaración de dudas sobre aspectos del código desarrollado, siendo necesario recurrir a diversas herramientas para que el alumno muestre al profesor los problemas que encuentra, y ralentizando el aprendizaje. Este reto es aún mayor al tratarse de alumnos de primer curso, que disponen en principio de menor experiencia y recursos, en cuanto a organización y trabajo autónomo. Del mismo modo, la realización de pruebas de evaluación de forma remota fuerza al uso de herramientas no pensadas originariamente para compartir códigos (Moodle) combinado con los equipos, propios o prestados, que los alumnos tienen en su casa para poder determinar el grado de aprendizaje práctico y de familiaridad con las herramientas de desarrollo.

Por otro lado, todos los alumnos tienen derecho a realizar las pruebas de evaluación pertinentes aunque no dispongan en casa del equipamiento necesario. Para ofrecer a dichos alumnos una evaluación análoga a la realizada por sus compañeros, se decide emplear un sistema provisional basado en la herramienta TeamViewer [4]. Con esta configuración, uno de los profesores de la asignatura pone su propio equipo Raspberry Pi a disposición remota del alumno para que éste pueda realizar una serie de tareas sobre un equipo real. Estas dificultades, soslayadas de forma razonable dadas las circunstancias y los medios disponibles, motivan la necesidad de implementar una solución robusta y estable para el acceso telemático a los equipos del laboratorio. Por este motivo, se plantea el desarrollo de dicha solución como parte de un Proyecto de Innovación Docente más amplio, que busca proponer mejoras para afrontar los retos de la enseñanza de estas asignaturas en el marco de la pandemia.

El comienzo del curso 2020-21 refuerza dicha necesidad, con la adopción de diversas medidas de higiene y seguridad que reducen el número de alumnos que pueden acceder a los recursos del laboratorio al mismo tiempo, lo que fuerza a desarrollar parte de las sesiones de forma telemática y dificulta el uso de los equipos por parte del alumnado fuera de los horarios asignados. Esto perjudica el trabajo autónomo del alumnado y las opciones que se les ofrecen para completar su aprendizaje, y se complica por el uso de diversos dispositivos de Entrada/Salida que incentivan al alumno a seguir las asignaturas [10], [11] pero incrementan la dificultad para disponer de un puesto propio en casa, debido a la baja disponibilidad de algunos dispositivos en los canales de venta minorista. Además, los protocolos sanitarios de actuación ante posibles contagios hacen que algunos alumnos se

vean obligados a no asistir a algunas sesiones o pruebas por haber estado expuestos al contagio por parte de un enfermo de COVID-19, o por haber contraído el virus ellos mismos.

Del mismo modo, para garantizar el alcance del nivel de conocimientos observado en años anteriores, se hace necesario establecer jornadas extendidas de evaluación para poder cumplir los requisitos de seguridad frente al COVID-19, con el consiguiente perjuicio para alumnado y profesorado.

Por todo ello, se hace atractiva la idea de disponer de un sistema de acceso remoto al laboratorio que permita interactuar en tiempo real tanto con los equipos Raspberry Pi como con diversos periféricos conectados a los mismos, maximizando el uso de los mismos en horarios de acceso restringido a las instalaciones. Además, la posibilidad de ofrecer un acceso concurrente al mismo dispositivo a alumno y profesor supone una mejoría clara para el proceso de aclaración de dudas, y para determinar con mayor precisión el grado de aprendizaje y las dificultades de cada alumno. Como colofón, un sistema de dichas características haría factible la realización de pruebas de evaluación a alumnos confinados de forma concurrente con el resto de sus compañeros. Esta posibilidad supondría un menor esfuerzo para el profesor a la hora de desarrollar dichas pruebas, y evita desigualdad en la evaluación de los alumnos frente a los compañeros que la realizan presencialmente.

Otra ventaja que supone un sistema de estas características es el menor número de equipos necesarios para la docencia, no sólo porque los alumnos pueden reutilizar el mismo de forma consecutiva sin necesidad de medidas de desinfección, sino también porque el profesorado puede utilizarlo incluso fuera del horario docente, no siendo necesario dedicar en exclusiva un equipo por cada docente para que pueda desarrollar nuevos contenidos o revisar dudas prácticas.

El resto de este artículo está organizado como sigue: la Sección II analiza los requisitos y necesidades a cubrir por este sistema de laboratorio remoto. La Sección III detalla las características e implementación de la herramienta, y la Sección IV muestra las características del manejo de la herramienta. Finalmente, la Sección V hace un resumen de los puntos más relevantes del trabajo, y detalla las posibles líneas de trabajo futuro.

II. OBJETIVOS

Como primer aspecto a considerar en el desarrollo de las herramientas para el sistema de laboratorio remoto se realiza una evaluación de las necesidades a cubrir, partiendo de un objetivo inicial de disponer de una aplicación o página web que permita interactuar con un escritorio remoto de una Raspberry Pi en tiempo real, así como con dispositivos hardware conectados al equipo. A este efecto, se consideran varios casos de uso, cada uno de ellos con algunos requisitos diferenciados:

- Uso por alumnos en remoto de forma autónoma.

