



Universidad
de Cádiz

Escuela Superior
de Ingeniería

TRABAJO DE FIN DE GRADO

GRADO EN INGENIERÍA INFORMÁTICA

**DISEÑO Y PROGRAMACIÓN DE LA INTERFAZ
WEB Y LA GESTIÓN DE LA CONEXIÓN CON
EL EXTERIOR, DE LOS LABORATORIOS
REMOTOS PARA EL APRENDIZAJE DEL
MICROCONTROLADOR LCP4088 Y LA PLACA
ARDUINO MEGA.**

AUTOR: JESÚS GÁLVEZ GUERRERO

Cádiz, julio 2022



Universidad
de Cádiz

Escuela Superior
de Ingeniería

TRABAJO DE FIN DE GRADO

GRADO EN INGENIERÍA INFORMÁTICA

**DISEÑO Y PROGRAMACIÓN DE LA INTERFAZ
WEB Y LA GESTIÓN DE LA CONEXIÓN CON
EL EXTERIOR, DE LOS LABORATORIOS
REMOTOS PARA EL APRENDIZAJE DEL
MICROCONTROLADOR LCP4088 Y LA PLACA
ARDUINO MEGA.**

DIRECTOR: ARTURO MORGADO ESTÉVEZ

CODIRECTOR: EDUARDO ALEJANDRO ROMERO BRUZÓN

AUTOR: JESÚS GÁLVEZ GUERRERO

Cádiz, julio 2022

Agradecimientos

Me gustaría poder ofrecer mis agradecimientos sobre todo a mis padres que han sido capaces de apoyarme en toda la duración de mis estudios de diversas formas.

A mis tutores Arturo Morgado y Eduardo Romero.

A mis compañeros durante la carrera por hacer la estancia mas amena y ayudarme en aquellos momentos que se me hacia un poco más complicada.

A mis compañeros del Laboratorio de Robótica Aplicada, aula D12 de la Escuela Superior de Ingeniería por hacerme un hueco para trabajar allí y ayudarme si tenía alguna duda.

Resumen

Durante esta época que ha sido la pandemia del virus conocido como Covid-19 todos nos hemos visto obligado a tener que permanecer en nuestros hogares el mayor tiempo posible, esto ha desembocado en que muchas personas no pudieran realizar sus rutinas diarias, ya sea para poder trabajar como para poder asistir a los centros de educación pertinentes. Lo cual ha sido un gran inconveniente para todos, en el caso de los alumnos se han visto obligados a tener que seguir con sus estudios de una forma no presencial, propiciando que sea mas complicado el estudios de ciertas asignaturas, sobretodo si requieren de ciertos componentes físicos con los que trabajar y que sus centros de estudios no han podido proveer a cada alumno de forma individual.

Con esto en mente se ha estudiado sobre una solución para poder ofrecer a los miembros de la Escuela Superior de Ingeniería de la Universidad de Cádiz y se ha llegado a la conclusión de diseñar un sistema que actúe como un laboratorio remoto, en el que cada placa tendrá un equipo conectada a la misma que se comunicará con un servidor principal que actuará de interfaz web para que los miembros de la Universidad de Cádiz puedan conectar desde sus hogares ciertos componentes a algunos pines específicos de la placa con la que desee trabajar, principalmente Arduino o una LPC4088, y a su vez les permita subir los programas desde sus ordenadores a las mismas placas y ver como funcionan, permitiéndoles poder trabajar cómodamente desde sus casas sin tener el hardware de forma física.

Palabras claves

Se incluirán algunas palabras claves para intentar ayudar en una posterior búsqueda en archivos de organización de documentos:

- Arduino
- LPC4088
- Laboratorio Remoto
- Python
- Flask
- Web
- Sistema Distribuido

Índice

I	Prolegómeno	11
1.	Introducción	11
1.1.	Motivación	11
1.2.	Objetivos del proyecto	11
1.3.	Glosario de términos	13
2.	Situación inicial	15
2.1.	Arduino	15
2.2.	LPC4088	16
2.3.	Situación del laboratorio LPC4088	21
3.	El estado del Arte	23
3.1.	Weblab-Deusto	23
3.2.	Educon2019	27
3.3.	MIT iLab	29
3.4.	Remotely controlled laboratory de Universidad Palacký	30
3.5.	Recolab	33
4.	Planificación	36
4.1.	Metodología	36
4.2.	Fases del proyecto	36
4.3.	Costes	39
4.4.	Tecnologías/Herramientas utilizadas	39
4.5.	Tecnologías software empleadas	43
II	Desarrollo	49
5.	Análisis del proyecto	49
5.1.	Requisitos	49
5.2.	Casos de uso	51
6.	Diseño	80
6.1.	Decisiones de diseño	80
6.2.	Diseño de la arquitectura	81
6.3.	Diseño de la base de datos	82
6.4.	Diagrama de flujo	85
6.5.	Diseño de la web	87
7.	Implementación	103
7.1.	Estructura del código	103
7.2.	Enrutamiento de las vistas	107
8.	Pruebas del software	109

III	Epilogo	111
9.	Conclusiones	111
10.	Experiencia	113
11.	Trabajo futuro	113
12.	Manual de configuración uVision para poder conectar de forma remota al laboratorio remoto de la LPC4088	115
13.	Manual de instalación	118
13.1.	Instalación del software necesario para conectar los laboratorios remotos al servidor principal	119
Anexos		122
A.	Tabla de ejemplo del mapeado de interconexión de pines para los laboratorios remotos	122
Referencias		124

Índice de figuras

1.	Representación los objetivos del proyecto junto a la tecnología empleada en cada parte del mismo	12
2.	Placa Arduino MEGA	15
3.	Kit de desarrollo para Arduino	16
4.	Placa LPC4088	17
5.	Conector JTAG	18
6.	Conector JLink	18
7.	IDE uVision	19
8.	Ciclo de ciclo de compilación y programación de los microcontroladores de uVision	21
9.	Laboratorio LPC4088	21
10.	Logo Weblab Deusto	23
11.	WebCam Arduino de WebLab-Deusto(LabsLand)	24
12.	IDE Arduino de WebLab-Deusto(LabsLand)	25
13.	Interfaz FPGA de WebLab-Deusto(LabsLand)	25
14.	Interfaz VISIR de WebLab-Deusto(LabsLand)	26
15.	Laboratorio remoto con Arduino y RaspberryPI de la EDU-CON2019 de IEEE.	28
16.	Interfaz del laboratorio remoto con Arduino y Raspberry PI de la EDUCON2019 de IEEE.	28
17.	Arquitectura del ilab del MIT	29
18.	Logotipo de la Universidad Palacký	30
19.	Arquitectura de los laboratorios remotos de la Universidad Palacký	31
20.	Experimentos de los laboratorios remotos de la Universidad Palacký	32
21.	Arquitectura del servicio de Recolab	33
22.	Flujo de trabajo del servicio de Recolab	34
23.	Diagrama de Gantt	38
24.	Logo de OverLeaf y de LaTeX	40
25.	Logo de Visual Studio Code	40
26.	Icono del IDE de Arduino	41
27.	Icono del IDE de Keil uVision	41
28.	Logo de Jlink	42
29.	Placa Arduino Due	42
30.	Placa Arduino Pro Micro de Sparkfun	43
31.	Logotipo de Python	44
32.	Logo de Flask	45
33.	Logo de Jinja	47
34.	Logo de PostgreSQL	47
35.	Casos de uso respecto a los usuarios	51
36.	Casos de uso de gestión	60
37.	Casos de uso de trabajo en el laboratorio remoto	73
38.	Interfaz del cliente Forti Client para conectarse a la VPN de la UCA	81
39.	Arquitectura física del proyecto	81

40.	Diagrama de clases de la base de datos	83
41.	Diagrama de flujo de la aplicación web	85
42.	Ventana de inicio de sesión	87
43.	Ventana de registrar usuario	88
44.	Barra de navegación	89
45.	Panel de administrador	89
46.	Lista de usuarios	89
47.	Panel de eliminar usuario	90
48.	Panel de crear nuevo usuario	90
49.	Panel que lista todos los laboratorios remotos	91
50.	Panel que creación de laboratorios remoto	91
51.	Tabla de mapeado de pines de la LPC4088	92
52.	Tabla de mapeado de pines de la Arduino MEGA	93
53.	Panel que lista los permisos de acceso a los laboratorios remotos	93
54.	Panel para reservar un espacio de tiempo del laboratorio remoto	94
55.	Panel que lista las reservas de los laboratorios remotos	94
56.	Panel de modificación de laboratorios remoto	95
57.	Panel de eliminación de laboratorios remoto	96
58.	Panel de usuarios	96
59.	Panel de gestión de PDF	97
60.	Panel de gestión de PDF de una carpeta específica	97
61.	Panel que lista todos los laboratorios remotos (vista de usuario)	98
62.	Panel del laboratorio remoto de Arduino MEGA	99
63.	Panel del laboratorio remoto de la LPC4088	101
64.	Estructura del código del servidor principal	103
65.	Estructura de la carpeta de los módulos de la aplicación web	105
66.	VPN de la UCA cuando esta conectada	115
67.	uVision: Abrir configuración del proyecto	116
68.	uVision: Acceder pestaña Debug de la configuración del proyecto	116
69.	uVision: Acceder a la configuración del Debugger JLink	117
70.	uVision: Acceder a la configuración del Debugger JLink	117
71.	Activar entorno virtual	119
72.	Activar Yawcam	120

Índice de cuadros

1.	Tabla resumen de WebLab-Deusto	27
2.	Tabla resumen del laboratorio remoto presentado en la EDUCON19.	29
3.	Tabla resumen de los laboratorios controlado remotamente de la Universidad Palacký	32
4.	Costes de personal del proyecto	39
5.	Costes asociados a servicios del proyecto	39
6.	Tabla de requisitos funcionales	49
7.	Tabla de requisitos no funcionales	50
8.	CU-001: Acceder a la web	52
9.	CU-002: Iniciar sesión	52
10.	CU-003: Restablecer contraseña	53
11.	CU-004: Modificar email	54
12.	CU-005: Modificar contraseña	55
13.	CU-006: Cerrar sesión	55
14.	CU-007: Registrar usuario	56
15.	CU-008: Verificar email	57
16.	CU-009: Dar/Quitar rol de Administrador	57
17.	CU-010: Crear usuario	58
18.	CU-011: Eliminar usuario	59
19.	CU-012: Pedir acceso a laboratorio remoto	61
20.	CU-013: Realizar reserva	62
21.	CU-014: Revisar reserva	63
22.	CU-015: Eliminar reserva	64
23.	CU-016: Añadir laboratorio remoto	65
24.	CU-017: Dar/Quitar acceso a laboratorio remoto	66
25.	CU-018: Modificar laboratorio remoto	67
26.	CU-019: Eliminar laboratorio remoto	68
27.	CU-020: Gestionar PDF	69
28.	CU-021: Subir PDF	70
29.	CU-022: Mover/Copiar PDF	71
30.	CU-023: Eliminar PDF	72
31.	CU-024: Descargar PDF	72
32.	CU-025: Acceder a lab remoto	74
33.	CU-026: Cambiar PDF visible	75
34.	CU-027: Cambiar mapeado de pines	75
35.	CU-028: Subir archivo Arduino MEGA	76
36.	CU-029: Consola serie Arduino MEGA	77
37.	CU-030: Activar conexión LPC	78
38.	CU-031: Simular teclado/joystick	78
39.	CU-032: Abrir tutorial conectar uVision	79
40.	Revisión de requisitos funcionales	112

Parte I

Prolegómeno

1. Introducción

1.1. Motivación

Debido a la situación que se está viviendo con la pandemia producida por el COVID, nos hemos dado cuenta que realizar las tareas de prácticas de laboratorio de las asignaturas que requieran equipos hardware específico se complica cuando el alumno no puede acceder al Centro de educación y que, además, se rija por el horario que tenga de clases o que pueda estar presente un profesor con el alumno.

En mi caso me fue de muchísima utilidad el poder llevarme a mi hogar la placa y poder seguir trabajando desde el mismo, sobretodo para partes donde por cualquier motivo se me complicaba alguna aplicación específica y tenía que pasarme mas tiempo del normal revisando y arreglando errores para que funcione correctamente y me parece muy problemático que no todos los alumnos puedan disfrutar de ese beneficio.

Al disponer de un laboratorio remoto, al estudiante le va a resultar más interesarse el aprendizaje de estas placas de desarrollo basadas en microcontroladores con el fin de testear y completar sus prácticas.

En este momento se dispone de un prototipo básico de laboratorio remoto en el laboratorio E16 de la ESI donde se experimentan distintas implementaciones hardware y software con el fin de tener un laboratorio remoto funcional. Hasta el momento, se ha experimentado con interfaces hardware básicas a modo de prototipo para placas de desarrollo para el microcontrolador LPC4088.

