



***Facultad
de
Ciencias***

**DESARROLLO DE UN LABORATORIO
DE DOCENCIA HARDWARE A
DISTANCIA**

(Development of a remote setup for
teaching with real hardware)

Trabajo de Fin de Grado
para acceder al

GRADO EN INGENIERÍA INFORMÁTICA

Autor: Vladimir Kirilov Mateev

Director: Pablo Fuentes

Julio - 2021

Resumen

Este trabajo de fin de grado comprende el desarrollo de un sistema de laboratorio remoto para la realización de sesiones prácticas usando equipos Raspberry Pi bajo el sistema operativo RISC OS.

Esta iniciativa surge debido a la situación de aislamiento domiciliario que se ha vivido durante tiempos de pandemia, lo cual ha conllevado la necesidad de reestructurar la docencia práctica en las asignaturas del área de Estructura y Organización de Computadores, donde dichas prácticas están centradas en el uso de hardware real. El objetivo del proyecto es, por tanto, proporcionar a los alumnos de un mecanismo para que puedan desarrollar las prácticas de forma telemática como si estuviesen en el laboratorio de las asignaturas.

Como parte del trabajo desarrollado se ha hecho un análisis de las necesidades a cubrir con el sistema y un estudio de las soluciones disponibles, con el fin de seleccionar la más adecuada. El sistema permite que el alumno pueda interactuar de forma remota con el escritorio de la Raspberry Pi, gestionar su alimentación (principalmente para poder reiniciar el equipo remoto en caso de fallo) e interactuar con dispositivos hardware periféricos. Asimismo, el sistema facilita que alumno y profesor puedan visualizar el mismo equipo de forma simultánea en tiempo real, lo que facilita la resolución de dudas y la realización de pruebas de evaluación.

Para su desarrollo, se ha dividido el sistema en sus componentes hardware y software. La parte hardware del sistema se organiza a su vez en cuatro grandes módulos: alimentación del sistema, lógica de control, visualización e interconexión. Se ha seleccionado al efecto un sistema basado en placas Arduino con gestión de relés, una fuente de alimentación que abastece al sistema y a los equipos Raspberry Pi, una webcam y un switch Ethernet a 10/100Mbps. La parte software del sistema comprende un front-end con una interfaz gráfica accesible a través de un navegador, un back-end para gestionar el manejo del sistema, y un servidor ligero para transmitir órdenes a la lógica de control mediante peticiones HTTP.

Tras el estudio, se ha implementado un prototipo del sistema para verificar su adecuado funcionamiento. El prototipo está dotado de dos equipos remotos y dos interfaces para pines de entrada de propósito general conectados a uno de los equipos. Cabe destacar que el desarrollo del sistema ha sido exitoso, alcanzando los requisitos planteados en la fase de análisis.

Palabras clave: Docencia remota, laboratorio, Estructura y Organización de Computadores, Raspberry Pi, RISC OS, Entrada/Salida, Arduino

Abstract

This final degree project aims to develop a remote laboratory system based on Raspberry Pi devices running RISC OS.

This initiative arises due to the request for home isolation experienced by the students because of the COVID-19 pandemic. Which has led to the need to restructure practical teaching in Computer Structure and Organization courses, where these practices are focused on the use of real hardware. The target of this work is thus to provide the students with a mechanism to carry on remotely the practical sessions as if they were physically on the lab.

An analysis of the system requirements has been performed as part of the work, as well as a study to select the most suitable solution to address them. The system enables students to interact with the desktop of the remote Raspberry Pi device, manage its power supply (mainly to allow for device restart in case of failure) and handle peripheral hardware modules connected to the Raspberry Pi. The system also allows for a student and a lecturer to visualize the desktop of the same remote device simultaneously in real time, providing an easy mechanism to solve questions and conduct tests.

The system has been split into its hardware and software components for its development. The hardware side is itself organized in four large modules: power supply, control logic, visualization and interconnect. A system based on Arduino boards with relay management has been proposed, using a single power supply for the Raspberry Pi devices and the system components, a webcam and a 10/100Mbps Ethernet switch. The software side of the system comprises a front-end with a GUI reachable through a web browser, a back-end to manage the system, and a light server to send orders to the control logic via HTTP requests.

Upon completion of the study, a prototype has been implemented to ensure proper operation of the system. The prototype is equipped with two remote Raspberry Pi devices and two general purpose input interfaces attached to one of the devices. It should be noted that the system has been successfully developed, complying with the requirements established during the analysis phase.

Keywords: Remote teaching, laboratory, Computer Structure and Organization, Raspberry Pi, RISC OS, Input/Output, Arduino

Índice general

1. INTRODUCCIÓN	6
1.1. Motivación	6
1.2. Desarrollo realizado	9
2. DISEÑO HARDWARE	11
2.1. Estudio previo	11
2.1.1. Módulo de control	12
2.1.2. Módulo de visualización	18
2.1.3. Red de interconexión	19
2.1.4. Módulo de alimentación	19
2.2. Implementación de la solución elegida	20
2.2.1. Módulo de alimentación	20
2.2.2. Módulo de control basado en placas Arduino	23
2.2.3. Red de interconexión	24
2.2.4. Módulo de visualización	25
2.3. Escalabilidad del sistema	25
3. DISEÑO SOFTWARE	27
3.1. Estudio previo	27
3.1.1. Escritorio remoto	28
3.1.2. Desarrollo de la plataforma web	29
3.1.3. API del módulo de control	31
3.2. Implementación de la solución elegida	31
3.2.1. Laravel	32
3.2.2. Conexión remota	33
3.2.3. Interfaz gráfica	34
3.2.4. Módulo de visualización	36
3.2.5. API de control	37
3.3. Acceso a la plataforma	38
4. RESULTADOS Y EVALUACIÓN DEL SISTEMA	39
5. CONCLUSIONES	42
5.1. Trabajo futuro	44
REFERENCIAS	45
ANEXO A. Código API del módulo de control	48
ANEXO B. Código para la comprobación del módulo de relé	51
ANEXO C. Código para la comprobación de los controles de la página web	52
ANEXO D. Publicación en las XXXI Jornadas de Paralelismo SARTECO	

Índice de figuras

2.1.	Diagrama de bloques con los módulos HW de la herramienta.	12
2.2.	Enchufe inteligente Meross MSS210	13
2.3.	DC Power Distribution Unit (Nigel VH)	14
2.4.	Hilitand Relay controller	15
2.5.	Arduino Uno R3	16
2.6.	Arduino Ethernet 2 Shield. Módulo de expansión que permite añadir una interfaz <i>Ethernet</i> al dispositivo Arduino	17
2.7.	Arduino 4 Relays Shield. Módulo de expansión que incluye cuatro relés manejables mediante las salidas digitales del dispositivo Arduino.	17
2.8.	Fuente de alimentación OMRON S8FS-G10005CD	20
2.9.	Enchufe 3P IEC320 C14	21
2.10.	Esquema de la placa PCB	22
2.11.	VKUSRA USB 3.0 Hub	22
2.12.	Configuración de pines de las placas Arduino. El zócalo situado a la izquierda corresponde con la placa Arduino UNO, el zócalo central pertenece al Ethernet 2 Shield, y el zócalo de la derecha es del 4 Relays Shield.	23
2.13.	Esquema de la red de interconexión	24
2.14.	Trust Exis Webcam	25
3.1.	Maqueta de las vistas de la interfaz	31
3.2.	Vista del formulario de acceso	34
3.3.	Vista del listado de dispositivos	35
3.4.	Vista de escritorio	36
3.5.	Vista de profesor	36

1. INTRODUCCIÓN

El presente proyecto se sitúa en el contexto de las prácticas relativas a las asignaturas del área de Estructura y Organización de Computadores en las titulaciones de Grado en Ingeniería Informática y Grado en Ingeniería de Tecnologías de Telecomunicación de la Universidad de Cantabria.

Las prácticas permiten al alumno afianzar los conceptos adquiridos en las clases magistrales. Desde el año 2018, en la Universidad de Cantabria se emplean equipos Raspberry Pi [21] corriendo el sistema operativo RISC OS [22] en una configuración de puesto autónomo. Este enfoque, combinado con el uso de dispositivos hardware sencillos, propicia una mayor implicación del alumnado y facilita su trabajo de forma autónoma.

Sin embargo, a raíz de las medidas de confinamiento y distanciamiento social como consecuencia de la pandemia por el virus COVID-19, se observa que el acceso del alumno a los equipos de laboratorio se ve significativamente reducido. Del mismo modo, la capacidad del alumno para plantear dudas al profesor de forma directa y mostrando los errores experimentados sobre el equipo se ve mermada.

Este trabajo tiene como objetivo proporcionar al alumnado un mecanismo de acceso a los equipos Raspberry Pi del laboratorio de forma telemática, con capacidad para controlar el arranque de los equipos y para manejar dispositivos hardware conectados a los mismos. Asimismo, el trabajo pretende facilitar al alumnado el proceso para mostrar a los profesores dudas prácticas sobre sus desarrollos permitiendo el acceso concurrente de profesor y alumno al mismo equipo remoto.

1.1. Motivación

Las asignaturas del área de Estructura y Organización de Computadores tienen como objetivo formativo el aprendizaje de la arquitectura de un computador y de sus principios básicos, incluyendo la relación con el entorno del computador a través de la Entrada/Salida. Las prácticas en laboratorio de las asignaturas ayudan a los alumnos a comprender los conceptos explicados en las sesiones teóricas, aplicándolos en un escenario real de hardware físico. Las sesiones prácticas están concebidas para el aprendizaje del funcionamiento de la arquitectura de un procesador a través de la programación de códigos en lenguaje ensamblador, y del manejo de la Entrada/Salida mediante la programación de controladores, también llamados *drivers*. El uso de equipos físicos (frente al uso de simuladores y emuladores) incrementa además la motivación del alumno, facilitando el seguimiento de las asignaturas [12][16][17].

Para el desarrollo de estas sesiones prácticas se optó por dispositivos Raspberry Pi, en concreto el modelo 1B+. Este dispositivo utiliza un procesador de arquitectura ARM, la cual goza de gran relevancia y una amplia cuota de mercado en el segmento de procesadores móviles [11]. ARM es una arquitectura de tipo RISC (*Reduced Instruction Set Computing*, Computación con conjunto de instrucciones reducido), lo que reduce la complejidad de la arquitectura del dispositivo y la hace más pedagógica, ya que facilita la interacción del alumno. Estos dispositivos tienen un bajo coste y presentan una alta disponibilidad, lo que facilita su adquisición por parte de los alumnos. No obstante, su compra es opcional, y los alumnos tienen a su disposición en el laboratorio equipos propiedad de la Universidad para la realización de las prácticas.

Los dispositivos Raspberry Pi cuentan con un ecosistema de dispositivos hardware periféricos que proporcionan una variedad muy amplia, útil de cara a que los profesores planteen nuevas prácticas. Además, conviene resaltar que estos equipos pueden funcionar de forma autónoma, ejecutando un sistema operativo completo sin necesidad de un PC externo.

Para la realización de estas prácticas se emplea el sistema operativo RISC OS, en virtud de su sencillez de manejo gracias a una interfaz gráfica de usuario (*Graphical User Interface*, GUI) y de la posibilidad de manejar diversos dispositivos periféricos sin necesidad de imponer capas de abstracción. Otra de las principales características de este sistema operativo es que incluye una suite de software que permite al alumno tanto escribir su código [25] como realizar la compilación y generación de ejecutables mediante la herramienta de compilación GCC [15]. Adicionalmente dispone de un depurador de código ensamblador llamado !UCDebug [13], desarrollado en la propia Universidad de Cantabria, que permite que los alumnos puedan ejecutar los códigos de manera sencilla, teniendo el control de la ejecución y del estado de la máquina. Por otro lado, resulta conveniente señalar que RISC OS permite al alumno poder analizar el comportamiento de su código con interferencias mínimas por parte del sistema. Como contrapartida, se pueden producir errores que hagan que el sistema pase a un estado bloqueado en el momento en que se ejecute el código, obligando al alumno a reiniciar la máquina.

El puesto del laboratorio para las sesiones prácticas se compone de un dispositivo Raspberry Pi, junto con RISC OS como sistema operativo y su suite de software disponible, conectado a los periféricos necesarios como pueden ser el monitor, el teclado o el ratón y a dispositivos hardware sencillos. Sin embargo, es preciso señalar que algunos de los alumnos no disponen de un monitor externo, dado que utilizan un portátil, lo que limita su capacidad de trabajar con un equipo propio de forma autónoma. Para solventar este problema, se ofrece a los alumnos la opción de manejar el equipo de forma remota a través de una conexión de computación virtual en red (*Virtual Network Computing*, VNC). Cabe destacar que la compra de un equipo Raspberry Pi es voluntaria, y que un porcentaje significativo del alumnado decide trabajar exclusivamente con los equipos disponibles en el laboratorio.

La pandemia causada por el virus SARS-CoV-2 supone una disrupción significativa de la forma de trabajo en las sesiones prácticas. En primer lugar, el confinamiento domiciliario impuesto en marzo de 2020 hace necesaria una reorganización de las asignaturas, puesto que no estaba permitido el acceso a las instalaciones de la Universidad. Estas circunstancias fuerzan al cambio apresurado a un modelo de docencia exclusivamente online.

Inmediatamente antes del comienzo del confinamiento se ofertó un sistema de préstamos de equipos para que aquellos alumnos que no dispusiesen del equipo en su propia casa pudiesen solicitarlo y hacer uso de ellos. Esta acción propició el hecho de que los alumnos pudiesen continuar con su aprendizaje, con resultados favorables pese a las circunstancias adversas. Sin embargo, se observaron dos limitaciones: en primer lugar, al no poder los alumnos consultar dudas a los docentes de forma presencial, el proceso de depuración y por ende el aprendizaje se veían ralentizados. Por otro lado, algunos de los alumnos que no disponían de un dispositivo propio no llegaron a solicitar el préstamo, de modo que no pudieron seguir las prácticas. Asimismo, para realizar las pruebas de evaluación se decide combinar una herramienta docente (Moodle) con el uso de la Raspberry Pi, lo que supone una limitación para los alumnos que no tenían equipo en casa.