- Análisis y aclaración de dudas prácticas por parte de un profesor a un alumno.
- Realización de pruebas de evaluación.

La herramienta está orientada a su uso en varias asignaturas del área de Estructura y Organización de Computadores, en cuyas sesiones prácticas se abordan dos ámbitos separados: el aprendizaje de la arquitectura de un computador a través de la programación a bajo nivel en lenguaje ARM, y el manejo de la Entrada/Salida con diversos dispositivos periféricos conectados a los pines de entrada/salida de propósito general (*General Purpose Input/Output*, GPIO) de una Raspberry Pi. Para asemejar la experiencia de uso de la herramienta a la del puesto de trabajo del laboratorio se necesita ofrecer un mecanismo para interactuar con el entorno gráfico del sistema corriendo sobre el equipo, un medio para modificar los valores de entrada de diversos pines GPIO de un equipo, y una forma de visualizar la salida que se observa en un dispositivo periférico conectado al equipo.

Además, las características de uso del sistema operativo RISC OS hacen necesaria la existencia de un mecanismo para reiniciar el equipo en caso de error, tal y como se detalló en la Sección I. Para realizar esta operación, dado que se presupone un estado no responsivo del sistema, se ha de interrumpir y reanudar la alimentación del equipo Raspberry Pi, forzando así el arranque del sistema. Dado que este procedimiento supone manipular la alimentación de la Raspberry Pi, se contempla como ventaja añadida la implementación de sendos mecanismos de encendido y apagado, facilitando así una reducción en el consumo energético del laboratorio remoto.

Para cubrir los casos de uso con múltiples actores (alumno y profesor), es necesario que el sistema ofrezca un acceso concurrente a un mismo equipo, permitiendo la visualización e interacción con el equipo y su entorno (dispositivos periféricos) a ambos usuarios al mismo tiempo.

Además, la herramienta debe proporcionar mecanismos de gestión y acceso autenticado, para restringir el empleo del sistema sólo a usuarios autorizados, y para realizar la asignación de los recursos disponibles (los equipos Raspberry Pi) de forma selectiva a determinados usuarios. Esto último permite, por ejemplo, que si cada equipo está configurado con un dispositivo periférico diferente, un profesor pueda limitar el acceso a cada alumno según su grado de avance en las sesiones, o según el tipo de sesión práctica propuesta. Por consiguiente, el sistema debe disponer de un mecanismo de gestión de usuarios, distinguiendo entre varios tipos con distintos privilegios de acceso, para que los profesores puedan añadir o eliminar usuarios y recursos.

Todos estos servicios conllevan la necesidad de emplear un equipo que actúe de servidor de acceso al servicio, puesto que el uso de soluciones puramente software sobre cada equipo Raspberry Pi se descarta de forma inmediata por las limitaciones y riesgos de seguridad de RISC OS. Para reducir el coste asociado

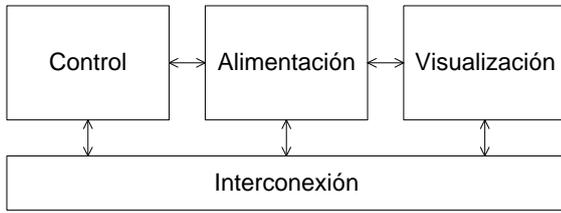


Fig. 1: Diagrama de bloques con los módulos HW de la herramienta.

a la implementación de la herramienta, se pretende maximizar el número de dispositivos conectados a un mismo servidor.

III. SISTEMA DE ACCESO REMOTO AL LABORATORIO HARDWARE

En esta Sección se detalla el desarrollo de la herramienta para proporcionar un acceso remoto a un laboratorio basado en equipos Raspberry Pi corriendo el sistema operativo RISC OS, y con dispositivos de Entrada/Salida conectados a los equipos. El desarrollo se divide en dos partes que, si bien están interconectadas para ofrecer la funcionalidad requerida, están claramente diferenciadas: los componentes hardware y software.

A. Diseño hardware de la herramienta

Al comienzo del desarrollo de la herramienta se identifica que una forma ágil y sencilla de permitir la interacción remota del alumno con el escritorio del equipo Raspberry Pi es empleando una conexión VNC similar a la que los alumnos usan para manejar sus propios equipos desde un PC cuando carecen de un puesto completo, tal y como se describe en la Sección I. Esta decisión permite ahorrar hardware adicional para hacer posible la visualización e interacción con el entorno gráfico, más allá de la necesidad de una conexión entre el equipo Raspberry Pi y la máquina que actúe de servidor para el acceso a la herramienta.