La intención es tener un laboratorio completamente funcional donde tenga también interfaz gráfica software que aún no está desarrollada.

1.2. Objetivos del proyecto

Este trabajo de fin de grado se basa en el diseño e implementación de una interfaz web que permita a los estudiantes poder realizar desde sus hogares una conexión a través de un laboratorio remoto basado en el aprendizaje de los microcontroladores LPC4088 y Arduino MEGA, necesarias para poder completar sus ejercicios prácticos de algunas asignaturas y así el alumno no se vea limitado solamente al horario de las clases en las cuales puede utilizar los microcontroladores.

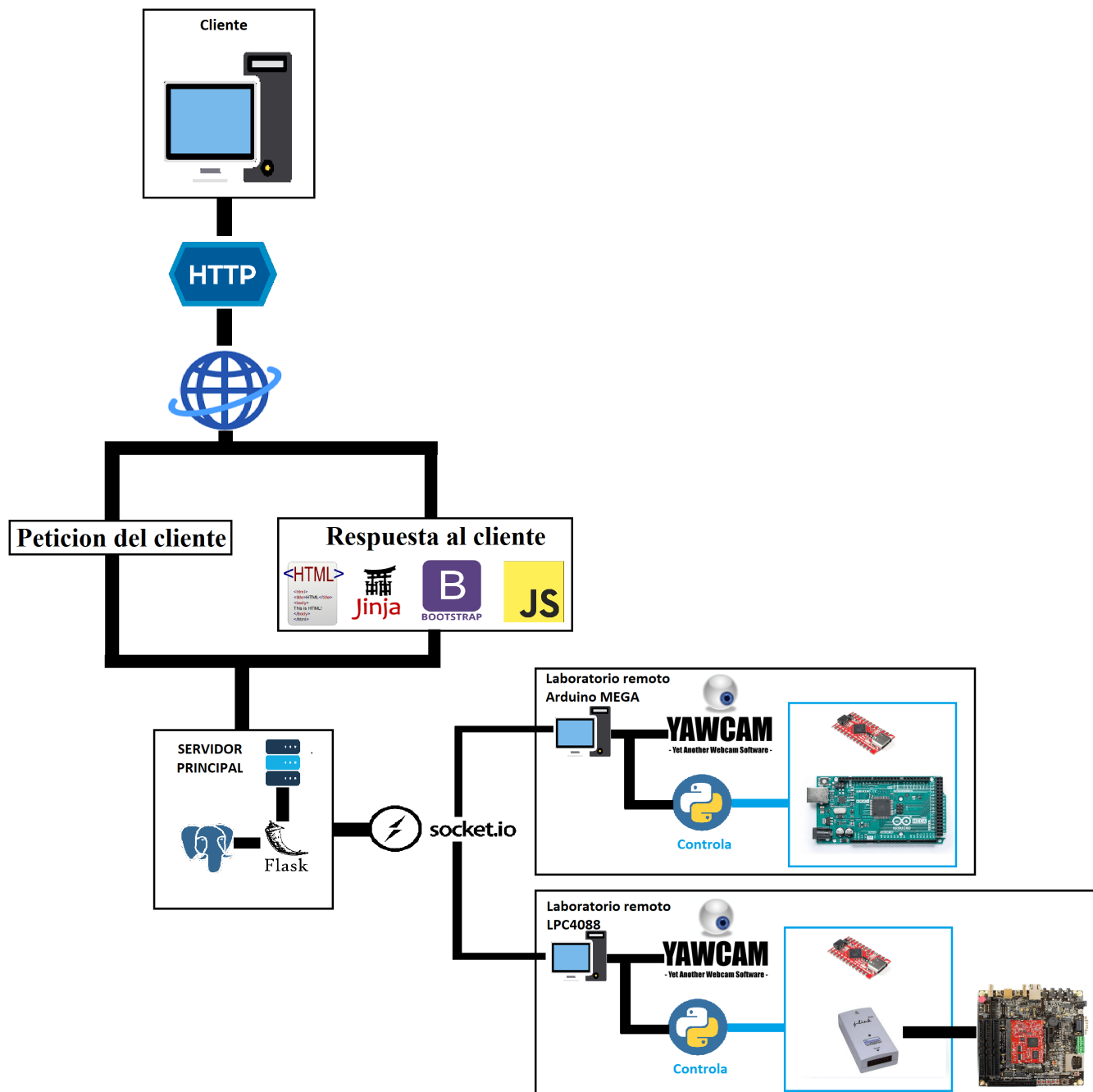


Figura 1: Representación los objetivos del proyecto junto a la tecnología empleada en cada parte del mismo

Los objetivos específicos del desarrollo software que tiene este trabajo de fin de grado son:

- Crear una interfaz web para poder comunicar el cliente con el servidor principal.
- Permitir crear, gestionar y verificar usuarios para poder decidir qué usuarios pueden acceder o no a cada laboratorio remoto.
- Recordar los datos a un usuario si se ha olvidado de las credenciales de acceso.
- Permitir añadir o eliminar nuevas placas basadas en microcontroladores.
- Diseñar un sistema que permita gestionar qué practicas se podrán visualizar mientras se está trabajando con los microcontroladores.
- Crear un sistema de reservas que permita que solamente un usuario y los administradores puedan acceder de forma simultánea a cada laboratorio.
- Controlar que los usuarios no se mantengan conectado al microcontrolador fuera de su tiempo asignado.
- Comunicar el ordenador que hace de servidor con los distintos laboratorios remotos.
- Visualizar los distintos dispositivos físicos (sensores, actuadores y el microcontrolador) en la interfaz web.
- Permitir comunicar la interfaz web con la placa de control del laboratorio remoto.
- Permitir subir archivos de código al equipo conectado con la placa Arduino MEGA para poder programar el microcontrolador.
- Permitir la comunicación con el equipo que tenga el microcontrolador Arduino MEGA actuando como consola serie.
- Diseñar una interacción que permita al usuario activar la conexión con el equipo conectado al JLink para poder trabajar con el Entorno de Desarrollo Integrado Keil uVision de manera remota.
- Virtualizar teclado y joystick y enviar la señal al controlador del laboratorio remoto.
- Realizar la guía de configuración del uVision para conectarse con el laboratorio de forma remota.

1.3. Glosario de términos

En este apartado se describen algunos términos que son necesarios conocer antes de comenzar a leer esta memoria:

- **Python:** Lenguaje de programación, interpretado, empleado durante la realización del proyecto.

- **Front-end:** Parte de la aplicación que sera vista por el usuario y con la que podrá interactuar.
- **Back-end:** Parte de la aplicación que estará del lado del servidor y que se encargara de la mayoría de la lógica de la misma aplicación.
- **Framework:** son herramientas que nos dan un esquema de trabajo y una serie de utilidades y funciones que nos facilita y nos abstrae de la construcción de páginas web dinámicas.
- **Modelo-Vista-Controlador(MVC):** El patrón MVC es una manera o una forma de trabajar que permite diferenciar y separar lo que es el modelo de datos (los datos que van a tener la App que normalmente están guardados en BD), la vista (página HTML) y el controlador (donde se gestiona las peticiones de la app web).
- **Flask:** Framework de Python empleado para el desarrollo de todo el Back-end de la aplicación web.
- **Flask-X:** Módulo específico diseñado para añadir una funcionalidad X a Flask.
- **Token CSRF:** Es una seguridad ante un tipo de exploit malicioso llamado Cross-Site Request Forgery(CSRF) en el cual una pagina web recibe comandos maliciosos de un usuario identificado del que se tiene confianza por permisos, usando el token CSRF se puede prevenir este tipo de exploits de seguridad.
- **HTML:** Es el componente más básico de la Web. Define el significado y la estructura del contenido web.
- **Jinja2:** Motor de plantillas incluido en Flask que permite simplificar el desarrollo del Front-end en conjunto con añadir lógica a las sentencias HTML.
- **Script:** secuencia de comandos que se usa para designar a un programa relativamente simple.
- **Javascript:** Lenguaje de programación ligero usado para implementar lógica en el Front-end mediante Scripts.
- **WSGI:** Tambien llamado Web Server Gateway Interface, es un estándar de Python que describe el como un servidor web debe comunicarse con una aplicación web.
- **Arduino:** Plataforma de electrónica de software y hardware fáciles de programar y de utilizar.
- **LPC4088:** Placa de desarrollo basado en el uso de un microcontrolador ARM.

2. Situación inicial

En este apartado se va a analizar la situación actual de los laboratorios donde se realizan las practicas de microcontroladores:

2.1. Arduino

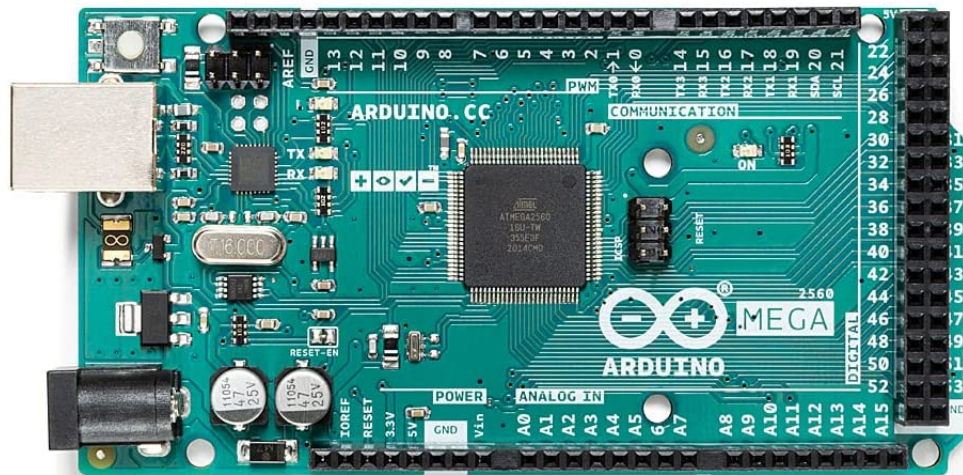


Figura 2: Placa Arduino MEGA

Arduino es una plataforma de creación de electrónica de código abierto, la cual está basada en una mentalidad de hardware y software libre, flexible y fácil de utilizar para cualquier usuario que desee introducirse en este mundo con pocos conocimientos. Aunque esta mentalidad no exime a la plataforma de poder realizar proyectos mas complejos que satisfarán las ideas de cualquier usuario avanzado.

El Arduino MEGA utiliza un microcontrolador ATmega2560 que permite un diseño sencillo y económico. Una de las ventajas que ofrece este modelo es que dispone de USB nativo por hardware y por lo tanto no necesita de ningún conversión serie-USB para poder realizar la comunicación. También permite a la placa ser utilizada y programada como un dispositivo de entrada para emular un teclado, ratón, etc.

Soporta 54 entradas digitales y 16 entradas analógicas y dado que el puerto de comunicación USB es emulado, deja el puerto serial hardware libre para la programación! De esta forma ya no ocurren conflictos de programación mientras tenemos periféricos serial conectados a la placa.

Características:

- Microcontrolador: ATmega2560
- Tensión de funcionamiento: 5V
- Alimentación recomendada: 7-12V

- Pines I/O Digitales: 54
- Canales PWM: 15
- Entradas analógicas: 16
- Corriente Máxima de los pines I/O: 20 mA
- Corriente continua del pin 3.3V: 50 mA
- Memoria Flash: 256 KB (8 KB usados para el bootloader)
- SRAM: 8 KB
- Velocidad: 16 MHz

En este caso las practicas de Arduino se pueden realizar en cualquier laboratorio, puesto que son placas de tamaño reducido que vienen con su kit de desarrollo en un pequeño maletín, permitiendo que se pueda usar cualquier laboratorio sin problemas, pero en muchos casos los maletines que incluyen el kit de desarrollo y el Arduino deben quedarse en posesión del profesorado al terminar la clase de practicas, impidiendo al alumnado poder seguir practicando en su hogar o fuera del horario de clases.

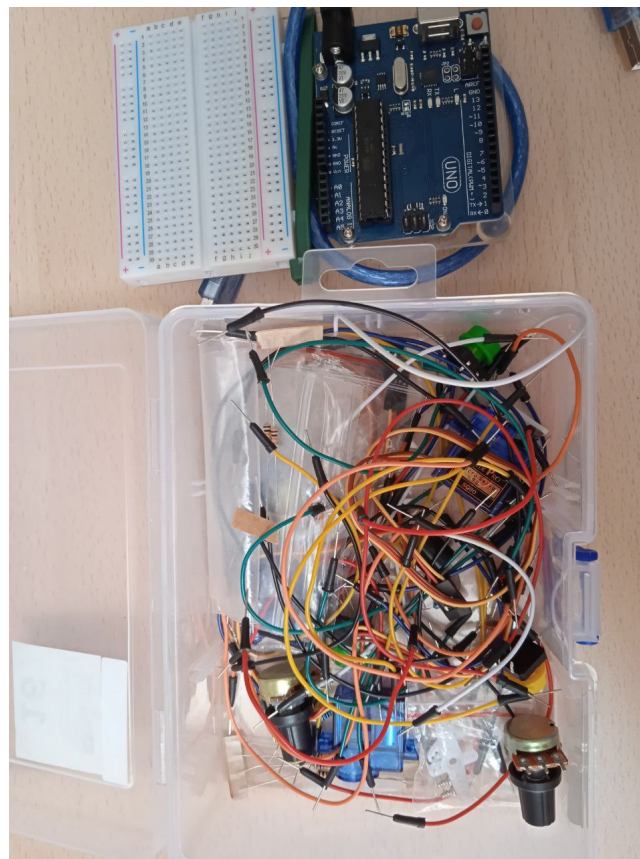


Figura 3: Kit de desarrollo para Arduino

2.2. LPC4088

El segundo microcontrolador que se usará en el proyecto es el ARM Cortex M4 de NXP, LCP4088.

