



***Facultad
de
Ciencias***

**AMPLIACIÓN DE UN PROTOTIPO DE
LABORATORIO REMOTO PARA DOCENCIA
HARDWARE**

(Expansion of a prototype for remote practical
sessions)

Trabajo de Fin de Grado
para acceder al

GRADO EN INGENIERÍA INFORMÁTICA

Autor: Jaime Palacios Mediavilla

Director: Pablo Fuentes

Junio - 2022

Agradecimientos

Después de 4 años de carrera, me gustaría agradecer a aquellas personas sin las cuales, seguramente no hubiese sido posible llegar hasta aquí.

En primer lugar, darle las gracias a mi familia por cuidarme, confiar en mí y siempre dejarme hacer aquello que me gustase.

Gracias a mi pareja, Sara, un pilar fundamental durante toda mi carrera. Por haberme apoyado y animado en todo momento, independientemente de las circunstancias.

A los que comenzaron siendo mis compañeros de clase y que se han convertido en amigos, indispensables en cada día. Gracias por haber hecho todo este largo trayecto más ameno, después de incontables horas trabajando juntos y darme un buen recuerdo de ello.

Por último, agradecer en especial a mi director, Pablo, por todo el trabajo y la ayuda que me ha dedicado. También a Vladimir, por ofrecerme continuar este proyecto y ayudarme siempre que lo necesitaba.

Resumen

Este trabajo de fin de grado comprende la escalabilidad y acondicionamiento de un sistema de laboratorio remoto para la realización de sesiones prácticas usando equipos Raspberry Pi bajo el sistema operativo RISC OS. La iniciativa inicial del proyecto surge debido a la situación de aislamiento domiciliario vivido al comienzo de la pandemia, lo cual ha conllevado la necesidad de reestructurar la docencia en las asignaturas del área de Estructura y Organización de Computadores, donde hay un fuerte componente práctico centrado en el uso de hardware real. En dichas sesiones prácticas se hace uso de un puesto basado en equipos de bajo coste Raspberry Pi, facilitando al alumno la adquisición de su propio hardware para fomentar el trabajo autónomo. Tras la reanudación de la docencia presencial, se observaron nuevas limitaciones de aforo debidas a las medidas de distanciamiento sanitario, manteniendo la motivación del proyecto. Posteriormente, la pandemia y otras circunstancias sociopolíticas han dado lugar a la actual crisis de desabastecimiento de semiconductores, que se traduce en un déficit de producción para el nivel de demanda experimentado. En el caso del laboratorio de las asignaturas, este déficit de semiconductores impide la libre adquisición de los equipos por parte del alumnado.

Debido a estas limitaciones de acceso a las instalaciones, se desarrolló un proyecto para estudiar la viabilidad de un sistema que permitiese el acceso remoto a puestos del laboratorio. Como parte de dicho trabajo, se creó un entorno hardware para el control de la alimentación de los equipos, y un entorno software para la utilización de los mismos a través de una interfaz web. Dicho proyecto demostró la viabilidad de un sistema de estas características, pero presentaba carencias importantes: problemas de escalabilidad a un número de equipos similar al disponible en el laboratorio, y falta de funcionalidades (gestión de reservas, mecanismos de intercambio de ficheros) básicas para el uso por parte de alumnos. El presente trabajo surge de la necesidad de corregir las carencias observadas en el primero, y para alcanzar unos requisitos mínimos de uso.

Como parte del trabajo desarrollado se ha hecho un análisis de las limitaciones del sistema original y un estudio de las soluciones disponibles con el fin de seleccionar la más adecuada. Para ello, se han establecido dos grandes bloques de mejoras y ampliaciones necesarias: las relativas al software del sistema, centradas principalmente en la funcionalidad ofrecida al alumno; y las correspondientes al hardware del prototipo, para su ampliación de cara a un sistema de uso generalizado. El sistema permite que el alumno pueda programar una reserva de un equipo remoto o utilizarlo libremente si no está ocupado. Asimismo, el sistema facilita que el alumno pueda compartir sus ficheros locales con el equipo remoto, pudiendo así continuar las prácticas con total facilidad. Por último, se ha mejorado el sistema de visualización y añadido funciones de gestión a la herramienta.

Tras el estudio de las limitaciones de escalabilidad del sistema, se han establecido una serie de propuestas de mejora del mismo. Como parte de la evaluación de la adecuación de la herramienta a las necesidades de los alumnos se ha desarrollado una nueva versión del prototipo del sistema para verificar su adecuado funcionamiento, desplegando la herramienta en la asignatura de Introducción a los Computadores. El prototipo está dotado de cuatro equipos remotos y dos interfaces para pines de entrada de propósito general conectados a cada uno de los equipos. Cabe destacar que el desarrollo del sistema ha sido exitoso, alcanzando los requisitos planteados en la fase de análisis y observándose una satisfacción generalizada de los alumnos con la herramienta.

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

Abstract

This final degree project comprises the scalability and the conditioning of a remote laboratory setup to conduct practical sessions using Raspberry Pi devices under RISC OS. The initial premise of the project arises with the lockdown measures at the beginning of the COVID-19 pandemic. Those measures led to a restructure of the courses in Computer Structure and Organization, where there is a strong practical component focused on the use of actual hardware. In these practical sessions, the workstation is based on low-cost Raspberry Pi devices, to help the students to acquire their own hardware to promote autonomous work. After the resumption of face-to-face teaching, social distancing measures reduced the room capacity, and the motivation for the project stood. Subsequently, the pandemic and other sociopolitical circumstances have led to the current semiconductor shortage crisis, resulting in a production deficit for the level of demand experienced. In the case of the subject laboratory, this deficit of semiconductors prevents the free acquisition of the equipment by the students.

Due to the access limitations to the facilities, an initial work was performed to assess the viability of a system to remotely access the lab workstations. As part of the work, a hardware environment was created to manage the power supply of the remote devices and a software environment to interact with them through a web interface. The project proved the feasibility of such a system, but carried significant deficiencies: scalability problems to reach a similar number of devices to the lab, and a lack of basic features for student use (device bookings, file exchange). This work stems from the need to address the deficiencies detected in the previous project, and to reach minimum requirements for its use in the courses.

As part of the work developed, an analysis of the limitations of the original system and a study of available solutions have been made in order to select the most appropriate one. To do this, two large blocks of necessary improvements have been established: those related to the system software, mainly focused on the features offered to the student; and those corresponding to the prototype's hardware, for its expansion towards a system for general use. The system allows the student to schedule a reservation of a remote device, or to freely use it if available. Likewise, the system also allows the student to share his local files with the remote device, thus being able to continue the practical sessions with total ease. Lastly, the visualization module has been improved and management functions have been added to the tool.

After studying the scalability limitations of the system, a series of proposals for its improvement have been established. As part of the evaluation of the tool adequacy to the student needs, a new version of the system prototype has been deployed for the students of one course, to verify its proper functioning. The prototype is equipped with four remote Raspberry Pi devices and two terminals connected to the general purpose input pins of each device. It should be noted that the development of the system has been successful, reaching the requirements set out in the analysis phase and observing a general level of student satisfaction with the tool.

Keywords: Remote teaching, laboratory, Computer Structure and Organization, Raspberry Pi, RISC OS.

Índice general

1. INTRODUCCIÓN	7
1.1. Motivación	7
1.2. Desarrollo Realizado	9
2. DISEÑO HARDWARE	12
2.1. Escenario previo y estudio de limitaciones	12
2.1.1. Módulo de alimentación	13
2.1.2. Módulo de control	13
2.1.3. Módulo de red de interconexión	15
2.1.4. Módulo de visualización	16
2.1.5. Resumen de limitaciones	17
2.2. Propuesta de mejora e implementación de la solución elegida	17
2.2.1. Módulo de control	18
2.2.1.1. Estrategia 1: Uso de la placa Arduino MEGA	18
2.2.1.2. Estrategia 2: Uso de Registros de Desplazamiento	19
2.2.1.3. Implementación de la solución elegida	21
2.2.1.4. Interfaz del módulo de control	22
2.2.2. Módulo de alimentación	22
2.2.3. Módulo de red de interconexión	22
2.2.4. Integración del sistema	22
2.2.5. Módulo de visualización	24
3. DISEÑO SOFTWARE	25
3.1. Escenario previo y estudio de limitaciones	25
3.2. Sistema de gestión de reservas temporales	27
3.2.1. Requisitos del sistema de reservas	27
3.2.2. Diseño e implementación de la solución elegida	28
3.3. Compartición de ficheros	31
3.3.1. Estudio previo	31
3.3.2. Implementación de la solución elegida	32
3.4. Módulo de visualización	35
3.4.1. Estudio previo	36
3.4.2. Implementación de la solución elegida	36
3.5. Gestión del sistema	37
3.5.1. Gestión de usuarios	37
3.5.2. Gestión de roles y permisos	39
3.5.3. Notificación de incidencias	39
3.5.4. Estadísticas de uso	40
4. EVALUACIÓN DEL SISTEMA	41
4.1. Análisis de uso de la herramienta	42
4.2. Encuestas de estimación de uso y satisfacción	42
5. CONCLUSIÓN	44
5.1. Trabajo futuro	45
REFERENCIAS	47
ANEXO A. Código API del módulo de control	49
ANEXO B. Código para la comprobación del registro de desplazamiento	51

Índice de figuras

2.1. Esquema de los módulos utilizados	12
2.2. Fuente de alimentación utilizada	13
2.3. Conjunto de placas Arduino utilizadas	14
2.4. Arduino MEGA 2560 Rev3	18
2.5. Registro de desplazamiento de 8 bits	19
2.6. Esquema de las conexiones realizadas	21
2.7. Esquema de PCB diseñado	23
3.1. Paneles de la interfaz en el prototipo inicial del sistema	26
3.2. Pasarela de reserva	29
3.3. Panel principal	30
3.4. Panel de las reservas del usuario	30
3.5. Panel de la gestión de ficheros	34
3.6. Panel de la exportación e importación de usuarios	38
3.7. Panel para cambiar la contraseña	38
3.8. Panel de actividad de usuarios	39
3.9. Vistas sobre la notificación de incidencias	40
3.10. Panel de estadísticas	40
4.1. Situación actual del prototipo	41

1. INTRODUCCIÓN

El presente proyecto se sitúa en el contexto de las prácticas de laboratorio relativas a las asignaturas del área de Estructura y Organización de Computadores. Estas asignaturas tienen un fuerte componente práctico, y como parte del proceso de aprendizaje se pide a los alumnos que plasmen sus conocimientos en el desarrollo de programas sencillos en lenguaje ensamblador. Desde el año 2018, en la Universidad de Cantabria se emplean equipos Raspberry Pi [27] corriendo el sistema operativo RISC OS [28] en una configuración de puesto autónomo. Este enfoque, combinado con el uso de periféricos hardware sencillos y de bajo coste, genera una mayor implicación del alumnado y facilita su trabajo de forma autónoma.

Sin embargo, a raíz de las medidas de distanciamiento social y las consiguientes reducciones de aforo debidas a la pandemia por el virus COVID-19, se observa que el acceso del alumno a los equipos de laboratorio se ha visto significativamente reducido. Aunque dichas restricciones se han ido eliminando con el paso del tiempo, se han observado nuevas circunstancias que dificultan el trabajo autónomo de los alumnos. La causa más significativa es la llamada “crisis de semiconductores”, que ha reducido de forma drástica la disponibilidad de equipos para su adquisición por parte de los alumnos. También cabe destacar el significativo aumento del coste de la energía en el último año, que ha motivado la búsqueda de medidas de ahorro energético por parte de los centros, entre las que se baraja la reducción del acceso a los espacios de la Facultad fuera del horario lectivo.

Por estos motivos, desde el año 2020 se ha buscado el desarrollo de un sistema de acceso remoto a los equipos de laboratorio, con el fin de facilitar el trabajo autónomo de los alumnos de forma similar a la dinámica de trabajo presencial. Como parte de un Trabajo de Fin de Grado de un alumno del Grado en Ingeniería Informática [24], se ha desarrollado un prototipo inicial del sistema para estudiar su viabilidad. Los primeros resultados observados han sido alentadores, pero se observan deficiencias significativas en el sistema inicial de cara a su uso por parte de los alumnos, que es preciso paliar antes de su implantación generalizada.

Este trabajo, que se enmarca dentro de un proyecto de innovación docente, tiene como objetivo proporcionar la escalabilidad necesaria al proyecto inicial, analizando las necesidades de los usuarios y las limitaciones de la implementación original. El análisis comprende las características hardware de los componentes del sistema, que dificultan la escalabilidad de la herramienta para alcanzar un número de equipos similar a los disponibles en el laboratorio presencial. También tiene una importante vertiente software, para implantar funcionalidades básicas tales como un sistema automatizado de reservas o un mecanismo de intercambio de archivos con los equipos remotos. Por último, se pretende evaluar el funcionamiento del prototipo con alumnos de la asignatura de Introducción a los Computadores, para valorar la implantación del sistema de forma estable.

1.1. Motivación

Las asignaturas del área de Estructura y Organización de Computadores tienen como objetivo formativo el aprendizaje básico de la arquitectura de un computador, incluyendo la relación con el entorno del computador a través de la Entrada/Salida. Se trata de asignaturas de formación básicas en los Grados en Ingeniería de Computadores y en Ingeniería de Tecnologías de Telecomunicación con un alto número de alumnos por curso (en total, entre 90 y 120 alumnos por cuatrimestre). Las asignaturas de este área tienen un fuerte componente práctico, con sesiones

en el laboratorio que tienen por finalidad la familiarización del alumno con la arquitectura de un procesador mediante el uso de hardware real gestionado a bajo nivel, utilizando lenguaje ensamblador.

Para el desarrollo de dichas prácticas se emplean equipos Raspberry Pi, modelo 1B+. Estos equipos utilizan un procesador con arquitectura ARM, de tipo RISC (*Reduced Instruction Set Computing*, computación con conjunto de instrucciones reducido). La arquitectura ARM tiene una amplia cuota de mercado en los sectores móvil y de embebidos [16], lo que la hace particularmente atractiva a los alumnos. Por otro lado, las arquitecturas de tipo RISC ofrecen un gran interés pedagógico, ya que tienen una menor complejidad aparente que facilita el aprendizaje de los conceptos básicos. Además, los equipos Raspberry Pi son compatibles con una enorme variedad de periféricos de bajo coste que permiten diversificar el desarrollo de las prácticas, fomentando el interés y la participación de los alumnos.

Uno de los objetivos del tipo de puesto empleado en estas prácticas es facilitar el trabajo autónomo del alumno. Para ello, además de seleccionar un hardware de coste reducido, se ha elegido un sistema operativo que permite gestionar el hardware de forma autónoma (sin necesidad de un equipo externo), pero permitiendo el manejo de periféricos a muy bajo nivel. La metodología de trabajo de las prácticas que realizan los alumnos de forma individual, consiste en el desarrollo de códigos sencillos en lenguaje ensamblador ARM con la ayuda de un editor de textos en RISC OS. Posteriormente se han de ensamblar y enlazar dichos códigos, generando binarios ejecutables que se llevarán a un proceso de depuración con ayuda de la herramienta UCDebug [17]. En la asignatura de Introducción a los Computadores se busca aprender el funcionamiento de un procesador mediante el desarrollo básico códigos en ensamblador, y en Estructura de Computadores se ve el manejo del hardware subyacente. Para el aprendizaje de la entrada/salida con hardware se emplean diversos periféricos en las prácticas, como la placa BerryClip, que dispone de medios de entrada (pulsadores) y salida (LEDs, un zumbador). Por ello, dado que la realización de las prácticas puede solaparse, tendremos equipos disponibles con diferentes periféricos conectados, creando una diferenciación entre ellos. Como parte del entorno de prácticas se usa el sistema operativo RISC OS, que facilita gestionar el hardware del sistema a muy bajo nivel. Sin embargo, esto implica que en ocasiones pueden producirse bloqueos del equipo, que sólo pueden resolverse desconectando y volviendo a conectar la alimentación. Además, el nivel de seguridad es muy inferior al proporcionado por sistemas operativos modernos (como Windows, Linux, macOS...), lo cual requiere el aislamiento de los equipos del entorno de red en la medida de lo posible, para impedir que se convierta en un vector de ataque.

La pandemia causada por el virus SARS-CoV-2 ha supuesto una disrupción significativa de la forma de trabajo en las sesiones prácticas. En primer lugar, el confinamiento domiciliario impuesto en marzo de 2020 hizo necesaria una reorganización de las asignaturas, puesto que no estaba permitido el acceso a las instalaciones de la Universidad. Se optó por el préstamo de equipos a los alumnos para que pudiesen realizar las prácticas durante el confinamiento. Aunque esta estrategia mitigó el problema, se observaron limitaciones en la docencia remota: dificultad a la hora de plantear y resolver dudas de los alumnos, e imposibilidad de seguir el itinerario lectivo para alumnos que no solicitaron préstamo de equipos. A causa de la situación, se consideró necesario desarrollar un sistema que permitiese replicar el uso de los puestos de laboratorio de forma remota.

Tras la reanudación de la docencia presencial, se han observado nuevos retos que justifican la importancia del sistema descrito. En primer lugar, durante los cursos académicos 2020-21 y

2021-22 se han mantenido una serie de medidas de seguridad que han restringido el aforo de los espacios universitarios, forzando a emplear docencias a distancia durante la mitad de las sesiones prácticas (rotando a los alumnos entre docencia presencial y a distancia). De forma similar a lo experimentado durante el confinamiento, se observa que al capacidad de ayuda a los alumnos se reduce de forma significativa cuando el profesor no tiene forma de observar el trabajo realizado por el alumno durante la sesión.

Por otro lado, y debido a los problemas de producción por la pandemia y a la creciente demanda de semiconductores se ha generado una crisis de abastecimiento. Esta crisis ha paralizado todo el mercado dependiente de los sistemas digitales, que hoy en día están presentes en una inmensidad de campos de la vida cotidiana, como la automoción, la domótica, los electrodomésticos, teléfonos móviles... En el caso concreto del laboratorio, la falta de producción supone que no hay equipos Raspberry Pi 1B+ disponibles en los proveedores habituales hasta el año 2023.

Todo esto impide que, los alumnos puedan adquirir por cuenta propia los equipos para la realización de las prácticas fuera del laboratorio. Tampoco es factible el préstamo de equipos a cada uno de los alumnos, pues en el laboratorio no se disponen de suficientes Raspberry Pi. Tampoco es recomendable el acceso de los alumnos al laboratorio fuera del horario lectivo, ya que es necesario que un profesor se encargue de gestionar el reparto y recogida de los equipos; además, esto supone un incremento del gasto energético para el centro, en concepto de iluminación y calefacción.