Tras determinar las especificaciones del sistema a partir del análisis de la Sección II, se observa que el desarrollo hardware puede dividirse en cuatro módulos: alimentación, visualización, control e interconexión, tal y como se detallan en la Figura 1. El módulo de alimentación debe proporcionar la energía necesaria para alimentar a los dispositivos necesarios, tanto los equipos en sí (la Raspberry Pi) como al hardware adicional necesario para permitir el manejo remoto. El módulo de visualización se encarga de proporcionar mecanismos para observar las salidas de los dispositivos periféricos. El módulo de control gestiona la entrada a los pines GPIO de la Raspberry Pi, así como el control sobre la alimentación de la misma. Por último, el módulo de interconexión vincula todos los demás módulos entre sí para que el sistema pueda trabajar de forma coordinada.

Se decide implementar la parte de control mediante módulos Arduino [1], empleando placas de interfaz Ethernet y de gestión de relés. La elección de la plata-

forma Arduino permite mantener unos costes reducidos a la par que garantizar la flexibilidad y escalabilidad del diseño. Además, dispone de una comunidad de desarrollo de código libre amplia y variada, facilitando el reuso de desarrollos software previos para minimizar el tiempo y coste de la implementación de la herramienta. La elección del módulo de relés se justifica por su capacidad de conmutar terminales de muy distinto tipo, tanto los de alimentación de los equipos como pines de E/S, lo cual permite dimensionar el sistema para trabajar con mayor o menor número de equipos según las necesidades de entrada/salida de cada uno de ellos para un tipo de sesión de laboratorio. Para una primera prueba de concepto se selecciona un único módulo de 4 relés, lo que proporcionaría acceso a un máximo de 4 equipos sin usar pines de E/S, o un máximo de 3 pines de E/S conectados a un mismo equipo Raspberry Pi.

Por otro lado, la placa de interfaz Ethernet ofrece una conectividad con el equipo servidor muy sencilla, permitiendo emplear en la placa Arduino un pequeño servidor que acepte peticiones HTTP para controlar las distintas acciones del módulo de control. Se trata de una elección lógica puesto que la interconexión de la Raspberry Pi con el servidor ha de realizarse también mediante interfaces Ethernet, para emplear el protocolo VNC para comunicar al alumno con el escritorio del equipo. Utilizando un switch Ethernet a 10/100 Mbps se pueden unir el módulo de control y los equipos Raspberry Pi al equipo servidor de la herramienta. El diseño puede adaptarse a un mayor o menor número de puertos en el switch según los equipos ofrecidos en el laboratorio remoto, necesitando reservar siempre al menos dos puertos para la conexión del equipo servidor y del módulo de control. La elección de un switch con puertos más lentos que el máximo disponible en soluciones comerciales es deliberada, puesto que las interfaces del modelo Raspberry Pi 1B+ usado en el laboratorio no pueden operar a mayor velocidad, y el volumen de comunicaciones con el módulo de control es despreciable en comparación, por lo que un menor coste resulta más interesante.

Para garantizar el aislamiento entre los elementos hardware de la herramienta y los alumnos que hacen uso de ella, el conexionado con el equipo servidor se realiza en una red separada de la que une al servidor con el exterior. Esta decisión favorece la seguridad del sistema, dado que RISC OS es un sistema operativo que permite un acceso muy completo a la arquitectura del procesador y que carece de mecanismos de seguridad para impedir que los equipos se conviertan en puertas de acceso a la red corporativa de la Universidad. Dado el sobrecoste de un switch gestionable que permita realizar el aislamiento de forma virtual, se opta por emplear un equipo servidor con dos interfaces Ethernet, asignando subredes IP completamente diferentes a cada una de ellas.

El módulo de alimentación de la herramienta ha de garantizar que se alcanzan los requisitos de los equipos Raspberry Pi, $\pm 5,1$ V y 2.5A por equipo;

inicialmente se opta por una fuente de alimentación de 5V y 16A, que permite conectar el máximo de cuatro equipos que nos impone el módulo de control empleado, más las placas Arduino que gestionan la lógica de control.

Por último, el módulo de visualización se implementa con una emisión en *streaming* del vídeo captado por una webcam colocada frente al equipo y a sus dispositivos periféricos. Se trata de una solución sencilla, de bajo coste, y que permite trabajar con cualquier tipo de dispositivo periférico que emplee una salida visual, flexibilizando así el trabajo con múltiples clases de dispositivos hardware. Se valora como alternativa de futuro la virtualización completa de los pines GPIO del equipo, empleando un software de adquisición de datos y control de instrumentación con un desarrollo visual como LabVIEW [13], [6], para modelar el comportamiento de cualquiera de los dispositivos hardware empleados en las sesiones prácticas. Esta solución permitiría una flexibilidad completa a la hora de trabajar con los dispositivos, pero resulta más complejo y costoso de implementar, y se aleja del objetivo de involucrar al alumno en el aprendizaje mediante hardware real explicado en la Sección I. Se decide por tanto descartar dicha implementación, dejándola como opción de trabajo futuro.