Es un microcontrolador de 32 bits que incluye varias funciones de comunicación como podrían ser: USB, Ethernet, LCD, CAN, I2C e I2S. El chip utiliza una arquitectura Harvard, lo cual significa posee de forma separada la memoria de datos y de programa físicamente, cada una con su propio bus de la CPU, lo cual aumenta la eficiencia del microcontrolador.

Otras características son:

- Procesador con una frecuencia de hasta 120 Mhz.
- 512KB de memoria FLASH.
- 96KB de memoria SRAM.
- Fuente de alimentación de 3,3V.
- Hasta 165 pines de propósito general de E/S.
- Múltiples niveles de interrupción para ejecutar código en situaciones puntuales o bajo condiciones específicas.

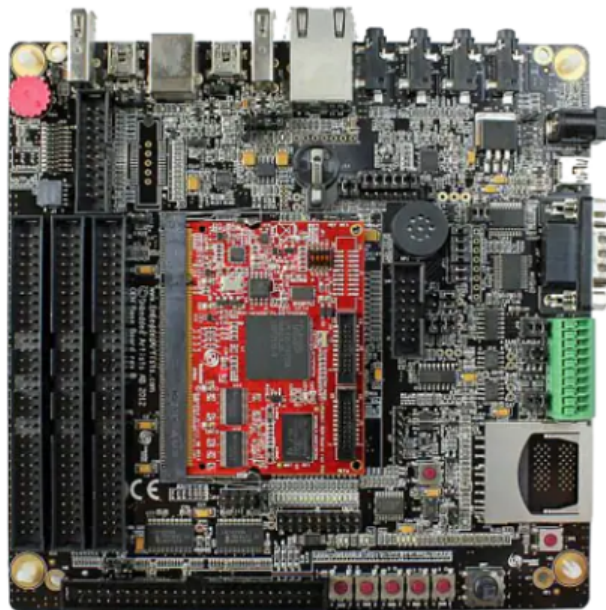


Figura 4: Placa LPC4088

Además, tanto las placas que se encuentran en el laboratorio como la mostrada anteriormente en la figura 4 se encuentran en un kit de desarrollo de la empresa OEM *Embedded Artist EAB* que permite añadir un conector SWD/JTAG, además incluir botón de Reset, LEDs, sensor de temperatura y mas cosas.

El conector JTAG que añade este kit de desarrollo permite que se pueda utilizar en entorno de desarrollo de uVision para poder programar y depurar de forma cómoda el microcontrolador LPC4088.

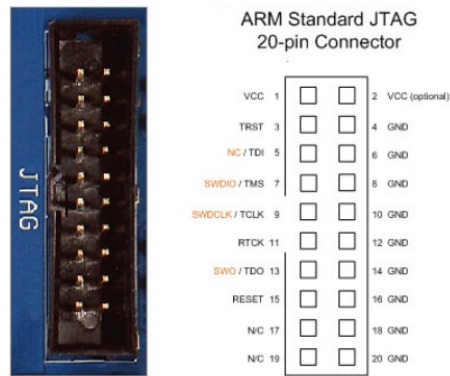


Figura 5: Conector JTAG

Para poder realizar esta programación y depuración del microcontrolador necesitaremos conectar el equipo a la placa con un JLink el cual sera el encargado de programar el código que escribamos en el entorno de desarrollo, de uVision, dentro del microcontrolador de la LPC4088.



Figura 6: Conector JLink

El dispositivo utilizado en la escuela es la versión destinada a educación, con lo cual no se puede usar para otros fines que no sean estos y utilizar el dispositivo además permite que las instrucciones sean mandadas de forma mas rápida que si directamente el ordenador fuera el

encargado de traducir y mandar las instrucciones a la LPC4088.

Aparte con el dispositivo *SEGGER* pone a la disposición de sus usuarios distintos componentes software para la utilización del JLink como son:

- J-Link que es una consola de línea de comando para mandar comandos directamente al dispositivo JLink6.
- Servidor Remoto Jlink que son 2 herramientas, una con interfaz gráfica y otra que es una consola de comandos que nos permiten conectarnos de forma remota al dispositivo JLink sin tener que estar conectados por un cable físico.

Aunque estas herramientas no se usen en los laboratorios nosotros aprovecharemos la segunda herramienta para poder hacer que el entorno de desarrollo del alumnado se pueda conectar de forma remota al Jlink.

Como hemos comentado antes para poder programar la placa de la LPC4088 utilizaremos el entorno de desarrollo de *Keil* uVision, el cual se encargará de compilar el código C/C++ y de programar el controlador LPC4088 con la ayuda del JLink.

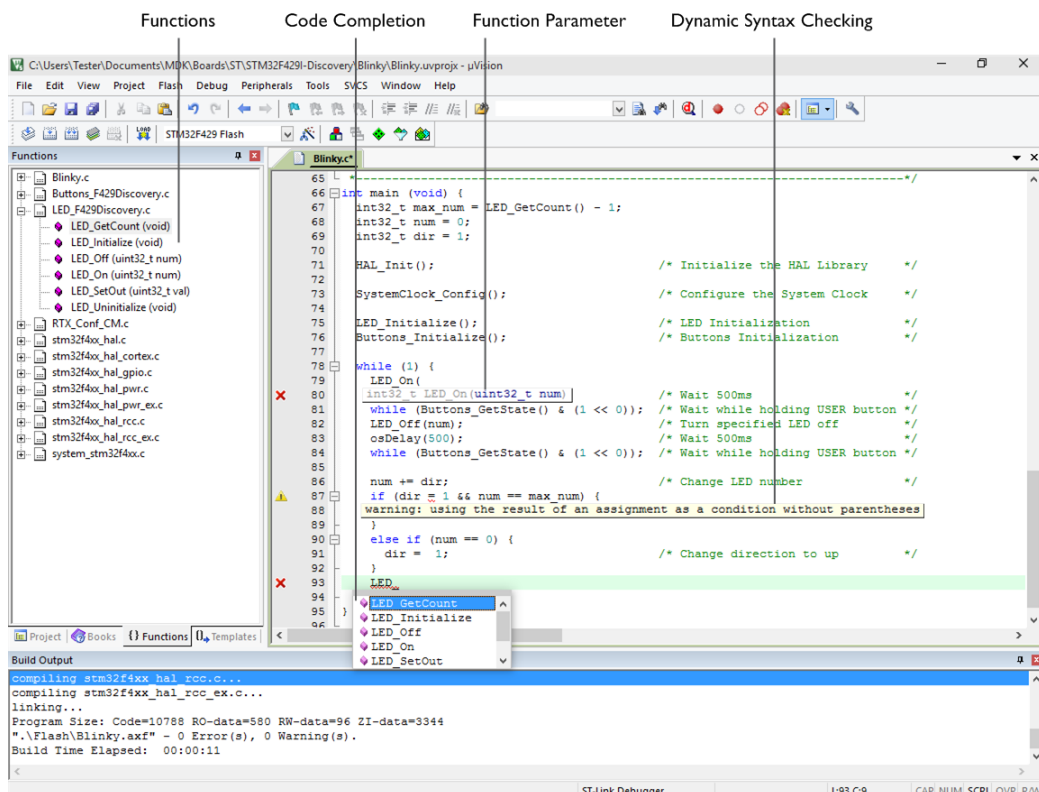


Figura 7: IDE uVision

Para ejecutar un programa en un microcontrolador uVision necesita saber qué microcontrolador se está programando y la forma en que desea

configurar los ajustes de la herramienta. Estas configuraciones se guardan en la configuración del proyecto.

El código fuente del proyecto se escribe en C/C++ y cuando este código se compila en su correspondiente compilador para C o C++, y se comprueba que no posee errores de compilación, se puede traducir y programar en el microcontrolador de acuerdo a las configuraciones del proyecto.

La salida del compilador será un archivo de tipo objeto que se compile y estos mismos se mandan al enlazador que los enlazará de acuerdo a los archivos DLL(Data Definition Language) y generará un archivo ejecutable que puede ser usado para poder programar el microcontrolador o en su defecto realizar simulaciones si no se posee la placa o si quiere comprobar ciertas entradas de datos que pueden ser algo más laboriosas de introducir en los dispositivos físicos.

Si todos los pasos se han realizado correctamente y sin ningún problema entonces una vez programado el microcontrolador este se podrá controlar desde el mismo IDE, gracias al JLink, usando el modo de depuración del mismo.

El modo de depuración de uVision permite trabajar en 2 diferentes modos, el "modo de simulación", en el cual el mismo IDE se encargará de simular las entradas y salidas del programa sin la necesidad del dispositivo físico, lo cual permite probar la lógica básica del software sin necesidad del microcontrolador mismo, y el "modo de objetivo", en el cual se cargará el programa al dispositivo físico y se podrá terminar de comprobar que el código programado funciona correctamente, revisar el funcionamiento paso a paso del código para comprobar que no tenga fallos de ejecución y luego ya entregar el producto final.

El archivo que uVision envía al dispositivo JLink y que luego este utiliza para programar el microcontrolador utiliza las herramientas del compilador de ARM que constan de:

- **Armcc:** Compilador ARM de C/C++
- **Microlib:** Módulo de código(en forma de biblioteca) de C/C++ altamente optimizada para ser utilizada en los microcontroladores programables con uVision
- **Armasm:** Assembly-Macro de ARM que se encargará de generar todos los archivos objetos para los distintos archivos de código C/C++
- **ArmLink:** El enlazador de ARM que se encargará de juntar todos los archivos objetos en un solo archivo para programar el microcontrolador

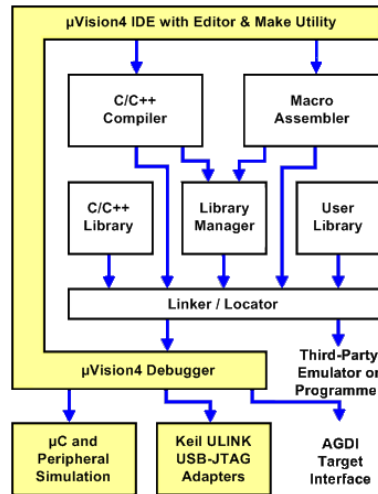


Figura 8: Ciclo de ciclo de compilación y programación de los microcontroladores de uVision

2.3. Situación del laboratorio LPC4088

La situación actual en la que se encuentran los laboratorios de la Escuela Superior de Ingeniería donde se realiza las tareas asignadas al microcontrolador LPC4088 es tal que la siguiente:

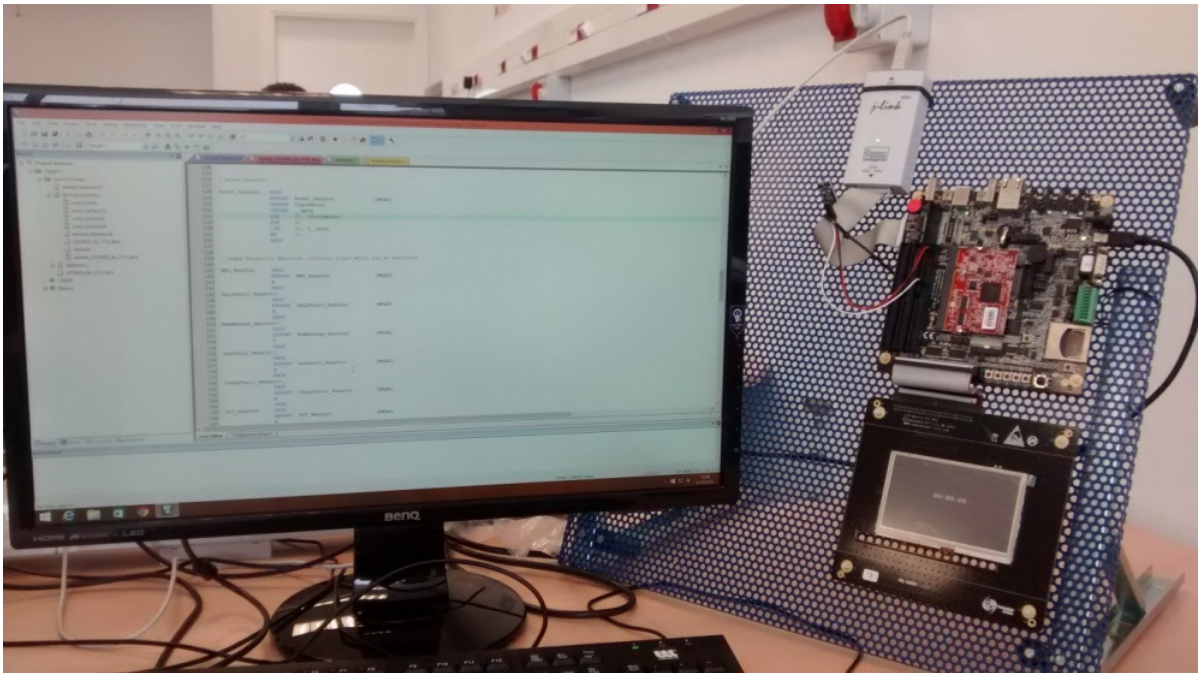


Figura 9: Laboratorio LPC4088

En esta imagen podemos apreciar que el microcontrolador, LPC4088, junto al dispositivo Jlink y otros aparatos físicos conectados al mismo, en este caso una pantalla LCD táctil resistiva, se encuentran en una plataforma voluminosa y conectada directamente a los ordenadores del laboratorio, lo cual obliga a que si se quiere trabajar con este microcontrolador se tenga que hacer de forma local en los ordenadores del

laboratorio, que cuentan con el entorno de desarrollo de Keil uVision.