Todo esto hace que surja un gran interés por la disponibilidad de un mecanismo de acceso remoto que permita compartir los equipos Raspberry Pi empleados en las sesiones prácticas con alumnos y profesores fuera del horario lectivo. De este modo, los alumnos pueden trabajar de forma autónoma sin necesidad de adquirir un equipo propio o de acceder al laboratorio fuera del horario lectivo. Este sistema debe permitir al alumno replicar la metodología de trabajo presencial, aportando una serie de funciones básicas: control de la alimentación de los equipos remotos, control de los periféricos conectados y visualización de la salida de los periféricos.

El proyecto inicial surge en el marco de un proyecto de innovación docente cofinanciado por el Vicerrectorado de Ordenación Académica y de Profesorado de la Universidad de Cantabria, con el fin de facilitar la enseñanza y el trabajo remoto sin renunciar al uso de elementos hardware. Como parte de dicho proyecto, y a través del esfuerzo de un alumno del Grado en Ingeniería Informática como parte de su Trabajo de Fin de Grado, se desarrolla e implementa un prototipo inicial del sistema, para analizar su viabilidad. Aunque el prototipo reunía los requisitos mínimos para afirmar que el sistema era viable, carecía de funciones básicas imprescindibles para el uso de los alumnos, y no garantizaba la escalabilidad del sistema. Este trabajo pretende paliar las diferencias observadas y aportar las funcionalidades necesarias para su uso por parte de los alumnos. No obstante, este proyecto inicial posee algunas limitaciones significativas de cara a su utilización en un entorno académico: necesidad de asignar los recursos de forma manual, imposibilidad de intercambiar ficheros entre el PC del alumno y el equipo remoto, alta latencia en la visualización de la salida de periféricos, y gestión manual individual de los usuarios sin capacidad de manejo en bloque.

1.2. Desarrollo Realizado

Para la realización de este proyecto se han marcado una serie de objetivos a cumplir para cubrir las necesidades básicas identificadas: escalabilidad hardware del sistema para uso de múltiples equipos remotos, implementación de un sistema automatizado de reserva de equipos,

implementación de un mecanismo de gestión de ficheros entre el equipo del alumno y el equipo remoto y mejora del sistema de visualización de los equipos remotos y sus periféricos.

Para alcanzar los objetivos marcados se han de seguir tres pasos claramente diferenciados:

- Evaluar las limitaciones del sistema, tanto de escalabilidad como de funcionalidad.
- Analizar diferentes soluciones para dichas limitaciones, e implementar la más adecuada.
- Verificar la viabilidad del sistema mediante el despliegue del mismo para su libre uso por los alumnos de la asignatura de Introducción a los Computadores.

Los objetivos planteados se han cumplido satisfactoriamente a lo largo de la realización del proyecto y con la antelación suficiente para realizar una prueba de uso real con los alumnos de la asignatura de Introducción a los Computadores (segundo cuatrimestre, segundo curso del Grado en Ingeniería Informática). Además, se han realizado desarrollos adicionales sobre la herramienta, no contemplados al inicio del proyecto.

El objetivo de la herramienta es servir de complemento para las prácticas desarrolladas en el laboratorio, paliando las consecuencias de la pandemia y la crisis de semiconductores, gracias al acceso remoto y la compartición de recursos.

El sistema original se divide en dos grandes componentes: hardware y software. El apartado hardware del proyecto inicial se estructura en cuatro módulos, cuyo funcionamiento se resume de forma breve a continuación. En la Sección 2.1 se detalla el funcionamiento inicial de cada módulo y sus características.

- Módulo de alimentación: suministra la energía necesaria para alimentar los equipos Raspberry Pi y otros módulos del sistema.
- Módulo de control: basado en lógica Arduino [1], realiza el control básico de la alimentación de los equipos remotos (encendido, apagado y reinicio) y de la entrada de propósito general.
- Módulo de red de interconexión: permite la comunicación entre todos los módulos, los equipos remotos y el servidor de la herramienta.
- Módulo de visualización: permite al alumno visualizar la salida de dispositivos periféricos conectados al equipo remoto.

El proyecto inicial está formado por los 4 módulos mencionados y un servidor que gestiona todo el sistema. El servidor emplea el sistema operativo Ubuntu en su versión de soporte a largo plazo (LTS, *long term support*) de 2020. Los equipos remotos Raspberry Pi utilizan la misma imagen del sistema operativo RISC OS empleada en el laboratorio presencial.

Se espera que la herramienta pueda ser publicada como proyecto de código libre para su uso en otras universidades o centros educativos. Por ello, se evitará el uso de cualquier hardware o software propietario que requiera de licencias privativas.

Tras completar el análisis de las limitaciones, se ha hecho un estudio de las soluciones hardware y software disponibles en el mercado y compatibles con el proyecto original. En lo referente al hardware, se ha rediseñado el módulo de control, manteniendo el uso de lógica Arduino. Respecto al software, se ha ampliado el desarrollo original empleando los mismos entornos de desarrollo iniciales. Se ha implementado un sistema de reservas, un gestor de ficheros compartidos, se ha expandido la gestión de usuarios y equipos y se han aplicado mejoras al módulo de

visualización para reducir su latencia.

En base al análisis de limitaciones y escalabilidad se ha ampliado el prototipo original. Se ha empleado una metodología de desarrollo de las mejoras software con un enfoque iterativo-incremental, en el que se han aplicado bloques de cambios sobre el sistema y se han evaluado antes de introducir otros. De esta forma, se ha añadido y modificado funcionalidad sobre el prototipo inicial sin necesidad de romper la funcionalidad previa. Posteriormente se ha procedido a su evaluación, primero mediante pruebas propias y después permitiendo el acceso a los alumnos de la asignatura de Introducción a los Computadores. Los resultados obtenidos tanto por parte de los alumnos como de los profesores responsables demuestran que la herramienta está lo suficientemente madura como para ser útil como mecanismo de apoyo para las sesiones prácticas.

El resto de la memoria está estructurado como sigue. En primer lugar, se ha detallado el hardware del sistema, tanto el escenario previo y sus limitaciones como el estudio de soluciones y su posterior implementación. A continuación se analiza el apartado software, en el que se hace un estudio previo de las funcionalidades carentes y se propone e implementa una solución para cada una de ellas, así como otras mejoras adicionales. Posteriormente se hace un breve análisis de la utilización del sistema por parte de los alumnos y los resultados de las encuestas de satisfacción con el sistema. Finalmente, se exponen las conclusiones extraídas del trabajo y las posibles líneas de trabajo futuro.

2. DISEÑO HARDWARE

Este capítulo se centra en analizar las necesidades a cubrir para dotar al sistema de escalabilidad hardware. Para ello, es necesario tener en cuenta el amplio número de alumnos que hacen uso del laboratorio (entre 90 y 120 en cada cuatrimestre), tal y como se explicó en el Capítulo 1. Sin embargo, dado que la versión inicial al comienzo de este trabajo era un prototipo de uso para verificar la viabilidad de la herramienta, el número de recursos en la misma era muy reducido. El prototipo inicial permitía conectar un máximo de cuatro equipos, reduciéndose a sólo uno o dos en caso de querer añadir pulsadores hardware que pueda controlar el alumno a distancia, dado que se usan los mismos recursos para gestionar alimentación de equipos y entradas hardware. La configuración usada para las pruebas de viabilidad en el prototipo inicial disponía de dos equipos Raspberry Pi, pero con dos entradas conectadas únicamente a uno de los equipos.

Por lo tanto, el objetivo es tener la capacidad de soportar un laboratorio completo con la posibilidad de implementar dos pulsadores en varios de los equipos, como es el caso de la de forma similar al uso que se hace de la placa BerryClip [6] en la realización de algunas de las prácticas. Por ello, el número mínimo de equipos remotos que podemos conectar debería de ser de aproximadamente 16 Raspberry Pi con placas BerryClip, es decir, un total de 32 pulsadores. No obstante, actualmente los equipos conectados en el sistema se están reaprovechando de los empleados en el laboratorio, por lo que hay limitaciones estrictas en el número de equipos de los que se puede disponer. Además, el uso del sistema se concibe para compartir los recursos entre múltiples usuarios durante bloques flexibles de tiempo, por lo que el número de equipos puede ser más reducido. Así, inicialmente se buscará poder llegar a 16 Raspberry Pi y la mitad de ellas con dos pulsadores, dejando la mitad de equipos disponibles para la utilización de otros periféricos de entrada que no requieren interacción con el usuario (por ejemplo sensores de temperatura o humedad) y periféricos de salida (como pantallas LCD modelo HD44780U). Por lo tanto, se necesitan 32 conexiones gestionables.

El capítulo está organizado en dos secciones. La primera sección detalla las características del prototipo inicial y analiza las limitaciones de cara a alcanzar la escalabilidad justificada en la Sección 1.1. La segunda sección estudia posibles soluciones para alcanzar los objetivos de escalabilidad planteados, y presenta los cambios efectuados en la implementación del prototipo en base a dichas soluciones.

2.1. Escenario previo y estudio de limitaciones

En esta sección se analiza el escenario previo con el que se ha trabajado y aplicado cambios en algunos aspectos para lograr la escalabilidad deseada. La estructura se compone de diferentes módulos, que se detallan de forma individual en las secciones posteriores. Estos módulos son cuatro: alimentación, control, interconexión y visualización. La Figura 2.1 ilustra las relaciones entre módulos, cuyas características se indican a continuación:

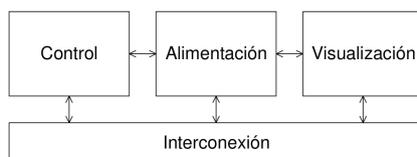


Figura 2.1: Esquema de los módulos utilizados

- Alimentación: se encarga de suministrar la energía al sistema. Originalmente se dispone de una fuente de alimentación de 80W que proporciona la energía necesaria para alimentar hasta ocho equipos Raspberry Pi y los módulos de control del sistema indicados a continuación.
- Control: gestiona el manejo básico de la alimentación de equipos Raspberry Pi (encendido, apagado y reinicio) y de entradas de propósito general. En el prototipo original se emplea una placa Arduino modelo UNO R3 [5] y dos módulos de expansión, uno para añadir conectividad Ethernet a la placa Arduino, y otro con cuatro relés.
- Red de interconexión: facilita la comunicación entre el resto de módulos y con el servidor del sistema. En la versión preliminar del sistema se tiene una pequeña subred local a la que, mediante un switch Fast Ethernet, se conectan todas las Raspberry Pi y la placa Arduino con el servidor.
- Visualización: para poder observar la salida de dispositivos hardware conectados a los equipos remotos, se tiene conectada una webcam VGA con resolución 640x480 píxeles a través de una interfaz USB al servidor. La señal de video capturada por la webcam proporciona al alumno la imagen a tiempo real de los periféricos conectados a los dispositivos y le permite verificar que la actuación sobre los mismos en los programas desarrollados se ajuste a lo esperado.

2.1.1. Módulo de alimentación

El proyecto actual se alimenta a través de una fuente de alimentación de 80W OMRON S8FS-G10005CD, mostrada en la Figura 2.2. Esta fuente nos proporciona 5V y 16A, y dado que cada Raspberry Pi necesita una corriente de 1,8A, nos permite tener hasta aproximadamente ocho Raspberry Pi conectadas. Si además se emplea esta misma fuente para alimentar la placa Arduino, cuyas especificaciones de energía son similares a los equipos remotos, se reduce en uno el número máximo de equipos que se pueden alimentar. Esto es inferior al mínimo de 16 equipos conectados al sistema para la simulación de un laboratorio como se detalló al comienzo del capítulo. Para poder conectar más dispositivos es preciso adquirir una fuente de alimentación de mayor amperaje: para el tamaño objetivo del sistema sería necesario emplear una fuente de al menos 150W, o suplementar la fuente actual con una segunda fuente de alimentación de al menos 70W.



Figura 2.2: Fuente de alimentación utilizada

2.1.2. Módulo de control

El módulo de control consiste en un controlador de relés basado en hardware Arduino. Un controlador de relé es un dispositivo que permite actuar sobre los distintos relés de forma remota mediante comandos software, en este caso a través de una comunicación en red Ethernet. Estos relés gestionan los conectores de alimentación de los equipos Raspberry Pi y los pines de entrada GPIO de los equipos, permitiendo controlar el encendido y apagado de los equipos y

actuar sobre las entradas como si un pulsador estuviese conectado a las mismas.

La placa Arduino actúa como procesador de propósito general, actuando sobre los diversos relés, y se gestiona desde el servidor del sistema mediante una conexión Ethernet gracias a una placa de expansión. El control del sistema se realiza mediante una API propia y un servicio implementado en la placa Arduino.

En el propósito inicial del sistema se dispone de una placa Arduino UNO R3 que se conecta a un módulo Arduino 4 Relays Shield [2] que nos proporciona el control de cuatro relés. Este módulo nos limita a cuatro equipos Raspberry Pi como máximo, sin considerar actuación sobre pines de entrada de los equipos. Para disponer de conectividad entre la placa y el servidor se emplea un módulo Arduino Ethernet Shield Rev2 [3] que proporciona una interfaz Ethernet. La Figura 2.3 muestra la placa Arduino, los módulos de relés e interfaz Ethernet, y la conexión de la PCB de interfaces USB con los relés.



(a) Arduino UNO R3



(b) Arduino Ethernet Shield Rev2



(c) Arduino 4 Relay Shield



(d) Montaje de PCB de interfaces de alimentación USB conectada a la fuente

Figura 2.3: Conjunto de placas Arduino utilizadas

En el caso del manejo de alimentación de los equipos, se conectan los terminales de los relés a la fuente de alimentación y a una placa de circuito impreso (PCB, *Printed Circuit Board*) con conectores USB hembra. Esta disposición permite usar los mismos cables USB de alimentación que se emplean en el laboratorio, facilitando la compartición de recursos con el sistema remoto. Para la actuación sobre las entradas de propósito general, se conectan los relés a los pines GPIO de los equipos Raspberry Pi, conmutando un pin de entrada entre los pines de alimentación y tierra del conector.

Para que el alumno pueda tener control sobre el estado del dispositivo o las entradas de propósito general se gestiona el conjunto de placas Arduino mediante una interfaz *ad-hoc* que recibe y procesa peticiones HTTP del servidor actuando sobre los relés según la orden recibida. Las peticiones pueden ser de encendido, apagado o reinicio, en la que se realiza de forma automática el apagado y encendido conmutando el relé entre el terminal común y los otros dos (conectados a alimentación y tierra, respectivamente). Esta última funcionalidad es particularmente importan-

te, porque como se indicó en la Sección 1.1 los alumnos pueden producir fácilmente un bloqueo del equipo, siendo necesario reiniciar físicamente la alimentación del equipo.

Para el tratamiento de periféricos, concretamente pulsadores hardware con relés y la gestión de alimentación de los equipos remotos, el código de la API utilizado en la placa Arduino es el mismo. Arduino nos proporciona una gestión modular de los pines, lo que nos permite realizar diferentes funciones sin tener que modificar la declaración de los pines.

La placa Arduino Ethernet Shield Rev2 es capaz de soportar hasta ocho conexiones TCP abiertas de forma simultánea. Esto se debe a que en el protocolo TCP las conexiones requieren recursos dedicados hasta su cierre y por lo tanto se deben dedicar recursos limitados durante un periodo de tiempo. Si se llega a este límite, las nuevas conexiones serán rechazadas, lo cual puede llegar a suponer un problema, ya que limitaría el número de equipos sobre cuya alimentación o entradas se puede actuar. No obstante, dado que las peticiones HTTP se realizan mediante conexiones TCP no persistentes que se abren y cierran prácticamente al instante, la severidad de esta limitación es relativamente baja, ya que sólo se produciría si más de ocho alumnos mandan una petición y éstas llegan a la placa a la vez. Por lo tanto, no se ha contemplado como un problema crítico, y se mantiene el uso de esta placa.

Estudiando las limitaciones físicas del módulo de control de los dispositivos y los GPIO, se necesita un relé para cada conexión, y cada relé es controlado por una salida digital de la placa Arduino. La placa Arduino UNO R3 dispone de 14 salidas digitales en total, pero el módulo Arduino Ethernet 2 Shield se reserva tres salidas digitales para su comunicación con la placa Arduino UNO R3. Teniendo en cuenta esta limitación de 11 salidas digitales, si se deseara conectar a cada equipo un periférico de tipo pulsador sólo podríamos tener cinco equipos Raspberry Pi conectados al sistema. Cinco equipos son claramente insuficientes para un laboratorio de arquitectura como el empleado en las asignaturas de Estructura y Organización de Computadores, con capacidad para hasta 30 puestos presenciales y un uso agregado de más de 100 alumnos. Sin embargo, tenemos una limitación aún mayor en el sistema original por culpa del módulo de relés, ya que sólo se disponen de cuatro relés para gestionar tanto la alimentación de equipos remotos como entradas de propósito general.

Para escalar el módulo de control la única solución es aumentar linealmente el número de relés, ya que es la única forma de replicar la desconexión física de la alimentación. Una posibilidad es adquirir más módulos de relés como el empleado en el prototipo original, con capacidad tanto para conectarse a la placa Arduino como para gestionarlos de forma independiente con señales de control individuales. Otra posibilidad para poder escalar el número de relés con granularidad más fina es mediante el uso de relés discretos. Estos relés discretos se pueden adquirir de forma unitaria o en conjunto en una placa, pero no están preparados para su acople a una placa Arduino, si no que se gestionan desde cero, administrando la alimentación y controlando su funcionamiento a través de las patillas básicas de un relé (activación de la bobina para cerrar o abrir el circuito).

2.1.3. Módulo de red de interconexión

Se ha empleado un switch limitado a 100Mbps para reducir los costes de adquisición del sistema, dado que el modelo Raspberry Pi 1B+ sólo dispone de un conector de red limitado a dicha velocidad. Para el acceso de los alumnos al sistema de laboratorio remoto se emplea un PC actuando como servidor web conectado a la red interna de la Universidad mediante una segunda interfaz de red. Esta organización limita el acceso a los equipos Raspberry Pi y proporciona un

mayor nivel de seguridad, dadas las características del sistema RISC OS empleado en los equipos remotos que se expusieron en la Sección 1.

Además, el acceso a la herramienta se realiza a través de UnicanLabs. Esto nos da ciertas garantías de seguridad ya que el sistema no va a ser accesible desde cualquier lugar, si no que solo podrá hacerse dentro del entorno de la red de la Universidad con usuarios validados. Por lo tanto se tiene una capa de seguridad adicional sobre la que ya se proporciona por el propio sistema inicial.