No obstante, debe tenerse en consideración que todos los alumnos tienen derecho de realizar las pruebas de evaluación, por lo que era necesario utilizar un mecanismo alternativo para determinar su nivel de conocimientos alcanzado. Como medida de urgencia, uno de los profesores de la asignatura pone a disposición del alumno su propio equipo Raspberry Pi mediante una conexión remota de su PC controlando al equipo a través de una conexión VNC para completar una serie de tareas de evaluación. Debido a la dificultad que supone realizar este tipo de acceso, se hace necesario desarrollar una solución estable para el acceso a los equipos del laboratorio que no estén en uso.

En el curso sucesivo, la Universidad adoptó medidas de higiene y seguridad que permitieron a los alumnos regresar a las aulas, pero reduciendo de forma significativa el número de puestos disponibles en los laboratorios. Esto fuerza a desarrollar algunas de las sesiones de manera telemática y a prohibir el acceso de los alumnos al laboratorio fuera del horario de clases, dificultando así el aprendizaje. Esto afecta particularmente a la realización de prácticas de Entrada/Salida, ya que los dispositivos periféricos pueden presentar problemas de disponibilidad para su compra en pequeñas cantidades, y además el uso de cada dispositivo suele limitarse a un solo curso para ofrecer variedad e incentivar al alumnado a seguir la asignatura.

Las nuevas medidas afectan también a la asistencia de las sesiones, ya que si un alumno se contagia o ha estado en contacto estrecho con un positivo en COVID-19 debe aislarse en su domicilio. Esto también perjudica a la realización de pruebas de evaluación, tanto por la necesidad de repetir pruebas a alumnos confinados como por el alto número de grupos en que hay que dividir a los alumnos para respetar los límites de aforo, perjudicando así al alumno y al profesor.

Todo ello ha fomentado la idea de disponer de un sistema de acceso remoto al laboratorio. Este sistema debe permitir al alumno el uso del equipo junto con los diversos periféricos hardware, favoreciendo su empleo en horarios extraescolares. El sistema también debe permitir que el profesor tenga acceso a ellos con el objetivo de ayudar al alumno, aclarando las dudas y apreciar detalladamente el aprendizaje de éste. Esto haría factible su uso en pruebas de evaluación a alumnos que estén confinados de forma simultánea con las pruebas presenciales del resto, asignando las mismas tareas y evitando así diferencias de criterio en evaluación y corrección. Otra de las ventajas de este sistema es que los profesores pueden utilizar estos equipos fuera del laboratorio, sin necesidad de tener que llevar el dispositivo para hacer nuevos contenidos o aclarar prácticas.

1.2. Desarrollo realizado

Para el desarrollo de este trabajo se ha realizado un estudio previo de los requisitos del sistema para cubrir las necesidades básicas identificadas: manejo remoto del equipo Raspberry Pi ejecutando el sistema operativo RISC OS, control de la alimentación del equipo, manejo de la Entrada/Salida y acceso simultáneo de profesor y alumno sobre una misma máquina. En este estudio se ha determinado que el sistema ha de constar de disponer de una lógica de control para manipular el suministro eléctrico de los equipos Raspberry Pi y de los pines de entrada de propósito general, de una fuente de alimentación que abastezca a los equipos y a la lógica de control, de un mecanismo de visualización de la salida de dispositivos hardware conectados al equipo, y de un interconexión que facilite la comunicación de todos los elementos del sistema entre sí. El sistema debe comprender también un equipo que actúe como servidor del sistema para brindar el acceso a los alumnos a través de una interfaz software. Dicho servidor debe disponer de un front-end que gestione la interacción del alumno y de un back-end que traduzca dicha interacción a acciones sobre los componentes del sistema. Por último, el sistema debe gestionar la existencia de dos tipos de usuarios, alumno y profesor, con diferentes niveles de permisos y capacidad de asignación de usuarios a recursos del sistema (los equipos Raspberry Pi).

Tras completar el análisis de los requisitos, se ha hecho un estudio de las soluciones hardware y software disponibles en el mercado para cubrir las necesidades del sistema. Dentro de dicho estudio se han favorecido aquellas soluciones que sean abiertas para el diseñador y que dispongan de buena documentación y el respaldo de una comunidad de desarrollo. En lo referente al hardware, se han evaluado dispositivos de alimentación, lógica de control, interconexión de red y mecanismos de visualización de la salida de periféricos. Respecto al software, se han contrastado frameworks para el desarrollo de front-end y back-end, así como software de gestión de la lógica de control y del mecanismo de visualización.

Finalmente, se ha desarrollado una propuesta basada en placas Arduino [1] con gestión de relés, un conexionado mediante una red Ethernet de 10/100Mbps y una webcam para la emisión de vídeo con la salida de los dispositivos hardware conectados

al equipo. El sistema emplea una conexión VNC para el manejo remoto del equipo, a través de un cliente VNC ligero embebido en una página web. Dicha web también ofrece un interfaz de manejo de alimentación y entradas, así como un reproductor embebido para visualizar el flujo de vídeo de la webcam con la imagen de los dispositivos. También se han elegido dos frameworks para el desarrollo del front-end y el back-end del servidor del sistema, Bootstrap [18] y Laravel [19]. Por último, se ha desarrollado una sencilla interfaz de programación (*Access Point Interface*, API) para la programación de la lógica de control.

La metodología de trabajo empleada en el desarrollo del sistema ha sido la iterativa-incremental, organizando el trabajo en fases y buscando inicialmente ofrecer las funcionalidades más imprescindibles en el sistema, y añadiendo nuevas características a medida que se completaba el desarrollo. Se ha dado prioridad al diseño del hardware del sistema frente al software, considerando que los elementos físicos del proyecto van a restringir los desarrollos software que se pueden emplear.

En base a la propuesta desarrollada se ha implementado un prototipo del sistema, para realizar una evaluación de su adecuado funcionamiento y garantizar el cumplimiento de las necesidades identificadas. Como parte de la verificación, se han realizado pruebas de estrés del sistema a nivel de usuario. Los resultados alcanzados demuestran que el sistema es lo suficientemente maduro para ser empleado por los alumnos a lo largo del curso próximo, así como para la realización de pruebas de evaluación a alumnos confinados en caso de emergencia.

El resto de esta memoria está estructurado como sigue. En primer lugar, se ha detallado el hardware del sistema, tanto el estudio previo de sus requisitos como la implementación de la solución escogida. A continuación se analiza el apartado de software, nuevamente empezando por el estudio previo de las necesidades a cubrir y prosiguiendo con el desarrollo de la interfaz y software de gestión elegidos. Posteriormente se hace un breve análisis del rendimiento del sistema en base a un prototipo con dos equipos Raspberry Pi y dos terminales de entrada. Finalmente, se exponen las conclusiones extraídas del trabajo y se detallan las posibles líneas de trabajo futuro.

2. DISEÑO HARDWARE

Este capítulo se centra en realizar un estudio previo que satisfaga las necesidades hardware imprescindibles, para poder ver y controlar distintas Raspberry Pi de manera remota, llegando a la implementación más adecuada para el laboratorio remoto.

El acceso a los dispositivos Raspberry Pi debe ser de manera ininterrumpida desde la interfaz del sistema, requiriendo una alimentación constante a los dispositivos y módulos asociados al sistema, y una conectividad de red entre todos los elementos del sistema.

El usuario (alumno o profesor) con un dispositivo asignado deberá poder modificar el estado de dicha máquina. Como se comentó en el Capítulo 1, el sistema operativo RISC OS permite la ejecución de códigos que tomen el control de la CPU y, por ende, la dejen bloqueada. Por tanto, resulta necesario un mecanismo que permita encender, apagar o reiniciar el equipo cuando se desee.

En el momento en que se esté haciendo uso del dispositivo Raspberry Pi para ejecución de código, se debe permitir gestionar la entrada bajo demanda y observar la salida de propósito general. Por consiguiente, se requerirá un mecanismo que permita su visualización y control.

2.1. Estudio previo

El estudio previo se estructura en los módulos siguientes:

- **Control:** realiza el control básico para manejar el estado del dispositivo (encendido, apagado y reinicio) y el sistema de entrada de propósito general.
- **Visualización:** permite que el alumno pueda ver en cualquier momento el estado físico del dispositivo y los periféricos adicionales conectados a él.
- **Interconexión:** permite que todos los módulos se comuniquen entre sí y trabajen de manera conjunta.
- **Alimentación:** proporciona la energía necesaria para alimentar los dispositivos y ciertos módulos.

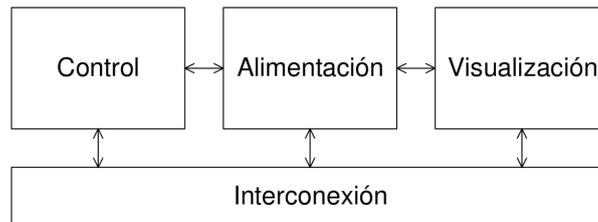


Figura 2.1: Diagrama de bloques con los módulos HW de la herramienta.

Para la realización de este estudio previo se ha partido del equipo **Raspberry Pi 1B+**, utilizado en los laboratorios de distintas asignaturas de la rama de Organización y Estructura de Computadores, con el fin de replicar la experiencia de trabajo en el laboratorio en un formato remoto.

2.1.1. Módulo de control

Como se indicó al comienzo del capítulo, el sistema necesita proporcionar un mecanismo que permita al alumno el control del estado del dispositivo. Esto se debe a que el alumno puede dejar de recibir respuesta por parte del sistema mientras está desarrollando sus prácticas, por lo que debería tener la capacidad de interrumpir y reanudar la alimentación del equipo, forzando de esta manera el arranque del sistema. No es posible implementar un mecanismo de reinicio del dispositivo mediante el sistema operativo, puesto que se parte de un escenario en el que éste no responde.

El módulo de control se subdivide en dos mecanismos:

- Mecanismo principal: encargado de las acciones de encendido, apagado y reinicio.
- Mecanismo secundario: encargado del control de la entrada y salida de propósito general (*General Purpose Input/Output*, GPIO).

La comunicación del servidor con estos mecanismos puede realizarse de varias maneras. Estas pueden ser: mediante peticiones en red o mediante conexiones periféricas al servidor, como por ejemplo con interfaces USB.

En dicho sistema resulta imprescindible disponer de una red, ya que en cualquier caso será necesario comunicar los dispositivos con el servidor, por lo que la implementación a través de peticiones en red es más sencilla y eficiente, reduciendo el volumen de cableado en el sistema.

Para la implementación de este módulo se han tomado en consideración varios dispositivos con los cuales sea posible realizar una conmutación, con la finalidad de facilitar el encendido y apagado de los dispositivos según uso remoto, así como permitir el reinicio en caso de cuelgue del sistema operativo.

Enchufes inteligentes



Figura 2.2: Enchufe inteligente Meross MSS210

Los enchufes o regletas inteligentes son dispositivos que se conectan a la red eléctrica y proporcionan una interfaz donde conectar los distintos aparatos, proporcionando así su propio control remoto gracias a un asistente de voz o aplicación móvil. Además, estos dispositivos pueden proporcionar cierta información, como puede ser el consumo eléctrico, o incluso permiten programar ciertas tareas o tiempos.

En la Figura 2.2 se muestra como ejemplo un enchufe inteligente de la marca *Meross* [5]. Este modelo en concreto puede soportar voltaje de 220 voltios con una intensidad de corriente de 16A y una potencia máxima de 3680W. El enchufe presenta un tamaño reducido e incorpora comunicación por WiFi para su uso en asistentes de voz o mediante su propia aplicación. Tiene un coste aproximado de 12 € por unidad.

Es un dispositivo visiblemente útil para domotizar una casa, por ejemplo, debido a su sencillez. Sin embargo, presenta numerosas desventajas de cara a nuestro proyecto:

1. Estos dispositivos suelen estar orientados a su uso por asistentes de voz, por lo que traen interfaces de programación de aplicaciones cerradas.
2. Cada enchufe o regleta contendrá un número máximo de equipos soportados de manera que, en el caso de que se necesite escalar el sistema o incluir más equipos, su coste se puede disparar, ya que su coste individual es elevado.
3. La comunicación remota suele realizarse por WiFi, por lo que ofrece un servicio más lento y vulnerable frente a una conexión cableada.
4. No proporciona una reducción del cableado de red, dado que será necesario conectar los equipos Raspberry Pi al servidor mediante Ethernet, por falta de interfaz de red inalámbrica en el modelo de Raspberry Pi 1B+ y de soporte en RISC OS.
5. Al tratarse de un enchufe resulta imprescindible utilizar tomas de corriente para conectar los dispositivos, en lugar de emplear una fuente de alimentación única.

Unidad de distribución de alimentación DC

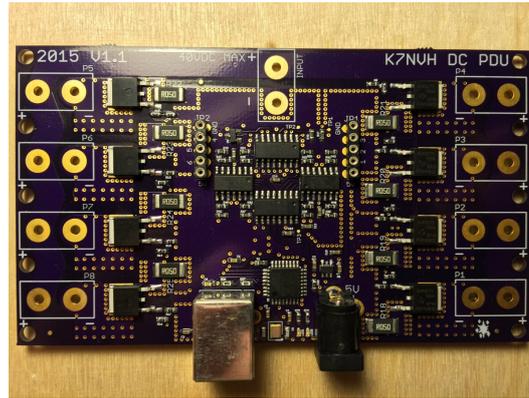


Figura 2.3: DC Power Distribution Unit (Nigel VH)

Una unidad de distribución de alimentación DC (*Power Distribution Unit*, PDU) es un dispositivo que toma como entrada una alimentación de corriente directa y la distribuye a un número de puertos independientes controlados de manera remota.