El dispositivo JLink⁶ se encuentra conectado a este equipo por medio de un cable USB y al kit de desarrollo del microcontrolador⁴ mediante el puerto JTAG⁵, mediante el cual se programa la memoria FLASH para que así pueda ejecutar el programa correspondiente. Además el kit de desarrollo del microcontrolador se encuentra conectado por un cable USB también al ordenador del laboratorio para proporcionar la alimentación necesaria al microcontrolador y al kit de desarrollo.

Mientras se usa de forma local el microcontrolador junto al kit de desarrollo es importante saber que el orden de conectar y desconectar los cables es importante para poder así mantener correctamente el funcionamiento de los dispositivos físicos:

Para conectar los cables se realiza en el siguiente orden:

- Primero se conecta el dispositivo JLink al ordenador.
- Segundo se conecta el kit de desarrollo al ordenador para proporcionar la alimentación.

Para desconectar los cables se realiza en el siguiente orden:

- Primero se conecta el kit de desarrollo al ordenador para proporcionar la alimentación.
- Segundo se conecta el dispositivo JLink al ordenador.

3. El estado del Arte

Los laboratorios remotos que se encuentran actualmente en funcionamiento por todo el mundo actualmente se apoyan en el uso de internet y nos encontramos que ofrecen sus servicios mediante una arquitectura cliente-servidor a través de la web.

El servidor se encarga de recoger las peticiones que todos los clientes envían desde sus respectivos navegadores y distribuirlas entre los distintos equipos que están conectados a los diferentes dispositivos físicos, como pueden ser Arduino, microcontroladores ARM como la LPC4088, FPGAs, etc...

A continuación se detallan algunos de los proyectos realizados hasta la actualidad en el ámbito de los laboratorios remoto:

3.1. Weblab-Deusto



Figura 10: Logo Weblab Deusto

Es uno de los laboratorios remotos actuales mas famosos y que se mantienen desde un tiempo y que tiene a su vez una gran variedad de productos que oferta para así poder trabajar con ellos, los cuales son programación de robots con Arduino, FPGAs, laboratorio virtual de circuitos eléctricos, máquinas virtuales, ...

Según la propia pagina de WebLab Deusto[1]:

¿Qué es WebLab-Deusto? “WebLab-Deusto es una iniciativa de la *Universidad de Deusto* que apunta a incrementar la experiencia de aprender mediante el uso y desarrollo de los laboratorios remoto. Para este fin varios laboratorios son ofrecidos de forma gratuito mediante internet, el software requerido esta disponible bajo una licencia Open Source y el equipo puede ser replicado.”

WebLab-Deusto es usado por distintas instituciones de educación de distintos grados, como serian facultades de ingeniería o institutos de educación secundaria de diferentes países a través de todo el mundo, de igual forma también es empleado en múltiples proyectos de investigación. Su software esta disponible en 10 idiomas.

¿Para qué sirve?

- **Acceder a laboratorios existentes:** WebLab-Deusto utiliza un sistema que llaman wCloud que pretende que los profesores afiliados

al mismo puedan registrar sus centros y emplear los laboratorios remotos disponibles en sus clases. Este método no requiere de ninguna instalación de software o hardware adicionales, solamente de tener acceso a un navegador web y a la página web de WebLab-Deusto, pero si no se desea esto siempre se puede descargar un sistema software proporcionado por ellos para poder utilizar sus servicios.

- **Crea tu propio laboratorio:** El software de WebLab-Deusto puede ser utilizado para poder crear un laboratorio remoto, si dispone del hardware necesario, y alojarlo en su página web, de esta forma WebLab-Deusto proporciona también la lógica necesaria para la autenticación de usuarios, la autorización de acceso e integración de sistemas de aprendizaje como sería Moodle.
- **Adquiere laboratorios:** El equipo de WebLab-Deusto ha diseñado y desarrollado WebLab-Box, que es un armazón industrial que contiene lo necesario para montar de forma sencilla un laboratorio electrónico que posteriormente puede ser añadido a su repositorio de laboratorios remotos en WebLab-Deusto o LabsLands también llamado ahora.

Los laboratorios de los que consta actualmente WebLab-Deusto son:

- **Robot:**

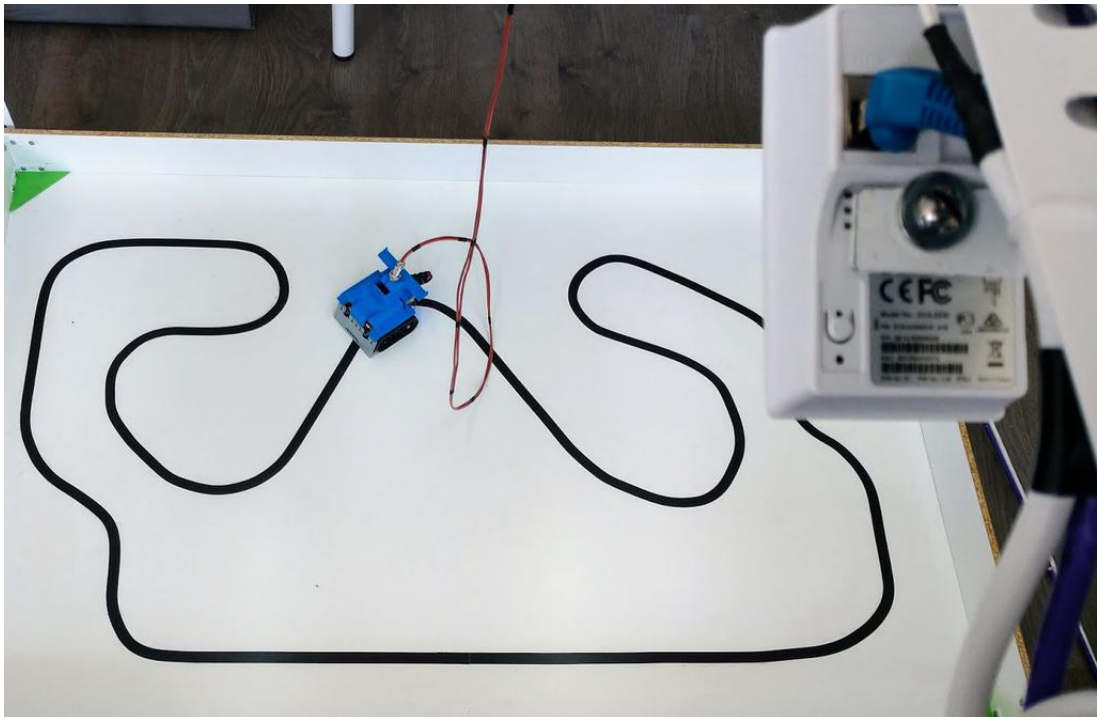


Figura 11: WebCam Arduino de WebLab-Deusto(LabsLand)

Laboratorio remoto en el cual puedes programar el microcontrolador Arduino de un pequeño robot que se encuentra en un laboratorio real, físico, empleando el lenguaje de programación propio

de Arduino o un lenguaje de programación visual por bloque, tipo Scratch, luego puedes cargar el programa en el robot controlado por el microcontrolador Arduino y ver como funciona mediante una webcam situada en el mismo laboratorio.



Figura 12: IDE Arduino de WebLab-Deusto(LabsLand)

- **FPGA:** Laboratorio remoto que permite trabajar a distancia con una Field Programmable Gate Array (FPGA), a través del software de Xilinx, mediante el cual puedes escribir un programa FPGA como lo harías normalmente. Una vez que el programa esté compilado sin errores, y listo para ser probado, simplemente debes subir el archivo binario “.bit” utilizando la interfaz web.

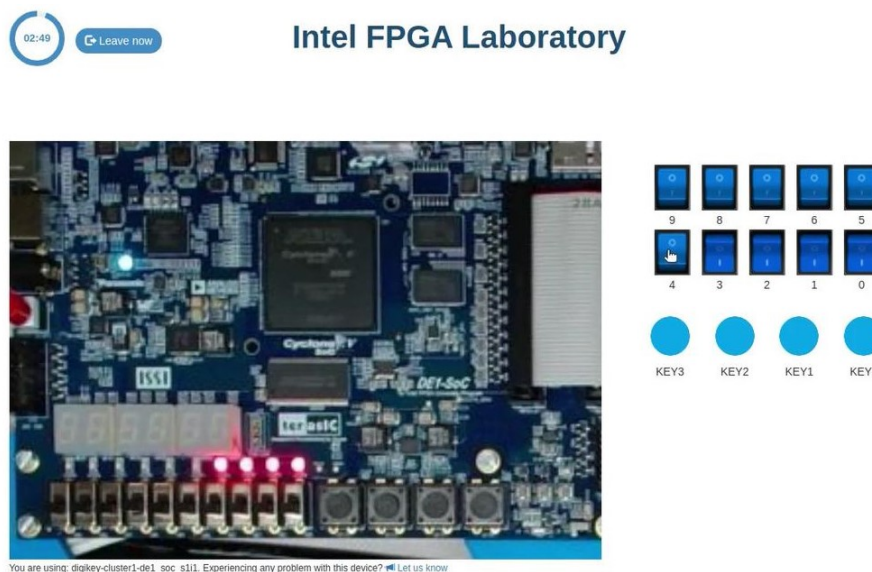


Figura 13: Interfaz FPGA de WebLab-Deusto(LabsLand)

En el caso del IDE se ve igual que la versión de Arduino¹²

- **VISIR:** El laboratorio remoto que ellos llaman VISIR (Virtual Instrument System In Reality) es un laboratorio remoto en el cual el usuario puede realizar diferentes circuitos de pruebas utilizando distintos componentes electrónicos básicos (resistencias, diodos, ...) y simularlo, de tal forma que no hay riesgo de estropear ningún componente, puedes montar el circuito sin tener los componentes para que cuando funcione sepas que componentes necesitas, además de que poder modificarlo de forma sencilla simplemente con unos clicks.

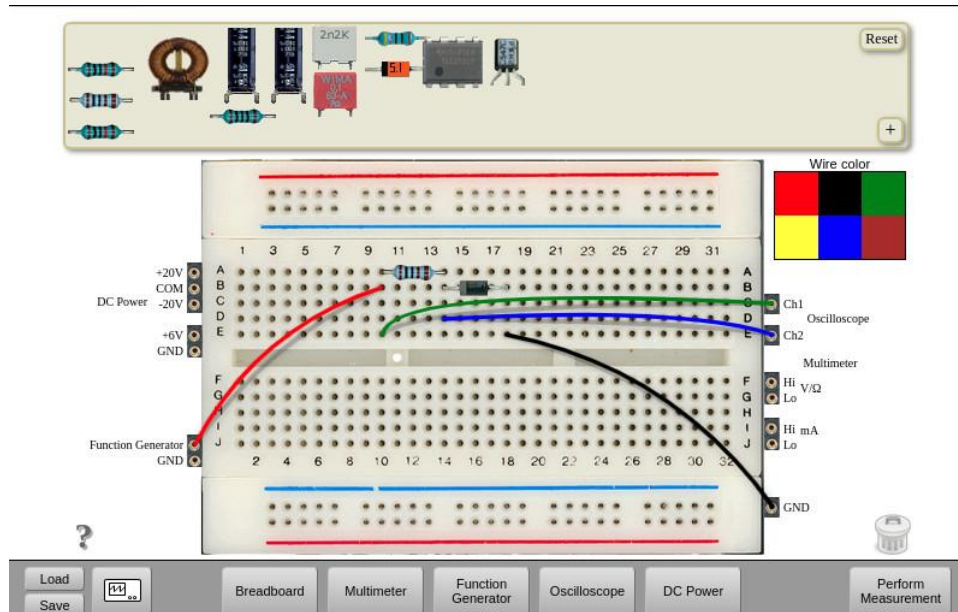


Figura 14: Interfaz VISIR de WebLab-Deusto(LabsLand)

Además de esto WebLab-Deusto consta de varios laboratorios físicos más que trabajan con distintos componentes, como acuarios, plantas con sensores térmicos,... que son externos a ellos pero que se pueden acceder desde su plataforma [LabsLand](#).