B. Diseño software de la herramienta

Tras determinar los componentes hardware de la herramienta, se completa un desarrollo software que integre los distintos elementos para ofrecer al alumno la gestión remota de un equipo Raspberry Pi y de los dispositivos conectados al mismo.

Por sencillez y coste de desarrollo se decide ofrecer la herramienta a través de una interfaz web, ya que una aplicación supondría multiplicar el esfuerzo de desarrollo por tantas veces como sistemas operativos se quieran soportar para no discriminar al alumnado. Además, las limitaciones de acceso que la Universidad de Cantabria impone sobre los equipos conectados a la red cableada corporativa hacen que resulte más sencillo la conexión mediante páginas web. Debido a dichas restricciones, y aprovechando la existencia de la herramienta UNICANLabs [8] que permite a los alumnos el acceso remoto con autenticación a los equipos Windows y Linux de las aulas, se implementa un servidor web accesible desde cualquier ordenador conectado a la red cableada, posibilitando el acceso mediante un PC convencional manejado remotamente. Esto simplifica los requisitos de seguridad, porque ningún usuario puede acceder al servicio web sin estar conectado de forma controlada a la red, y minimiza las complicaciones de acceso de los alumnos, porque aprovecha una herramienta que ya emplean activamente en otras asignaturas.

A partir de las necesidades observadas en el análisis de la Sección II, se decide que el sistema debe contar con varias vistas claramente separadas: un panel de acceso a la herramienta, exigiendo el uso de unos credenciales de usuario; un panel donde se muestren

los distintos equipos y se permita la selección de uno en concreto; un panel de gestión de recursos reservado al profesor, para gestionar los usuarios y equipos disponibles; y por último un panel de manejo de un equipo en concreto, ofreciendo una vista del escritorio del mismo y una serie de controles para interactuar con el equipo y con los dispositivos periféricos que haya conectados.

Dado que las necesidades de gestionar equipos y usuarios conllevan el uso de una base de datos que gestione dicha información, y que la interacción a través del panel principal con la Raspberry Pi y los dispositivos periféricos descartan el uso de una página web estática, se opta por emplear el entorno de desarrollo Laravel [14], basado en el lenguaje de programación PHP. Se trata de un *framework* de código libre y gratuito que sigue el Modelo-Vista-Controlador, dando una estructura clara y definida al código. Además, implementa un mapa de objeto relacional (*Object Relational Mapping*, ORM) que simplifica el manejo de la base de datos subyacente, modelándolas con una estructura orientada a objetos y evitando la necesidad de emplear peticiones SQL explícitas. Por último, el entorno Laravel cuenta con una amplia documentación, y permite añadir nuevas funcionalidades fácilmente mediante un gestor de paquetes modular.

La interacción con el escritorio del equipo Raspberry Pi remoto se gestiona mediante una conexión VNC similar a la que emplean aquellos alumnos que disponen del equipo propio Raspberry Pi pero no de una configuración de puesto autónomo con monitor, teclado y ratón. Esta conexión aprovecha el servidor VNC instalado en el equipo y permite el envío de la imagen correspondiente al escritorio remoto, así como la recepción de la entrada a través del teclado y el ratón. Para ello, se escoge el cliente noVNC [3] por su facilidad para integrarlo en el marco de una página web preexistente y por la calidad y extensión de su documentación, gracias al respaldo de una fuerte comunidad de desarrollo.

El manejo del entorno de control que gestiona el encendido y apagado de los equipos y de los pines GPIO usados como entrada se realiza a través de una sencilla API desarrollada *ad-hoc*. La API recibe peticiones de tipo HTTP que se gestionan desde un pequeño servicio web ejecutado en la placa Arduino, y modifica el estado del módulo de control en consecuencia. Dichas peticiones sólo pueden realizarse desde el equipo servidor del sistema de laboratorio remoto, y se limitan a aquel usuario que esté conectado al panel de interacción con la Raspberry Pi afectada (o con los dispositivos periféricos involucrados), para evitar acciones maliciosas entre alumnos.

IV. USO DE LA HERRAMIENTA

En esta Sección se analiza la dinámica de uso de la herramienta desarrollada, mostrando las diversas vistas disponibles.

La herramienta muestra en primer lugar un panel de acceso que permite introducir datos de usuario y

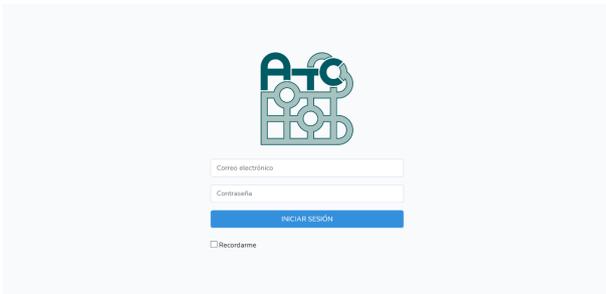


Fig. 2: Vista del panel de acceso a la herramienta.