Este dispositivo es similar a una regleta de enchufes, pero con una alimentación de corriente directa en lugar de corriente alterna. Además, permite medir y controlar el uso de energía de cada puerto.

En la Figura 2.3, se puede apreciar una implementación desarrollada por *Nigel VH* [24]. Este dispositivo permite inyectar a 8 puertos individuales una tensión entre 0 y 40V y suministrar hasta una corriente de 3A por cada puerto. Asimismo, incluye sensores de corriente y tensión en cada puerto. Su precio oscila en torno a los 120€.

Este tipo de dispositivo tiene numerosas ventajas, como por ejemplo:

1. Cada implementación puede proporcionar distintos interfaces de comunicación: Ethernet, USB.
2. Incluye sensores en cada puerto para obtener los valores medidos, los cuales pueden estar disponibles en el apartado de administración de la plataforma, con el fin de controlar o hacer monitorización de los dispositivos.
3. Permite la creación de una interfaz de programación (API) propia a medida.
4. Permite controlar un número grande de dispositivos por placa.

No obstante, también reúne diversas desventajas como:

1. Estar orientado a proporcionar energía y no permitir hacer conmutaciones entre distintos terminales.
2. Su precio es bastante más elevado, dependiendo del fabricante y el número de puertos. Sin embargo, a pesar de que la inversión inicial es mayor, permiten alimentar múltiples dispositivos simultáneamente.

Este dispositivo podría tratarse de una solución apropiada de cara a los requisitos del módulo de control.

Sin embargo, sólo proporciona la funcionalidad de control del suministro de alimentación, y no permite conmutar conexiones entre terminales, por lo que no es apropiado para el control de entrada de propósito general, donde un pin con entrada activa está conectado a tierra.

Controlador de relé

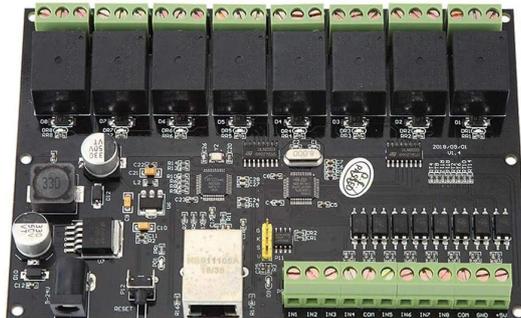


Figura 2.4: Hilitand Relay controller

Un controlador de relé es un dispositivo que controla los distintos relés o *relays*, incluidos para poder conmutar entradas de manera remota con una comunicación en red Ethernet.

Para hacer uso del dispositivo se incluye un software que ejecuta un servicio HTTP que procesa peticiones. Dependiendo de los parámetros de la petición, se permite activar o desactivar el relé correspondiente.

En algunos casos, el fabricante del dispositivo puede proporcionar el código fuente de dicho software, lo que permite realizar modificaciones sobre él. Esto permite añadir nuevas funcionalidades, entre las que se encuentran la de activar o desactivar un cierto relé durante un tiempo concreto, obtener el estado del relé, etc.

Este controlador presenta varias ventajas como:

1. Permite la conmutación entre dos elementos, ya sean terminales de entrada de propósito general o líneas de alimentación.
2. Presenta un número de relés elevado, permitiendo con un único módulo el manejo de múltiples dispositivos.
3. La comunicación se hace a través de una conexión en red.
4. Permite escalar el tamaño del sistema fácilmente.
5. El coste por dispositivo del controlador es bastante bajo.

Por otra parte, cuenta con diversos inconvenientes:

1. Requiere de una alimentación externa de mayor voltaje que los dispositivos Raspberry Pi.
2. Al tratarse de dispositivos de propósito muy específico, su disponibilidad en el mercado es baja, particularmente en canales de venta minorista.

El controlador de relé es un gran candidato para su uso en el proyecto, puesto que puede satisfacer los requisitos tanto de gestión de alimentación de la Raspberry Pi como de manejo de puertos de entrada.

Controlador de relé basado en módulos Arduino



Figura 2.5: Arduino Uno R3

Se trata de un dispositivo microcontrolador equipado con interfaces de entrada y salida, tanto analógica como digital, en la que se pueden conectar distintos tipos de módulos de expansión y circuitos propios, con el fin de desarrollar diversos proyectos.

Arduino tiene una gran comunidad activa por detrás que trabaja constantemente en la plataforma. Esta ofrece un ecosistema rico en módulos destinados a diversos propósitos SW/HW que se ajustan a las necesidades del proyecto con cambios mínimos. Además, al tratarse de un proyecto de código libre se pueden realizar dichos ajustes fácilmente, empleando el código fuente disponible.

El hecho de que Arduino sea modular es una gran ventaja, debido a que se pueden seleccionar aquellos módulos que sean necesarios para cumplir con las necesidades del proyecto y dejar abierta la posibilidad de poder cambiar la configuración elegida y cubrir necesidades futuras. Con la ayuda de dos módulos de expansión (Figuras 2.6 y 2.7) es posible replicar una placa controladora de relés, con software propio, ajustado a nuestras necesidades y con una escalabilidad superior.

Cabe también destacar que, gracias a la popularidad de Arduino, es sencillo obtener módulos en una gran variedad de proveedores, lo que facilita el proceso de compra del hardware necesario para el proyecto. Asimismo el coste de los recursos hardware es bajo.



Figura 2.6: Arduino Ethernet 2 Shield. Módulo de expansión que permite añadir una interfaz *Ethernet* al dispositivo Arduino

Con un *software* sencillo es posible poner en marcha un servidor que acepte distintas peticiones *HTTP*, permitiendo controlar cualquier tarea de manera remota.



Figura 2.7: Arduino 4 Relays Shield. Módulo de expansión que incluye cuatro relés manejables mediante las salidas digitales del dispositivo Arduino.

Arduino es también un gran candidato para el uso en el proyecto. Gracias a la modularidad que ofrece, se puede obtener a partir de él, un simple prototipo para una primera versión del módulo de control.

2.1.2. Módulo de visualización

El módulo de visualización pretende ofrecer al alumno una réplica de la experiencia de uso durante las sesiones de prácticas realizadas en el laboratorio, en donde es necesario poder interactuar con un dispositivo hardware conectado a la Raspberry Pi.

Este módulo permitirá mostrar el resultado de una alteración de un pin de entrada, realizada de manera remota, por medio del módulo de control y la observación de los resultados alcanzados en los dispositivos periféricos. Para su implementación, se barajan dos grandes alternativas: la virtualización de los dispositivos hardware periféricos, y la emisión en directo de vídeo con el estado de los periféricos.

Virtualización de periféricos

La virtualización de periféricos consiste en emplear un software de adquisición de datos y control de instrumentación para digitalizar los pines de entrada y salida de propósito general, proporcionando un desarrollo visual de cada dispositivo periférico y modelando su comportamiento. Esta solución proporciona una flexibilidad completa donde el alumno debería seleccionar el periférico a utilizar y podría comprobar el funcionamiento de sus códigos.

Por otro lado, su desarrollo resulta mucho más complejo y costoso, ya que requiere de un complejo hardware dedicado para poder captar y reproducir las entradas y salidas. Además introduce una barrera entre el alumno y el hardware real, puesto que lo que se estaría utilizando es un dispositivo emulado en lugar de uno real, con las diferencias pedagógicas que dicho cambio entraña.

Por tanto, se opta por descartar esta implementación para una versión inicial del sistema, dejándola como posibilidad a explorar como parte del trabajo futuro, tal y como se puede observar en la Sección 5.1.

Emisión en directo

La emisión en directo, también llamada *streaming*, consiste en emitir en tiempo real el vídeo capturado por una webcam, colocada de manera que enfoque a los distintos dispositivos Raspberry Pi.

La interfaz del sistema para cada equipo Raspberry Pi tendría conexión a un solo canal. Este canal se corresponde con el área del vídeo donde se ve dicho equipo, de forma que cada alumno solo ve la parte correspondiente a los dispositivos que está manejando.

Se trata de una solución sencilla, de bajo coste y que permite trabajar con cualquier dispositivo periférico que emplee una salida visual, flexibilizando así el trabajo con múltiples tipos de dispositivos hardware.

2.1.3. Red de interconexión

Para poder coordinar el funcionamiento de los diversos módulos del sistema, se necesitará una conexión de red que permita la comunicación de dichos módulos, los dispositivos Raspberry Pi y el servidor de la plataforma.

Para ello, serán necesarios un adaptador de red o una tarjeta de interfaz de red y un *switch* o enrutador Ethernet.

El uso de un adaptador de red nos permitirá proveer al servidor de una interfaz de red adicional, de modo que se produzca un aislamiento de las conexiones con la red externa y con los equipos Raspberry Pi y la lógica de control. Normalmente, un equipo dispone de al menos una interfaz, la cual será utilizada para conectarlo con el exterior. El uso de una segunda interfaz proporciona un aislamiento físico sin necesidad de recurrir a un switch programable, que tiene un mayor coste de adquisición. Este aislamiento garantiza que los equipos Raspberry Pi no sean directamente accesibles desde el exterior, dado que no presentan una seguridad robusta frente a ataques porque el sistema operativo RISC OS es antiguo y permite el manejo a muy bajo nivel, siendo inherentemente poco seguro. De este modo se evita que los equipos se conviertan en un vector de ataque/puerta de acceso a la red de la Universidad.

El switch nos permitirá conectar los elementos internos del sistema y los dispositivos Raspberry Pi. Para su correcta elección, resulta imprescindible prestar atención a las características de red soportadas por los elementos. El switch debe emplear el protocolo de red Ethernet puesto que es el único tipo de interfaz de red del que disponen los equipos Raspberry Pi. En este caso, es suficiente que la velocidad máxima de transmisión soportada por el switch sea de 100Mbps, puesto que los equipos Raspberry Pi no pueden operar a velocidades superiores y el resto de módulos del sistema tienen un flujo de transmisión de datos realmente pequeño, dado que se tratan de mensajes de control cuyo volumen va a ser despreciable frente a las comunicaciones con los equipos.

2.1.4. Módulo de alimentación

El módulo de alimentación está compuesto únicamente por una fuente de alimentación que cumple con los requisitos para suministrar energía en el sistema.

El módulo de alimentación puede usarse para proporcionar suministro exclusivamente a los distintos dispositivos Raspberry Pi, pero también puede alimentar a otros módulos del sistema, por ejemplo como el de control. En ese caso, los módulos no necesitarían hacer uso de su propia fuente externa de alimentación proporcionada por el fabricante, simplificando el montaje del sistema.

Se planteó suministrar energía a por lo menos 4 dispositivos Raspberry Pi para el prototipo inicial, permitiendo de esta manera obtener una prueba de concepto con un número máximo de hasta 4 dispositivos. Por defecto, se pretende alimentar los dispositivos Raspberry Pi, para lo cual será imprescindible tener en cuenta las características recomendadas por el fabricante.

Atendiendo al modelo 1B+ que se está empleando, la intensidad de corriente recomendada es de 1.8A para su correcto funcionamiento [7]. Sin embargo, esto dependerá del uso que le demos. Si se añaden distintos periféricos al dispositivo, este drenará corriente adicional. Según el fabricante se nos recomienda alimentar el dispositivo mediante su adaptador de corriente universal con una intensidad de corriente de 2.5A y un voltaje de $\pm 5.1V$, para alimentar a cuatro dispositivos, será necesario contar con una fuente de alimentación de al menos 5V y 12A.

En caso de alimentar también demás módulos del sistema, se deberá consultar su características e intercambiar por una fuente de mejores prestaciones, capaz de soportar el sistema completo.

2.2. Implementación de la solución elegida

A raíz del estudio previo se decide implementar un sistema basado en una placa Arduino UNO, empleando placas adicionales para implementar la lógica de control, como una placa de interfaz de red Ethernet y una placa de gestión de relés. De esta forma se pretende mantener unos costes reducidos a la par que garantizar la flexibilidad y escalabilidad del diseño. Estas placas Arduino comparten alimentación con los equipos Raspberry Pi. La gestión de la alimentación de los equipos se complementa con una placa PCB a la que se conectan los cables de alimentación USB. El resto del sistema se complementa con un adaptador de red USB, un switch Ethernet a 10/100Mbps, y una webcam de bajo coste.

2.2.1. Módulo de alimentación

El módulo de alimentación esta constituido por una fuente de alimentación y una placa PCB que proporciona una interfaz de alimentación donde se conectarán los distintos dispositivos.



Figura 2.8: Fuente de alimentación OMRON S8FS-G10005CD

La fuente de alimentación escogida, mostrada en la Figura 2.8, cuenta con una entrada de 100 a 240 V en corriente alterna que proporciona una salida de corriente continua de 5V y un rango de corriente entre 0 y 16A entregando una potencia nominal de 80W. Aunque cuenta con un rango de corriente mayor al determinado en el estudio previo, esto, nos permitirá alimentar el conjunto completo de elementos del laboratorio remoto, incluyendo el módulo de control.



Figura 2.9: Enchufe 3P IEC320 C14

La fuente de alimentación no trae consigo ningún tipo de cableado para su conexión a la red eléctrica. Para realizar dicha conexión, se ha optado por un enchufe de tipo *3 Pin IEC320 C14* como el mostrado en la Figura 2.9. Este enchufe incorpora un interruptor y protección a sobre-tensión gracias al fusible de 10A que incorpora.