Como vemos WebLab-Deusto es una enorme plataforma que abarca una gran cantidad de opciones, todas gratuitas y Open Source que se lleva desarrollando desde el año 2000, lo cual les ha permitido ir creciendo a un ritmo constante.

En nuestro caso como se quiere intentar depender lo mínimo posible de fuentes externas para poder mantener y modificar nuestro producto todo lo necesario usaremos WebLab-Deusto como una de fuentes de inspiración y referencia, pero no usaremos su software para desarrollar nuestro laboratorio remoto.

En cuanto a las tecnologías que utiliza WebLab-Deusto se basa en el uso de Python para programar su servidor y los equipos conectados a

los microcontroladores, una base de datos que utiliza SQLAlchemy como herramienta de gestión de la base de datos desde Python, mientras que la base de datos en si misma utiliza MySQL por ser un sistema robusto y que permite varios procesos en paralelos, también emplean para cosas puntuales SQLite, que a pesar de trabajar como MySQL solo permite 1 acceso simultaneo a la base de datos pero es mas ligera y veloz en casos puntuales, el protocolo HTTP, apoyado con JSON, para la comunicación entre el cliente y el servidor. Además de poseer un sistema de reservas para asegurar que cada usuario pueda disponer de su propio tiempo con los dispositivos físicos con los que desee trabajar.

Ventajas	Desventajas
Robustez	Dependencia de la plataforma
Variedad de laboratorios con los que trabajar	Inflexible a la hora de configurar entornos personalizados si usas sus laboratorios
Puedes usar su software para hacer tu propio laboratorio	Complejidad para montar tu propio laboratorio
Sistema de reservas	

Cuadro 1: Tabla resumen de WebLab-Deusto

3.2. Educon2019

También conocida como IEEE Global Engineering Education Conference.

Con este apartado veremos que no todos los laboratorio remotos actuales son de tan amplia arquitectura como es el laboratorio de WebLab-Deusto. Durante la conferencia que se dio en 2019 se presentó un proyecto que constaba de un laboratorio remoto con el cual se puede trabajar con Arduino y que usaba una Raspberry PI como servidor central.

Al utilizar una Raspberry PI como servidor central la cantidad de peticiones simultaneas y de almacenamiento de servidor se ven algo mas limitadas que teniendo un equipo actuando de servidor o un servidor por derecho haciendo esa misma labor, pero igualmente es funcional, puesto que su objetivo es dar acceso a una cantidad limitada de dispositivos hardware y no a una mega infraestructura que comprende distintos laboratorios de todo el mundo.

La arquitectura de este laboratorio remoto se expreso que era de la siguiente forma:

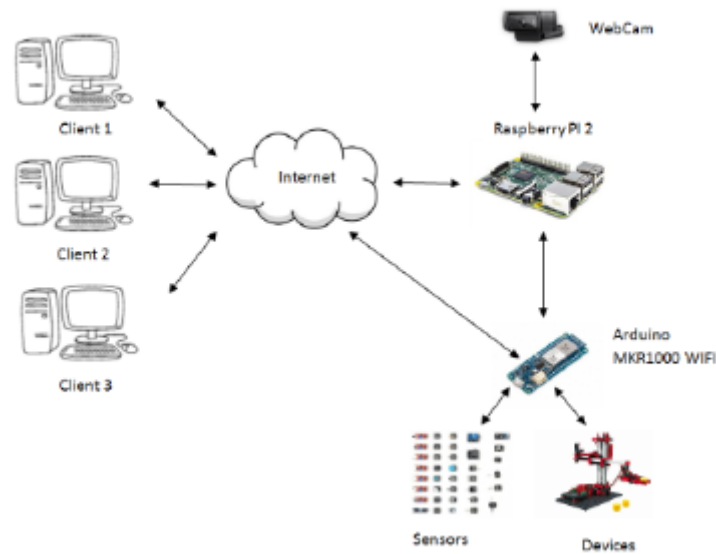


Figura 15: Laboratorio remoto con Arduino y RaspberryPI de la EDU-CON2019 de IEEE.

La interfaz web que emplea el servidor principal de este laboratorio remoto se ha escrito de forma sencilla utilizando PHP para el Backend y HTML, CSS y JavaScript para el Frontend, y como podemos ver en la siguiente figura¹⁶ es una interfaz web sencilla que simplemente muestra una webcam y una ventana de interacción con el laboratorio remoto, en ese caso para poder comprobar la conexión y poder subir archivos al mismo.

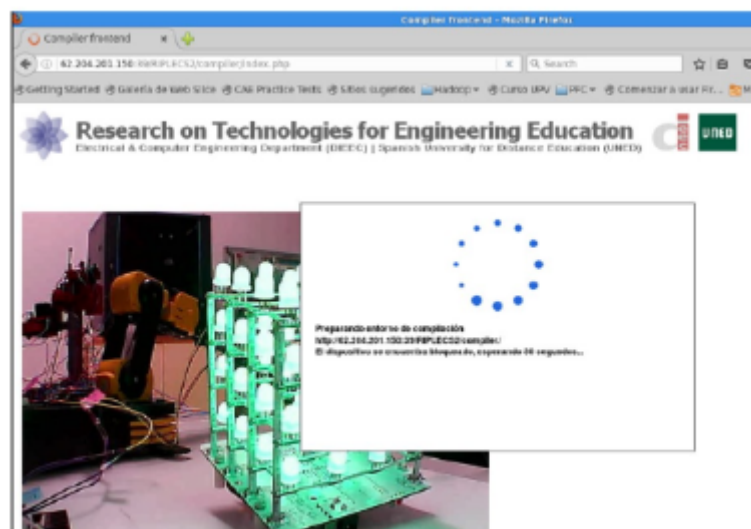


Figura 16: Interfaz del laboratorio remoto con Arduino y Raspberry PI de la EDU-CON2019 de IEEE.

Como podemos ver este laboratorio es bastante mas modesto que toda la red de laboratorios y software que posee WebLab-Deusto, pero aun así hace su función, de forma mas limitada que otros laboratorios remotos

pero hace la función para la que se diseñó, pero igualmente contiene sus desventajas, por ser un proyecto más modesto que otros.

Ventajas	Desventajas
Sencillez Velocidad de construcción Facilidad de añadir nuevos dispositivos	Seguridad Escalabilidad Solamente 1 “servidor” central

Cuadro 2: Tabla resumen del laboratorio remoto presentado en la EDUCON19.

3.3. MIT iLab

Proyecto original del MIT que se encargaba de conectar distintos laboratorios físicos a través de internet con los alumnos para poder enriquecer la enseñanza de la ciencia y la ingeniería ampliando en gran medida la gama de experimentos con los que sus alumnos pueden trabajar durante su formación académica. Pero tristemente terminó sus servicios en 2019, aunque ahora se encuentra asociado a otros laboratorios remotos y virtuales en otra plataforma.

Los encargados de este proyecto instalaron en el MIT distintos laboratorios remotos sobre microelectrónica, ingeniería química, cristalización de polímeros, ingeniería estructural y procesamiento de señales como casos de estudio para comprender los complejos requisitos de funcionamiento de los distintos tipos de experimentos y ampliar su uso a grandes grupos de estudiantes en el MIT y en todo el mundo.

La arquitectura compartida del iLab del MIT se ve de la siguiente forma:

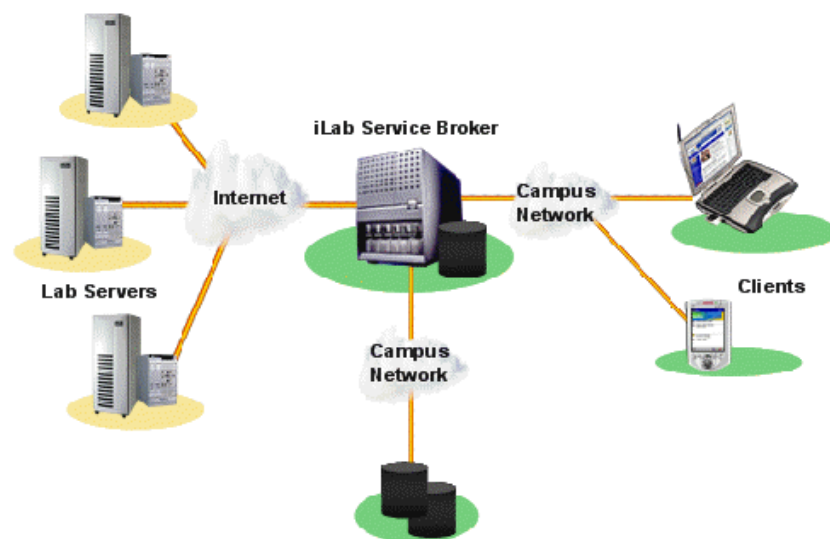


Figura 17: Arquitectura del ilab del MIT

La arquitectura del proyecto de MIT iLab se podía separar en tres módulos distintos conectados mediante el protocolo de servicios web.

- **El servidor de laboratorio** es operado por el propietario del laboratorio y se conecta con el hardware del laboratorio y se encarga de su funcionamiento.
- **El cliente de laboratorio** se ejecuta en el ordenador del usuario final y proporciona la interfaz web que utilizara el mismo para trabajar con el laboratorio.
- **El Service Broker** es el encargado de la comunicación entre el cliente del laboratorio y el servidor de laboratorio y proporciona servicios de almacenamiento y administración que es utilizado por toda la infraestructura de forma compartida dentro de una misma universidad.

Como en estos momentos se encuentra cerrado el servicio no se pondrá ninguna tabla resumen del mismo.

3.4. Remotely controlled laboratory de Universidad Palacký

Este es un proyecto realizado por la Universidad Palacký, situada en Chequia, y es la universidad más antigua de Moravia y la segunda universidad más antigua del país.



Univerzita Palackého
v Olomouci

Figura 18: Logotipo de la Universidad Palacký

Según la Universidad Palacký, un experimento remoto es todo aquel experimento real con instrumentos y equipos reales de laboratorio con el

cual se puede trabajar o controlar desde el ordenador personal de un profesor o estudiante a través de Internet sin tener que estar en el laboratorio físicamente.

La universidad Palacký tiene disponible sus distintos experimentos de forma gratuita en todo momento, las 24 horas del día los 7 días de la semana, y lo que buscan no es sustituir los experimentos clásicos hechos completamente de forma presencial, sino que lo que es poder ampliar la enseñanza de las ciencias y la ingeniería con nuevas alternativas.

Su arquitectura es definida por ellos de la siguiente forma:

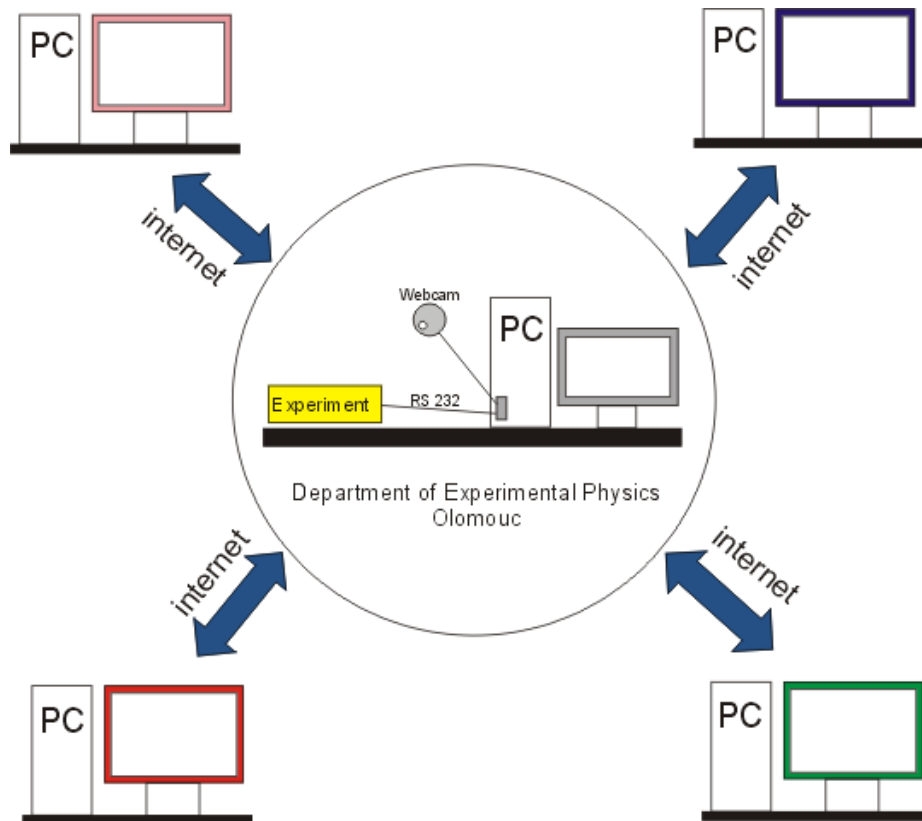


Figura 19: Arquitectura de los laboratorios remotos de la Universidad Palacký

En este tipo de arquitectura nos conectamos directamente a cada experimento y no a un servidor central que redirige nuestras peticiones e interacciones hacia el experimento en sí.