2.1.4. Módulo de visualización

El módulo de visualización en el prototipo inicial consiste en una cámara web conectada por interfaz USB al PC. A pesar de que el uso de cámaras web con conexión USB sigue siendo común, existen alternativas cuyo uso es cada vez más frecuente, como las cámaras IP (inalámbricas o cableadas). Estas cámaras IP poseen una interfaz de red con una dirección IP propia, por lo que son capaces de emitir vídeo sin necesidad de un PC externo y software adicional para realizar la emisión de la señal de vídeo, a diferencia del sistema actual.

La calidad de la imagen captada por la webcam es reducida, con una resolución máxima de 640x480 píxels y un alto nivel de ruido de imagen que se traduce en pérdida de nitidez y detalle en el vídeo resultante. La falta de calidad y resolución puede suponer la pérdida de detalles en la observación de la salida de los periféricos, pero a grandes rasgos cumple su cometido, si consideramos que algunas de las salidas de periféricos hardware se limitan a encender o apagar un LED. Por otro lado, la señal de vídeo tiene una latencia de imagen elevada, en torno a varios segundos, lo que en ocasiones puede ser un inconveniente para el alumno a la hora de verificar el correcto funcionamiento de su código. Sin embargo, dicha latencia proviene del procesado de la imagen y no de su captación, por lo que esta limitación se tratará posteriormente en la Sección 3.4

Sin embargo, como parte del estudio se ha descartado la transición al uso de cámaras IP, manteniendo el uso de la webcam USB. Esto es debido a que el sistema de laboratorio remoto ya requiere un PC para actuar como servidor del sistema, por lo que el uso de una cámara IP sólo permite reducir su carga de cómputo, no prescindir del mismo. Además, para evitar el paso de vídeo por el servidor se necesita que cada cámara sea asignada una dirección individual desde el Servicio de Informática de la Universidad. Finalmente, el coste de una cámara IP es muy superior al de una webcam convencional, dado el hardware adicional que incluye.

Otro problema que surge con la escalabilidad del número de equipos remotos es el encuadre de todos los periféricos conectados en el ángulo de visión de una sola cámara web. Actualmente los equipos remotos conectados pueden ser retransmitidos conjuntamente. Los periféricos están diferenciados mediante etiquetas colocados en los mismos, que identifican a qué equipo remoto pertenecen. Dado que la visualización de los periféricos no es información sensible que pueda comprometer el código de un alumno, no hay problema en que se visualicen de forma generalizada.

Sin embargo, la solución adoptada en el prototipo presenta limitaciones de escalabilidad con el número de equipos remotos. La cámara web del prototipo tiene un campo de visión que limita la captura a aproximadamente cuatro equipos, siendo suficiente para los dos equipos disponibles originalmente pero no para un sistema de mayor tamaño. Por lo tanto, en la Sección 2.2.5 se analizan algunas soluciones que se pueden adoptar en el módulo de visualización para su uso con un número elevado de equipos remotos.

2.1.5. Resumen de limitaciones

Cada uno de los módulos descritos en las secciones anteriores supone una serie de limitaciones hardware descritas de cara a la escalabilidad del sistema. A continuación se recoge un resumen de estas limitaciones para el posterior planteamiento de las soluciones a cada una de ellas.

- Alimentación: la potencia que suministra la fuente no permite superar ocho equipos remotos conectados sin incumplir las recomendaciones del fabricante [21].
- Control: este módulo supone la limitación más restrictiva debido al reducido número de relés y de pines digitales disponibles en la placa Arduino. En el prototipo inicial este módulo sólo permite gestionar cuatro actuadores totales a repartir entre alimentación de equipos remotos y entradas de periféricos hardware. En la Sección 2.2.1 se explicará con mayor detalle las causas que impiden aumentar el número de relés soportados con la placa Arduino UNO, y que requieren el análisis de soluciones alternativas para aumentar la escalabilidad del sistema.
- Red de interconexión: el número de puertos del switch es limitado, con un total de siete interfaces a repartir entre el PC servidor del sistema, la placa Arduino, y los equipos remotos. Pese a que la configuración del prototipo inicial impide gestionar más de cinco equipos remotos, esta limitación puede resolverse fácilmente con la adquisición de switches adicionales o la sustitución por un switch con mayor número de puertos, siendo la capacidad del prototipo superior a lo permitido por el módulo de control.
- Visualización: la principal limitación del módulo es la latencia de vídeo desde la captura hasta su visualización en el PC del alumno. Sin embargo, esta latencia se debe al procesamiento de la imagen en el servidor, y se analizará en la Sección 3.4. Respecto a la escalabilidad del sistema, a medida que se aumenta el número de equipos remotos en el sistema se necesita un mayor campo de visión para poder capturarlos a todos, estando el límite de la implementación original en torno a cuatro equipos.

Por último, durante la realización de las pruebas de uso por parte de los alumnos se ha observado una limitación adicional debida al equipo empleado como servidor. Este equipo se ha reaprovechado tras considerarlo obsoleto para su uso en un laboratorio, y tiene carencias de rendimiento debidas a sus características de procesador, memoria y disco y a un uso constante prolongado. Aunque esto no representa una limitación de escalabilidad del sistema como las estudiadas anteriormente, sí demuestra una carencia que es necesaria paliar en el montaje del sistema final. Por tanto, para la versión final de la herramienta se contempla la necesidad de adquirir un nuevo equipo para su uso como servidor. Es de esperar que esto también impacte en la escalabilidad futura del sistema, ya que una mayor capacidad del servidor facilita poder gestionar las conexiones a un mayor número de equipos remotos de forma simultánea. Dado que la observación de esta limitación se ha producido durante la realización del trabajo y aún no se dispone de las características del futuro equipo, se ha dejado el análisis de las limitaciones de rendimiento del equipo como línea de trabajo futuro, como se discute en la Sección 5.1.

2.2. Propuesta de mejora e implementación de la solución elegida

En esta sección vamos a explorar dos posibles soluciones para el problema de escalabilidad en el módulo de control, destacando sus ventajas y desventajas sobre el proyecto. También se estudiará una solución para los módulos de alimentación, interconexión y visualización. Por último, se detallará como se han implementado las opciones elegidas.

2.2.1. Módulo de control

Para aumentar la escalabilidad del sistema se proponen dos soluciones, cada una enfocando el problema desde un punto de vista diferente. La primera propuesta parte de aplicar la misma estrategia del prototipo inicial, pero empleando placas y módulos diferentes que soporten un mayor número de relés. Para ello, se sustituiría la placa Arduino UNO por una Arduino MEGA, de mayores prestaciones y coste. En la otra solución se intenta alcanzar la escalabilidad manteniendo la misma placa Arduino UNO, pero empleando una estrategia diferente mediante el uso de registros de desplazamiento.

Durante el desarrollo del prototipo inicial, se barajaron otras alternativas al uso de placas Arduino para implementar el módulo de control, tales como controladores de relés de propósito específico, unidades de distribución de alimentación, o enchufes inteligentes. Sin embargo, estas alternativas se descartaron por coste o falta de disponibilidad hardware, limitaciones que aún siguen presentes e incluso agravadas. Por ello, esas alternativas no se han tenido en cuenta y se ha decidido seguir empleando placas Arduino para la ampliación del proyecto. Puede leerse una descripción pormenorizada de dichas limitaciones en el trabajo de Fin de Grado en el que se desarrolló el prototipo inicial [24]

2.2.1.1. Estrategia 1: Uso de la placa Arduino MEGA

La placa Arduino MEGA 2560 Rev3 [4], mostrada en la Figura 2.4, es una ampliación de la placa Arduino UNO empleada originalmente en el prototipo, pero que incluye más salidas digitales, concretamente 54 en lugar de las 14 salidas de la placa UNO. También dispone de más memoria SRAM y EEPROM, lo cual nos permite atender las conexiones TCP simultáneas con un mayor margen de operación en cuanto a memoria disponible, si bien es cierto que la placa UNO no tiene problemas con el tráfico de datos que se generan en el prototipo original. El resto de las características relevantes para el proyecto son iguales a la placa Arduino UNO.

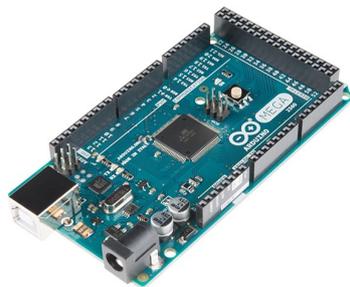


Figura 2.4: Arduino MEGA 2560 Rev3

Esta placa nos daría terminales suficientes para cumplir nuestro objetivo de escalabilidad (16 equipos Raspberry Pi con dos entradas de propósito general). No sería necesario modificar la estructura hardware del sistema dado que el único cambio significativo es el aumento de terminales de salida. Tampoco es necesario hacer cambios sustanciales sobre el código utilizado en la placa Arduino UNO, ya que es totalmente compatible entre ambas placas. Sin embargo, su uso tiene algunas desventajas de cara a nuestro proyecto:

- Para dedicar dos botones por dispositivo (como la placa BerryClip usada en las asignaturas), necesitamos tres pines por cada dispositivo, limitando a 16 dispositivos aproximadamente. Esto supone que para ofrecer más equipos remotos en el sistema necesitamos

añadir placas Arduino adicionales.

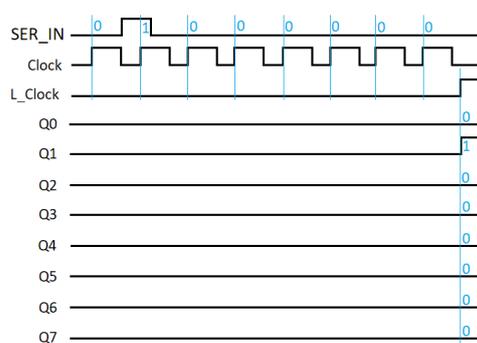
- El coste de adquisición de la placa es de aproximadamente 36€, frente a los 21€ que cuesta la placa Arduino UNO R3 original.

Mediante la Arduino MEGA se puede alcanzar el límite de dispositivos propuesto para un laboratorio. Sin embargo, en el caso de querer seguir aumentando el número de equipos ofrecidos en el sistema es necesario emplear múltiples módulos de control basados en placas Arduino o adoptar una estrategia diferente, ya que no hay placas de mayor número de salidas digitales.

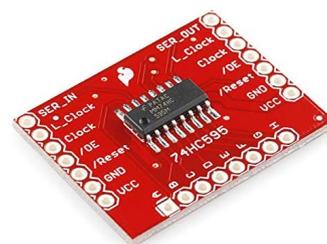
2.2.1.2. Estrategia 2: Uso de Registros de Desplazamiento

La otra solución contemplada para garantizar la escalabilidad del módulo de control es el uso de registros de desplazamiento. Un registro de desplazamiento es un circuito digital que emplea un serie de biestables (*flip-flops*) en cascada, permitiendo convertir una secuencia de bits serie en salidas paralelas. Por cada uno de los bits recibidos, se activa o desactiva la salida correspondiente a dicha posición, según si el bit tiene un valor 0 ó 1. De este modo, existe una relación exponencial entre el número de entradas empleadas y el número de salidas que ofrece. Estos dispositivos son muy comunes y baratos, por lo que no hay problemas de abastecimiento o de coste de adquisición del sistema. La Figura 2.5a muestra un ejemplo del funcionamiento de un registro de desplazamiento, convirtiendo una entrada dada en la activación de ciertas salidas. Como se puede apreciar en cada señal de reloj se marca el valor del bit de datos, en el momento que se activa el flanco de la señal “L_Clock” los ocho bits de datos que se encuentren en el *buffer* de datos salen de forma paralela por su salida correspondiente. Por ejemplo, se observa que los dos primeros bits son '0', '1' y '0', haciendo que las señales Q0, Q1 y Q2 tomen valores bajo, alto y bajo respectivamente.

El registro de desplazamiento en concreto es el modelo SparkFun 74HC595 [13], el cual se encarga de recibir una entrada de datos en serie de ocho bits y sacarlos en paralelo simultáneamente. Este modelo, ilustrado en la Figura 2.5b, es uno de los modelos más utilizados junto a las placas Arduino para realizar este tipo de operaciones.



(a) Cronograma del funcionamiento básico



(b) Implementación 74HC595 de SparkFun

Figura 2.5: Registro de desplazamiento de 8 bits

La salidas de este chip se encuentran a un voltaje de aproximadamente 3.3V, coincidiendo con el voltaje que reciben los terminales GPIO de las Raspberry Pi [20]. Esto nos va a aportar una gran utilidad ya que podemos conectar directamente la salida del registro de desplazamiento con los terminales de GPIO en los que queramos simular un pulsador. De esta forma, ampliamos el número de relés disponibles para el control de alimentación de los equipos remotos. No obstante, para evitar problema de variaciones de tensión es necesario tener el registro de desplazamiento

conectado a la misma tierra que los equipos remotos. Como las tomas de alimentación y tierra del registro de desplazamiento provienen de la placa Arduino, esto lo logramos alimentando la placa Arduino, por la misma fuente de alimentación que tienen los equipos remotos. Para el control de la alimentación de las Raspberry Pi sigue siendo necesario el uso de relés debido a que no es compatible con salidas de 5V y 1.8A.

El esquema de conexiones de este chip, está compuesto por 7 entradas y 8 salidas de datos, y 7 salidas de control para expansión. Las 8 salidas de datos serán las que controlen los relés, mientras que las 7 salidas de expansión irán conectadas a las 7 entradas del siguiente chip indicándose por el mismo nombre, facilitando significativamente la ampliación del circuito. Las entradas incluyen alimentación, tierra y 5 señales para gestionar el funcionamiento del registro. De éstas cabe destacar 3: una para introducir la serie de datos a paralelizar (SER_IN), otra para marcar la actualización de las salidas (L_Clock) y una de reloj (Clock) para interpretar a las demás. Las 2 entradas restantes, nos permiten realizar funciones especiales sobre el registro de desplazamiento: una deshabilita las salidas y otra pone todas las salidas a 0. Estas señales irán conectadas a alimentación y tierra respectivamente dado que carece de interés para el proyecto. Por lo tanto, sólo tendremos 5 conexiones a la placa Arduino para gestionar las 32 salidas bus-cadas, siendo necesarios 4 registros de desplazamiento encadenados.

Este dispositivo implica replantear la estructura hardware del prototipo inicial debido a que la conmutación de los relés ya no se realizaría desde la placa Arduino UNO R3, si no que se controlaría desde las salidas del registro de desplazamiento. Sin embargo, esta solución tiene numerosas ventajas:

- El número de salidas digitales que debemos dedicar del Arduino UNO R3 es fijo (tres salidas de información y dos de alimentación) por lo que el número de pines digitales deja de ser una limitación para el número de dispositivos a conectar.
- Este modelo está preparado para poder conectarse en cascada con otros registros de desplazamiento y por lo tanto poder aumentar el número de chips sin necesidad de aumentar la complejidad del circuito. Esto es posible porque el registro recibe una señal serie, cuya longitud puede variarse sin modificar el número de entradas desde la placa Arduino.
- Proporciona una salida de 3.3V, lo cual nos permite controlar los relés y poder prescindir de un relé por cada botón, permitiendo poder conectar muchas más Raspberry Pi con los mismos recursos.
- Ofrece mucha más flexibilidad en el uso de relés, ya que se puede optar por placas como el Arduino 4 Relays Shield o por relés individuales, según necesidades del sistema, coste de adquisición y disponibilidad de compra.
- Arduino tiene una gran comunidad de desarrollo, contando con numerosas librerías que nos permiten gestionar las salidas del registro de desplazamiento tal y como se trataban en el sistema inicial, como pines digitales. Esto permite reusar el código de gestión del prototipo casi por completo.
- El coste económico en el mercado es de aproximadamente 1€ por chip.

No obstante, esta solución hace necesario emplear una placa de pruebas para poder montar el conjunto de registros de desplazamiento conectados, o el desarrollo de una placa PCB dedicada. Sin embargo, esta necesidad no supone una desventaja frente a la primera solución, ya que para la implementación con Arduino MEGA también es necesaria una placa de pruebas para la actuación sobre las entradas de los equipos Raspberry Pi (replicando el comportamiento de

pulsadores hardware). Este registro de desplazamiento es un gran candidato para su uso en el proyecto, puesto que no sólo satisface los requisitos para el control de un mayor número de relés sino que también prescinde de su uso para la gestión de entradas de propósito general en los equipos remotos. Por último, el precio de esta solución es muy inferior al de la placa Arduino MEGA, más aún si se considera que podemos reaprovechar el módulo de expansión de relés de la implementación en el prototipo original.

2.2.1.3. Implementación de la solución elegida

A raíz del estudio previo se decide añadir al sistema actual dos registros de desplazamiento, con un coste total de aproximadamente 4€. Esta implementación nos ofrece un total de 16 salidas de control mediante tres salidas digitales de la placa Arduino UNO R3, más dos de alimentación.

La interconexión es sencilla y no requiere de ningún hardware adicional más allá de los cables necesarios para un mayor número de equipos o entradas a los mismos. Como se ha mencionado en la Sección 2.2.1.2, gracias a que los registros de desplazamiento están preparados para conectarse en cascada, simplemente se conectan las salidas de un chip con las entradas del siguiente. El segundo chip y sucesivos reciben tanto las señales de datos como alimentación y tierra de los anteriores, por lo que estos chips no tienen ninguna conexión con la placa Arduino. En la Figura 2.6 puede observarse un ejemplo del esquema de conexionado del montaje. Se puede apreciar las conexiones del registro de desplazamiento detalladas en la Sección 2.2.1.2, el control de los cuatro relés con las salidas del chip y se ha añadido el control de los GPIO2 y GPIO3 de la Raspberry Pi, que hace la función de pulsador hardware.

En cuanto a la placa de relés y la placa de puertos USB, se han mantenido las placas ya presentes por motivos de presupuesto. Si se deseara en un futuro incrementar la capacidad del sistema simplemente se necesitaría incrementar el número de relés y puertos USB. Debido a esto, se tienen cuatro equipos remotos conectados.

Para poder aprovechar las salidas del registro de desplazamiento como simulación de pulsadores hardware para la Raspberry Pi, simplemente hemos de conectar la salida directamente al GPIO que deseemos, ya que tanto los equipos como la lógica de control Arduino comparten la misma referencia de alimentación.