Como se explicó en la Sección 2.1.1, se pretende controlar la alimentación de los equipos Raspberry Pi mediante conmutación con relés. Una forma rápida de hacer este conexionado es cortar los cables de alimentación microUSB a USB que se usan en el laboratorio, eliminando el extremo USB tipo A y conectando los cables internos directamente a los terminales de la fuente de alimentación y al relé correspondiente. No obstante, es una solución poco elegante, que obliga a cortar y rehacer cables (incumpliendo las recomendaciones eléctricas del fabricante del cable), y que impide reutilizar posteriormente los cables para conectar los equipos en el puesto de laboratorio convencional.

Una alternativa eficaz es el uso de una placa PCB con conectores hembra USB tipo A. Los conectores se unen a los terminales de la fuente de alimentación a través de un circuito conectado a los relés del módulo de control. Cuando los relés conmutan a un estado activo, se cierra el circuito entre el conector USB hembra y la fuente de alimentación, de modo que un cable USB conectado a la Raspberry Pi pueda alimentarla. De esta forma se pueden usar los cables para alimentar la Raspberry Pi en los puestos de laboratorio remoto y convencional, y el conexionado es muy sencillo, como se observa en la Figura 2.10.

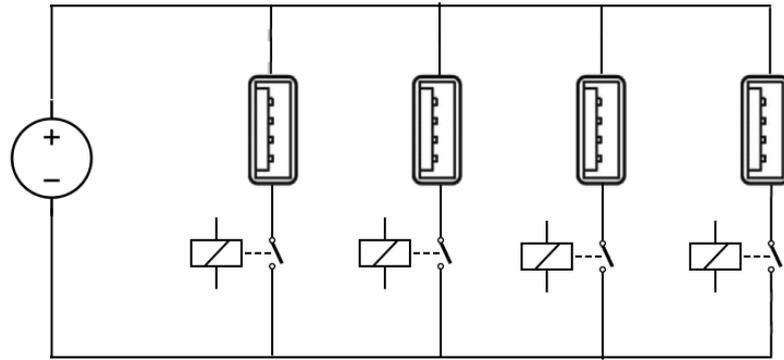


Figura 2.10: Esquema de la placa PCB

La creación de una placa PCB específica requiere de un mayor tiempo, debido a que se debe realizar el estudio previo del circuito y la implementación puede tardar varias semanas en fábrica. Una alternativa ágil es partir de una implementación de placa preexistente y modificarla para que cumpla con la funcionalidad deseada.



Figura 2.11: VKUSRA USB 3.0 Hub

Como solución práctica se ha optado por partir de un hub USB que expande un único puerto en varios, en este caso 4, como se ilustra en la Figura 2.11, y que permite interrumpir la conexión a cada puerto USB de forma independiente.

Para poder emplearlo como placa PCB de alimentación, se han eliminado el resto de componentes hasta dejar solo las líneas de alimentación y las clavijas de los puertos USB. Se han sustituido los conmutadores originales de cada puerto por terminales conectados a cada relé de control mediante cables. Por otro lado, las líneas originales de alimentación obtenidas del conector USB macho se han unido directamente a la fuente de alimentación, soldando dos cables a las líneas de voltaje de la placa. De este modo, la placa PCB recibe 5V que luego transmite a cada puerto, dado que están conectados de manera paralela.

2.2.2. Módulo de control basado en placas Arduino

Se ha escogido utilizar una placa *Arduino UNO* junto con dos periféricos adicionales: el módulo *Arduino Ethernet 2 Shield* para habilitar la conexión en red y el módulo *Arduino 4 Relays Shield* que nos proporciona un total de cuatro relés para poder realizar las conmutaciones necesarias en el módulo de control.

El diseño de estos periféricos adicionales replica los pines de conexionado originales de la placa *Arduino*, para facilitar el conexionado, por lo que es posible la conexión de ellas de manera apilada. Pero, si el propio periférico tiene una configuración de uso de pines similar a otro periférico ya conectado, apilar estos ya no nos sirve. Para ello se debería realizar un ajuste de la conexión de este a otros pines digitales correspondientes mediante cables de extensión, respetando siempre la normativa del fabricante.

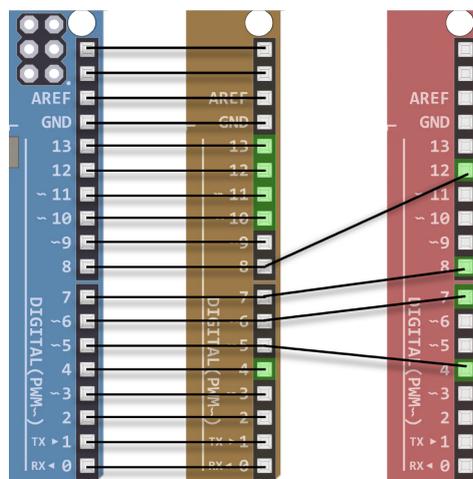


Figura 2.12: Configuración de pines de las placas Arduino. El zócalo situado a la izquierda corresponde con la placa *Arduino UNO*, el zócalo central pertenece al *Ethernet 2 Shield*, y el zócalo de la derecha es del *4 Relays Shield*.

En la Figura 2.12 se puede observar la configuración de los pines utilizados para conectar los distintos módulos a la placa *Arduino*. La placa de la izquierda es la propia placa *Arduino*, la placa del medio es el módulo de *Ethernet* y la placa de la derecha es el módulo de relés. Como se puede observar, ambas placas de expansión utilizan una serie de pines comunes, por lo tanto no se pueden apilar las tres placas y se debe conectar una de ellas mediante cables de extensión.

El módulo *Arduino Ethernet 2 Shield* proporciona un puerto de red *Ethernet* que estará conectado a nuestro switch. El módulo *Arduino 4 Relays Shield* tendrá conectado parte de los relés a los conmutadores originales sustituidos de la placa *USB* modificada. Estos relés se encargarán de dar energía a los dispositivos *Raspberry Pi*, realizando las tareas de encender, apagar o reiniciar dichos dispositivos cuando recibe la señal por parte de la placa *Arduino UNO*. Los demás relés se utilizarán para gestionar la entrada de propósito general.

Dependiendo de cómo se utilicen estos relés del módulo de la placa Arduino, se podrá hacer uso de distintas configuraciones. Por ejemplo, si queremos hacer uso remoto de cuatro Raspberry Pi prescindiendo del control de entrada y salida, simplemente deberemos conectar todos los módulos de relé a la placa USB y conectar en ella todos los dispositivos. Si queremos utilizar una entrada por dispositivo, deberemos de tener dos relés conectados a la placa USB y conectar los dos dispositivos a ella y el resto de relés se mapearan en una entrada de cada dispositivo.

Por último tendremos un pequeño servicio HTTP ejecutándose en nuestra placa Arduino UNO, capaz de manejar la placa de relés de manera remota mediante peticiones HTTP con un *token* asociado a un relé y el estado a poner dicho relé. Este servicio se verá con mayor detalle en la Sección 3.2.5.

2.2.3. Red de interconexión

Como se ha indicado en la explicación previa, se ha empleado una subred local dedicada para conectar los dispositivos Raspberry Pi, junto con el módulo de control y el servidor que alberga la plataforma web, de manera que estos puedan comunicarse entre sí, pero no desde el exterior. Por otro lado, el servidor web estará también conectado a la red interna de la Universidad, permitiendo así que los alumnos se puedan conectar a la plataforma. Para ello, el equipo servidor cuenta con dos interfaces de red, incorporando un adaptador de red con conexión USB para la interfaz adicional, dado que el equipo empleado como servidor sólo cuenta de forma nativa con una interfaz de red. El conexionado resultante se puede observar en la Figura 2.13.

De esta manera, cuando un alumno realiza cualquier petición del módulo de control, ésta se procesa en el servidor que tiene comunicación con dicho módulo, y los dispositivos del sistema remoto (equipos Raspberry Pi y placas Arduino) quedan aislados.

Teniendo en consideración que los dispositivos conectados a la red interna pueden funcionar como mucho a una velocidad de 100Mbps, los conectaremos mediante un switch que soporte dichas características.

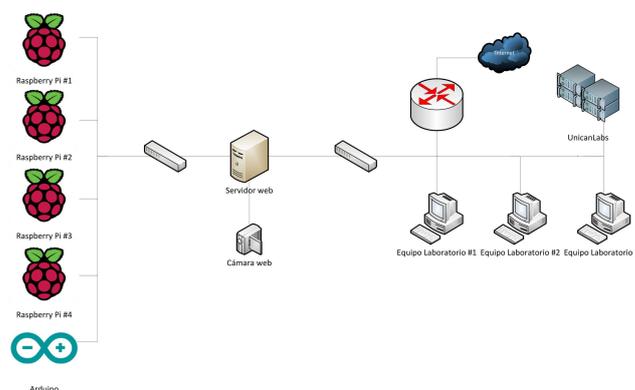


Figura 2.13: Esquema de la red de interconexión

2.2.4. Módulo de visualización

Uno de los aspectos importantes es ofrecer al alumno una visualización en tiempo real del estado de los periféricos conectados al equipo Raspberry Pi. De esta forma, el alumno podrá comprobar sus códigos que hagan uso de estos dispositivos periféricos.

Por lo tanto, se decidió que para este prototipo se utilizaría una cámara web conectada al servidor que permitiese la emisión en directo de la imagen capturada enfocando a los dispositivos periféricos. Esta solución es mucho más rápida de implementar, de menor coste ya que nos puede servir cualquier cámara web del mercado y permite observar el comportamiento de multitud de dispositivos hardware sin necesidad de reconfigurarla.



Figura 2.14: Trust Exis Webcam

La Figura 2.14 ilustra el modelo de cámara web escogido. Se trata de una cámara web de bajas prestaciones que cuenta con una interfaz USB para el vídeo y una interfaz Jack para el sonido. No se trata de una cámara de muy alta resolución y calidad, pues simplemente necesitamos que cumpla con la necesidad básica de poder verse los periféricos conectados a la máquina Raspberry Pi.

2.3. Escalabilidad del sistema

El prototipo está implementado para inicialmente proporcionar soporte a un número limitado de dispositivos, con el objetivo de escalar el sistema una vez haya sido verificado, de cara a su uso en las asignaturas.

Para escalar el sistema deberemos de tener en cuenta ciertos aspectos previos, como son el número de dispositivos Raspberry Pi que serán conectados y el número de terminales de entrada de propósito general que serán utilizados.

Una solución compleja que ofrece una mayor calidad de cara a un producto final es fijar el número de dispositivos Raspberry Pi soportados por cada nodo y diseñar una placa PCB personalizada que incorpore la fuente de alimentación y el módulo de control. Esta implementación se menciona más adelante en la Sección 5.1.

Otra solución barajada para escalar el sistema, es el uso de una placa Arduino de versión superior que ofrezca un mayor número de pines digitales, los cuales permitan controlar un número mayor de relés sin la necesidad de añadir circuitos adicionales. Tener un número mayor de pines digitales permitiría también su uso para hacer un control de las entradas de propósito general de los equipos. Para ello sería necesario utilizar un circuito externo basado en un divisor de tensión para reducir la tensión, debido a que los pines de la placa Arduino funcionan con una tensión de 5V y el equipo Raspberry Pi utiliza 3.3V para los pines de entrada/salida.

Si se decide el uso de una placa Arduino como la empleada en el prototipo con un número limitado de pines digitales, se puede conseguir ampliar estos pines mediante un registro de desplazamiento. Este componente convierte los datos obtenidos en serie en salidas paralelas. Existen implementaciones en el mercado como puede ser el circuito integrado SN74HC595 [8]. Con este circuito se puede aumentar a un total de 8 salidas más por cada integrado ocupando solamente tres pines digitales de la placa Arduino. Es decir, si tenemos en serie 2 circuitos integrados tendremos 16 salidas disponibles, si se hacen uso de 3 circuitos integrados tendremos 24 salidas disponibles.

3. DISEÑO SOFTWARE

Este apartado pretende cubrir los distintos componentes de la parte *software* que permitirán el desarrollo de un panel remoto donde los alumnos podrán hacer uso de los dispositivos Raspberry Pi.

Para ello, se hará un análisis de las necesidades básicas que dichos componentes han de cubrir, así como las posibles soluciones que se han contemplado a este efecto. Posteriormente, se detallará la solución escogida y su implementación.

3.1. Estudio previo

Se pretende desarrollar un sistema software que cubra dos tareas fundamentales:

- La gestión de la interacción del equipo servidor con el hardware previamente estudiado y elegido en el capítulo anterior para la implementación del sistema.
- Un mecanismo de interacción del alumno con el hardware físico disponible en los laboratorios, es decir, los equipos Raspberry Pi y sus dispositivos hardware periféricos.

La interacción se permitirá a través de un servicio web que proporcionará acceso a una Raspberry Pi. El servicio web deberá proporcionar un mecanismo que permita la autenticación del usuario y una vista que muestre un listado de los distintos equipos Raspberry Pi que estén disponibles en cada momento, permitiendo que el usuario seleccione uno de ellos para trabajar con el mismo.

Una vez accedido al equipo, se pasará a una vista que contendrá el escritorio remoto del dispositivo y un panel de control que servirá para gestionar la entrada de propósito general y observar la salida de los diversos dispositivos periféricos conectados al equipo. Además deberá contar con un panel de actuación del estado de la propia Raspberry Pi, para poder realizar el encendido/apagado y reinicio cuando sea necesario, tal y como se detalló en el Capítulo 1.

También se contempla la disponibilidad de un sistema de gestión de reservas para que cada alumno pueda utilizar el dispositivo en una franja horaria. Por último, se considera la posibilidad de disponer de un mecanismo para que los profesores puedan controlar y gestionar el acceso y uso de los recursos.

Como parte del estudio previo se estudiarán las diferentes posibilidades o componentes que nos permitan el acceso remoto al escritorio y su control de teclado y ratón correspondientes, el lenguaje a utilizar para el desarrollo web, la implementación de la interfaz gráfica y la API del módulo de control.