contraseña para acceder al servicio, como se ilustra en la Figura 2. Según el tipo de usuario que se tenga asignado (alumno o profesor), se ajustará de forma acorde la vista de las máquinas disponibles. Tras acceder, la herramienta conduce a un panel de selección donde se pueden ver los equipos (denominados “máquinas” en la herramienta) que están conectados al sistema, como en el ejemplo de la Figura 3. Desde ahí, si un alumno tiene acceso asignado a una máquina, podrá entrar al escritorio de manejo de la misma usando el botón “Entrar”. En el caso de un usuario de tipo profesor, puede acceder a cualquier máquina en todo momento, independientemente de si está siendo usada por un alumno o no, aunque dicha circunstancia se puede observar porque la etiqueta de la máquina cambia al estado “Ocupado”. Esto permite que un profesor pueda comprobar el trabajo que realiza un alumno en tiempo real, pudiendo así observar el progreso y aclarar cualquier duda que le surja a un alumno como si ambos estarían presencialmente en el laboratorio; debe tenerse en cuenta que la herramienta no proporciona un mecanismo de comunicación entre profesor y alumno, por lo que la aclaración debe realizarse a través de una herramienta externa, como puede ser los servicios de correo o mensajería instantánea que proporcione la Universidad.

La vista de escritorio de un equipo en concreto es la más compleja e importante de la herramienta, tal y como se muestra en la Figura 4. La vista se compone de un panel superior que permite conmutar entre el escritorio en sí, un gestor de archivos para transmitir ficheros entre el equipo remoto y el del alumno, y un gestor de reservas; por el momento, la única opción implementada es la vista de escritorio. A continuación hay una columna de paneles de gestión a la izquierda, y la vista del escritorio remoto a la derecha. Los paneles de gestión son tres: uno para encender/apagar/reiniciar el equipo, así como para comprobar su estado actual, junto con un identificador del equipo; uno con el vídeo captado por la webcam, con la salida de los dispositivos periféricos en uso; y uno de gestión de entrada a través de los puertos GPIO, con tantos botones como pines estén configurados en el sistema como entrada para dicho equipo. Los botones de entrada GPIO funcionan como un pulsador convencional conectado a cada pin, conmutando la entrada a tierra mientras el botón esté pulsado, y restableciéndole a 3.3V cuando se suelta el botón. Éste es el comportamiento que se observa con un botón genérico como los pulsadores

de la placa BerryClip [2], un dispositivo comúnmente usado en las sesiones prácticas de las asignaturas.

La última pantalla disponible en la herramienta es la vista del profesor, accesible en cualquier vista desde el menú con el nombre del usuario situado en la parte superior derecha, siempre que éste tenga permisos de profesor. En el caso de que sea un alumno, desde este menú sólo aparecerá la opción de desconectarse del sistema.

Desde esta vista el profesor puede gestionar los usuarios, máquinas y permisos, así como acceder a un registro de estado con los eventos ocurridos en el sistema. El profesor puede añadir o quitar usuarios, o modificar su rol, lo cual permite gestionar la entrada de nuevos profesores en el sistema. En el apartado de máquinas se puede configurar los equipos disponibles, eliminando uno en caso de fallo, o añadiéndole si se quiere ampliar el sistema. Lógicamente, la incorporación de un nuevo equipo al sistema conlleva el conexionado asociado (alimentación, control, interconexión) para que funcione correctamente.

Es importante recalcar que la herramienta desarrollada no tiene aplicación práctica únicamente para la configuración de puesto de laboratorio utilizado en la Universidad de Cantabria, puesto que el manejo de la alimentación para el arranque de los equipos Raspberry Pi y el funcionamiento de los pines GPIO es independiente del sistema operativo que corran. Del mismo modo, para el manejo del escritorio remoto únicamente es necesario disponer de un servidor VNC en el equipo del laboratorio, disponible en la práctica totalidad de sistemas operativos. Así, los conceptos básicos de la herramienta se podrían aplicar a otras configuraciones, ampliando la utilidad del sistema.

V. CONCLUSIONES

Las asignaturas de Estructura y Organización de Computadores han tenido tradicionalmente un fuerte componente práctico. Con la llegada del COVID-19 y las medidas higiénicas y de seguridad asociadas al mismo, se ha hecho necesario adaptar la docencia a un escenario semipresencial o incluso puramente telemático. Esta circunstancia es particularmente restrictiva para la docencia de laboratorio, particularmente cuando se emplea un sistema basado en equipos físicos, por la dificultad de replicar el puesto en casa. Además, la realización a distancia de las prácticas dificulta el planteamiento de dudas frente a un escenario presencial, donde el profesor puede observar con todo detalle el puesto del alumno para así determinar las causas de error.