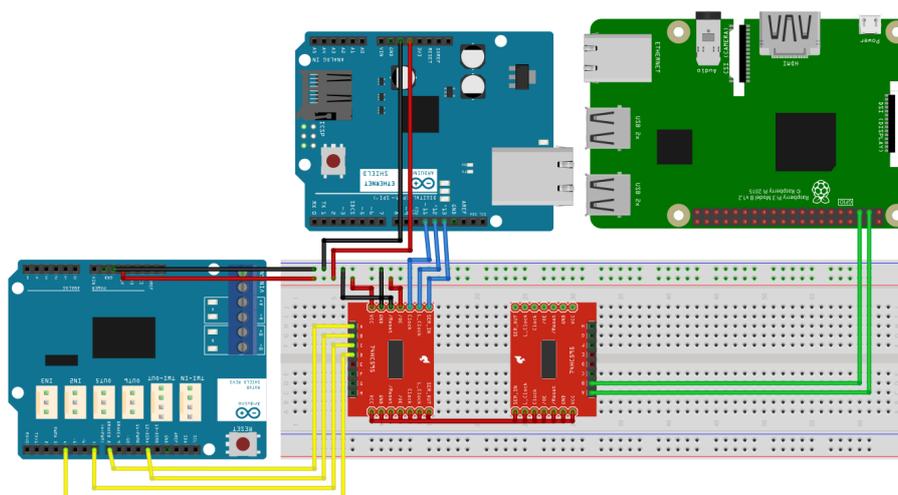


Figura 2.6: Esquema de las conexiones realizadas

2.2.1.4. Interfaz del módulo de control

El código utilizado por la API del servicio web para recibir las peticiones HTTP y controlar las salidas del registro de desplazamiento es similar al código previo. Las salidas del registro de desplazamiento se tratan como una salida digital de la placa, gracias al uso de una librería para Arduino [30]. Por lo tanto, para implementar el registro de desplazamiento solo hay que definir las características del mismo, como el número de registros de desplazamiento empleados y el mapeado de pines entre el registro de desplazamiento y la placa Arduino.

Cada vez que se actualiza el valor de las salidas, el registro de desplazamiento almacena todos los valores de las salidas actuales en un *buffer*. Por lo tanto, sólo es necesario indicar al chip las actualizaciones que se quieren hacer. Cada vez que se reciba una petición HTTP del servidor, se indicará al registro el nuevo valor de la salida que cambie, manteniendo el mismo comportamiento de la API en el prototipo inicial.

2.2.2. Módulo de alimentación

Debido a que la fuente de alimentación actual tiene una potencia de 80W, nos limita a un máximo de ocho equipos remotos, tal como se comentó en la Sección 2.1.1. El registro de desplazamiento 74HC595 puede ser alimentado por la placa Arduino UNO R3 a través de su salida de 5V y obtiene la referencia a tierra del mismo, por lo que no es necesario utilizar ninguna fuente de alimentación externa ni modificar la actual.

Sin embargo, la placa Arduino UNO R3 tendrá que ser alimentada por la fuente de alimentación del sistema y no por otra fuente, como se comentó en la Sección 2.2.1.2. De esta forma tendremos todo el sistema conectado a una tierra común con el objetivo de no tener variaciones de tensión.

Para el uso experimental de la herramienta con el alumnado se van a utilizar cuatro Raspberry Pi, por lo que se mantiene la fuente de alimentación actual. Tras verificar el uso de la herramienta con los cambios realizados durante el proyecto, se adquirirá el hardware restante para montar el sistema final de cara a su uso el próximo curso.

2.2.3. Módulo de red de interconexión

Adquiriendo más y/o mejores recursos se pueden mitigar las limitaciones del módulo de red de interconexión. Se plantea la adquisición de un switch complementario al actual u otro con mayor número de puertos para aumentar el número de equipos remotos que se pueden conectar. En cuanto al ancho de banda del switch, se ha medido el consumo de red del prototipo para verificar si su uso por parte del alumno requiere una capacidad superior a los 100Mbps disponibles. Sin embargo, se ha observado que la concurrencia de alumnos es pequeña, teniendo los mayores consumos medios en torno a 500Kbps en las horas punta. Por lo tanto, el consumo de red no supone un problema y se mantiene el ancho de banda del switch empleado en la implementación original.

2.2.4. Integración del sistema

El prototipo inicial del sistema está implementado por los módulos descritos en la Sección 2.1. Estos módulos están físicamente separados, y se conectan mediante cables para conformar el sistema completo. El uso de este cableado para conectar los distintos componentes del sistema, como la placa Arduino, los registros de desplazamiento, los relés o la fuente de alimentación, resulta un inconveniente para la escalabilidad del sistema. Para mejorar la integridad y porta-

bilidad del sistema se contempla la fabricación de una PCB que tenga integrados ocho relés y ocho puertos USB para la conexión de los equipos Raspberry Pi a la fuente de alimentación y tres registros de desplazamiento 74HC595 con sus correspondientes interconexiones ya impresas en la placa. La PCB incluye también una interfaz para las salidas del tercer registro de desplazamiento, para poder conectar en serie otra placa PCB si se desea expandir el sistema.

Con este diseño, podríamos necesitar cableado únicamente para conectar la PCB a la placa Arduino UNO que se encarga de gestionarla, a la fuente de alimentación, y a los equipos remotos. También es posible integrar la placa Arduino UNO en la PCB, reduciendo aún más las dimensiones del sistema. El objetivo final sería poder montar el sistema completo en una estructura de tipo *rack*, facilitando su colocación en el laboratorio.

Se ha realizado el diseño para un máximo de ocho equipos remotos porque es el máximo que podemos emplear con la fuente de alimentación actual. La alimentación de estos ocho equipos remotos se gestiona a través de un registro de desplazamiento y las entradas de periféricos gestionables a través de los otros dos registros de desplazamiento. Esto nos permite tener dos entradas por cada equipo remoto. De este modo, el sistema se puede escalar en bloques de ocho equipos Raspberry Pi, añadiendo una nueva fuente de alimentación y un nuevo conjunto de relés y registros de desplazamiento que se encadenan al anterior, compartiendo el control con la misma placa Arduino. Alternativamente, se puede contemplar rediseñar la PCB para ampliar el número de equipos soportados, empleando una fuente de alimentación de mayor amperaje.

En la Figura 2.7 se muestra un esquemático realizado para la PCB a modo de ejemplo para poder visualizar el sistema. En la parte inferior de la placa se ubican los tres registros de desplazamiento encadenados. El registro de la izquierda se conecta a los relés, situados en la parte superior izquierda de la PCB. Cada relé se conecta a su vez a los pines de alimentación en los interfaces USB hembra. Los terminales de los dos registros de desplazamiento restantes están diseñados para unir bloques de pines macho, facilitando el conexionado a los pines GPIO de los equipos remotos mediante cables para prototipado.

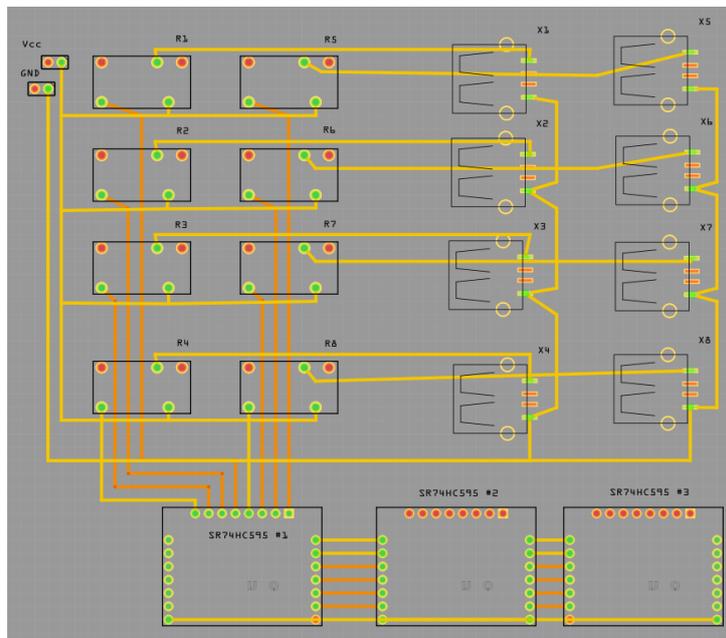


Figura 2.7: Esquema de PCB diseñado

2.2.5. Módulo de visualización

Como se expuso en la Sección 2.1.4, la cámara empleada en el prototipo original permite visualizar un máximo de cuatro equipos remotos. Para garantizar la escalabilidad del módulo con el resto del sistema se valoran dos opciones: aumentar el número de cámaras web, o adquirir una cámara de gran angular.

Aumentar el número de cámaras web implica modificar el tratamiento de las imágenes desde el servidor. Actualmente todos los alumnos visualizan la misma retransmisión, por lo que el uso de múltiples cámaras requiere una asignación de cada equipo remoto a una de las cámaras web conectadas al servidor. El coste aproximado de una cámara web de angular normal es de 30€, y se puede compartir una cámara web para mostrar 3 o 4 periféricos (dependiendo del tamaño de estos), por lo que el coste por equipo oscilaría en torno a 9€.

El uso de una cámara de gran angular nos permite tener un mayor campo de visión, donde éste puede ir desde los 80º hasta los 120º. Con este campo de visión podemos mostrar aproximadamente 6 periféricos. Por tanto, el uso de una cámara de gran angular no elimina la necesidad de disponer de múltiples cámaras para un alto número de equipos remotos, como se contempla para el sistema final. El coste aproximado de una cámara de gran angular es de 200€, por lo que el coste por equipo remoto sería de aproximadamente 33€.

Teniendo en cuenta los costes por equipo remoto de cada tipo de cámara, y la necesidad de modificar el sistema para emplear múltiples cámaras en cualquiera de los dos casos, el uso de cámaras de gran angular es económicamente contraproducente aunque reduzca el número de dispositivos de captura de vídeo conectados al servidor. Respecto a los recursos computacionales que emplea cada una de las cámaras, el uso de red para la transmisión de imagen será bastante similar ya que el número de píxels en el cómputo global será parecido. Para la utilización de CPU, en el equipo actual que alberga el servidor se observa un aumento del 5% en el momento que se comienza a transmitir la señal de vídeo. En el caso de añadir 3 o 4 cámaras web no tendríamos un consumo de CPU muy significativo, por lo que la diferencia en número de cámaras entre una solución y otra no va a tener excesivo impacto en el rendimiento del sistema.

3. DISEÑO SOFTWARE

Este capítulo cubre el análisis de escalabilidad de los distintos componentes software del prototipo inicial del sistema de laboratorio remoto para alcanzar las necesidades planteadas en la Sección 1.1. El objetivo es cumplir con la funcionalidad principal de la herramienta: proporcionar al alumno una experiencia de uso análoga al trabajo presencial en un puesto del laboratorio de Arquitectura de Computadores. Por lo tanto, la parte software de la herramienta debe prepararse para su uso por un número elevado de alumnos, lo cual es inviable en el prototipo inicial ya que los equipos remotos deben ser asignados a cada alumno de forma manual.

A lo largo del capítulo también se describen las funcionalidades que faltan en el prototipo inicial para facilitar el manejo del sistema tanto a alumnos como a profesores, y cuya carencia ha sido advertida durante las pruebas preliminares de uso por parte de los profesores. Estas funcionalidades se pueden clasificar en cuatro grandes grupos: un sistema de reservas temporales automatizado, un mecanismo de compartición de ficheros con los equipos remotos, un módulo de visualización de salida mejorado, y un sistema de gestión de usuarios y permisos.

Para paliar cada una de las deficiencias existentes en la versión preliminar del sistema, primero se realiza un análisis de las necesidades a cubrir. A continuación, se estudian posibles soluciones para cubrir dichas necesidades, y se elige la más adecuada. Finalmente, se describe la implementación de los cambios.

3.1. Escenario previo y estudio de limitaciones

En esta sección se va a hacer un breve resumen de la implementación software inicial del sistema de laboratorio remoto para después destacar las limitaciones observadas que tiene y que se han solventado durante la realización del trabajo.

El sistema consta de una interfaz web para interactuar con la herramienta, y de un *backend* para gestionar los recursos y operar sobre la lógica de control de sistema. El *backend* emplea el *framework* Laravel [26], un entorno de desarrollo de código libre basado en el lenguaje de programación PHP. Dicho entorno sigue el patrón de diseño Modelo-Vista-Controlador, y proporciona gran parte de los componentes y herramientas necesarias para realizar cada una de las partes del sistema.

En cuanto a la organización del sistema, para el tratamiento de los datos se utiliza una base de datos que contiene principalmente usuarios y equipos. El PC que actúa como servidor del sistema es el encargado de gestionar los accesos de los usuarios a los equipos Raspberry Pi remotos, a través de la subred descrita en la Sección 2.1. Para permitir al usuario manejar el escritorio del equipo remoto se utiliza noVNC [11], un cliente de comunicaciones VNC (*Virtual Network Computing*, computación virtual en red). El cliente noVNC, diseñado para permitir su embebido en páginas web, está basado en JavaScript y requiere soportar WebSockets, un protocolo de comunicaciones bidireccionales persistentes sobre una conexión TCP [35]. Esto nos permite mantener una comunicación entre el servidor y la Raspberry Pi a través de la subred privada para permitir la compartición de escritorio.

Para la interfaz gráfica del sistema de laboratorio remoto se hace uso del *framework* Bootstrap [25] que se basa en el lenguaje básico HTML5 [9]. La interfaz del prototipo inicial se compone de cuatro vistas, mostradas en la Figura 3.1



Figura 3.1: Paneles de la interfaz en el prototipo inicial del sistema

- Panel de validación (Figura 3.1a): en este panel el usuario introduce sus credenciales (correo electrónico y contraseña) para poder acceder al sistema. Es la primera vista que se muestra al acceder al sistema.
- Panel principal (Figura 3.1b): este panel nos muestra los equipos Raspberry Pi que hay en el sistema. Junto a cada equipo se muestra su estado (ocupado o disponible), permitiéndonos entrar sólo en aquellos que estén asignados a nuestro usuario. De forma contra-intuitiva, el sistema original marcaba los equipos asignados como “ocupados”, mientras que los equipos “libres” no estaban asignados a ningún alumno y por tanto no podían ser utilizados. Esta nomenclatura se ha actualizado como parte de las mejoras de uso de la herramienta, como se describirá en la Sección 3.2.2
- Panel del dispositivo (Figura 3.1c): este panel nos permite visualizar e interactuar con el equipo remoto, así como con el periférico hardware que esté conectado al mismo. La gestión de la alimentación del equipo se realiza mediante un bloque con tres botones, permitiéndonos cada uno apagar, encender o reiniciar el dispositivo. Los botones de encendido y apagado sólo están disponibles según el estado del equipo en cada momento; por ejemplo, si el equipo está encendido, no se da la opción de encenderlo, para evitar que la interacción con el relé de control interrumpa la alimentación por un instante. La interacción con los pines de entrada (marcados como GPIO 2 y GPIO 3) se hace mediante un bloque con un botón para cada uno. La visualización de los periféricos se realiza mediante un pequeño reproductor de vídeo.
- Panel de administración (Figura 3.1d): este panel, destinado a los profesores, contiene cada una de las entidades que tiene el sistema, como usuarios o máquinas, pudiendo realizar operaciones básicas sobre cada uno de ellos (listar, crear, editar y eliminar). Cada acción sobre usuarios debe realizarse de forma manual e individual (alumno por alumno).

El sistema dispone de un mecanismo básico de verificación de usuarios, basado en el uso de una dirección de correo electrónico y una contraseña. Además, el sistema no realiza ningún tipo de

verificación para prevenir ataques externos. Sin embargo, la herramienta sólo está accesible desde equipos de la propia Universidad de Cantabria o mediante la herramienta UnicanLabs [19]. Como ambos métodos de acceso involucran un nivel mínimo de garantía de seguridad, no se contempla como parte del trabajo un estudio de mecanismos de protección frente a ataques externos.

A pesar de que el prototipo inicial de la herramienta era funcional, tenía ciertas limitaciones que impedían que su uso pudiese aplicarse fácilmente a una clase con numerosos alumnos. Para afrontar dicho escenario a lo largo de este capítulo, analizaremos y resolveremos las limitaciones observadas. De forma preeminente se observan 4 deficiencias: asignación manual de alumnos a cada dispositivo, falta de compartición de ficheros, módulo de visualización con latencias variables y elevadas, y gestión de usuarios limitada.

3.2. Sistema de gestión de reservas temporales

En esta sección vamos a tratar la limitación de las asignaciones manuales de los equipos para su uso por los alumnos. Este enfoque es inviable en un entorno con un número de usuarios como el descrito anteriormente. Para corregirlo, se pretende desarrollar un sistema de reservas automatizado, que permita a los alumnos gestionar los recursos sin necesidad de la intervención manual de un profesor. Dicha funcionalidad permite liberar de carga al profesor y dar más flexibilidad en la utilización de los equipos, incentivando su uso.

El formato de las prácticas seguido en las asignaturas de Introducción a los Computadores y Estructura de Computadores es individual, como se mencionó en la Sección 1. Por lo tanto restringiremos el uso de los equipos remotos por medio de reservas a un solo alumno.

En primer lugar, se lleva a cabo un estudio previo en que se analiza y concreta los parámetros básicos que seguirá el sistema. Posteriormente, se detalla la implementación del mismo en la herramienta.

3.2.1. Requisitos del sistema de reservas

Para alcanzar la funcionalidad deseada, el sistema software de reserva de recursos debe cubrir dos tareas fundamentales:

- Reservar un dispositivo en una determinada fecha y hora.
- Permitir el acceso inmediato a un dispositivo siempre y cuando dicho dispositivo se encuentre libre en ese momento.

Para organizar la reserva de equipos en bloques de tiempo limitado debemos tener una perspectiva del uso que va a tener la herramienta por parte de los alumnos. Dicha perspectiva cubre varios aspectos:

- Duración de la reserva, contemplando el posible tiempo de utilización del equipo remoto.
- Rango máximo temporal en el que se puede hacer una reserva, es decir, hasta que día podemos reservar.
- Límite de reservas por día y dispositivo.

El primer punto está concebido para facilitar el uso de los equipos de forma flexible, facilitando un uso razonable pero sin acaparar recursos. Los otros dos puntos evitan estrategias de reserva “preventivas” que dejen el sistema completamente reservado a futuros. Además, dichos límites

evitan posibles usos erróneos del sistema como realizar una reserva para dentro de un mes, estimándose que dichos casos no van a ser realistas.

Para tener una visión más precisa del uso del sistema es conveniente realizar un sondeo a los alumnos acerca de sus previsiones de trabajo. Se contempla como grupo de partida el alumnado de la asignatura de Introducción de Computadores, de primer curso del Grado en Ingeniería Informática, ya que la mayoría de ellos no disponen al inicio de su propia Raspberry Pi. No obstante, no es posible realizar las encuestas con la suficiente antelación para implementar el sistema de reservas antes del comienzo de las clases. Como solución de compromiso, se han adoptado una serie de valores consensuados con el profesorado de las asignaturas involucradas, y se ha realizado una encuesta a los alumnos al comienzo de la asignatura. Los resultados de dicha encuesta y su análisis se describen en la Sección 4.2.