3.1.1. Escritorio remoto

Los servicios de escritorio remoto tienen en común tres componentes:

- Servidor.
- Cliente.
- Protocolo de comunicación.

En la actualidad existen diversas formas para poder realizar la conexión del escritorio de manera remota, existiendo soluciones software de fácil acceso que implementan su propios componentes de forma privada, normalmente enfocado para el uso en empresas. Algunos ejemplos de este tipo de software son TeamViewer, Anydesk, Ammy Admin o, Microsoft RDP (*Remote Desktop Protocol*).

Este tipo de soluciones permiten el envío simultáneo del escritorio remoto junto con una comunicación de sonido y/o vídeo, así como funciones para gestionar el nivel de visualización y acceso remoto. Por otro lado, podemos encontrar protocolos de libre acceso como RFB (*Remote Frame Buffer*) utilizado por los servicios VNC (*Virtual Network Computing*).

Para escoger el servicio adecuado deberemos tener en cuenta que el sistema operativo utilizado en el laboratorio es *RISC OS*, que carece de implementaciones de las soluciones comerciales más relevantes, como las anteriormente descritas. Esto limita las posibilidades a software basado en RFB, como el servicio VNC.

Debido a la falta de desarrollos software en *RISC OS*, es necesario emplear un equipo intermedio que ofrezca el escritorio de la Raspberry Pi de forma remota. Esta funcionalidad puede incorporarse al mismo equipo que actúa como servidor del laboratorio remoto y se encarga de las reservas y accesos a los equipos y el HW asociado, pero esto supone un handicap para el rendimiento por la funcionalidad extra asumida.

Por lo tanto nos centraremos en el protocolo *RFB* utilizado por el servicio *VNC*. Este servicio se ha utilizado ya previamente por alumnos que han cursado las asignaturas de Estructura y Organización de Computadores en cursos anteriores para el trabajo autónomo en casa. Muchos de los alumnos que decidieron adquirir el equipo no disponían en casa de un puesto de trabajo similar al del laboratorio, con monitor, teclado y ratón externos, por lo que se hacía preciso proporcionar una forma de acceder al escritorio de *RISC OS* desde un PC convencional, como por ejemplo un portátil. La existencia de un servidor VNC para *RISC OS* de código abierto y con licencia gratuita de uso [20] hacen posible el manejo desde un PC externo que corra cualquier sistema operativo moderno (Windows, macOS, distribuciones Linux), conectando ambos a través de un cable Ethernet. Esto solventa además las carencias que en ocasiones supone *RISC OS* sobre la Raspberry Pi, como la lentitud de manejo de ficheros PDF con la documentación de referencia de las asignaturas, ya que es posible acceder en paralelo al fichero PDF y al escritorio remoto desde el PC.

Este servidor VNC para RISC OS recibe constantes actualización para su mejora, e incluye diversas opciones que hacen que el uso del dispositivo sea más completo, como múltiples conexiones al escritorio, uso de teclas de atajos, o uso del portapapeles. La conexión simultánea de varios accesos al mismo escritorio posibilita que varios alumnos puedan trabajar de forma coordinada sobre la misma Raspberry Pi, o que muestren fácilmente a un profesor el escritorio de forma remota para la resolución de dudas. Por dichas razones, se ha decidido integrar este servidor como parte de nuestro proyecto.

Para hacer uso del servidor de escritorio remoto, se ha de barajar un cliente **VNC** que pueda ser embebido en un navegador, con el fin de integrarlo en nuestra plataforma. Para ello se han tenido en consideración dos clientes *VNC* de código abierto:

- Apache Guacamole [10]: cuenta con una aplicación web lista para su uso sin necesidad de embeber en un proyecto. Permite conectarse a distintos protocolos como: VNC, RDP, SSH. Además se puede hacer uso de su propia API para la integración en paginas web.
- noVNC [6]: cuenta con una aplicación web lista para embeber en la página web, incluyendo todas las características de uso remoto programadas. Permite solo la conexión con servidores VNC. Asimismo ofrece una API para la integración con una documentación sencilla.

Ambos pueden satisfacer la integración del escritorio remoto en nuestra plataforma. Pero, se ha decidido utilizar noVNC por su amplia comunidad y uso en distintas corporaciones, además de proveer de una documentación sencilla.

3.1.2. Desarrollo de la plataforma web

Para el desarrollo de la aplicación web se ha tenido en cuenta que deberemos de utilizar lenguajes de programación web que permitan crear una página dinámica, permitiendo así una mejor gestión a la hora de añadir los equipos, asociarlos a los distintos alumnos, incluir sistemas de reserva y sistemas de autenticación. Como parte del estudio, se han considerado los siguientes lenguajes de programación web:

- PHP: se trata de un lenguaje de código abierto muy popular que es especialmente adecuado para el desarrollo web.
- Node.js: se trata de un proyecto de código abierto que proporciona un entorno en tiempo de ejecución multiplataforma basado en el lenguaje de programación de JavaScript.

Ambos mecanismos de programación son igualmente válidos, pero dependiendo del proyecto que realicemos nos vamos a decantar por uno o por otro. Después de un estudio de ambos se ha elegido PHP, por tener estas ventajas sobre Node.js [26]:

1. Los entornos de desarrollo para PHP son más fáciles de configurar.
2. A igualdad de funcionalidad implementada, se requiere menos código en PHP que en Node.js, por lo que el desarrollo es mucho más rápido.

3. PHP tiene un mayor soporte entre los distintos proveedores de alojamientos en Internet.
4. Hay mayor disponibilidad de librerías para PHP, y además muchas de ellas están mejor documentadas.
5. Hay una buena cantidad de frameworks que soportan PHP.

Frameworks

En la actualidad es muy común el uso de un entorno de trabajo o *framework* a la hora de desarrollar una aplicación de escritorio o web. Esto es debido a que nos proporciona una estructura o patrones definidos que podremos aprovechar como punto de partida en nuestro proyecto.

En este trabajo se ha decidido seleccionar un framework que responda a nuestras necesidades básicas, de modo que no sea necesario implementar desde cero la funcionalidad del sistema, sino que puedan reutilizarse desarrollos preexistentes sabiendo que están comprobados y ofrecen soporte. El desarrollo será además mucho más rápido, evitando códigos repetitivos como programar consultas a la base de datos. Asimismo, podremos hacer uso de plantillas de formularios o vistas ya disponibles, lo que facilita únicamente tener que modificar lo que no corresponda. Por otro lado, ciertos frameworks se basan en patrones de diseño que facilitan la organización del proyecto, como el patrón de Modelo-Vista-Controlador (MVC).

Se puede encontrar dos grandes grupos de tipos de framework, según la labor que desarrollan:

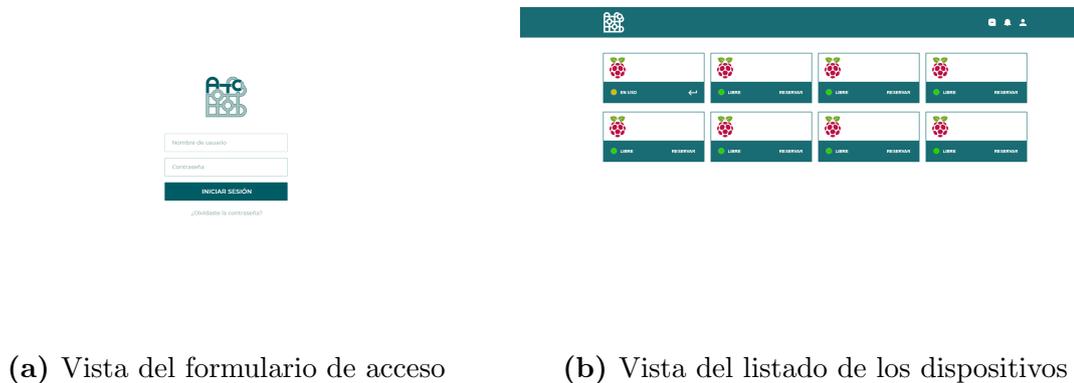
- Front-end: parte encargada de la capa de presentación y la interacción con los usuarios. Se deberá proporcionar al usuario una interfaz simple y fácil de usar. Ésta se ejecuta en el lado del cliente y suele estar codificada en lenguajes como HTML, CSS, Ajax, etc.
- Back-end: parte encargada de la capa de acceso a datos. Proporciona el procesamiento de la información de los datos obtenidos y de su posterior entrega. Ésta se ejecuta en el lado del servidor junto con su conexión a la base de datos y es inaccesible de forma directa por el usuario. Se suelen utilizar lenguajes como Python, Node.js, PHP, etc.

Implementación de la interfaz de usuario

La implementación de la interfaz de usuario se ha realizado mediante un framework front-end para realizar dicha de manera más intuitiva, sencilla y estable. Para ello se escogió desde un primer momento *Bootstrap* por la amplia documentación que contiene y ofrece a sus usuarios y su gran reputación siendo utilizada por grandes organizaciones.

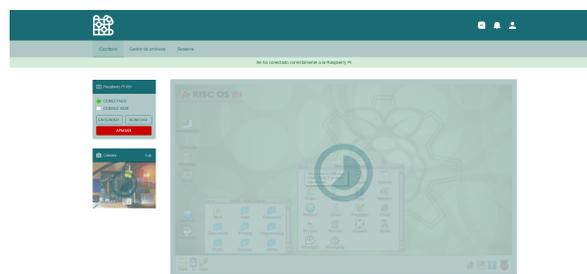
Para cumplir las necesidades para la interfaz de usuario, se realizó un boceto de diseño previo a la implementación. Éste cuenta con tres vistas básicas ilustradas en la Figura 3.1: formulario de acceso, listado de los dispositivos Raspberry Pi y el escritorio, donde

estarían los elementos de control, visualización y escritorio remoto. Se puede encontrar una descripción en detalle de las características de cada vista en la explicación de la interfaz desarrollada, en la Sección 3.2.3.



(a) Vista del formulario de acceso

(b) Vista del listado de los dispositivos



(c) Vista de escritorio del dispositivo

Figura 3.1: Maqueta de las vistas de la interfaz

3.1.3. API del módulo de control

La lógica de control del sistema se conecta al servidor del sistema mediante conexiones Ethernet, como se explicó en la Sección 2.2. Para ello se ha puesto en funcionamiento un pequeño servicio web dentro de la Arduino y se ha contemplado el desarrollo de una pequeña API que mediante peticiones de tipo HTTP es la encargada de realizar las opciones de apagado, reinicio y encendido de las máquinas y sus correspondientes entradas de propósito general. Esta API, cuya implementación se detalla en el apartado de implementación de la Sección 3.2.5, sólo es accesible por parte del servidor y los usuarios no pueden interactuar con ella de forma directa.

3.2. Implementación de la solución elegida

En esta sección se detallan las tecnologías software utilizadas para el desarrollo del sistema, incluyendo una plataforma web donde los alumnos podrán hacer uso de los dispositivos de manera remota.

Inicialmente se detallan los entornos de trabajo utilizados para el desarrollo de la página web, y se continua con la descripción de la implementación de un cliente VNC embebido en el sistema, la implementación de la cámara web para el módulo de visualización y por último la API de gestión del módulo de control.

3.2.1. Laravel

Analizados los frameworks disponibles como parte del estudio previo, se determina que la mejor estrategia es utilizar Laravel como framework para el back-end.

Laravel es un framework de código libre basado en el lenguaje de programación PHP. Se trata de un proyecto activo y que recibe actualizaciones asiduas, con una amplia documentación disponible en su página web. De entre sus características, las más relevantes (y que han contribuido a su elección para este proyecto) son:

- Utiliza el patrón de diseño **Modelo-Vista-Controlador**, que proporciona una estructura clara, sencilla y rápida de desarrollar además de ofrecer modularidad en las distintas partes.
- Dispone de un modelo de programación denominado **Eloquent ORM**, basado en un mapa de objeto relacional que simplifica el uso de consultas en la base de datos transformándola en una serie de funciones que realizan estas tareas a través de un lenguaje de alto nivel orientado a objetos.
- Al igual que la mayoría de frameworks, dispone de un motor de vista, denominado **Blade**. Este cuenta con su propio lenguaje de programación para hacer uso de variables, estructuras control y bucles.
- Presenta una interfaz de línea de comandos, llamada **Artisan**, la cual nos proporciona un medio para la interacción con la aplicación pudiendo crear modelos de controladores, objetos e incluso realizar tareas en la base de datos.
- Cuenta con una amplia cantidad de librerías propias que se podrán utilizar para ahorrar el tiempo de desarrollo. Además, permite añadir nuevas librerías y/o funcionalidades mediante el gestor de paquetes **Composer**.
- Proporciona una interfaz para hacer consultas sobre la base de datos sin necesidad de programar peticiones explícitas en SQL, además ofrece métodos dedicados para las tareas de migración, sembrado de datos para generar datos masivos y la construcción de esquemas.

Haciendo uso de este framework se dividió el desarrollo de la plataforma web en las siguientes partes:

1. Sistema de autenticación: comprende la autenticación del alumno en la plataforma. Está basado en la propia librería que incluye Laravel, proporcionando una implementación rápida, segura y comprobada.

2. Sistema de gestión de las máquinas: comprende el objeto de la clase máquina, el cual tiene una serie de atributos específicos para hacer uso de la funcionalidad del escritorio remoto y el módulo de control.
3. Sistema de administración: comprende la implementación de usuarios y tipos de roles para diferenciar entre alumno y profesor. Éste último tendrá a su disposición un panel donde poder controlar a los alumnos y las máquinas. Está basado en el paquete *laravel-admin* [23].

3.2.2. Conexión remota

La conexión remota con el dispositivo Raspberry Pi se basa en la obtención del escritorio en tiempo real mediante un cliente VNC, diseñado para ser embebido como parte de un página web, de modo que el alumno pueda acceder al escritorio remoto como parte del interfaz del laboratorio remoto.