Para solucionar dichas limitaciones, se ha desarrollado e implementado un sistema de laboratorio telemático basado en la misma configuración que las prácticas de laboratorio convencionales: equipos Raspberry Pi 1B+ ejecutando el sistema operativo RISC OS y la herramienta de depuración UCDebug. Este sistema permite al alumno observar e interactuar con el escritorio de un equipo remoto, gestionar su alimentación para forzar el reinicio del equipo en

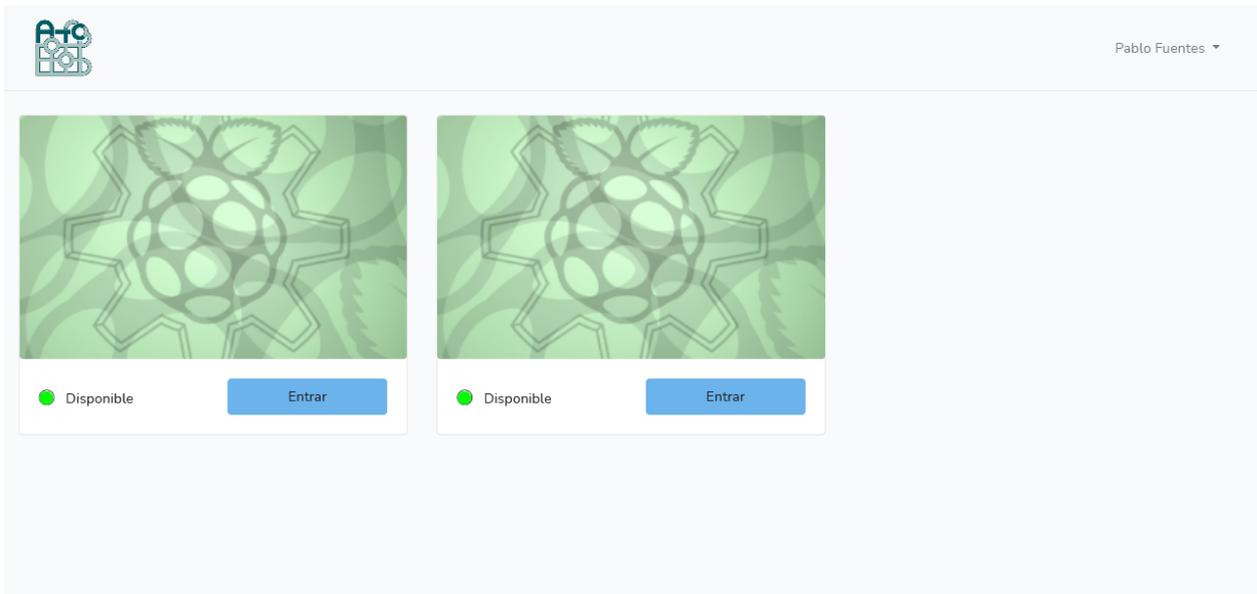


Fig. 3: Vista inicial de la herramienta, con las máquinas existentes.