Dado que el uso mayoritario de la herramienta es para complementar las sesiones de laboratorio, y que para la realización de dichas sesiones se presta el equipo Raspberry Pi, se establece que el tiempo de sesión en el sistema remoto será de una hora. Además, teniendo en cuenta que los turnos de laboratorio son de dos horas, y dado que el objetivo de la herramienta es servir de complemento, se supone que una hora es suficiente. Sin embargo, los alumnos pueden utilizar los equipos siempre y cuando estén libres, por lo que pequeñas diferencias en este valor no deberían tener un impacto significativo. En cuanto al límite de reservas, se restringe a una reserva por día y Raspberry Pi. No se permite realizar otras reservas de cualquier equipo para el mismo día en el que ya existe una reserva, a excepción del uso libre. Este límite permite a los alumnos planificar un bloque de trabajo suficiente sin perjudicar al uso general del sistema.

Respecto a la utilización libre de los equipos, se puede tratar como una reserva que se realiza automáticamente al entrar en un dispositivo que esté libre en ese instante. Esta estrategia permite seguir garantizando fácilmente la exclusión mutua entre los alumnos sin necesidad de implementar otros mecanismos, y su uso queda registrado para conocimiento tanto del alumno como del profesor.

3.2.2. Diseño e implementación de la solución elegida

Como se ha analizado previamente en la Sección 3.2.1, se utilizará un tiempo de sesión de reserva de una hora, siendo posible ampliarlo en bloques consecutivos siempre y cuando no exista una reserva del equipo tras el final de la sesión actual. La ampliación de reserva se ofrece informando al usuario con un mensaje emergente antes de que acabe la sesión, permitiendo extenderla directamente, sin la necesidad de salir de la vista del equipo remoto. Por último, las reservas se pueden hacer en un intervalo máximo de una semana.

Para la realización de las reservas se plantean dos posibles implementaciones: una vista dedicada de reservas común a todos los equipos, o desde la vista principal de la herramienta (desde la que se accede a los equipos remotos). En caso de emplear un sistema común de reservas, se contempla que los alumnos deberán seleccionar la fecha y hora deseados, y desde ahí elegir uno de los equipos remotos disponibles para dicho bloque. En el segundo caso, se contempla actualizar la vista ya existente con los equipos remotos, y que desde ahí el alumno elija un equipo a reservar y posteriormente seleccionar la fecha y hora.

Se ha seleccionado la segunda opción, integrando el sistema de reservas en el panel principal existente. De este modo, se permite empezar a trabajar con un equipo remoto si está libre, o reservar el dispositivo para más adelante. Esta decisión se fundamenta en que un posible uso

futuro del sistema es disponer de equipos Raspberry Pi con diferentes dispositivos periféricos conectados, como se detalló en la Sección 1, pese a que en la actualidad dicha diferenciación no se emplee. Con dicho enfoque, el alumno debe primero seleccionar el equipo remoto que se ajuste a sus necesidades, antes de poder reservar un bloque temporal.

Para el proceso de reserva se ha seguido un proceso similar a la actual reserva de uso del comedor de la Facultad de Ciencias [12]. Para ello, una vez seleccionada la Raspberry Pi que se desea reservar, el alumno ha de seleccionar el día de la reserva y a continuación puede visualizar las horas disponibles para dicho dispositivo y día en forma de botones. Este proceso es familiar al alumno, por lo que la pasarela de reserva resultará más intuitiva y fácil de utilizar. Dicho proceso se representa en la Figura 3.2



(a) Panel para la selección de día

(b) Panel para la selección de hora

Figura 3.2: Pasarela de reserva

Como parte del trabajo, y para implementar el sistema de reserva propuesto, se ha modificado el aspecto del panel principal de la herramienta, mostrado en la Figura 3.3. Como parte de los cambios se ha modificado la información mostrada acerca del estado de la Raspberry Pi para hacerlo más detallado, estableciendo cuatro posibles estados: reservada, ocupada, disponible o asignada. El estado reservada se da cuando el usuario tenga el equipo reservado en ese mismo instante en el que se consulta. Los estados ocupada y disponible se han redefinido para solventar la confusión original detallada previamente en la Sección 3.1, en la que un alumno sólo podía acceder a aquellos equipos que estaban marcados como “ocupados” y no a los “libres”. Tras el cambio de nomenclatura, se indica la Raspberry Pi como ocupada cuando el equipo esté siendo utilizado por otro usuario, y por lo tanto no se puede acceder; y disponible cuando no esté siendo utilizada por nadie y se puede utilizar libremente. Por último, a pesar de implementar las reservas, se ha conservado la asignación manual de equipos, en cuyo caso se dará el estado asignada. Esta reserva se contempla para casos específicos en que un profesor quiera asignar el equipo a un alumno por un período indeterminado.

Además, se han añadido varios botones al bloque de información de cada equipo remoto, todos ellos relacionados con el sistema de reservas. Desde estos botones se puede efectuar una reserva, cancelarla, liberar un equipo que tengamos en uso (bien por acceso libre o bien por reserva explícita) y observar si hemos realizado alguna reserva del mismo, indicando la fecha y hora. En el caso de equipos que estén actualmente libres, se dispone de un botón adicional que permite acceder al equipo, dejando reservado su uso de forma automática desde el instante que se entra hasta la siguiente hora.

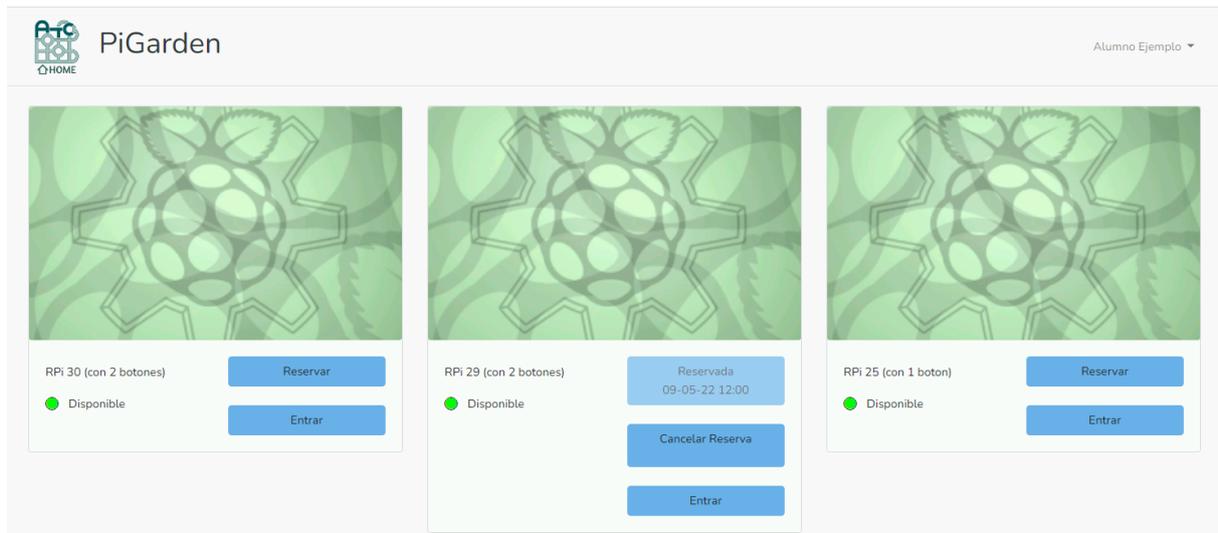


Figura 3.3: Panel principal

De forma complementaria, se ha añadido en las opciones del usuario una nueva página en la que el alumno podrá ver su historial de reservas, crear nuevas reservas o cancelar las futuras, mostrado en la Figura 3.4. Esto es útil cuando el alumno tiene varias reservas en diferentes días.

ID Máquina	Usuario	Dia	Hora Inicio	Hora Fin	Programada	
29	Alumno Ejemplo	17-06-22	14:00:00	15:00:00	Si	
29	Alumno Ejemplo	06-05-22	20:14:20	20:17:47	No	
29	Alumno Ejemplo	04-04-22	19:14:01	19:16:05	No	
30	Alumno Ejemplo	04-04-22	19:13:10	19:14:01	No	
29	Alumno Ejemplo	03-04-22	13:49:04	13:55:07	No	
29	Alumno Ejemplo	03-04-22	14:00:00	14:05:24	No	
24	Alumno Ejemplo	26-03-22	17:34:39	17:49:10	No	

Figura 3.4: Panel de las reservas del usuario

Finalmente, el uso de reservas automáticas de equipos que estén disponibles proporciona un manejo similar a las reservas programadas. Esto, además de facilitar el seguimiento del uso de los equipos, permite asegurar el uso sin miedo a expulsión durante el período actual. Además, al compartir el tratamiento con las reservas periódicas, también ofrece al alumno la posibilidad de extender la utilización si no hay ninguna reserva pendiente. Estas reservas automáticas no se incluyen en el cómputo de restricciones de las reservas programadas descrito en la Sección 3.2.1, ya que asumen un aprovechamiento de un recurso sin utilizar (sin perjudicar a otros usuarios del sistema).

3.3. Compartición de ficheros

El sistema original no dispone de ningún mecanismo para poder compartir ficheros entre el equipo local del alumno y el equipo remoto. Esto empobrece la experiencia de usuario e impide el flujo de trabajo empleado en el laboratorio.

La dinámica de trabajo prevista para la herramienta es que los alumnos dispongan de códigos no finalizados o sin depurar, los verifiquen y corrijan gracias al uso de un equipo remoto, y se lleven consigo el código ya finalizado, como se indicó en la Sección 1.1. Para ello, es preciso disponer de un mecanismo que permita compartir ficheros entre el PC del alumno y el equipo Raspberry Pi remoto.

En esta sección se detalla el estudio de las necesidades básicas y las tecnologías software utilizadas para dotar a todos los usuarios de la capacidad de compartir ficheros con los equipos Raspberry Pi remotos. El objetivo es que cada alumno pueda tener un espacio de usuario con sus ficheros en el servidor que podrá cargar y descargar en cualquier momento. Además, es necesario implementar una estrategia para compartir dichos ficheros entre el servidor del sistema y los equipos Raspberry Pi remotos.

Por último, se pretende que el profesor tenga la capacidad de compartir sus ficheros con equipos remotos que estén siendo usados por un alumno. El principal objetivo de esta funcionalidad es poder solventar dudas y establecer el correcto funcionamiento de los periféricos, que en ocasiones pueden estar dañados y dar resultados erróneos para un programa válido. Sin embargo, dicha compartición sólo debe producirse cuando el profesor lo indique, de forma que el alumno no pueda acceder a dichos ficheros sin la intervención del profesor y obtener el resultado de las prácticas por ingeniería inversa.

3.3.1. Estudio previo

La compartición de ficheros se realizará en dos pasos diferenciados: entre el PC local del alumno y el servidor, y entre el servidor y el equipo remoto. El servidor del sistema emplea un sistema operativo Linux, lo que facilita la compatibilidad con múltiples protocolos de compartición de ficheros, tanto con el PC del alumno como con los equipos Raspberry Pi remotos. Sin embargo, los equipos remotos emplean el sistema operativo RISC OS que, si bien facilita el manejo a bajo nivel de periféricos hardware, tiene una disponibilidad de software significativamente reducida. Por tanto, el objetivo principal es encontrar un protocolo para la compartición de ficheros en red que disponga de cliente y/o servidor para RISC OS y que también sea común a Linux. Para ello, se contemplan cuatro posibles protocolos de comunicación: SMB, NFS, FTP y SCP.

El protocolo FTP [8] (*File Transfer Protocol*, protocolo de transferencia de ficheros) tiene como principal ventaja las altas velocidades de transferencia. Sin embargo, los ficheros de programas en lenguaje ensamblador o los binarios ejecutables que se generan durante el desarrollo de las prácticas en el laboratorio de arquitectura son de apenas kilobytes, por lo que la velocidad de transferencia no tiene un impacto significativo a la hora de elegir el protocolo. Por otro lado, el cliente FTP disponible para RISC OS (FTPc [18]) no está implementado en la imagen del sistema que usan los equipos Raspberry Pi del laboratorio. Además, este cliente proporciona una interfaz gráfica distinta de la del explorador de archivos igual a la de RISC OS, lo que hace menos intuitivo su uso.

Por otro lado, existe la posibilidad de utilizar el comando SCP [32] (*Secure Copy Protocol*, protocolo de copia segura) para la transferencia de ficheros. Dicho comando funciona a través

de sesiones de shell seguras (SSH). El objetivo diferencial de esta opción es gestionar los ficheros desde el servidor hacia la Raspberry Pi y no al contrario, a diferencia del resto de opciones. Esto facilita la integración con el resto de la herramienta, ya que las órdenes de transferencia se realizan desde el servidor. También permite al alumno tener la misma mecánica de trabajo que en el puesto físico de laboratorio, ya que los ficheros se alojan físicamente en la Raspberry Pi durante la sesión del alumno. Sin embargo, RISC OS no tiene un soporte completo y sencillo para comunicaciones SSH, por lo que esta posibilidad es la más compleja de implementar.

Por último, la imagen de sistema de RISC OS trae por defecto la aplicación Acorn Omni-Client (también denominado “Omni”) [14], un cliente de transferencia de ficheros en red que es capaz de funcionar con los protocolos SMB1 [23] y NFS [10] (*Network File System*, sistema de ficheros de red). Tenemos otro cliente NFS para RISCOS, que no viene por defecto en la imagen del sistema y que por tanto no tenemos en los equipos remotos, como Sunfish [29]; no obstante, su desarrollo es mucho menor que Omni y ya no dispone de soporte, por lo que puede tener un comportamiento inestable. A diferencia de FTPc, el cliente Omni muestra los contenidos de los directorios en línea como si fuese un directorio físico en la Raspberry Pi, utilizando el explorador de archivos nativo de RISC OS.

El protocolo SMB (*Server Message Block*, bloque de mensajes de servidor) permite compartir archivos, impresoras y más recursos entre nodos de una red de computadores que usen el sistema operativo Windows. Este protocolo se ubica en la capa de aplicación, sustentado por la pila de protocolos de transporte y red TCP/IP. Pese a ser un protocolo orientado a la comunicación entre equipos bajo Windows, es posible la interconexión con equipos Linux mediante SAMBA [33], una implementación de código libre de la parte de servidor del protocolo SMB obtenida por ingeniería inversa.

Tras analizar las limitaciones de disponibilidad y uso, escogemos Acorn OmniClient para la implementación del sistema de compartición de ficheros. En cuanto al protocolo a utilizar, tanto NFS como SMB1 cubren las necesidades básicas requeridas. Sin embargo, el soporte del protocolo SMB en Linux a través de la suite SAMBA es mucho más completo, permitiéndonos configuraciones más avanzadas, por lo que escogemos el protocolo SMB1.

Cabe destacar que SMB1 es una versión obsoleta cuyo soporte se mantiene exclusivamente por razones de retrocompatibilidad, siendo mucho más aconsejable el uso de las versiones SMB2 ó SMB3, que lamentablemente no están soportadas en RISC OS. Sin embargo, las comunicaciones mediante SMB entre los equipos Raspberry Pi y el servidor se producen exclusivamente a través de una subred privada. Además, se dispone de la garantía de que los usuarios han sido validados de forma previa al acceso al servidor, ya que éste sólo resulta accesible desde equipos de la Universidad o mediante la herramienta UnicanLabs. Adicionalmente, se contempla que la información que se va a transferir son principalmente programas y ficheros ejecutables generados por los alumnos, que no constituyen información particularmente sensible.

3.3.2. Implementación de la solución elegida

La implementación se divide en dos partes. La configuración del protocolo SAMBA en el servidor y la configuración del cliente Omni en RISC OS con SMB1. Primero trataremos la configuración del servidor en la que diferenciamos dos grupos de directorios:

- Ficheros de los usuarios en el servidor web.
- Carpetas compartidas de los equipos remotos.

Para cada usuario tendremos un directorio en el servidor sobre el que podrá realizar acciones básicas, como subir ficheros, descargarlos o eliminarlos. Además, para cada equipo remoto tendremos un directorio que será compartido por SMB con el equipo correspondiente.

Esta estructura se podría simplificar conectando el cliente SMB de los equipos remotos con el directorio privado del alumno conectado a dicho equipo en cada momento. Sin embargo, esto implica modificar la configuración del cliente Omni cada vez que un alumno se conecte a la Raspberry Pi, lo cual no es posible debido a que no podemos controlar la Raspberry Pi remotamente desde el servidor. Otra opción consiste en crear una estructura de compartición indirecta de tal forma que podemos dejar una configuración de Omni fija en los equipos remotos que apunten a un directorio y poder tener varios enlaces simbólicos dentro de dicho directorio, lo que permite compartir ficheros de diferentes usuarios en el mismo equipo. El vínculo entre los directorios de los equipos remotos en el servidor y los directorios privados sí que se puede hacer de forma automática, al estar únicamente en el ámbito local del servidor.

Para conectar ambas ubicaciones se emplean enlaces simbólicos, de tal forma que para compartir los ficheros del usuario con los del dispositivo no es preciso transferirlos de un directorio a otro mediante una copia local, ni tenemos problemas de consistencia. Además, nos permite tener los ficheros de múltiples usuarios compartidos en el mismo dispositivo de forma simultánea. De este modo permitimos que alumno y profesor puedan acceder a sus ficheros en un mismo equipo remoto (por ejemplo, para hacer una prueba de verificación de un periférico hardware) sin añadir complejidad a la hora de diferenciar los propietarios de los ficheros.

En los equipos remotos, la configuración de Omni es bastante sencilla, simplemente indicaremos la información en red del servidor y el nombre del recurso compartido. Dicha configuración ha de realizarse manualmente, pero sólo es necesaria la primera vez que se conecta el equipo al sistema de laboratorio remoto, dejando almacenados los datos de conexión en el equipo para futuras conexiones. Además, se ha automatizado el proceso para que en cada arranque el cliente de Omni se conecte por defecto al servidor, liberando al alumno de intervenir en el proceso de conexión. De este modo, cada vez que el alumno accede a un equipo remoto ya dispone por defecto de su directorio de ficheros compartidos en el servidor. El directorio compartido se muestra mediante un icono en la barra inferior de iconos del sistema, análogo al que se tiene para el directorio de trabajo en el puesto físico del laboratorio.