En nuestro caso se ha escogido noVNC, un cliente de VNC basado en JavaScript que funciona en la mayoría de los navegadores actuales. Además permite la mayoría de características del protocolo VNC como:

- Escalar, recortar y cambiar el tamaño del escritorio.
- Soporte de los protocolos SSL/TLS.
- Renderizar los escritorios con distintos formatos de imagen, incluyendo compatibilidad con versiones básicas y modernas.
- Renderizado del cursor, de modo que el alumno pueda visualizar la posición y movimientos del puntero sobre el escritorio remoto.
- Portapapeles para copiar y pegar textos, tanto desde el equipo local del alumno al equipo remoto como en sentido inverso.

Para hacer uso de este cliente se requiere soportar WebSockets. WebSocket es un protocolo de comunicaciones que permite crear una comunicación bidireccional persistente entre el navegador y el servidor sobre una conexión TCP.

Para abrir un WebSocket utilizaremos un paquete proporcionado por los propios creadores de noVNC llamado *websockify* [9], que nos proporciona una implementación simple con un proxy. Este proxy lo utilizaremos para mapear las direcciones ip de los dispositivos Raspberry Pi de la red interna. De esta manera, se podrá visualizar el escritorio remoto sin necesidad de que los clientes estén en la misma subred interna que los equipos Raspberry Pi. Para ello, cada dispositivo Raspberry Pi será asignado un token de identificación único que redirigirá la conexión en el WebSocket al servidor VNC correspondiente.

Por último, deberemos de embeber el cliente noVNC en la vista correspondiente de nuestra plataforma web mediante la API que nos proporcionan en la documentación del cliente.

3.2.3. Interfaz gráfica

Para el diseño de la interfaz gráfica de nuestra plataforma web se hace uso del framework Bootstrap, como se indica en el estudio previo de la Sección 3.1.2. Este framework permite crear los contenedores y la estructura de manera sencilla y proporciona una adaptación dependiendo del monitor del alumno.

Cuando un alumno accede a la plataforma, la primera vista que se encuentra es un formulario de acceso que permite introducir sus credenciales de uso (correo electrónico y contraseña), como se identifica en la Figura 3.2. Una vez realizada la autenticación pasa a visualizar el listado de dispositivos.

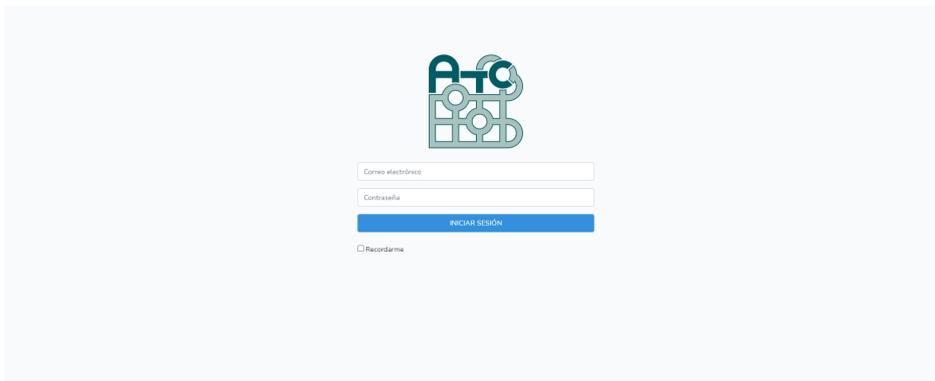


Figura 3.2: Vista del formulario de acceso

La vista con el listado de dispositivos se adapta según el tipo de usuario, que puede ser alumno o profesor. En ella se presentan los dispositivos Raspberry Pi registrados en el sistema y puedan ser utilizados, como se aprecia en la Figura 3.3.

En caso de que un profesor haya asignado un dispositivo a un alumno en concreto, este podrá acceder al equipo remoto mediante el botón “Entrar”. Si el usuario es de tipo profesor puede acceder a cualquier máquina en cualquier momento, sin importar que esté siendo usada por un alumno. Esta característica permite a un profesor ver el trabajo realizado por el alumno y aclarar cualquier duda o problema como si ambos estuviesen presencialmente en el laboratorio.

Los dispositivos que están calificados como “Disponibles” son aquellos que no han sido asignados a ningún alumno, mientras que si están marcados como “Ocupado” tienen una asignación activa en ese momento.

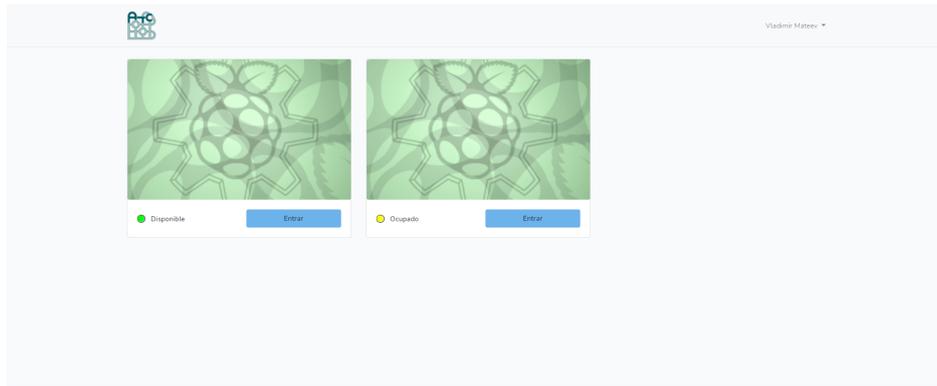


Figura 3.3: Vista del listado de dispositivos

Cuando un alumno o profesor accede a uno de los dispositivos mediante el botón “Entrar”, este pasa a la vista de escritorio (Figura 3.4). Esta vista se puede dividir en tres secciones:

- Panel superior: este panel contiene un menú que permite navegar por las distintas opciones del dispositivo, como el gestor de archivos donde se podrán hacer transferencias de ficheros hacia la Raspberry Pi o el gestor de reservas en caso de querer aumentar el tiempo de reserva o eliminarla. Por ahora la única opción que está desarrollada es la vista del escritorio.
- Panel lateral: este panel situado en la parte izquierda recopila los módulos de control y visualización. Mediante la parte superior del panel podemos encender, apagar o reiniciar el dispositivo. A través de la parte inferior del panel podemos controlar las entradas de propósito general y hacer una visualización de los periféricos conectados gracias a la emisión en directo de la cámara.
- Escritorio: este panel está situado en la parte central de la vista y permite visualizar e interactuar con el escritorio remoto. Para facilitar la usabilidad del sistema y el desarrollo de los alumnos se puede maximizar el tamaño de este panel, poniéndole en pantalla completa.

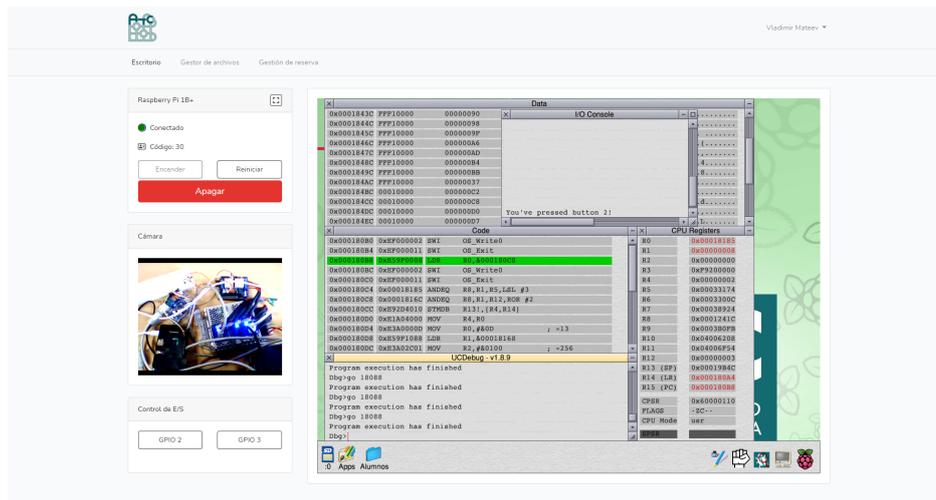


Figura 3.4: Vista de escritorio

Se dispone también de una vista de profesor (Figura 3.5) que se accede en el desplegable cuando pinchamos sobre el nombre de usuario. Esta vista, que no está disponible para usuarios de tipo alumno, habilita la posibilidad de introducir nuevos usuarios al sistema, asignarles el rol correspondiente (alumno o profesor) y asignarles el dispositivo a utilizar. También ofrece la posibilidad de añadir nuevos dispositivos Raspberry Pi al sistema o eliminarlos en caso de fallos de dichos dispositivos o reconfiguración del sistema.

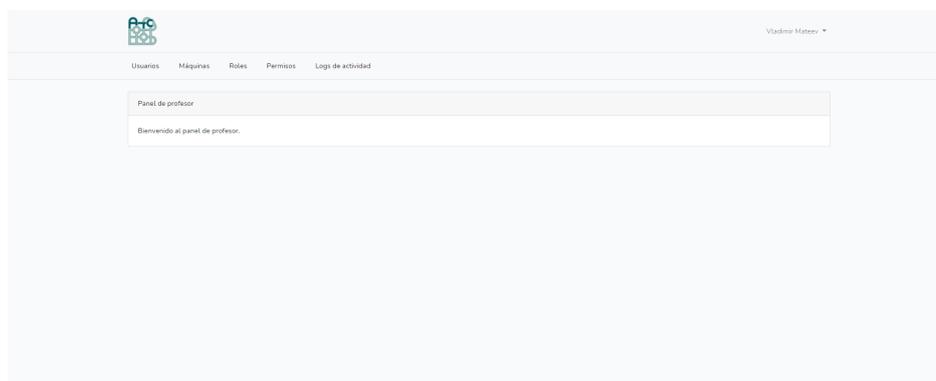


Figura 3.5: Vista de profesor

3.2.4. Módulo de visualización

En la sección 2.2.4 se había contemplado el uso de una cámara web para que el alumno pueda ver los dispositivos periféricos conectados al dispositivo que estuviera haciendo uso, asimismo del conexionado utilizado para dotar al servidor de su uso.

Para capturar la señal de vídeo de la cámara para su posterior emisión en directo, se ha hecho uso del software FFmpeg [3]. Este software nos permite utilizar el protocolo HLS (HTTP Live Streaming), con el podremos embeber la señal de vídeo dentro de

una página web. Dicho método nos evita tener que desarrollar un cliente basado en conexiones socket para hacer uso de otros protocolos como RTSP (*Real Time Streaming Protocol*, protocolo de emisión de vídeo en tiempo real). Para implementar el módulo de visualización en la interfaz es necesario incorporar un reproductor compatible con HLS. En nuestro caso, se ha seleccionado el reproductor JW Player [4] de entre las diversas soluciones de código libre disponibles.

En la Figura 3.4 podemos ver el reproductor en el apartado de “Cámara”, dentro del panel lateral izquierdo de la vista de escritorio. El rendimiento de la señal de vídeo cumple con las necesidades del sistema y permite el flujo de trabajo del alumno, ya que se puede visualizar con un nivel de detalle razonable el equipo y los dispositivos conectados, aunque presenta una latencia apreciable para el usuario.

3.2.5. API de control

Es necesario disponer de un mecanismo que permita que el servidor que alberga la plataforma pueda mandar peticiones al módulo de control. Esto permite que el alumno pueda controlar la alimentación de la máquina, apagándola, encendiéndola e incluso reiniciándola cuando sea necesario, como por ejemplo cuando cuando el sistema operativo se vuelva no responsivo. También permite controlar los terminales GPIO configurados como entrada en la Raspberry Pi, uniendo el pin del equipo con una alimentación de 3.3V proveniente del propio conector GPIO.

En la Sección 2.2 se analizan los componentes hardware que realizan estas funciones mediante el módulo de control formado por una placa Arduino y dos placas adicionales que implementan el uso de relés controlables de forma remota. La placa de relés es muy sencilla de utilizar puesto que sólo se necesita encender o apagar el pin digital adecuado para activar o desactivar dicho relé. Sin embargo, el encendido o apagado de dicho pin requiere un desarrollo software que maneje los pines asignados manualmente.

Para poder realizar el control de manera remoto deberemos crear un pequeño servicio HTTP que correrá sobre la placa Arduino UNO. Este servicio recibirá peticiones y modificará consecuentemente el valor del pin digital. Para que el sistema pueda recibir las peticiones, es necesario hacer uso de la placa de expansión con interfaz de red Ethernet. Para hacer uso de esta placa necesitaremos hacer uso de dos librerías:

- Ethernet.h: permite utilizar la placa Arduino como cliente o como servidor, aceptando conexiones entrantes. Debido a las limitaciones de la placa se soporta un total de 8 conexiones de manera concurrente, ya sean entrantes, salientes o bidireccionales.
- SPI.h: permite la comunicación entre la placa Arduino y la placa de Ethernet.

Cuando se enciende el módulo de control este aplica la configuración de red asignada en el código y posteriormente pone en funcionamiento el servicio HTTP que escuchará cualquier petición proveniente de la plataforma. El servicio responde a una sencilla API

desarrollada para este sistema. Cuando el módulo de control recibe una petición del tipo `/token={id}&status={on/off/restart}` determinará la acción a realizar y cerrará la conexión con el servidor.

El campo `token` contiene el identificador único del relé que sirve para identificar qué pin digital deberá encender o apagar la placa Arduino UNO, mientras que el campo `status` indica la acción a realizar (encendido, apagado o reinicio). En caso de recibir una petición de reinicio, el módulo de control procede a interrumpir brevemente la alimentación del equipo Raspberry Pi y a continuación reanudarlo, forzando el arranque automático del sistema.