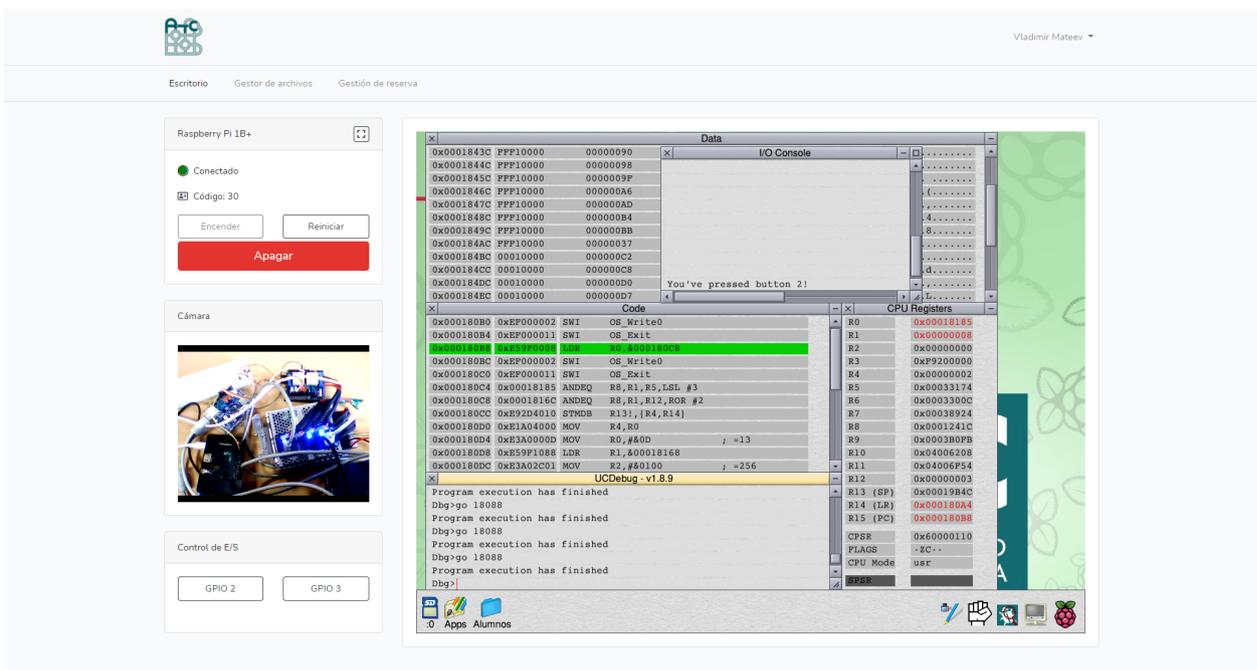


Fig. 4: Vista de escritorio de un equipo, con el entorno del sistema operativo mostrado en tiempo real, así como diversos paneles para gestionar el arranque del equipo y el manejo de la Entrada/Salida.

caso de error, modificar la entrada de pines GPIO configurados al efecto, y observar la salida obtenida para su programa. Para facilitar la aclaración de dudas, se permite la conexión simultánea de un alumno y uno o varios profesores al mismo equipo, pudiendo así observar el estado de la máquina para aclarar dudas con mayor precisión y celeridad. Finalmente, el sistema permite al profesor gestionar los usuarios y equipos disponibles, pudiendo restringir el acceso de un alumno a un equipo concreto si es necesario.

El sistema emplea una lógica de control basada en módulos Arduino con conmutación por relés, que se manejan a través de peticiones HTTP gestionadas por un servidor ligero corriendo en la placa Arduino. Tanto la lógica de control como los equipos Raspberry Pi se conectan mediante cable Ethernet a un equipo PC convencional que actúa como servidor del

sistema de laboratorio remoto y gestiona el acceso de los alumnos a cada equipo. El acceso a la herramienta se realiza desde una página web basada en el lenguaje de programación PHP. La vista de escritorio de la herramienta combina el cliente noVNC para embeber el escritorio del equipo remoto, con una serie de paneles de manejo que permiten el encendido/apagado del equipo remoto, la visualización de la salida de los dispositivos HW conectados al equipo, y el manejo de los pines GPIO configurados como entrada.

Como parte del trabajo se ha implementado una prueba de concepto de la herramienta, con capacidad para dos equipos Raspberry Pi y dos pines GPIO configurados como entrada, para comprobar la validez del sistema.

Se valoran diversas mejoras a introducir en el sistema como líneas de trabajo futuro. Un ejemplo de

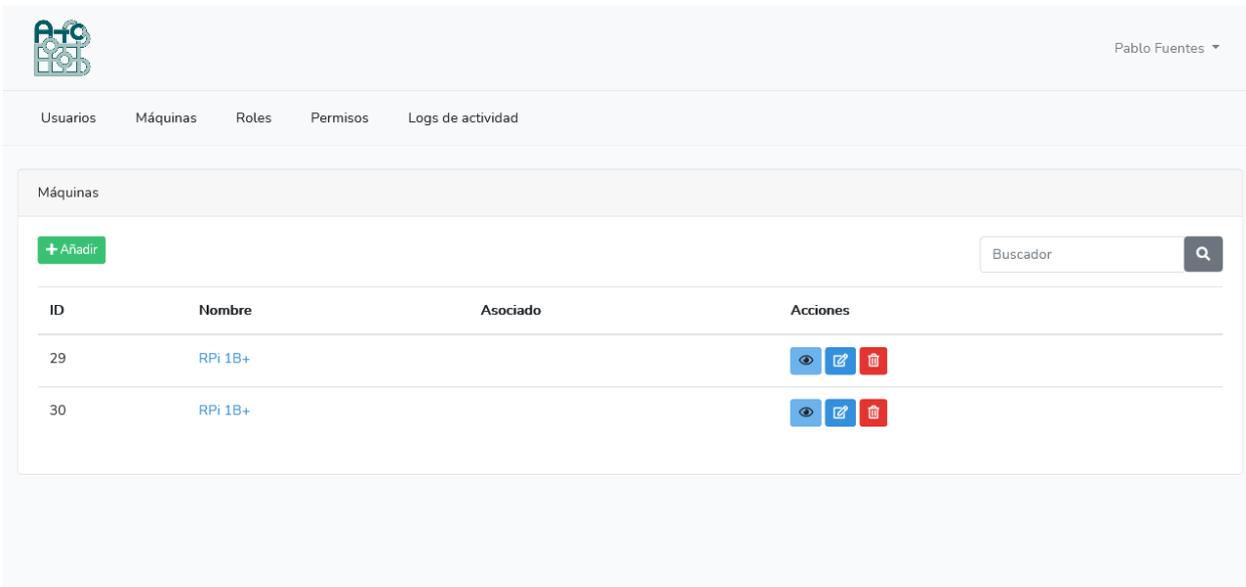


Fig. 5: Vista del panel de gestión del profesor, con diversas pestañas para manejar los usuarios y equipos disponibles, así como un control de permisos de la herramienta y registros de actividad.