Interfaz del sistema de compartición de ficheros en la herramienta

Como se ha indicado a lo largo de la sección, para poder compartir ficheros con los equipos remotos, estos se alojan en el servidor del sistema. Para disponer de los ficheros en el servidor es necesario un mecanismo para transferirlos entre el servidor y el PC del alumno. El mecanismo implementado es un desarrollo completamente nuevo que utiliza las herramientas ofrecidas por el *framework* Laravel para guardar y descargar ficheros en el servidor. En la implementación del sistema podemos diferenciar tres partes:

- Vista de la interfaz de la herramienta desde la que se interactúa con los ficheros.
- Gestión de transferencia de ficheros entre el PC del alumno y el servidor.
- Scripts para enlazar los directorios de los equipos remotos al directorio del alumno.

En cuanto a la interfaz, se dispone de una única vista en la que tenemos todo lo necesario para gestionar los ficheros, mostrada en la Figura 3.5. En primer lugar se muestra un cuadro en el que se puede subir un fichero de forma individual al directorio del usuario. Después se listan

los ficheros almacenados en el espacio del usuario, dando la opción de eliminar o descargar cada fichero. Además, se da la posibilidad de descargar todos los ficheros en un archivo ZIP. Para generar el fichero ZIP en el servidor se utiliza el comando de Linux 'zip -r', que nos permite comprimir una carpeta de forma recursiva manteniendo la jerarquía de los ficheros.

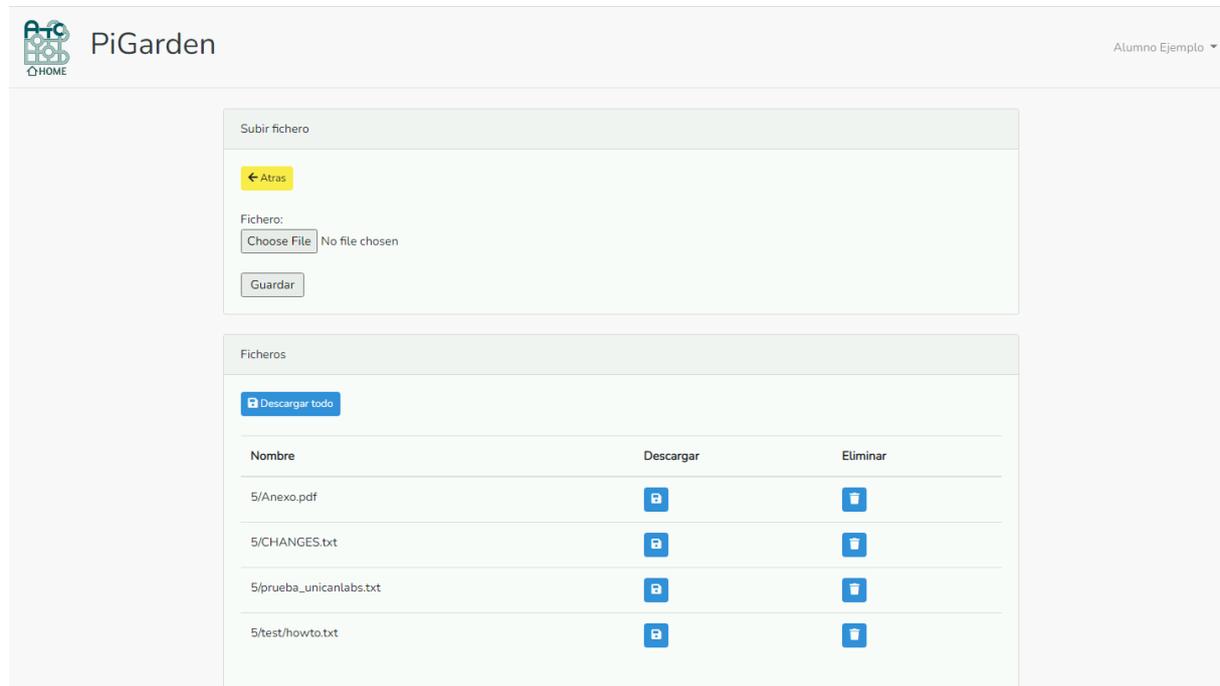


Figura 3.5: Panel de la gestión de ficheros

Respecto a la transferencia de ficheros entre el PC del alumno y el servidor, se realiza mediante un módulo que proporciona Laravel por defecto para la gestión de ficheros en los directorios de almacenamiento del servidor, denominado *Storage*. Se emplea dicho módulo tanto para la carga como la descarga de ficheros desde el servidor. Cada vez que se recibe un fichero a través de la interfaz se emplea este módulo, para almacenar el fichero en la carpeta asignada al alumno en el servidor.

Como se ha indicado al comienzo de la sección, los equipos Raspberry Pi remotos están configurados para conectarse a un directorio compartido en red, alojado en el servidor. Dentro de este directorio se crea un enlace simbólico a la carpeta del alumno que esté conectado en ese momento al equipo remoto. Para enlazar de forma automática al directorio del alumno cuando éste se conecta, se emplea un script en el servidor que crea dicho enlace simbólico mediante el comando de Linux 'ln -s'. En el momento en el que el alumno deja de utilizar el equipo remoto, el enlace simbólico se rompe a través de otro script que usa el comando de Linux 'unlink'.

La interfaz para la gestión de ficheros en el servidor es independiente de la vista del equipo remoto. Para acceder al apartado de gestión de ficheros existe una vista auxiliar accesible desde el menú de la herramienta, en la esquina superior derecha. De esta forma el alumno puede interactuar con los ficheros sin la necesidad de entrar en un dispositivo. Del mismo modo, el alumno puede acceder a sus ficheros (los programas que ha desarrollado) incluso si todos los equipos remotos están en uso.

También se da al profesor la posibilidad de poder compartir o no su propio directorio de ficheros

correspondiente. Como se indicó al comienzo de la Sección 3.3, el profesor puede necesitar de forma puntual acceder a sus propios ficheros, principalmente para verificar el correcto funcionamiento de un dispositivo periférico o para enseñar un código correcto para resolver una duda del alumno. Sin embargo, dichos ficheros pueden contener información sensible como para estar accesibles de forma automática cuando un profesor accede al equipo remoto empleado por un alumno. Con la solución empleada, el profesor puede interactuar con los equipos cuando haya un alumno conectado sin tener que compartir sus ficheros, y decidir cuándo hacerlo desde la propia consola. Para ello, se ha implementado un botón adicional en la vista del equipo remoto, disponible exclusivamente para usuarios de tipo profesor. Dicho botón permite comenzar y detener la compartición de sus archivos con el equipo remoto visualizado.

Por último, se ha garantizado la privacidad de los ficheros de los alumnos, evitando que los alumnos puedan acceder a directorios que no les pertenecen. Por defecto, el servidor restringe el acceso de la vista de ficheros únicamente al usuario conectado. Además, desde los equipos remotos únicamente se puede acceder al directorio del alumno que tiene la reserva actual. Sin embargo, un alumno malintencionado podría intentar acceder a los programas de un alumno que esté conectado a otro equipo remoto, manipulando la configuración de Omni para que se conecte al directorio de otro equipo. Para impedir esta posibilidad, se ha limitado desde la configuración de SAMBA, el acceso a cada recurso compartido a la dirección IP del equipo correspondiente. Por tanto, cada equipo remoto únicamente puede conectarse al directorio reservado a ese equipo.

3.4. Módulo de visualización

En este capítulo se estudian las limitaciones del módulo de visualización en el prototipo inicial del sistema. El principal problema observado es una gran vulnerabilidad en la latencia de la señal de vídeo que muestra la salida de los periféricos, con valores observados entre 2 y 15 segundos. Esta latencia, además de tener una gran desviación, es en ocasiones lo suficientemente grande como para afectar a la verificación de los códigos de los alumnos: cuando el cambio en la salida se muestra decenas de segundos después de producirse, el alumno puede pensar que su código no ha funcionado como debería.

El sistema original emplea una cámara web Trust Exis Webcam. Se trata de un modelo de muy bajo coste y prácticamente obsoleto, que ofrece una resolución de imagen de 640x480 píxeles. Para capturar la señal de vídeo de la cámara se hace uso del software FFmpeg [7]. Mediante este software también se realiza la emisión en directo del vídeo capturado, utilizando para ello el protocolo HLS (*HTTP Live Streaming*, transmisión en vivo en HTTP) que nos permite embeber la señal de vídeo dentro de la página web. Este protocolo guarda fragmentos de vídeo de escasos segundos y los va mostrando en la página web de forma sucesiva. Podemos, por tanto, afirmar que la versión original el prototipo no hace realmente una emisión en directo, si no la reproducción consecutiva de fragmentos de vídeo.

Todo esto nos da como resultado una emisión con una latencia mínima de dos segundos. No obstante, en ocasiones dicha latencia varía, pudiendo llegar hasta los quince segundos. Esto es inviable para el desarrollo de una práctica que precise de visualizar la salida de los periféricos. A continuación, se analizan posibles soluciones a estos problemas. Posteriormente se detalla la implementación de la solución elegida.

3.4.1. Estudio previo

El protocolo utilizado para mostrar la señal de vídeo en el prototipo inicial es HLS. Este protocolo simplifica la implementación de una emisión de vídeo a través de la generación de pequeños fragmentos independientes, pero introduce una latencia alta con gran variabilidad. Además, la grabación continua de fragmentos de vídeo y su posterior procesamiento supone una carga computacional añadida que no tenemos si retransmitimos directamente el vídeo capturado.

Uno de los protocolos más utilizados para la retransmisión de vídeo es RTSP (*Real Time Streaming Protocol*, protocolo de transmisión en tiempo real). Este protocolo nos ofrece latencias bajas y funciona con una transmisión directa, sin almacenar ningún fragmento de vídeo. Sin embargo, RTSP no está soportado en HTML5, que es el lenguaje que se emplea para el desarrollo de las vistas del proyecto. Esta falta de soporte implica la necesidad de utilizar un reproductor de vídeo embebido, ya sea software propietario (como QuickTime Player [15]) o software libre (como WebRTC [22])

El uso de software propietario se descarta dado que uno de los objetivos del proyecto es que sea completamente libre. El objetivo es que la herramienta pueda utilizarse en otros centros sin ningún tipo de restricción legal o comercial, como se comentó en el Capítulo 1. Por lo tanto, debido al uso de licencias, todo el *framework* QuickTime, incluyendo el QuickTime Player, está completamente descartado para su implementación en el sistema. WebRTC es un proyecto libre y de código abierto que a través de una API proporciona a los navegadores web capacidades de comunicación en tiempo real. Sin embargo, su implementación supone crear una infraestructura de red compleja, que por carga de trabajo y tiempo no es posible llevar a cabo en el proyecto.

Por último, podemos retransmitir en directo la señal de un dispositivo de captura de vídeo mediante el reproductor y *framework* VLC [34]. Es un programa completamente libre y de código abierto, por lo que es adecuado para su uso en el proyecto. VLC nos permite convertir un dispositivo de captura de vídeo en una retransmisión encapsulada y codificada para embeber en un navegador web a través de HTTP. Esto nos permite utilizar las etiquetas de vídeo que proporciona HTML de forma nativa, y prescindir de un reproductor de vídeo embebido. VLC tiene una interfaz fácil de utilizar y también dispone de comandos para el manejo desde consola lo que nos permite automatizar la retransmisión.

Tras analizar las soluciones disponibles, se escoge VLC para la retransmisión en directo de la señal de vídeo de los periféricos desde el panel del equipo remoto.

3.4.2. Implementación de la solución elegida

Para la implementación de VLC simplemente tenemos que configurar unos pocos parámetros: la URL a través de la que VLC va a transmitir la señal de vídeo, el códec (formato de compresión) empleado, y el protocolo de encapsulamiento.

El encapsulamiento utilizado es OGG, desarrollado por la Fundación Xiph.Org [36]. Este encapsulamiento es libre y abierto, sin restricciones de patentes de software. OGG es compatible con las versiones actuales del navegador Firefox y algunas versiones de Google Chrome. Se ha elegido este encapsulamiento ya que funciona junto al códec Theora desarrollado por la misma organización. Este encapsulamiento nos proporciona un bajo consumo de CPU y una reducción de la latencia con respecto a otros encapsulamientos, ya que está basado en la compresión con pérdidas.

Tras implementar el cambio del módulo de visualización se ha observado un ligero aumento en la calidad de imagen. El cambio más significativo es la reducción y estabilización de la latencia de la transmisión, durante las pruebas realizadas se ha obtenido una latencia mínima de un segundo y máxima de dos segundos. Esto supone una diferencia notable con respecto a la implementación original, que en ocasiones llegaba a 15 segundos de latencia. Durante las pruebas del módulo, se ha observado que el consumo de red asociado es muy reducido, de aproximadamente 120KB/s. Como referencia, la conexión del servidor que se emplea con el resto de la red de la Universidad tiene un ancho de banda de 1 Gbps, 4 órdenes de magnitud mayor que el ancho de banda del vídeo emitido, por lo que no supondrá ningún problema de uso de red.

3.5. Gestión del sistema

La gestión de usuarios en el prototipo original es bastante limitada, obligando al profesor a añadir y eliminar usuarios uno por uno. Este procedimiento no es escalable para su uso en una plataforma con cientos de alumnos por curso. En esta sección se detallan las funcionalidades de gestión de usuarios y equipos que se han incorporado sobre la versión original del sistema con el objetivo de mejorar la experiencia de uso tanto por parte del profesorado como del alumnado.

El sistema original ofrece las operaciones básicas para su administración. No obstante, tiene un número limitado de funciones de automatización de tareas y de una gestión de roles casi inexistente. Dado que los roles en el prototipo original carecen de permisos, únicamente hay una diferenciación entre alumno y profesor. Los usuarios de tipo profesor pueden administrar la herramienta por completo y tienen libre acceso a los equipos remotos, mientras que el alumno únicamente puede acceder a equipos remotos que estén libres o que haya reservado. En esta sección se explican los cuatro bloques de funcionalidad añadidos para garantizar que el sistema cumpla los requisitos para su uso por parte del alumno: gestión en bloque de los usuarios, ampliación de los roles de usuario e incorporación de permisos por rol, desarrollo de un sistema automatizado de notificación de incidencias, y generación de estadísticas de uso.

3.5.1. Gestión de usuarios

La primera de estas funcionalidades está destinada a la gestión y supervisión de los usuarios del sistema. En el prototipo inicial del sistema no existe modalidad para añadir varios alumnos en bloque, debiendo el profesor de hacerlo manualmente para cada uno de los alumnos. En una asignatura como Introducción a los Computadores, que tiene aproximadamente cien alumnos, este método de introducción de usuarios no es viable.

Para corregir esta deficiencia, se ha desarrollado un sistema para poder importar y exportar los alumnos desde una hoja de cálculo. De esta forma, el profesor puede crear una hoja de cálculo con los datos de todos los alumnos (o de aquellos que quiera añadir) y generar para cada uno una contraseña aleatoria con las herramientas de los gestores de hojas de cálculo. Esta modalidad es particularmente útil porque los profesores disponen de opciones de exportación en hoja de cálculo tanto en Moodle como en el Campus Virtual de la Universidad, no siendo necesario introducir los datos de los alumnos de forma manual. Para importar los usuarios en la herramienta, el profesor debe dirigirse al panel de administración de alumnos y elegir la opción de importar alumnos mostrando la vista mostrada en la Figura 3.6. La herramienta solicita la carga del fichero y añade automáticamente los alumnos si la hoja de cálculo cumple con los campos requeridos (correo de usuario, nombre de usuario, contraseña). También se podrá indicar de forma opcional el rol del usuario; como este método se va a implementar mayoritariamente para añadir alumnos, la herramienta asigna por defecto el rol de alumno a los nuevos usuarios, pero

se deja la opción de asignar roles distintos en la importación.

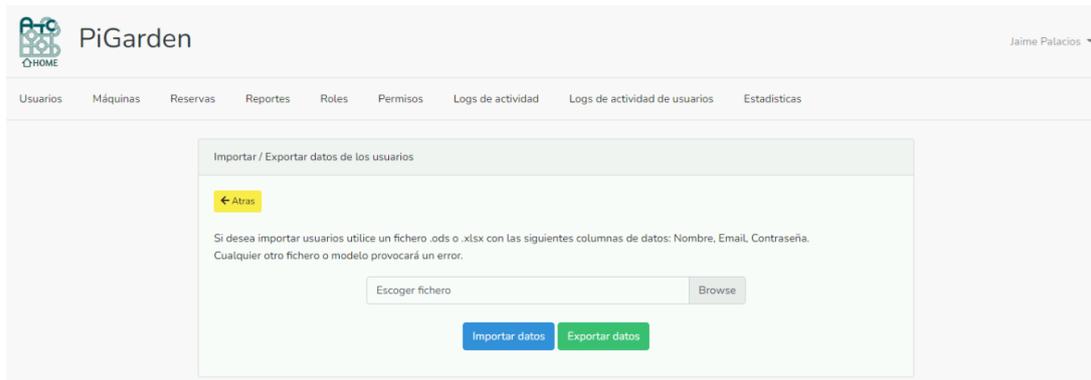


Figura 3.6: Panel de la exportación e importación de usuarios

Para la importación y exportación de hojas de cálculo desde el framework de Laravel se ha utilizado el paquete maatwebsite/Excel [31]. Este paquete nos proporciona los métodos necesarios para poder crear hojas de cálculo con datos extraídos de la base de datos y para leer los datos de una hoja de cálculo para su posterior inserción en la base de datos.

Otra de las funciones no disponibles en el sistema original es que un usuario pueda modificar su contraseña sin intervención del administrador. Dado que la contraseña inicial de los usuarios es generada de forma aleatoria o establecida por el profesor, es conveniente que el alumno pueda sustituirla por otra de su elección. Como parte de los cambios introducidos, se ha implementado esta función completamente nueva con las herramientas ya disponibles del lenguaje PHP. Mediante un nuevo panel, mostrado en la Figura 3.7, se requiere al usuario que indique su antigua contraseña e introduzca la nueva dos veces (para evitar fallos de escritura). Luego el servidor simplemente comprueba que la contraseña actual es igual a la introducida, y almacena la nueva contraseña (codificada con un *hash* para garantizar la confidencialidad) en la base de datos.