Estos tokens se generan y asignan de manera manual de forma que el servicio sea lo más simple y ligero posible, para garantizar la funcionalidad en el prototipo. Es también posible la generación de estos identificadores de forma dinámica desde el servidor web, para lo que se deberá implementar un mecanismo de intercambio de datos mediante lenguajes como es JSON. Esta funcionalidad no se ha empleado en el desarrollo del prototipo del sistema, y se deja reservada como parte del trabajo futuro.

El código de la API desarrollada está descrito en el Anexo A.

3.3. Acceso a la plataforma

Por cuestiones de seguridad, los equipos conectados a la red cableada de la Universidad no son directamente accesibles desde el exterior, siendo necesario un acceso dedicado o el uso de una VPN, no disponible para el alumnado. Esto impide acceder directamente al servidor del sistema de laboratorio remoto, ya que hace falta realizar un proceso de certificación de seguridad para garantizar que el equipo no sea vulnerable. Sin embargo, desde el Servicio de Informática de la Universidad se ofrece la herramienta UNICANLabs, que permite el manejo de forma remota de los equipos de las salas de informática y las aulas de laboratorio de los centros [14]. Estos equipos sí pueden acceder al servidor del sistema, ya que pertenecen a la misma red cableada.

La forma de uso propuesta para el sistema es que el alumno acceda a un equipo del centro mediante UNICANLabs, y desde ahí se conecte al entorno web del sistema de laboratorio remoto. Esta forma de trabajo consume más recursos de lo deseable, ya que ocupa un equipo sólo como medio de acceso, pero proporciona un mecanismo sencillo y ya disponible para que los alumnos puedan hacer uso de la herramienta de forma fácil y segura. Al ser necesario acceder al sistema de UNICANLabs de forma autenticada, se evita que un usuario malicioso pueda explotar una vulnerabilidad en el equipo servidor, ya que todos los accesos quedan registrados.

4. RESULTADOS Y EVALUACIÓN DEL SISTEMA

Este capítulo realiza un breve recorrido del prototipo desarrollado, mostrando los resultados obtenidos tras el uso del mismo y la evaluación de los distintos componentes que lo forman y la disponibilidad del sistema.

Teniendo en consideración el estudio previo de los componentes hardware en la Sección 2.1 y el estudio previo de las tecnologías software en la Sección 3.1, se llega al desarrollo de un prototipo funcional para verificar que el sistema cumple con los requisitos marcados al comienzo del trabajo. Este prototipo proporciona acceso a dos dispositivos Raspberry Pi con dos terminales de entrada de propósito general conectados a uno de los equipos.

Dicho prototipo está compuesto por varios módulos hardware y por una plataforma web que proporciona la interfaz de acceso al mismo. El sistema está compuesto por el módulo de control basado en una placa Arduino UNO en conjunto con dos módulos adicionales que proporcionan una conexión en red y una gestión de relés. Asimismo, el sistema integra la emisión en directo de una cámara web permitiendo al alumno ver en cualquier momento la salida de los periféricos conectados al dispositivo. Por último se dispone de un módulo de alimentación basado en una fuente de alimentación y una interfaz de alimentación por USB. Todos estos módulos están conectados entre ellos mediante una red Ethernet local.

Este prototipo se ha utilizado para realizar diversas pruebas de funcionamiento y disponibilidad. En primer lugar, se han realizado pruebas sobre el módulo de alimentación para determinar la disponibilidad que nos ofrece y su correcto funcionamiento, entregando al equipo el voltaje de manera constante y el uso de corriente dependiendo del uso del equipo. Para ello, se conectaron cuatro dispositivos Raspberry Pi de manera simultánea y mediante el uso de un multímetro USB comprobando que efectivamente el voltaje disponible en los puertos USB era correcto y se mantenía de manera estable dentro de los márgenes establecidos por el fabricante. Para evaluar la disponibilidad del sistema, se ha mantenido alimentado el prototipo durante más de 10 semanas, sin observar ningún problema de caída de tensión repentina o de falta de suministro eléctrico al sistema.

Para evaluar el módulo de control se realizaron diversas pruebas del correcto funcionamiento de los relés. Para ello se utilizó el código incluido en el Anexo B que enciende los relés de forma secuencial hasta tener todos activos, y posteriormente los apaga consecutivamente. Este código, ejecutado desde la propia placa Arduino, nos ha permitido

asegurarnos de que el módulo funciona correctamente aun aumentando la velocidad de encendido. De esta manera también nos aseguramos que no exista ningún error en la configuración de los pines, ya que las limitaciones descritas en la Sección 2.2 impiden que se pueden apilar todas las placas, forzando a unir el último módulo expansor mediante cables, lo cual puede llevar a errores de conexionado.

Para evaluar el módulo de visualización, se ha comprobado que funciona correctamente y muestra una calidad de vídeo aceptable para observar la salida de un dispositivo hardware conectado al equipo asignado. Se ha podido apreciar que al utilizar el protocolo HLS se produce un tiempo de retraso perceptible a la vista, que se pretende solventar como parte del trabajo futuro, pero que no impide su uso de cara al desarrollo de las sesiones prácticas.

Por último, se ha comprobado la conexión entre los dispositivos Raspberry Pi, la placa Arduino UNO y el servidor en la red local interna, verificando la disponibilidad de todos ellos y midiendo los tiempos de respuesta, que son inferiores a 1 ms. Se trata de un valor muy reducido y despreciable frente a los varios segundos que tarda en arrancar el equipo Raspberry Pi al reiniciarlo. Además, esta latencia adicional no perjudica a la usabilidad del sistema, puesto que en el puesto de laboratorio para reiniciar el equipo el alumno ha de soltar físicamente el cable de alimentación de la Raspberry Pi y volver a reconectarlo, dado que no hay ningún botón para ello.

Por otro lado, el sistema cuenta con un servidor que proporciona una interfaz mediante la cual el alumno puede conectarse y hacer uso del equipo asignado. No se han realizado pruebas de estabilidad del entorno web, ya que al estar desarrollado mediante un framework, como ya se explicó en la Sección 3.1.2, se considera que ya ha sido validado. Sí que se ha comprobado que el sistema funcione adecuadamente y provea la funcionalidad esperada, por ejemplo, que se pueda manejar el escritorio remoto a través de la conexión VNC.

Para comprobar el correcto funcionamiento de los controles del estado de alimentación se ha utilizado el código incluido en el Anexo C, que produce un bloqueo del sistema operativo mediante un bucle infinito con lectura bloqueante de un caracter. Este tipo de código acapara los caracteres introducidos a través del teclado, de modo que no se puede detener el flujo de ejecución en el depurador mediante el comando al efecto. Al tratarse de un bucle infinito, acapara la ejecución de la máquina y sólo puede detenerse mediante un reinicio forzado del dispositivo. Esta prueba ha permitido verificar que el servidor es capaz de enviar de manera correcta las peticiones al módulo y de esta manera iniciar, apagar o reiniciar la maquina, permitiendo recuperar con éxito su uso después de que se vuelva no responsiva.

Asimismo, se ha comprobado que el servidor soporte a varias personas trabajando de manera concurrente sobre el sistema e incluso sobre la misma máquina, sin observar ningún error y asegurando que los controles sigan funcionando acorde a las especificaciones. Se ha comprobado además que el escritorio remoto basado en el servicio VNC

funciona de manera óptima no observándose tiempos de respuesta perceptibles para el usuario.

Es importante considerar que el funcionamiento correcto del sistema está supeditado a una buena conexión de red hasta el servidor del sistema. En el caso de alumnos que utilicen la red local, por ejemplo conectándose desde uno de los equipos de las aulas de informática o de los laboratorios, la conexión es estable y rápida, no observándose fallas en el uso del sistema. Para verificar el funcionamiento a través de UNICANLabs, de forma similar a la conexión que haría un alumno no presencial, varios profesores de las asignaturas del área han comprobado que el acceso sea correcto desde sus conexiones de casa. El comportamiento ha sido correcto en todos los casos, no observándose ninguna latencia perceptible, incluso empleando una conexión inalámbrica para la conexión en el hogar.

5. CONCLUSIONES

El trabajo tiene como objetivo el desarrollo de un prototipo de laboratorio virtual para el manejo remoto de equipos Raspberry Pi bajo el sistema RISC OS, así como de dispositivos periféricos con los que se interaccionan en las sesiones prácticas. Esta configuración de puesto es la utilizada habitualmente en las asignaturas del área de Estructura y Organización de Computadores. Sin embargo, la situación sobrevenida a causa de la pandemia de SARS-CoV-2 ha forzado una serie de cambios y medidas que restringen a los alumnos el acceso a los recursos de laboratorio.

Los objetivos principales del trabajo son los siguientes:

- Evaluar las necesidades del sistema en base a los requisitos de uso y necesidades docentes de las asignaturas.
- Analizar diferentes soluciones que se puedan aplicar para elaborar el sistema.
- Escoger la propuesta más adecuada para el desarrollo del sistema, en base a simplicidad de desarrollo, facilidad de uso por parte de los alumnos, y criterios de seguridad de la plataforma.
- Implementar un prototipo del sistema en base a la solución escogida.
- Verificar la viabilidad del sistema mediante el prototipo.

Tomando como referente tres casos de uso del sistema (realización de prácticas en las que se manejen dispositivos periféricos, consulta de dudas a un profesor con acceso concurrente, y realización de pruebas de evaluación) se han identificado diversos requisitos del sistema. En primer lugar, el sistema debe permitir al usuario el acceso en tiempo real a los dispositivos situados en la Universidad y la capacidad de poder iniciar, apagar y reiniciar la máquina cuando sea necesario. También debe poderse interactuar con el escritorio del equipo remoto. Asimismo, se pretende tener la capacidad para hacer uso de la entrada y salida de propósito general, siendo posible su control y visualización con la compatibilidad de cualquier periférico asociado a este. El sistema debe proporcionar una interfaz de manejo sencilla y compatible con cualquier sistema operativo moderno, para flexibilizar el acceso del alumno. También debe disponer de herramientas para la gestión de usuarios y máquinas, así como un control de permisos para restringir el uso de los alumnos. Por último, el sistema ha de permitir que un alumno y un profesor puedan acceder de forma simultánea y en tiempo real a la misma máquina, de modo que el profesor pueda observar la forma de trabajo del alumno.

Para la implementación se ha escogido una solución basada en el uso de módulos

Arduino para implementar una lógica de control mediante relés. Dichos relés están conectados al circuito de alimentación de los equipos, y a los pines de entrada/salida de los mismos. De esta forma, cada relé puede usarse para gestionar la alimentación de un equipo o el valor de un terminal configurado como entrada. El manejo de esta lógica de control se ha realizado a través de un ligero servidor de peticiones HTTP respondiendo a una API desarrollada como parte del trabajo.

El circuito de alimentación de los equipos Raspberry Pi se ha construido a partir de una placa PCB modificada y de una fuente de alimentación que también suministra a la lógica de control del sistema. Para la visualización de la salida de los dispositivos hardware periféricos conectados a las máquinas se ha empleado una webcam cuyo vídeo se muestra en un reproductor embebido como parte de la interfaz del sistema. La comunicación de los elementos del sistema se ha realizado mediante conexiones Ethernet, empleando un switch operando a 10/100Mbps.

La interfaz del sistema se ha desarrollado como un entorno web, accesible desde cualquier navegador, construido a partir de un framework preexistente. Como parte de la interfaz, el usuario tiene acceso a los controles necesarios para el control del sistema, los cuales engloban los controles de inicio y apagada y el manejo de los distintos pines de entrada y salida. Asimismo, se dispone de un escritorio remoto a través de una conexión VNC mediante un cliente ligero embebido en la interfaz del sistema. Permitiendo observar el escritorio del equipo como si este se tratase de estar conectado físicamente a un monitor. Este método tiene la ventaja de haber sido empleado por los alumnos que usan equipos Raspberry Pi propios sin disponer de un puesto autónomo completo (monitor, teclado y ratón).

Por último, en lo que respecta a la seguridad de la plataforma y el reparto de las máquinas disponibles, se ha optado por el uso de un framework que garantiza la fiabilidad y seguridad necesarias, al haber sido empleado con anterioridad por otros desarrolladores en diversas aplicaciones. Para evitar que las máquinas puedan convertirse en un mecanismo de acceso a la red interna de la Universidad se han aislado físicamente los elementos del sistema en una subred separada, permitiendo únicamente el acceso al exterior a través del equipo servidor. También se ha establecido un sistema de reserva de recursos y asociación con los usuarios.

La realización de este trabajo me ha permitido aplicar a un caso práctico los conocimientos adquiridos a lo largo del Grado, como la organización de la parte software del proyecto o la configuración de la solución de red elegida. También me ha permitido explorar nuevos temas de aprendizaje, particularmente en lo relativo a la programación de entornos web.

Para finalizar, quiero destacar que el presente trabajo ha dado lugar a una publicación aceptada para su exposición en un congreso de ámbito nacional, las XXXI Jornadas de Paralelismo SARTECO. Puede encontrarse una versión preliminar del artículo en el Anexo D.