dichas mejoras es un sistema de gestión de reservas para impedir el acaparamiento de los recursos por parte de los alumnos, limitando el tiempo de acceso y permitiendo la reserva de un equipo para un período concreto. Otra posibilidad es implementar un mecanismo para compartir ficheros entre el PC del alumno y el equipo remoto, permitiendo así guardar copia de los desarrollos realizados y partir de códigos ya realizados. De cara a facilitar la gestión del profesor, se valora la introducción de un sistema de inserción de nuevos usuarios en bloque, a partir de un fichero, de forma similar a las herramientas que el docente ya cuenta para el seguimiento y evaluación de los alumnos (Moodle, Campus Virtual...). Además, se pretende expandir la prueba de concepto a un laboratorio completo, aumentando el número de máquinas y de pines GPIO de entrada disponibles en el sistema.

Finalmente, se plantea poner a disposición todos los aspectos del desarrollo SW, así como una breve documentación de cara al conjunto HW-SW, como un proyecto de código abierto para que pueda ser adoptado por otras universidades que estén interesadas.

AGRADECIMIENTOS

Los autores agradecen la colaboración de Fernando Vallejo, Carmen Martínez y Cristóbal Camarero. Este trabajo ha sido parcialmente financiado por la V Convocatoria de Proyectos de Innovación Docente, del Vicerrectorado de Ordenación Académica y Profesorado de la Universidad de Cantabria.

REFERENCIAS

[1] Arduino. [Último acceso: Junio 2021]. URL: <https://www.arduino.cc/>.
 [2] BerryClip+ User Guide. [Último acceso: Junio 2021]. URL: <https://bitbucket.org/MattHawkinsUK/rpispyp-berryclip-plus/downloads/>.
 [3] noVNC - the open source VNC client. [Último acceso: Junio 2021]. URL: <https://novnc.com/info.html>.
 [4] TeamViewer, solución integral de acceso, control y soporte remoto. [Último acceso: Junio 2021].

[5] Sergio Barrachina Mir, Germán Fabregat Lluca, and José Vicente Martí Avilés. Utilizando arduino due en la docencia de la entrada/salida. In *Actas de las XXI Jornadas de la Enseñanza Universitaria de la Informática*, pages 58–65. Universitat Oberta La Salle, 2015.
 [6] Robert H Bishop. *LabVIEW 8*. Pearson, 2007.
 [7] Cristóbal Camarero, Fernando Vallejo, David Herreros, and Pablo Fuentes. UCDebug, an ARM Debugger for RISC OS. [Último acceso: Junio 2021]. URL: <https://fuentesp.github.io/UCDebug/>.
 [8] Universidad de Cantabria. UNICANLabs. [Último acceso: Junio 2021]. URL: https://sdei.unican.es/Paginas/servicios/salas_aulas/UNICANLabs.aspx.
 [9] Free Software Foundation, Inc. GCC, the GNU Compiler Collection. [Último acceso: Junio 2021]. URL: <http://gcc.gnu.org/>.
 [10] Pablo Fuentes, Cristóbal Camarero, Fernando Vallejo, and Carmen Martínez. La importancia del uso de hardware real en la docencia de estructura y organización de computadores. page 480, May 2019.
 [11] Pablo Fuentes Saez, Cristóbal Camarero Coterillo, María del Carmen Martínez Fernández, Fernando Vallejo Alonso, et al. Tecnología low-cost para motivar al alumno. In *Actas de las XXV Jornadas sobre la Enseñanza Universitaria de la Informática*, volume 4, pages 319–326.
 [12] Francisco J. Gallego-Durán, Rosana Satorre-Cuerda, Patricia Compañ-Rosique, and Carlos Villagrà-Arnedo. El código máquina mola. In *Actas de las XXIV Jornadas de la Enseñanza Universitaria de la Informática*, pages 149–156. Universitat Oberta de Catalunya, 2018.
 [13] National Instruments. ¿Qué es LabVIEW? [Último acceso: Junio 2021]. URL: <https://www.ni.com/es-es/shop/labview.html>.
 [14] Taylor Otwell. Laravel, the PHP Framework for Web Artisans. [Último acceso: Junio 2021]. URL: <https://laravel.com/>.
 [15] Hernik Bjerregaard Pedersen, Crispian Daniels, David Llewellyn-Jones, and Jeffrey Lee. VNC server for RISC OS. [Último acceso: Junio 2021]. URL: http://www.phlamethrower.co.uk/riscos/vnc_serv.php.
 [16] Raspberry Pi Foundation. Raspberry Pi. [Último acceso: Enero 2019]. URL: <https://www.raspberrypi.org/>.
 [17] Guttorm Vik, John Whittington, Carl Hetherington, and Fred Graute. StrongEd - a programmer's text editor for RISC OS. [Último acceso: Junio 2021]. URL: <http://stronged.iconbar.com/>.