Actualmente la Universidad Palacký consta de 5 experimentos en su página y estos son:

- Volt-ampere characteristics of six different light bulbs
- Determination of acceleration of gravity g
- Study of water flow in the system of tubes
- Weather station in Olomouc
- Monitoring of radioactive background in Olomouc

Las ventajas que proponen que sus laboratorios controlados remotamente son:

Ventajas	Desventajas
Realización de experimentos desde cualquier parte del mundo	Laboratorios remotos completamente independientes
Sin restricciones de tiempo	Experimentos específicos y poco modificables en su mayoría
Rápido procesamiento gráfico de los valores medidos	Dependencia del tráfico del laboratorio para la devolución de resultados
Experimentos permanentemente listos para ser medidos	Un fallo en un componente implica el mal funcionamiento de todo el experimento
No hay peligro para los experimentadores durante los experimentos	

Cuadro 3: Tabla resumen de los laboratorios controlado remotamente de la Universidad Palacký



Figura 20: Experimentos de los laboratorios remotos de la Universidad Palacký

3.5. Recolab

Este proyecto apuesta principalmente por el uso de Matlab/Simulink para poder tanto ofrecer a sus usuarios la capacidad de controlar ciertos sistemas hardware instalados en sus laboratorios remotos, como el poder usar las capacidades de calculo de Matlab para poder realizar un laboratorio remoto simulado, donde todos los cálculos los realiza el servidor y envía el resultado al usuario sin pasar por otro equipo físico específico.

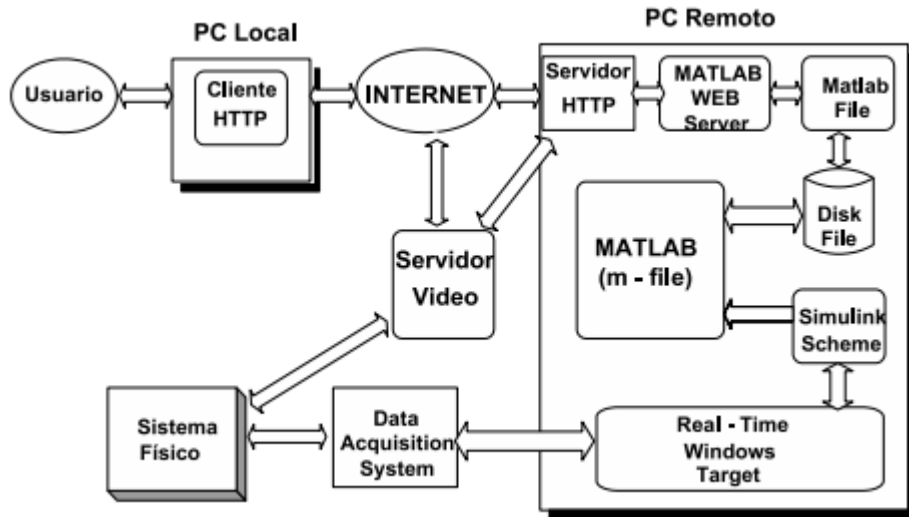


Figura 21: Arquitectura del servicio de Recolab

El funcionamiento interno del servicio de Recolab se puede resumir en 3 partes bien diferenciadas:

- **Comunicación con el cliente:** Utilizando la el protocolo HTTP mediante una pagina web programada con HTML, CSS, JavaScript, tanto para hacer peticiones como para recibir las respuestas del servidor.
- **Ejecución en el servidor:** Utiliza un servidor codificado en PHP que controla el acceso a los recursos y comunica la petición del cliente, recibida por el protocolo HTTP a las aplicaciones de Matlab/- Matlab Web Server/Simulink, y devuelve la respuesta el cliente con los resultados de sus experimentos utilizando el protocolo HTTP.
- **Ejecución en tiempo real:** Se ejecuta la simulación en el entorno de Matlab/Simulink o se codifica, programa y se ejecuta en un equipo hardware preparado para la tarea en específico que requiere el cliente.

Podemos ver este flujo de trabajo en la siguiente imagen ofrecida por Recolab[22]:

Como paso intermedio entre la comunicación del cliente y el servidor se encuentra una aplicación intermedia que se llama Matlab Web Server,

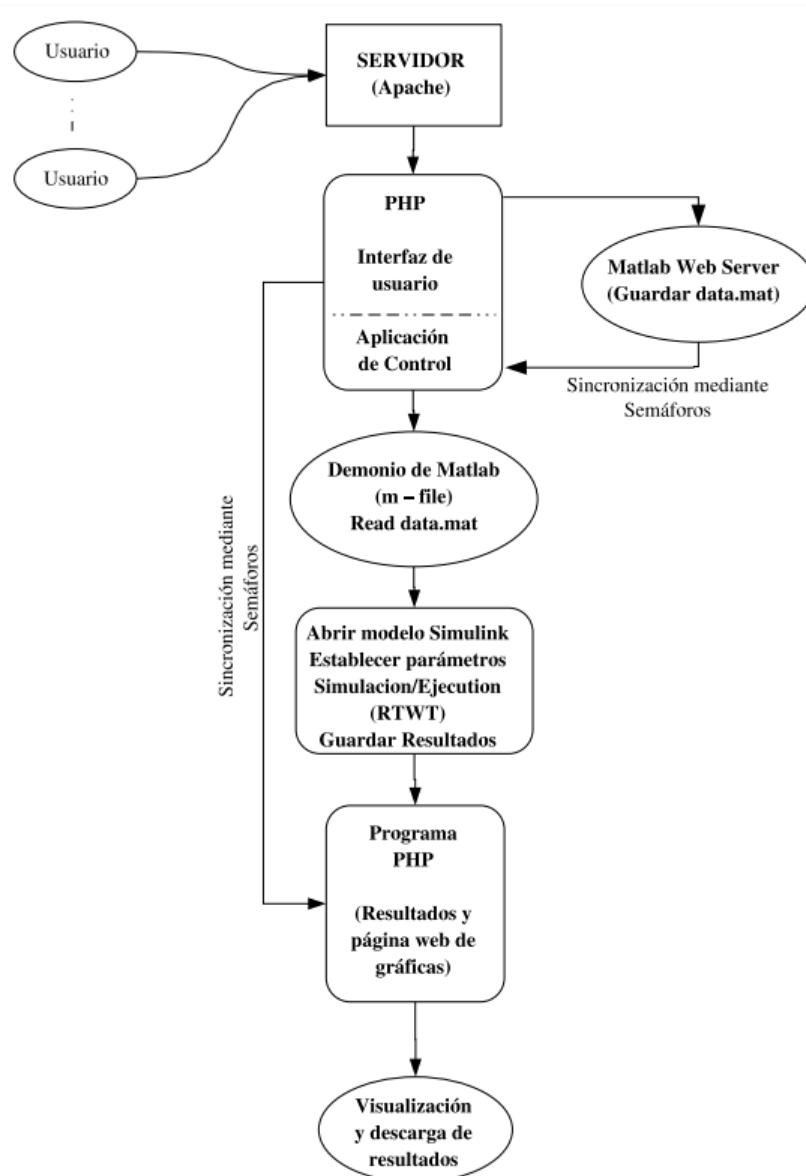


Figura 22: Flujo de trabajo del servicio de Recolab

que permite recopilar de forma sencilla los datos enviados por el cliente y los generados por Matlab, para así poder posteriormente tratarlos y enviárselos de forma correcta a la contra parte del sistema (los datos del cliente a Matlab y viceversa), además permite generar gráficas con los datos recibidos de Matlab y así poder mostrar estos datos de una forma mas visual al cliente.

Para realizar las tareas respectivas a la ejecución en tiempo real de parte del servidor se utilizan distintas herramientas como pueden ser Matlab, Simulink Real-Time Workshop y Real-Time Windows Target. Estas dos herramientas permiten que los datos generados por Simulink puedan ser enviados a un sistema de adquisición de datos (Real-Time Workshop) y por también a un equipo hardware conectado al mismo, para que así este sistema pueda ejecutarlo correctamente (Real-Time

Windows Target).

Aunque por desgracia el software no posee un medio de comunicación eficiente y directo entre Matlab Web Server y la sesión de Matlab que contiene los datos generados por la ejecución en tiempo real. Remarcamos que aunque Matlab Web Server permite ejecutar programas de Matlab no puede ejecutar esquemas Simulink mediante las herramientas Real – Time Workshop y Real – Time Windows Target. Por esto mismo es necesario establecer un mecanismo de comunicación entre una sesión de Matlab (que ejecuta en tiempo real el regulador elegido sobre el sistema físico) y Matlab Web Server.

Para solucionar este problema de logística se ha establecido una sincronización y transferencias de datos mediante una aplicación codificada en PHP que realiza esta labor mediante un sistema de ficheros.

Otro detalle que se debe remarcar sobre RecoLab es que solamente permite el acceso a trabajar con los dispositivos físicos a un solo usuario de forma simultanea, por lo que se encarga de que ningún otro usuario mas pueda acceder a este servicio mientras se esta empleando, pero si permite que varios usuarios usen de forma simultanea el servicio de simulación, Simulink, que no emplea ningún dispositivo físico.

4. Planificación

4.1. Metodología

Durante la realización de este proyecto se ha enfocado más en seguir una metodología del tipo ágil la cual tiene un enfoque iterativo en donde se divide el proyecto en distintos componentes y se van programando y entregando al cliente para así poder recibir su feedback, esto permite la evolución del proyecto de una forma mas cercana a la que el cliente tiene en su mente a pesar de los posibles cambios que puedan surgir a los requisitos del proyecto durante su desarrollo.

Concretamente, se ha optado por la metodología de desarrollo ágil SCRUM. Esta es una metodología de trabajo iterativa e incremental basada en la división y desarrollo de funcionalidades que se verifican y corrigen conforme se avanza en la consecución del proyecto reduciendo así las posibilidades de cometer errores.

Esta metodología de desarrollo permite que durante el proyecto se puedan ir probando los distintos bloques que componen el mismo de forma individual para poder así arreglar cualquier fallo o problema que surja sin llegar a tener que hacer un prototipo de todos los bloques del proyecto, con los costes de tiempo que puedan suponer esto.

4.2. Fases del proyecto

En el siguiente apartado del documento se especificarán las distintas fases que hubieron durante el desarrollo del proyecto y se indicará el diagrama de Gantt de la planificación del proyecto.

4.2.1. Fase de especificación de proyecto

Durante esta fase se estuvo hablando con los tutores para ver que proyecto desarrollar y que dirección debe tomar el mismo.

4.2.2. Investigación y búsqueda de información

Durante esta fase se ha estado estudiando los distintos laboratorios remotos que se han ido desarrollando en el tiempo para poder analizar como están desarrollados y tomar referencias de cómo desarrollar el proyecto.

También se ha estado estudiando las diferentes tecnologías disponibles para el desarrollo del proyecto y se ha decidido por cuál de todas ellas decantarse a la hora del desarrollo.

4.2.3. Implementación de la solución

Durante esta fase se ha implementado todo el software necesario para el alcance de la solución propuesta. Esta fase abarca desde la creación de la base de datos, a la implementación de la interfaz del usuario, del Backend del servidor, como los Scripts necesarios para poder comunicar con el servidor principal los equipos conectados a los microcontroladores.

4.2.4. Diagrama Gantt

En esta sección se muestra el diagrama de Gantt con el transcurso completo del proyecto.