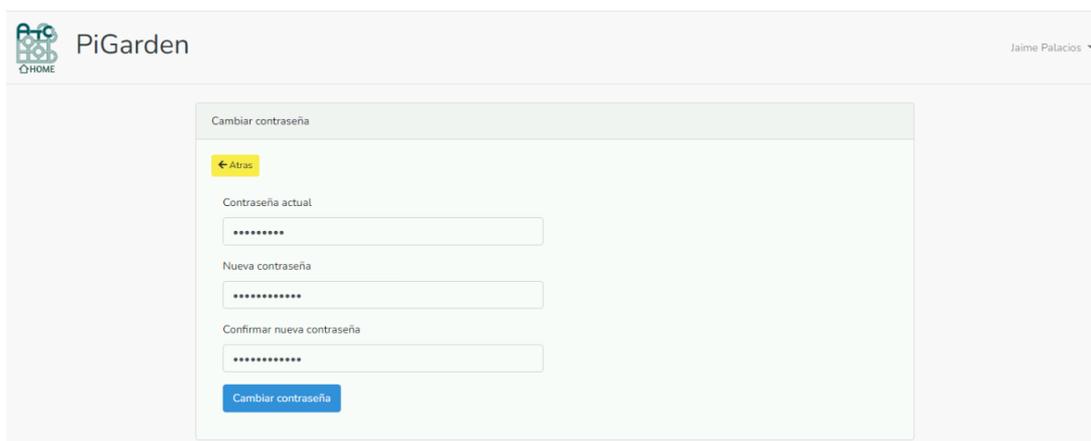


Figura 3.7: Panel para cambiar la contraseña

Por último, en la versión original del sistema no existe ninguna forma de controlar la actividad de los usuarios, ya sea por usos ilícitos de la herramienta o por interés del profesor. Para mitigar esta carencia se ha añadido una entidad más al panel de administración, mostrada en la Figura 3.8. Ésta consiste en un registro de las URL de la herramienta a las que ha accedido cada alumno,

junto a la dirección IP y la fecha de acceso. De esta forma, el profesor es capaz de conocer el uso exacto de la herramienta, pudiendo solventar conflictos en caso de que los halla o conocer la actividad de los alumnos. La implementación se ha realizado añadiendo dicha entidad a la base de datos con sus respectivos campos y tratándolo como el resto de las entidades administrables.

IP	URL	Usuario	Fecha	Acciones
193.144.197.163	http://gato.atc.unican.es:8080		23-05-22 16:51:15	[icon] [icon]
193.144.197.163	http://gato.atc.unican.es:8080/machine/delete-reserve/274		23-05-22 16:51:15	[icon] [icon]
193.144.197.163	http://gato.atc.unican.es:8080		23-05-22 16:51:10	[icon] [icon]
193.144.197.163	http://gato.atc.unican.es:8080/machine/console/24		23-05-22 16:50:09	[icon] [icon]
193.144.197.163	http://gato.atc.unican.es:8080		23-05-22 16:49:00	[icon] [icon]
193.144.197.163	http://gato.atc.unican.es:8080/machine/delete-reserve/273		23-05-22 16:49:00	[icon] [icon]
193.144.197.163	http://gato.atc.unican.es:8080		23-05-22 16:48:57	[icon] [icon]
193.144.197.163	http://gato.atc.unican.es:8080/machine/console/29/lapagar		23-05-22 16:47:59	[icon] [icon]

Figura 3.8: Panel de actividad de usuarios

3.5.2. Gestión de roles y permisos

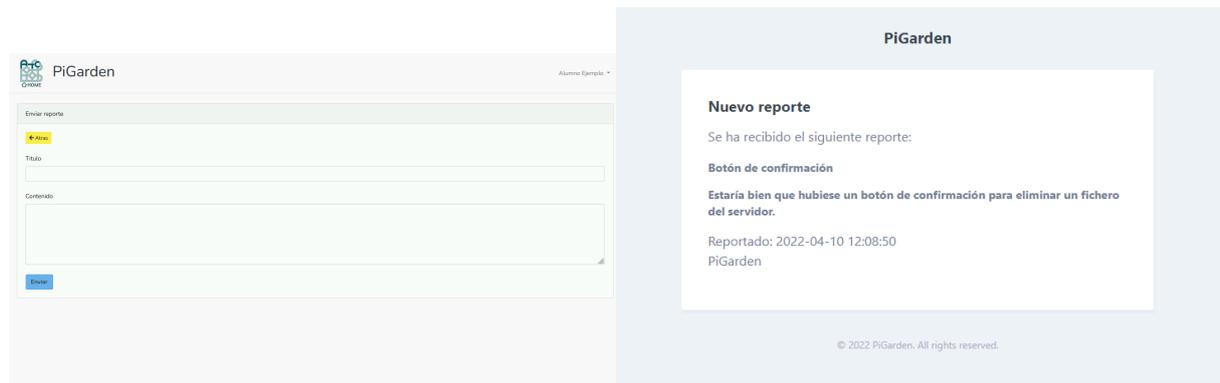
El sistema original dispone de dos roles, alumno y profesor. También tiene la capacidad de gestionar permisos, pero no contiene ningún permiso definido. Todas las distinciones se hacen en base al rol al que pertenece el usuario y no a sus permisos, ya que son inexistentes. Sin embargo, en las asignaturas involucradas existe profesorado asociado, que no es previsible que realice tareas de administración del sistema. Por lo tanto, se ha considerado que el rol de profesor no debe tener todos los permisos para administrar la herramienta, si no que dicho tarea debería pertenecer a un usuario de tipo administrador.

Se ha llevado a cabo la creación de tantos permisos como entes administrables hay en el sistema (maquinas, usuarios, reservas, actividad. . .). De esta forma podemos privar a los roles actuales y futuros de poder administrar ciertos aspectos. Se ha creado un rol administrador, el cual dispone de todos los permisos de gestión de la herramienta. Esto nos permite poder reducir los permisos del rol de profesor a los necesarios para administrar un laboratorio; por ejemplo, un profesor no puede incorporar o retirar equipos remotos, estando esta tarea reservada a los administradores.

3.5.3. Notificación de incidencias

El sistema se encuentra actualmente en fase de pruebas, desplegado con un número limitado de equipos remotos para su uso por los alumnos de la asignatura de Introducción a los Computadores en el presente curso 21-22. Dado que el sistema no ha alcanzado todavía un nivel estable, es posible que los usuarios experimenten ciertos problemas, ya sean funcionales o de usabilidad. Para poder obtener retroalimentación de los alumnos sobre el funcionamiento de la herramienta sin la necesidad de que el profesorado intervenga se ha diseñado un sistema de notificación de incidencias. Este sistema consiste en una vista adicional, mostrada en la Figura 3.9, en la que el alumno puede describir su incidencia, que se envía por correo electrónico al administrador del sistema, mostrado en la Figura 3.9. Para su implementación se han utilizado las herramientas que proporciona Laravel para el envío de correos electrónicos empleando el protocolo SMTP

(Simple Mail Transfer Protocol, protocolo de transferencia simple de correo).



(a) Panel para redactar incidencia

(b) Vista de un correo de incidencia

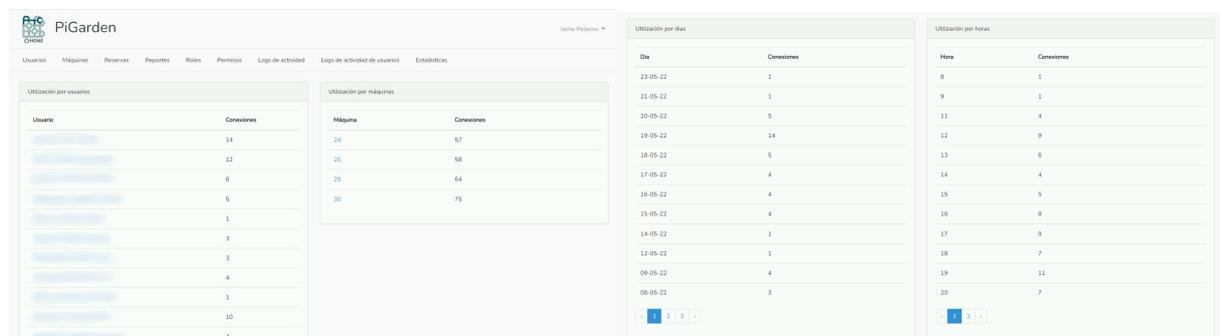
Figura 3.9: Vistas sobre la notificación de incidencias

De esta forma podemos obtener tanto problemas descubiertos por los alumnos como propuestas de mejora de usabilidad. La importancia de esta funcionalidad radica en que los alumnos, al trabajar con la herramienta como usuarios finales, son capaces de descubrir problemas que pasarían desapercibidos para el administrador y que pueden notificar de forma directa al responsable del sistema.

3.5.4. Estadísticas de uso

Por último, se ha considerado oportuno proporcionar al profesor algún tipo de retroalimentación sobre el uso de la herramienta por parte de los alumnos. En la Sección 3.5.1 se ha indicado la vista añadida para el seguimiento de la actividad de los usuarios de la herramienta. Sin embargo, esta información es muy pormenorizada y puede resultar compleja de analizar para sacar conclusiones generales sobre el uso de la herramienta.

Se ha implementado una nueva vista en el panel de administración, mostrada en la Figura 3.10 (se han censurado los datos de usuario por razones de privacidad), en la que se dan cuatro tipos de estadísticas: número de conexiones por cada alumno, máquina, día y hora. Esta vista se ha desarrollado completamente desde cero, utilizando las funciones ya disponibles de Laravel y PHP. El profesor puede ver estas estadísticas y conocer la utilización de la herramienta de una forma rápida y clara, así como establecer el volumen de trabajo realizado por un alumno.



(a) Parte superior del panel de estadísticas

(b) Parte inferior del panel de estadísticas

Figura 3.10: Panel de estadísticas

4. EVALUACIÓN DEL SISTEMA

Este capítulo realiza un breve recorrido de los cambios efectuados sobre el prototipo original, mostrando los resultados obtenidos tras el uso del mismo por parte de los alumnos, tanto en forma de estadísticas de uso como a través de encuestas de satisfacción.

Teniendo en cuenta las limitaciones observadas en el prototipo original, tanto en su parte hardware (Sección 2.1) como software (Sección 3.1), se ha alcanzado un prototipo final verificando que el sistema cumple con los requisitos marcados al comienzo del trabajo. Este prototipo puede ser desplegado sobre un laboratorio de arquitectura para el uso de Raspberry Pi de forma remota por parte de los alumnos.

La versión actual del prototipo, mostrada en la Figura 4.1, que se ha utilizado para la evaluación del sistema, está formada por cuatro equipos remotos Raspberry Pi. La alimentación de estos dispositivos está controlada por relés gestionados por un registro de desplazamiento. Los terminales de la GPIO están controlados por otro registro de desplazamiento, sumando un total de dos registros implementados en el prototipo.



Figura 4.1: Situación actual del prototipo

Para evaluar el correcto funcionamiento de los registros de desplazamiento y las librerías que se utilizan se ha diseñado un programa simple que actúa sobre unos LED conectados a las salidas de los registros mediante una placa de prototipado. El programa enciende los LED en sentido de izquierda a derecha, después los apaga y finalmente los enciende en el sentido contrario. El código completo del programa de pruebas puede observarse en el Anexo B. Tras completar el funcionamiento esperado en las salidas de los registros, se implantó su uso sobre el prototipo inicial del sistema. Por último, se ha evaluado el control de la alimentación de los equipos remotos desde la herramienta web tras el cambio del módulo de control. De esta forma nos aseguramos de que la integración del nuevo módulo en el sistema es correcta, dado que el uso de cableado descrito en la Sección 2.2.1.3 puede conducir a errores de conexionado.

Posteriormente, se han evaluado las funciones añadidas a la herramienta web: el sistema de reservas automatizado, la compartición de ficheros entre equipos finales, y la ampliación de las funciones de gestión de usuarios. Dichas funciones han sido verificadas durante su desarrollo, y después de forma externa por parte de profesores y alumnos. Pese a que durante el despliegue de las actualizaciones de la herramienta se comprobó su correcto funcionamiento, durante su uso

por parte de profesores y alumnos se han detectado pequeños fallos. La observación de fallos por parte de alumnos ha ayudado a perfeccionar la herramienta y ha validado el enfoque de pruebas con una versión final del prototipo como paso previo a la implantación del sistema final, que se prevé para el próximo curso 22-23.

Para evaluar el módulo de visualización, se ha comprobado que la calidad de imagen ofrecida es suficiente para verificar la salida de periféricos hardware, y que la latencia de vídeo entre captura y visionado por parte del alumno es estable y reducida. Los valores observados de latencia (entre 1 y 2 segundos) garantizan los requisitos de uso durante la realización de las prácticas.

4.1. Análisis de uso de la herramienta

La herramienta ha estado disponible para su uso por parte de los alumnos de la asignatura de Introducción a los Computadores durante el presente curso. En esta sección se analiza la carga de uso que ha recibido la herramienta a lo largo de dos meses de utilización por parte de los alumnos, desde el despliegue actual de la herramienta.

A través de las estadísticas que se generan, en la vista descrita en la Sección 3.5.4, hemos obtenido un resumen de la utilización de la herramienta durante dos meses (desde el 25 de marzo hasta el 25 de mayo).

En cuanto al número de conexiones por día, sin tener en cuenta el periodo vacacional de Semana Santa, ha habido un promedio de 1.5 conexiones por día. Sin embargo, hay un aumento notable del uso al final del mes de mayo. Esto puede ser debido a que las últimas prácticas requieren de un esfuerzo mayor y por lo tanto es más probable necesitar completar las prácticas fuera del horario de laboratorio.

La herramienta ha sido utilizada por 13 alumnos diferentes, de un total de 94 alumnos que disponían de acceso en la asignatura de Introducción a los Computadores. El despliegue de la herramienta con los cambios propuestos sobre el prototipo original se ha completado casi dos meses después de comenzar el cuatrimestre. Por lo tanto, es posible que parte del alumnado disponga de Raspberry Pi propia, bien por haberla adquirido al comienzo de la asignatura, o bien porque ya disponía de ella. Pese al problema de disponibilidad por la crisis de semiconductores que se discutió en la Sección 1.1 los alumnos han tenido la oportunidad de adquirir los equipos de segunda mano de otros alumnos de cursos superiores, aumentando ligeramente la disponibilidad. Sin embargo, dado que no todos los alumnos deciden vender sus equipos cuando superan las asignaturas involucradas, es de esperar que la falta de equipos se vuelva más acusada a corto plazo.

Asimismo, la reducción de restricciones de aforo ha facilitado el acceso físico de los alumnos a los equipos del laboratorio, disminuyendo la necesidad de realizar trabajo autónomo de forma remota. Teniendo en cuenta estos factores, se considera que el empleo por un 14% de los alumnos es un resultado satisfactorio, esperando que su uso aumente en el próximo curso por estar disponible desde el comienzo de las asignaturas.

4.2. Encuestas de estimación de uso y satisfacción

Con el objetivo de tener un conocimiento mayor y más cercano sobre la utilización de la herramienta por parte de los usuarios finales se han realizado dos encuestas a lo largo del desarrollo del proyecto. La primera encuesta trata sobre una estimación del uso que se va a dar a la

herramienta para poder ajustar los parámetros del sistema de gestión de reservas. Esta encuesta fue proporcionada a los alumnos de la asignatura de Introducción a los Computadores antes de completar el despliegue de la herramienta. La segunda encuesta se plantea con el fin de conocer la satisfacción y averiguar posibles mejoras o limitaciones que se hayan dado durante el uso del prototipo. Dicha encuesta se ha proporcionado a los alumnos al final del curso académico, dejando un amplio margen de tiempo para tener una muestra significativa de alumnos que hayan utilizado la herramienta y tengan una opinión formada sobre su experiencia.

La primera encuesta se realizó para conocer qué posible uso de la herramienta podía darse por parte de los alumnos y de esa forma poder ajustar los parámetros de reservas. En esta encuesta participaron 10 alumnos que, teniendo en cuenta el porcentaje de uso final de la herramienta, son datos coherentes. La primera pregunta consistía en si los alumnos disponían de una Raspberry Pi propia, en la que el 30 % negaban que dispusiesen de un equipo. Dada la crisis de semiconductores descrita en la Sección 1.1, es probable que esta cifra aumente al 50 % para alumnos de primer curso en el próximo año académico. La segunda pregunta tenía como objetivo ver si los alumnos conocían el servicio UnicanLabs. Esta pregunta es particularmente relevante, ya que los alumnos han de emplear obligatoriamente UnicanLabs para acceder a la herramienta desde equipos de fuera de la Universidad de Cantabria. El 80 % de los encuestados afirmaron que había utilizado UnicanLabs, dato positivo que da a entender que el paso intermedio por el servicio no es un obstáculo para el acceso a la herramienta. La tercera pregunta está dedicada al sistema de reservas desarrollado en la Sección 3.2, concretamente para conocer el tiempo de uso que se iba a dedicar a cada reserva. El 70 % respondieron que el tiempo que iban a dedicar a las prácticas en casa iba a ser de una hora al día, y otro 20 % indicaron que iban a emplear menos de una hora diaria, por lo que la elección de bloques de reserva de una hora puede considerarse acertada. La cuarta y última pregunta también está relacionada con el sistema de reservas, concretamente con la antelación de las reservas. El 60 % respondieron que las realizarían un día antes, y el 30 % que haría la reserva una semana antes. A la luz de los resultados de la encuesta, puede valorarse reducir el límite actual de antelación de reserva de una semana a dos o tres días.

La segunda encuesta, con el fin de conocer la experiencia de uso de los alumnos, se llevó a cabo dos meses después de proporcionar el acceso a los alumnos. En esta encuesta participaron 7 alumnos. Los resultados recogidos nos indican que el número de Raspberry Pi que hay instaladas es suficiente para el uso actual del sistema, ya que los alumnos siempre han visto entre 2 y 3 equipos libres de un total de 4 cada vez que se conectaban. Esta disponibilidad explica que el 67 % no haya utilizado el sistema de reservas, ya que siempre tenían un equipo disponible al conectarse. En cuanto al sistema de ficheros, el 100 % lo valoran como bueno o muy bueno, lo cual indica que su desarrollo es suficiente para cumplir su función. Por último, en cuanto al uso de UnicanLabs, el 80 % afirman que el acceso a la herramienta a través del servicio no ha tenido ningún tipo de impacto sobre la experiencia de uso. Por tanto, a medio plazo se puede mantener la mecánica de acceso actual sin perjudicar a los alumnos. El objetivo final sería poder prescindir del uso de UnicanLabs para simplificar el acceso a la herramienta, empleando una dirección pública. Esta actuación se plantea como línea de trabajo futuro, y se discute en la Sección 5.1.

Por parte del profesorado responsable de la asignatura sobre la que se ha desplegado la herramienta ha habido una satisfacción general con el uso del sistema. Por ello, se pretende continuar con su uso en el próximo curso académico en todas las asignaturas relacionadas, permitiendo acceder desde el comienzo de la asignatura. Con esto se espera que el número de alumnos que empleen la herramienta sea mucho mayor.