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 funcionalidades y cambios se contemplan de cara a su desarrollo en trabajos futuros, y de entre ellos cabe destacar los siguientes:

- Unificar el módulo de control con el módulo de alimentación, diseñando un circuito propio y haciendo uso del proyecto libre de Arduino para el desarrollo de un prototipo capaz de alimentar hasta 8 equipos Raspberry Pi, con un formato reducido para su uso en racks. Esto facilitaría la escalabilidad del sistema, ya que reduce los recursos necesarios para nuevos equipos Raspberry Pi.
- Establecer un sistema de reserva de alumnos con franja horaria o por disponibilidad de recursos, donde se procura tener una limitación del tiempo de uso de los equipos. Este sistema realizaría la expulsión automática de los usuarios que hayan excedido el tiempo de la reserva y gestionaría la reserva de slots futuros.
- Implementar un gestor de ficheros que permita a los alumnos intercambiar ficheros entre el equipo remoto y el PC propio, implementando un servidor SAMBA y aprovechando las funcionalidades disponibles en el framework Laravel para gestionar un portapapeles de archivos.
- Mejorar la retransmisión de la cámara web, empleando sockets que permitan la reproducción mediante protocolo RTSP (Real Time Streaming Protocol), con la finalidad de reducir la latencia de visualización.
- Hacer un estudio previo acerca del empleo de un software de adquisición de datos, y control de instrumentación para digitalizar los pines de entrada y salida de propósito general, realizando la implementación de una de las placas disponibles en laboratorio, como por ejemplo la BerryClip+ [2]. En un caso extremo, esta funcionalidad permitiría emular el comportamiento de cualquier tipo de dispositivo hardware, lo cual podría resultar útil de cara a la realización de pruebas de uso por parte del profesorado.

REFERENCIAS

- [1] Arduino. [Último acceso: Junio 2021]. URL: <https://www.arduino.cc/>.
- [2] BerryClip+ User Guide. [Último acceso: Junio 2021]. URL: <https://bitbucket.org/MattHawkinsUK/rpispys-berryclip-plus/downloads/>.
- [3] FFmpeg: A complete, cross-platform solution to record, convert and stream audio and video. [Último acceso: Junio 2021]. URL: <https://www.ffmpeg.org/>.
- [4] JW Player is the world's most popular embeddable media player. [Último acceso: Junio 2021]. URL: <https://github.com/jwplayer/jwplayer>.
- [5] Meross - Smart Wi-Fi Plug (MSS210). [Último acceso: Junio 2021]. URL: <https://www.meross.com/Detail/3/Smart%20Wi-Fi%20Plug>.
- [6] noVNC - the open source VNC client. [Último acceso: Junio 2021]. URL: <https://novnc.com/info.html>.
- [7] Raspberry Pi Power Supply Requirements. [Último acceso: Junio 2021]. URL: <https://www.raspberrypi.org/documentation/hardware/raspberrypi/power/README.md>.
- [8] SNx4HC595 8-Bit Shift Registers With 3-State Output Registers Datasheet. [Último acceso: Junio 2021]. URL: <https://www.ti.com/lit/ds/scls041i/scls041i.pdf>.
- [9] websockify: WebSockets support for any application/server. [Último acceso: Junio 2021]. URL: <https://github.com/novnc/websockify>.
- [10] Apache Software Foundation. Apache Guacamole™. [Último acceso: Junio 2021]. URL: <https://guacamole.apache.org/>.
- [11] Arm Holdings. Arm FY2018 Q2 Key Performance Indicators report. [Published October 2018; Revised November 2018]. URL: https://www.arm.com/-/media/global/company/investors/Financial%20Result%20Docs/Arm_SBG_Q2_2018_Financial_Data_NonFinancial_KPIs_FINAL.xlsx.
- [12] Sergio Barrachina Mir, Germán Fabregat Lluca, and José Vicente Martí Avilés. Utilizando arduino due en la docencia de la entrada/salida. In *Actas de las XXI Jornadas de la Enseñanza Universitaria de la Informática*, pages 58–65. Universitat Oberta La Salle, 2015.

-
- [13] Cristóbal Camarero, Fernando Vallejo, David Herreros, and Pablo Fuentes. UC-Debug, an ARM Debugger for RISC OS. [Último acceso: Junio 2021]. URL: <https://fuentesp.github.io/UCDebug/>.
- [14] Universidad de Cantabria. UNICANLabs. [Último acceso: Junio 2021]. URL: https://sdei.unican.es/Paginas/servicios/salas_aulas/UNICANLabs.aspx.
- [15] Free Software Foundation, Inc. GCC, the GNU Compiler Collection. [Último acceso: Junio 2021]. URL: <http://gcc.gnu.org/>.
- [16] Pablo Fuentes, Cristóbal Camarero, Fernando Vallejo, and Carmen Martínez. La importancia del uso de hardware real en la docencia de estructura y organización de computadores. page 480, May 2019.
- [17] Pablo Fuentes Saez, Cristobal Camarero Coterillo, María del Carmen Martínez Fernández, Fernando Vallejo Alonso, et al. Tecnología low-cost para motivar al alumno. In *Actas de las XXV Jornadas sobre la Enseñanza Universitaria de la Informática*, volume 4, pages 319–326.
- [18] Mark Otto, Jacob Thornton, et al. Bootstrap, The most popular HTML, CSS, and JS library in the world. [Último acceso: Junio 2021]. URL: <https://getbootstrap.com/>.
- [19] Taylor Otwell. Laravel, the PHP Framework for Web Artisans. [Último acceso: Junio 2021]. URL: <https://laravel.com/>.
- [20] Hernik Bjerregaard Pedersen, Crispian Daniels, David Llewellyn-Jones, and Jeffrey Lee. VNC server for RISC OS. [Último acceso: Junio 2021]. URL: http://www.phlamethrower.co.uk/riscos/vnc_serv.php.
- [21] Raspberry Pi Foundation. Raspberry Pi. [Último acceso: Junio 2021]. URL: <https://www.raspberrypi.org/>.
- [22] RISC OS Open Limited. RISC OS. [Último acceso: Junio 2021]. URL: <https://www.riscosopen.org/>.
- [23] Sohel Amin. Laravel Admin Panel. [Último acceso: Junio 2021]. URL: <https://github.com/appzcoder/laravel-admin>.
- [24] Nigel VH. DC Power Distribution Unit. [Último acceso: Junio 2021]. URL: <https://www.tindie.com/products/nigelvh/dc-power-distribution-unit/#specs>.
- [25] Guttorm Vik, John Whittington, Carl Hetherington, and Fred Graute. StrongEd - a programmer's text editor for RISC OS. [Último acceso: Junio 2021]. URL: <http://stronged.iconbar.com/>.

- [26] Peter Wayner. Node.js vs. php: An epic battle for developer mindshare. *InfoWorld*, 2019. URL: <https://www.infoworld.com/article/3166109/nodejs-vs-php-an-epic-battle-for-developer-mindshare.html>.

A. Código API del módulo de control

```
/* Librerias */
#include <SPI.h>
#include <Ethernet.h>

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

/* Configuracion Rele */
String ID[] = {"6y1kp", "0p3cz", "1m4aq", "9j3nf"};
int PIN[] = {6,7,8,9};
int NUM_RELES = 4;

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

void
pulse(int pin, int time)
{
  if (digitalRead(pin) == HIGH)
  {
    digitalWrite(pin, LOW);
    delay(time);
    digitalWrite(pin, HIGH);
  }
  else if (digitalRead(pin) == LOW)
  {
    digitalWrite(pin, HIGH);
    delay(time);
    digitalWrite(pin, LOW);
  }
}

void
setup()
{
  // Comunicacion serial
  Serial.begin(9600);
}
```

```
// Pins correspondientes a los rele
// en modo salida
int i;
for (i = 0; i < NUM.RELES; i++)
{
    pinMode(PIN[i], OUTPUT);
    digitalWrite(PIN[i], LOW);
}

// Inicializacion de la red
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);

                if (c == '\n')
                {
                    for (i = 0; i < NUM.RELES; i++)
                    {
                        if (uri.indexOf(" token="+ID[i]+"&status=ON") > -1)
                        {
                            digitalWrite(PIN[i], HIGH);
                        }

                        if (uri.indexOf(" token="+ID[i]+"&status=OFF") > -1)
                        {
                            digitalWrite(PIN[i], LOW);
                        }
                    }
                }
            }
        }
    }
}
```

```
        if (uri.indexOf("token="+ID[i]+"&status=PULSO") >
            -1)
        {
            pulse(PIN[i], 200);
        }
    }

    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 módulo de relé

```
unsigned char Pin[4] = {6,7,8,9};

// Configura los pines como salida
void
setup()
{
    int i;
    for(i = 0; i < 4; i++)
    {
        pinMode(Pin[i],OUTPUT);
    }
}

// Enciende los relés uno a uno y al cabo
// de un rato los desconecta
void
loop()
{
    int i;

    for(i = 0; i < 4; i++)
    {
        digitalWrite(Pin[i],HIGH);
        delay(400);
    }
    delay(2000);

    for(i = 0; i < 4; i++)
    {
        digitalWrite(Pin[i],LOW);
        delay(400);
    }
    delay(2000);
}
```

C. Código para la comprobación de los controles de la página web

```
.globl _start
.text

_start:
mov r5, #0
while:  cmp r5, #0
bne fin_while
swi 0x4
b while
fin_while:
swi 0x11

.end
```

D. Publicación en las XXXI Jornadas de Paralelismo SARTECO

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

Vladimir Mateev, Esteban Stafford y Pablo Fuentes ¹

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

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

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

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

I. INTRODUCCIÓN

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

II. OBJETIVOS

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

- Uso por alumnos en remoto de forma autónoma.

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

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

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

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

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

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

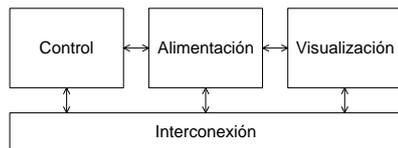


Fig. 1: Diagrama de bloques con los módulos HW de la herramienta.

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

III. SISTEMA DE ACCESO REMOTO AL LABORATORIO HARDWARE

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

A. Diseño hardware de la herramienta

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

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

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

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

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

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

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

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

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

B. Diseño software de la herramienta

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

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

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

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

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

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

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

IV. USO DE LA HERRAMIENTA

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

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



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

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

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

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

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

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

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

V. CONCLUSIONES

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

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

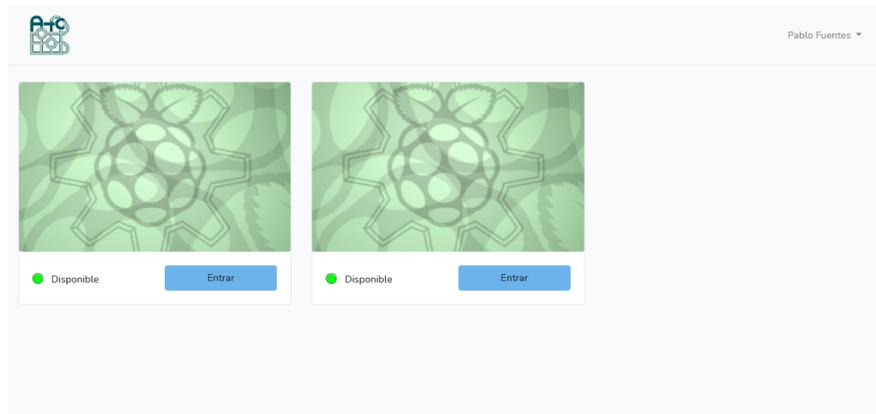


Fig. 3: Vista inicial de la herramienta, con las máquinas existentes.

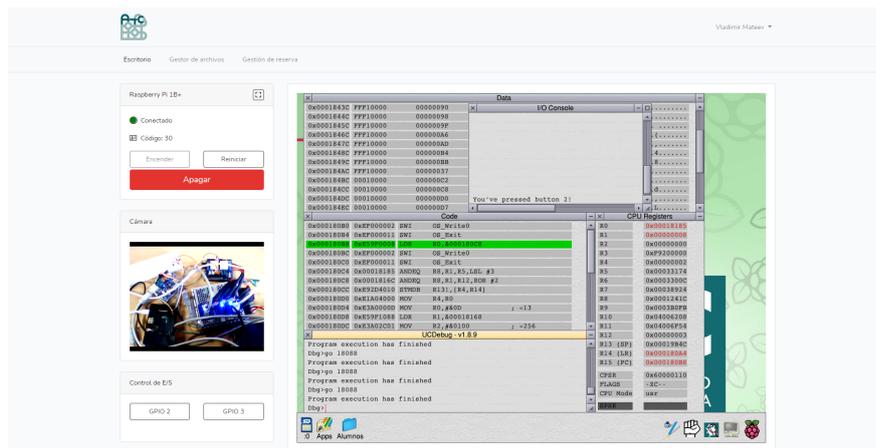


Fig. 4: Vista de escritorio de un equipo, con el entorno del sistema operativo mostrado en tiempo real, así como diversos paneles para gestionar el arranque del equipo y el manejo de la Entrada/Salida.

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

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

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

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

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

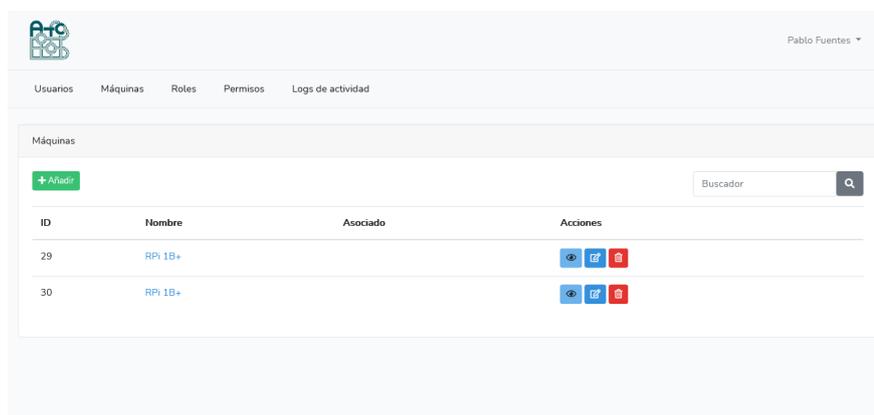


Fig. 5: Vista del panel de gestión del profesor, con diversas pestañas para manejar los usuarios y equipos disponibles, así como un control de permisos de la herramienta y registros de actividad.

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

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

AGRADECIMIENTOS

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

REFERENCIAS

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

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