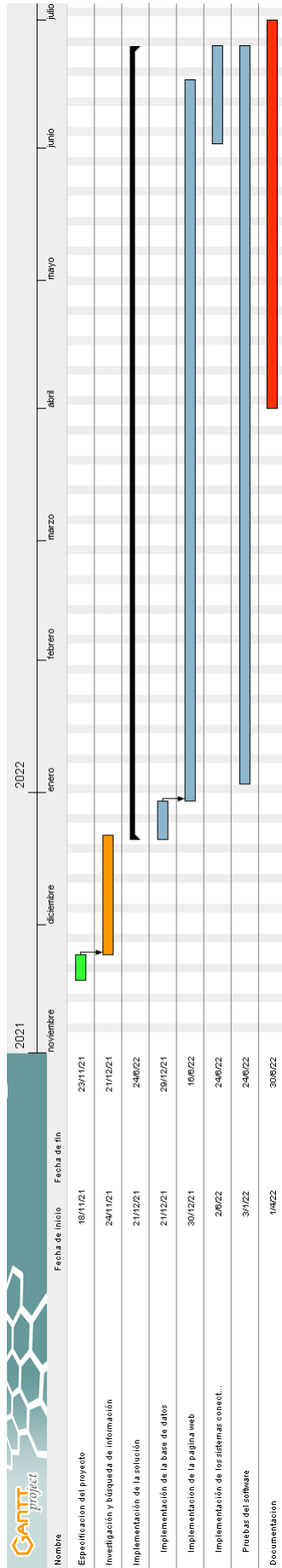


Figura 23: Diagrama de Gantt

4.3. Costes

En esta sección se hablarán de los costes implicados durante el desarrollo del proyecto, tanto humanos, materiales, como asociados a los servicios necesitados durante el mismo.

4.3.1. Costes de personal

El personal involucrado durante este proyecto consta de una sola persona y la experiencia de ese trabajador es considerado junior, por lo que tendría un salario mensual de aproximadamente 1.500€, el desarrollo ha abarcado un total de 7 meses por lo que el cuadro de costes de personal quedaría de la siguiente forma:

Concepto	Precio	Tiempo(en meses)	Total
Salario	1500€	7	10500€

Cuadro 4: Costes de personal del proyecto

4.3.2. Costes Materiales

No ha sido necesario la compra de nuevos materiales para la realización de este TGF, por lo cual no es necesario añadir ningún coste de material al análisis de los costes del trabajo.

4.3.3. Costes asociados a servicios

En este apartado se incluyen los costes asociados a servicios de terceros, como son el acceso a internet, el uso de electricidad y el transporte al lugar de trabajo, como la duración del proyecto ha sido de siete meses, debemos de tener en cuenta los gastos causados durante este intervalo de tiempo.

Concepto	Precio	Tiempo(en meses)	Total
Internet/mes	50€	7	350€
Electricidad/mes	45€	7	315€
Trasporte/mes	32€	7	244€
Total			879€

Cuadro 5: Costes asociados a servicios del proyecto

4.4. Tecnologías/Herramientas utilizadas

Durante este apartado se indicarán las distintas herramientas o tecnologías que se usaron durante el desarrollo del proyecto.

4.4.1. Herramientas de redacción de textos

Para el desarrollo de esta documentación se ha utilizado la plataforma de OverLeaf que permite la escritura de un documento científico con la utilización de LaTeX, que es un sistema de composición de textos, orientado a la creación de documentos escritos que presenten una alta calidad tipográfica. Por sus características y posibilidades, es usado de forma especialmente intensa en la generación de artículos y libros científicos que incluyen, entre otros elementos, expresiones matemáticas.



Figura 24: Logo de OverLeaf y de LaTeX

4.4.2. Herramientas usadas durante el desarrollo

Durante el desarrollo del proyecto se utilizó principalmente el editor de código de Visual Studio Code, el cual es un editor de código muy ligero y personalizable que te permite añadirle cualquier funcionalidad con extensiones que te descargas desde el mismo editor y que además permite poder trabajar perfectamente con un Workspace que abarque todo el proyecto base.

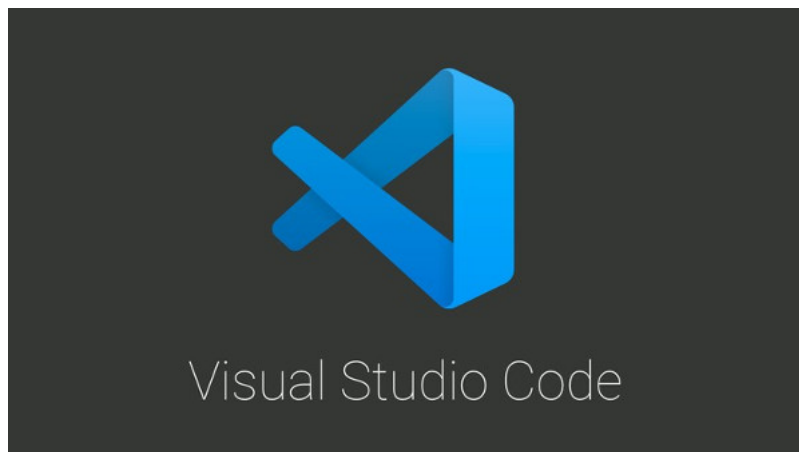


Figura 25: Logo de Visual Studio Code

Para programar y probar los programas en Arduino se utilizó el IDE (Entorno de Desarrollo Integrado) específico de Arduino que permite una programación rápida del microcontrolador, también gestionando de forma cómoda y sencilla que tipo de microcontrolador Arduino vas a

programar y el tipo de conexión tanto para poder subir el archivo a la memoria Flash de Arduino como poder ver la entrada/salida del puerto serie donde este conectado el microcontrolador Arduino.



Figura 26: Icono del IDE de Arduino

Para poder programar el microcontrolador LPC4088 se utilizará el IDE Keil uVision que sera el middleware (lógica de intercambio de información entre aplicaciones) usado para programar o compilar código C/C++ en el microcontrolador. Es creado por la empresa Keil y la versión utilizada actualmente es la versión 5 del entorno de desarrollo.



Figura 27: Icono del IDE de Keil uVision

Para poder conectar el IDE de Keil uVision con el microcontrolador LPC4088 de forma remota desde fuera de la red local usaremos el programa de Remote Server que ofrece SEGGER.

4.4.3. Hardware empleado

Para la realización del proyecto se han empleado distintos microcontroladores que son comentados con anterioridad en el apartado de [Situación Inicial](#) como pueden ser:

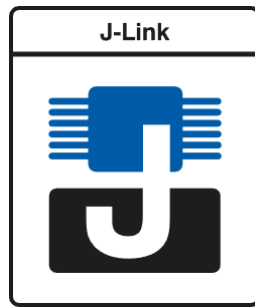


Figura 28: Logo de Jlink

- **Arduino:** Que como se comento antes Arduino es una plataforma de creación de electrónica de código abierto, la cual está basada en hardware y software libre, flexible y fácil de utilizar para los creadores y desarrolladores.

Durante el desarrollo del proyecto se han empleado tanto el modelo de Arduino MEGA[2] cuyas características se describen de mejor manera en [este apartado](#) que habla sobre el Arduino en la sección del estado inicial al proyecto.

A su vez también se utilizara dependiendo de si es el laboratorio remoto de la LPC4088 o del Arduino MEGA, un el Arduino Due o el Arduino Pro Micro junto a una PCB diseñada por otro alumno llamado Kick Cornelis Ouwens, que se encargará de permitir que el usuario pueda conectar los distintos sensores o actuadores a cualquier pin de Arduino o de la LPC4088 mediante un sistema de multiplexación, actuando de controladora del laboratorio remoto. Este modelo del microcontrolador Arduino cuenta con las siguientes

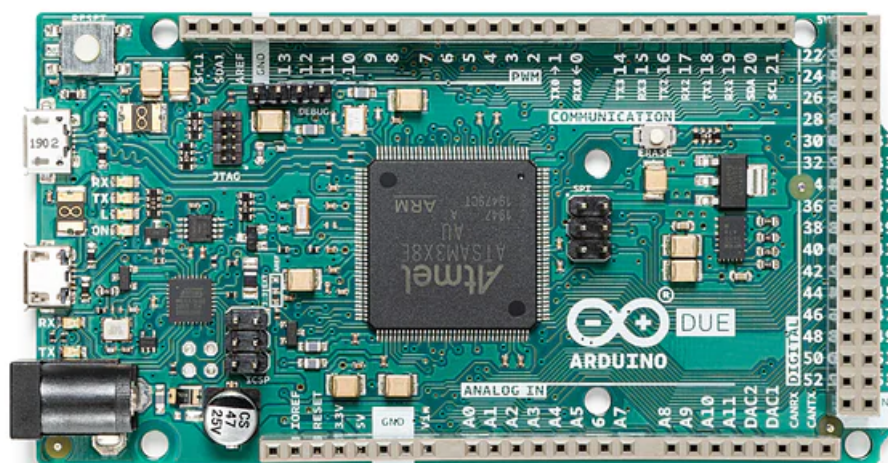


Figura 29: Placa Arduino Due

características:

- Microcontrolador: AT91SAM3X8E (32 bits ARM)

- Tensión de funcionamiento: 3.3V
- Alimentación recomendada: 7-12V
- Pines I/O Digitales: 54
- Canales PWM: 12
- Entradas analógicas: 12
- Corriente de salida de Corriente Continua total en todas las líneas I/O: 130 mA
- Corriente continua del pin 3.3V: 800 mA
- Memoria Flash: 512 KB
- SRAM: 96 KB (En dos bancos de memoria: 64KB and 32KB)
- Velocidad: 84 MHz

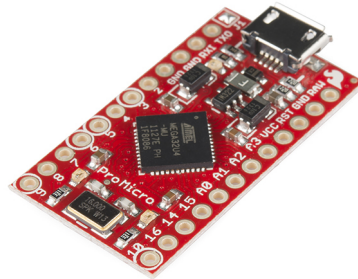


Figura 30: Placa Arduino Pro Micro de Sparkfun

Este modelo del microcontrolador Arduino cuenta con las siguientes características:

- Microcontrolador: ATmega32U4
 - Tensión de funcionamiento: 3.3V
 - 9x 10-bit ADC pins
 - Pines I/O Digitales: 12
 - Canales PWM: 5
 - Entradas analógicas: 4
 - Memoria Flash: 32 KB
 - SRAM: 2.5 KB
 - Velocidad: 8 Mhz
- **LPC4088[4]:** Que como se comento con anterioridad en [este apartado](#) es un microcontrolador ARM de 32 bits que utiliza la arquitectura Harvard, es decir que tiene la memoria de datos y de programa separadas físicamente. Además necesita de un dispositivo que es el JLink que se describe también en la sección indicada anteriormente.

4.5. Tecnologías software empleadas

Para la realización de este proyecto nos hemos centrado en desarrollarlo utilizando el lenguaje de programación Python, el cual es un lenguaje

de alto nivel de programación cuya filosofía hace hincapié en la legibilidad de su código mediante una sintaxis elegante y una asignación de tipos dinámica de variables, se utiliza para desarrollar aplicaciones de todo tipo, algunos ejemplos de estas aplicaciones pueden ser Instagram, Netflix, Spotify, entre otros.



Figura 31: Logotipo de Python

Python se trata de un lenguaje de programación multi-paradigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multi-plataforma.

Dentro del ámbito de Python se ha decidido a utilizar Flask para el desarrollo de la página web, el cual es un “micro” Framework escrito en Python y concebido para facilitar el desarrollo de Aplicaciones Web bajo el patrón Modelo-Vista-Controlador (MVC). El patrón MVC es una manera o una forma de trabajar que permite diferenciar y separar lo que es el modelo de datos (los datos que van a tener la App que normalmente están guardados en la base de datos), la vista (página HTML) y el controlador (donde se gestiona las peticiones de la app web). La palabra “micro” no designa a que sea un proyecto pequeño o que nos permita hacer páginas web pequeñas sino que al instalar Flask tenemos las herramientas necesarias para crear una aplicación web funcional pero si se necesita en algún momento una nueva funcionalidad hay un conjunto muy grande de extensiones (plugins o librerías) que se pueden instalar con Flask que le van dotando de funcionalidad.



Figura 32: Logo de Flask

Los módulos de Python y Flask que han sido empleados durante el desarrollo del proyecto son:

- `Flask.Login`[2]: Es una librería que provee la lógica básica para gestionar las sesiones de los usuarios, provee de forma sencilla de funciones que permitan manejar el acceso o login del usuarios, la desconexión o logout, y para gestionar correctamente la sesión del usuario durante largos periodos de tiempo.
- `Flask.SQLAlchemy`[3]: Esta librería se encarga de añadir soporte para SQLAlchemy apuntando a intentar simplificar el uso de SQLAlchemy, con Flask, aportando algunas funciones de ayuda para completar las tareas mas comunes que se puedan realizar y tambien aporta algunas ayudas extras para tareas menos comunes pero igualmente necesarias de realizar.
- `Flask.Migrate`[4]: Extensión de flask que maneja la gestión de la base de datos de SQLAlchemy, como por ejemplo añadir tablas, campos a la base de datos, etc., usando Alembic.
- `Flask.WTF`[5]: Esta librería integra la utilización de los WTForms en Flask, mientras incluye además soporte para los tokens de seguridad CSRF (Cross-Site Request Forgery).
 - `WTForm`[6]: Esta librería se encarga de aportar una forma flexible de validación y renderización de formularios en el desarrollo web con Python, puede ser empleado en prácticamente cualquier framework de Python y además soporta diferentes técnicas de seguridad como validación de datos, protección RSCF y más.
- `Bootstrap-Flask`[7]: Esta librería de Flask incorpora una gran variedad de macros de Jinja2 para poder emplear de forma sencilla Bootstrap junto con Flask y el motor de plantillas de Jinja2, además de permitir renderizar objetos relacionados con Flask de forma sencilla en la pagina HTML.
- `Flask-JSGlue`[8]: Esta librería de Flask permite la utilización de las funciones internas de Flask para el backend en el frontend, como por ejemplo en archivos de scripts de JavaScript.