5. CONCLUSIÓN

El trabajo tiene como objetivo la escalabilidad tanto hardware como software de un prototipo de laboratorio virtual para el manejo remoto de equipos Raspberry Pi bajo el sistema RISC OS. Asimismo, el trabajo comprende la introducción de funcionalidades que permitan un manejo más fluido y cómodo de la herramienta para los usuarios. Esta configuración de puesto es la utilizada habitualmente en las asignaturas del área de Introducción a los Computadores y Estructura de Computadores. A causa de la pandemia, se han visto reducidos los aforos de las aulas y espacios universitarios, restringiendo a los alumnos el acceso a los equipos del laboratorio. Esto, junto a la crisis de semiconductores que da lugar al desabastecimiento de equipos electrónicos, perjudica el trabajo autónomo de los alumnos y motiva el desarrollo de una herramienta que permita el uso remoto de equipos Raspberry Pi con RISC OS. En un trabajo previo se ha implantado un prototipo preliminar, confirmando la viabilidad del sistema, pero dicho prototipo presenta serias limitaciones de escalabilidad y uso.

Los objetivos principales de este trabajo consisten en evaluar las limitaciones del sistema, escoger e implementar la solución más apropiada para corregir las limitaciones identificadas, facilitar el uso de la herramienta por parte de los alumnos y profesores y verificar la viabilidad de la herramienta en una prueba real con alumnos.

Tomando como referencia el formato de trabajo de las prácticas de las asignaturas involucradas, se han identificado una serie de requisitos del sistema. En primer lugar, se debe poder aumentar el número de equipos Raspberry Pi conectados al sistema respecto al prototipo inicial, facilitando la escalabilidad del mismo para usos en entornos de grandes dimensiones. Asimismo, el sistema debe permitir al alumno el acceso a los equipos remotos siempre y cuando estos estén libres, y proporcionar la capacidad de programar una reserva para asegurarse la utilización del equipo en un futuro. También debe permitir al alumno transferir al equipo remoto los ficheros desarrollados durante la realización de las prácticas en el laboratorio. De esta forma, facilitamos el objetivo de la herramienta como complemento a las sesiones de laboratorio. Por último, es preciso mejorar las capacidades de gestión y supervisión del sistema para grandes cantidades de alumnos.

Para la escalabilidad hardware del sistema se han propuesto diversas mejoras como la integración del módulo de control en una placa PCB o la adquisición de más fuentes de alimentación, switches y cámaras para permitir el escalado de equipos. Estas mejoras no se han llevado a cabo por limitaciones temporales, pero se proponen como línea de trabajo futuro para el próximo curso académico. Para la implementación final se ha rediseñado el módulo de control de la herramienta, escogiendo una solución basada en el uso de registros de desplazamiento. Dichos registros nos permiten obtener múltiples bloques de 8 salidas independientes, a partir de una entrada en serie de múltiplos de 8 bits. De esta forma reducimos el número de conexiones necesarias para gestionar un número mayor de salidas. Estas salidas se pueden utilizar tanto para actuar sobre los relés que conmutan la alimentación de los equipos remotos como conectadas a los terminales GPIO de los equipos Raspberry Pi, replicando entradas de propósito general. Para la actuación sobre los registros de desplazamiento se ha conservado el uso de la placa Arduino UNO con un módulo de expansión de conectividad de red que había en el prototipo original.

En cuanto a los cambios sobre la parte software del sistema, se ha desarrollado un sistema de reservas automáticas que permite independizar al profesor de la tarea de permitir el acceso a

los alumnos a los equipos remotos. Este sistema consta de reservas de una hora de duración con un límite de una reserva por día y equipo. Además, se permite el uso libre de aquellos equipos remotos que se encuentren libres, sin la necesidad de que el alumno realice ninguna reserva.

Por otro lado, se ha implementado un mecanismo para compartir ficheros desde el equipo local del alumno con los equipos remotos a través de la herramienta. Esta funcionalidad permite hacer uso de los equipos remotos como complemento del trabajo realizado presencialmente en el laboratorio. Para lograr la compartición extremo a extremo, se han realizado dos pasos de compartición, uno entre el equipo remoto y el servidor y otro entre el servidor y el equipo del alumno. Para la comparación de directorios en red entre Raspberry Pi y servidor se utiliza el protocolo SMB1. Para el intercambio de archivos entre el servidor y el equipo del alumno se proporciona a cada alumno un directorio privado en el servidor, en el que puede subir, descargar y eliminar ficheros.

Por último, se han añadido diversas funcionalidades a la herramienta para dar un uso más fluido y cómodo a los usuarios. Entre ellos, cabe destacar la posibilidad de que los alumnos puedan modificar su contraseña de acceso al sistema sin intervención de un profesor o administrador. Los profesores y administradores pueden exportar e importar alumnos desde hojas de cálculo, ver la actividad detallada de los usuarios y tener un resumen de la utilización de la herramienta por parte de los alumnos. También se ha ampliado la información en la vista del listado de equipos remotos, para que el alumno conozca su estado detalladamente. Además, se han mejorado algunos paneles de administración como la configuración de los datos de los equipos remotos, en los que ahora se puede indicar un nombre para los periféricos de entrada conectados, que luego se verá reflejado en el panel del dispositivo.

Los cambios efectuados sobre el prototipo original han permitido la puesta en marcha del sistema para el uso del alumnado. Tras varios meses de uso por alumnos de la asignatura de Introducción a los Computadores, se ha verificado su utilidad como complemento para el trabajo autónomo de los alumnos. Las estadísticas de uso y los resultados de encuestas de satisfacción a los alumnos demuestran la aceptación del sistema y su adecuación a las expectativas.

En el terreno personal, la realización de este trabajo me ha permitido aplicar a un caso práctico los conocimientos adquiridos a lo largo del Grado, como la gestión de circuitos integrados y su programación, la organización de la parte software del proyecto, el uso de protocolos de compartición de ficheros en red o la utilización del sistema de composición de textos Latex. También me ha permitido explorar nuevos temas de aprendizaje, particularmente en lo relativo a protocolos de transmisión de vídeo por la red, programación de entornos web, integración de circuitos y administración del sistema, incluyendo el contacto con los usuarios finales.

5.1. Trabajo futuro

Durante la realización de este trabajo se han valorado diversas mejoras y ampliaciones que no se han podido implementar debido al tiempo y complejidad requeridos. Estas funciones y cambios se contemplan de cara a su desarrollo en trabajos futuros.

El primer objetivo planteado, con un mayor alcance de trabajo, consiste en el despliegue del sistema de forma estable. El sistema definitivo constará del conexionado de ocho equipos remotos Raspberry Pi. Será necesario migrar el sistema a un nuevo equipo que tenga mayores prestaciones hardware para un uso más estable. A diferencia del prototipo actual, ubicado en una sala

de acceso restringido a miembros del grupo de Arquitectura y Tecnología de Computadores, se pretende ubicar el nuevo sistema en el laboratorio donde se realizan las sesiones prácticas de las asignaturas. Para ello, es necesario estructurar los componentes del sistema (PC servidor, fuente de alimentación, lógica de control, red de interconexión, equipos remotos) en una composición de tipo *rack* o similar, para facilitar su desplazamiento y restringir el acceso. Como consecuencia del escalado de dispositivos, para el módulo de visualización, se tendrá que implementar algunas de las soluciones propuestas o evaluar nuevas con el objetivo de poder capturar todos los equipos.

El segundo objetivo de trabajo futuro propuesto consiste en la integración de los módulos del sistema en una PCB. Se ha diseñado un prototipo de dicha PCB, descrito en la Sección 2.2.4. Se propone un circuito formado por ocho relés y puertos USB que nos permite conmutar la alimentación de los equipos Raspberry Pi y tres registros de desplazamiento, donde uno se utiliza para el control de los relés y los dos restantes para gestionar la entrada de propósito general de los equipos remotos, consiguiendo así dos entradas por equipo. El objetivo consiste en la fabricación de la PCB, su testado e integración con el resto del sistema para su uso en la herramienta. Esta placa se propone para su uso en el despliegue del sistema comentado en el objetivo anterior.

Otro objetivo trata sobre el aspecto comentado en la Sección 3.2, sobre la ampliación del sistema de reservas para la compatibilidad con grupos. Se propone modificar el sistema de reservas individuales para permitir el uso compartido de un equipo por parte de un grupo de alumnos. De esta forma, el sistema facilitaría el trabajo con prácticas grupales, empleadas de forma esporádica en las asignaturas. Esta ampliación también facilitaría la cooperación entre alumnos en prácticas individuales, pudiendo comparar estrategias distintas para un problema dado.

También se propone una mejora del módulo de visualización. Actualmente se dispone de una webcam que captura varios equipos remotos, diferenciados entre ellos por etiquetas. Para mejorar la experiencia de usuario y calidad de la herramienta se propone el particionado de la señal de vídeo. Mostrando así, en cada panel de control del equipo remoto, la partición de vídeo que corresponde a dicho equipo.

Por otro lado, se propone como objetivo a futuro el diseño de un sistema motorizado controlable por el alumno para la verificación de prácticas que empleen un sensor de distancia. En algunas de las sesiones de la asignatura de Estructura de Computadores se emplea un sensor de distancias que funciona a través de infrarrojos, en las que el alumno debe desarrollar un controlador del sensor. Para verificar el correcto funcionamiento de su código, el alumno toma varias distancias de referencia y las coteja con las mediciones del sensor. Dichas pruebas no se pueden realizar en la herramienta actual de forma remota, ya que el sensor sólo podría colocarse con una distancia de medición predefinida. Para ello, se pueden emplear servomotores con movimientos previamente definidos y que el alumno pueda seleccionar una distancia conocida, haciendo que el sistema actúe sobre el servomotor para posicionar la referencia del sensor a dicha distancia. De esta forma mantenemos el enfoque de trabajo con elementos hardware reales gestionados de forma remota.

Por último, se propone permitir acceso a la herramienta desde la red pública. A este respecto, se barajan dos estrategias: la integración de la herramienta en UnicanLabs para poder acceder desde su página web, y que sea un servicio más al igual que las salas comunes y laboratorios que ya existen o la implementación propia de mecanismos de seguridad para ofrecer acceso directo al sistema desde el exterior sin la intervención de UnicanLabs.

REFERENCIAS

- [1] Arduino. [Último acceso: Junio 2022]. URL: <https://www.arduino.cc/>.
- [2] Arduino 4 Relay Shield Documentation. [Último acceso: Junio 2022]. URL: <https://docs.arduino.cc/hardware/4-relays-shield>.
- [3] Arduino Ethernet Shield Rev2 Documentation. [Último acceso: Junio 2022]. URL: <https://docs.arduino.cc/hardware/ethernet-shield-rev2>.
- [4] Arduino MEGA 2560 Rev3 Documentation. [Último acceso: Junio 2022]. URL: <https://docs.arduino.cc/hardware/mega-2560>.
- [5] Arduino UNO R3 Documentation. [Último acceso: Junio 2022]. URL: <https://docs.arduino.cc/hardware/uno-rev3>.
- [6] BerryClip+ User Guide. [Último acceso: Junio 2022]. URL: <https://bitbucket.org/MattHawkinsUK/rpisky-berryclip-plus/downloads/>.
- [7] FFmpeg: A complete, cross-platform solution to record, convert and stream audio and video. [Último acceso: Junio 2022]. URL: <https://www.ffmpeg.org/>.
- [8] File Transfer Protocol. [Último acceso: Junio 2022]. URL: <https://tools.ietf.org/html/rfc959>.
- [9] HyperText Markup Language, version 5. [Último acceso: Junio 2022]. URL: <https://html.spec.whatwg.org/multipage/>.
- [10] Network File System, version 4. [Último acceso: Junio 2022]. URL: <https://datatracker.ietf.org/doc/html/rfc3530>.
- [11] noVNC - the open source VNC client. [Último acceso: Junio 2022]. URL: <https://novnc.com/info.html>.
- [12] Solicitud de reserva para el comedor de la Facultad de Ciencias. [Último acceso: Junio 2022]. URL: <https://web.unican.es/centros/ciencias/solicitud-de-reserva-para-comedor>.
- [13] SparkFun 74HC595 8-Bit Shift Registers With 3-State Output Registers Datasheet. [Último acceso: Junio 2022]. URL: <http://cdn.sparkfun.com/datasheets/BreakoutBoards/MM74HC595.pdf>.
- [14] Acorn Computers. Acorn OmniClient. [Último acceso: Junio 2022]. URL: <https://www.riscosopen.org/wiki/documentation/show/Software%20information:%20omni>.
- [15] Apple Inc. QuickTime user guide. [Último acceso: Junio 2022]. URL: https://support.apple.com/es_ES/manuals/quicktime.
- [16] ARM. Arm limited, “arm limited roadshow slides q2 2020. [Último acceso: Junio 2022]. URL: https://group.softbank/system/files/pdf/ir/presentations/2020/arm-roadshow-slidesq2fy2020_01_en.pdf.
- [17] Cristóbal Camarero, Fernando Vallejo, David Herreros, and Pablo Fuentes. UCDebug, an ARM Debugger for RISC OS. [Último acceso: Junio 2022]. URL: <https://fuentesp.github.io/UCDebug/>.

-
- [18] Colin Granville. FTPc. [Último acceso: Junio 2022]. URL: <http://www.ftpc.iconbar.com/>.
- [19] Universidad de Cantabria. UNICANLabs. [Último acceso: Junio 2022]. URL: https://sdei.unican.es/Paginas/servicios/salas_aulas/UNICANLabs.aspx.
- [20] Raspberry Pi Foundation. Raspberry Pi Documentation - GPIO. [Último acceso: Junio 2022]. URL: <https://www.raspberrypi.com/documentation/computers/raspberry-pi.html#gpio>.
- [21] Raspberry Pi Foundation. Raspberry Pi Documentation - Power Supply. [Último acceso: Junio 2022]. URL: <https://www.raspberrypi.com/documentation/computers/raspberry-pi.html#power-supply>.
- [22] Google. WebRTC. [Último acceso: Junio 2022]. URL: <https://webrtc.org/>.
- [23] IBM. Server Message Block. [Último acceso: Junio 2022]. URL: <https://docs.microsoft.com/es-es/windows/win32/fileio/microsoft-smb-protocol-and-cifs-protocol-overview>.
- [24] Vladimir Kirilov Mateev Mateev. Desarrollo de un laboratorio de docencia hardware a distancia, 2021. Trabajo de Fin de Grado.
- [25] Mark Otto, Jacob Thornton, et al. Bootstrap, The most popular HTML, CSS, and JS library in the world. [Último acceso: Junio 2022]. URL: <https://getbootstrap.com/>.
- [26] Taylor Otwell. Laravel, the PHP Framework for Web Artisans. [Último acceso: Junio 2022]. URL: <https://laravel.com/>.
- [27] Raspberry Pi Foundation. Raspberry Pi. [Último acceso: Junio 2022]. URL: <https://www.raspberrypi.org/>.
- [28] RISC OS Open Limited. RISC OS. [Último acceso: Junio 2022]. URL: <https://www.riscosopen.org/>.
- [29] Simsso. NFS client for RISC OS. [Último acceso: Junio 2022]. URL: <https://www.riscos.info/index.php/Sunfish>.
- [30] Simsso. Shift Register 74HC595 Arduino Library. [Último acceso: Junio 2022]. URL: <https://github.com/Simsso/ShiftRegister74HC595>.
- [31] Spartner. Laravel-Excel. [Último acceso: Junio 2022]. URL: <https://github.com/SpartnerNL/Laravel-Excel>.
- [32] The OpenSSH Project. Secure Copy Protocol. [Último acceso: Junio 2022]. URL: <https://github.com/openssh/openssh-portable/>.
- [33] The Samba Team. Samba. [Último acceso: Junio 2022]. URL: <https://www.samba.org/>.
- [34] VideoLAN Organization. VLC Media Player. [Último acceso: Junio 2022]. URL: <https://www.videolan.org/>.
- [35] Vint Cerf and Robert Kahn. Transmission Control Protocol. [Último acceso: Junio 2022]. URL: <https://datatracker.ietf.org/doc/html/rfc793>.
- [36] Xiph.ORG. The Xiph open source community. [Último acceso: Junio 2022]. URL: <https://www.xiph.org/>.

A. Código API del módulo de control

```
/* Libraries */
#include <SPI.h>
#include <Ethernet.h>
#include <ShiftRegister74HC595.h>

/* LAN Parameters */
byte  MAC[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED}; // MAC
byte  IP[]  = {192, 168, 0, 3}; // Direccion IP
EthernetServer server(80); // Puerto

/* Create an instance of the ShiftRegister74HC595 */
int numberOfShiftRegisters = 2; // number of shift registers attached in
    series
int serialDataPin = 7;           // DS
int clockPin = 5;                // SHCP
int latchPin = 6;               // STCP
ShiftRegister74HC595<2> sr(serialDataPin, clockPin, latchPin);

/* Output Parameters */
String ID[] = {"6y1kp", "0p3cz", "1m4aq", "9j3nf",
              "4", "5", "6", "7", "8", "9", "10", "11", "12", "13", "14", "15"};
int NUM_DISP = 16;

/* HTTP Requests */
String uri = "";

void
setup()
{
    // Serial communication
    Serial.begin(9600);

    sr.setAllHigh();

    // Network initialization
    Ethernet.begin(MAC, IP);
    server.begin();
}

void
loop()
{
    int i;
    EthernetClient client = server.available();
    if (client)
    {
        while (client.connected())
        {
            if (client.available())
            {
                char c = client.read();
            }
        }
    }
}
```

```
    if (uri.length() < 100)
    {
        uri += c;
    }

    Serial.write(c);

    Serial.print(c);

    if (c == '\n')
    {
        for (i = 0; i < NUM_DISP; i++)
        {
            if (uri.indexOf(" token="+ID[i]+"&status=ON") > -1)
            {
                sr.set(i, HIGH);
                delay(250);
            }

            if (uri.indexOf(" token="+ID[i]+"&status=OFF") > -1)
            {
                sr.set(i, LOW);
                delay(250);
            }
        }

        client.println("HTTP/1.1 200 OK");
        client.println(" Content-Type: text/html");
        client.println("");
        break;
    }
}
}
delay(1);
uri="";
client.stop();
}
```

B. Código para la comprobación del registro de desplazamiento

```
#include <ShiftRegister74HC595.h>

// Create a global shift register object
// Parameters: <number of shift registers> (data, clock, latch pin)
ShiftRegister74HC595<2> sr(12, 9, 11);

void setup() {
}

void loop() {
  // setting single pins
  for (int i = 0; i < 8; i++) {
    sr.set(i, HIGH); // set single pin HIGH
    delay(250);
  }
  for (int i = 0; i < 8; i++) {
    sr.set(i, LOW); // set single pin HIGH
    delay(250);
  }
}
```