- Werkzeug[9]: Librería que contiene una colección de varias utilizadas para aplicaciones WSGI y actualmente es una de las librerías de utilidad para WSGI mas avanzadas.
- Functools[10]: Es una librería diseñada para implementar funciones que actúan o devuelven otras funciones. En general cualquier objeto que se pueda invocar puede ser tratado como una función según esta librería. En nuestro caso se ha usado esta librería para diseñar decoradores personalizados para las funciones de las vistas.
- Flask-SocketIO[11]: Esta librería permite a Flask implementar una comunicación bi-direccional de baja latencia entre el servidor y cualquier otro dispositivo. Del lado del dispositivo no servidor se puede usar cualquier librería que implemente también la comunicación SocketIO, ya sea por Python, Java, C++ o cualquier lenguaje, en nuestro caso se ha implementado esta comunicación entre los laboratorios remotos y el servidor con Python-SocketIO de parte del laboratorio remoto.
- AIOHTTP[12]: Esta librería permite manejar de forma asíncrona las peticiones HTTP de parte del cliente o del servidor.
- Flask-Mail[13]: Esta librería proporciona una interfaz sencilla para poder implementar una conexión SMTP y enviar correos electrónicos desde el servidor principal.

Externamente para los scripts que se implementan en los laboratorios remotos para establecer la conexión con el servidor principal se han empleado los siguientes módulos:

- Python-SocketIO[14]: Esta librería implementa la lógica necesaria para poder emplear el protocolo de comunicación de SocketIO.
- Asyncio[15]: Librería que implementa la lógica necesaria para poder escribir código concurrente en Python.
- PySerial_Asyncio[16]: Esta librería incluye funciones que permiten a Python conectarse a un puerto serie y utilizarlo para comunicarse con otros dispositivos, y la versión Asyncio de esta librería permite hacerlo de forma concurrente.

Junto a Flask para ayudarnos durante el desarrollo de la parte Frontend del proyecto viene incluido Jinja2 que es un motor de plantillas web para el lenguaje de programación Python. Fue creado por Armin Ronacher y tiene una licencia BSD (Berkeley Software Distribution), la cual es una licencia de software libre permisiva, parecida a la licencia de OpenSSL o la MIT License, aunque tiene menos restricciones que otras licencias como por ejemplo la GPL (General Public License) como por ejemplo permitiendo el uso de su código fuente en software no libre. Jinja es un motor de plantillas que proporciona expresiones similares a las de Python, al tiempo que garantiza que las plantillas se evalúan en una caja de arena. Es un lenguaje de plantillas basado en texto y puede utilizarse para generar cualquier tipo de marcado, así como el código fuente.



Figura 33: Logo de Jinja

Jinja2 esta internamente basada en Unicode y se ejecuta en una amplia cantidad de versiones de Python desde la versión 2.5 hasta las versiones actuales, incluyendo las versiones de Python 3, y algunas de las características de Jinja2 son:

- Incluye internamente un sistema de escape de código de fuentes externas para que no ejecuten scripts no autorizados.
- La herencia de plantillas permite utilizar el mismo o similar diseño para todas las plantillas o páginas.
- Para proporcionar un alto rendimiento a la hora de cargar la página traducirá el código Python a código Byte al principio de la carga del mismo.
- Facilita la depuración integrando la compilación y ejecución del código de las plantillas según el sistema estándar de seguimiento de Python.
- Incluye pequeñas ayudas que sirven para resolver tareas comunes en las plantillas.

Para la base de datos se ha decidido emplear sistema de base de datos objeto-relacional llamado PostgreSQL, que usa internamente el lenguaje SQL combinado con otras características para ampliar sus funciones. PostgreSQL es un sistema de base de datos altamente probado y con gran reputación gracias a su fiabilidad, la integridad de sus datos y por una gran comunidad Open Source que busca continuamente arreglar los fallos que puedan surgir e ir mejorando continuamente su funcionamiento.



Figura 34: Logo de PostgreSQL

La razón para usar PostgreSQL es debido a que ofrece una gran cantidad de herramientas para los desarrolladores como sería la creación de tipos personalizados que no sean los básicos ofrecidos por la mayoría de sistema de base de datos, como también la implementación de funciones personalizadas que se ejecuten a nivel de la base de datos o hasta poder escribir código en diferentes lenguajes de programación sin tener que recompilar la base de datos.

PostgreSQL intenta seguir el estándar SQL siempre que dicho ajuste no contradiga las características tradicionales o pueda llevar a decisiones arquitectónicas erróneas, porque la mayoría de las características requeridas por el estándar SQL están soportadas, aunque a veces impliquen cambios ligeros en la sintaxis o funciones empleadas.

También PostgreSQL ofrece una gran compatibilidad con las herramientas de Python para el trabajo con bases de datos, permitiendo que la mayoría de las tareas básicas que debe realizar el software en conjunto con la base de datos se haga de forma eficiente, cómoda y fiable, sin tener que preocuparse de tener que realizar alguna tarea intermedia para poder adaptar los datos recibidos de la base de datos o para mandarlos de vuelta a la misma.

Para trabajar con PostgreSQL se ha decidido utilizar el software Open Source de PGAdmin en su 4 versión. PGAdmin es un software escrito con Python y JQuery, compatible con todas las funciones de PostgreSQL, que sirve para la administración y desarrollo de las bases de datos relacionadas del usuario, además de poder usarse para varias tareas como serían el redactar consultas básicas de SQL para la base de datos, hasta monitoriza las mismas y configurar las arquitecturas mas avanzadas de bases de datos.

Parte II

Desarrollo

5. Análisis del proyecto

5.1. Requisitos

Durante este apartado se expondrán los requisitos funcionales y no funcionales que tiene que cumplir el proyecto.

5.1.1. Requisitos funcionales

Los requisitos funcionales son aquellas funciones o servicios mínimos que debe ofrecer el producto final, y que son indicadas por el cliente.

RF-01	Permitir crear usuarios
RF-02	Poder verificar el correo de los usuarios creados
RF-03	Permitir modificar contraseña y email del usuario
RF-04	Poder modificar la contraseña si el email del usuario esta verificado
RF-05	Permitir añadir nuevos laboratorios remotos
RF-06	Mostrar PDF de prácticas cuando se trabaje con laboratorios remotos
RF-07	Gestionar que PDF de prácticas se mostraran en cada laboratorio remoto
RF-08	Sistema de reservas individual para cada laboratorio remoto
RF-09	Gestionar el acceso a cada laboratorio remoto
RF-10	Comunicar el servidor principal con los laboratorios remotos
RF-11	Visualizar en la interfaz web la webcam del laboratorio remoto
RF-12	Panel de comunicación entre la interfaz web y la placa de control del laboratorio remoto
RF-13	Subir archivos de código “.ino” al laboratorio remoto del microcontrolador Aduino MEGA
RF-14	Comunicar la interfaz web con el Arduino MEGA actuando como consola Serie para entrada y salida
RF-15	Permitir activar el protocolo de servidor remoto JLink del laboratorio remoto del microcontrolador LPC4088
RF-16	Diseñar panel de simulación de teclado y joystick

Cuadro 6: Tabla de requisitos funcionales

5.1.2. Requisitos no funcionales

Son aquellos requerimientos que no provienen de las funciones o servicios que debe prestar el producto final, sino de aquellas características emergentes del propio sistema.

RFN-01	Interfaz sencilla
RFN-02	Poder integrar todo lo posible de la interfaz del laboratorio remoto en una sola pestaña
RFN-03	Limitar las reservas a 3 horas por reserva
RFN-04	Limitar las reservas a 1 reserva pendiente
RFN-05	Debe estar disponible ininterrumpidamente
RFN-06	Debe ser capaz de responder rápidamente ante cualquier acción del usuario

Cuadro 7: Tabla de requisitos no funcionales

5.2. Casos de uso

En este apartado describiremos los casos de uso del sistema, los cuales describen las interacciones entre los actores, que se pueden identificar 2 tipos de actores, el usuario alumno y el usuario administrador (que tiene los mismos permisos que el de alumno y otros adicionales para gestionar la aplicación web), y el sistema.

5.2.1. Casos de uso respecto a la gestión de usuarios

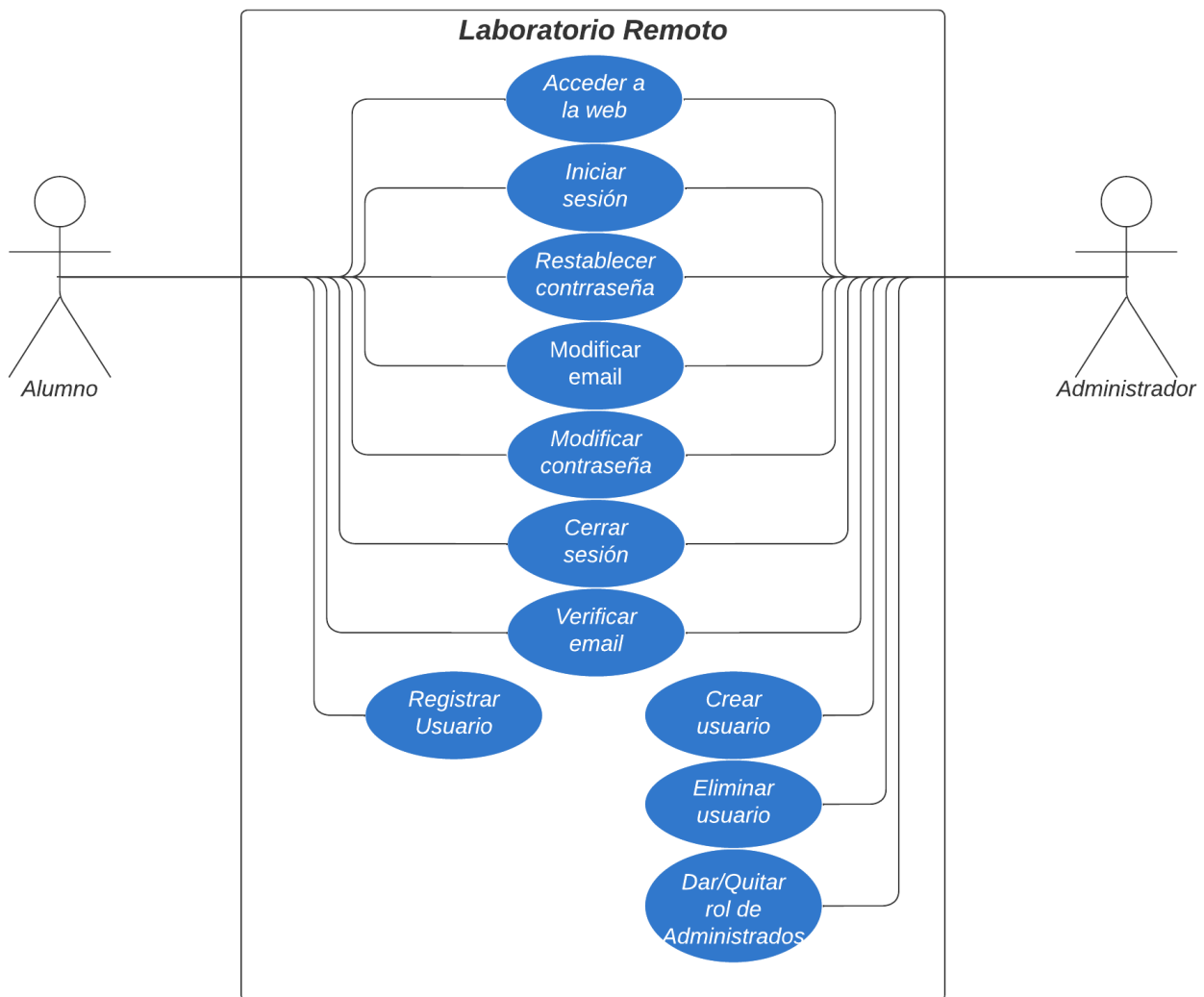


Figura 35: Casos de uso respecto a los usuarios

CU-001	Acceso a la web
Actores	Alumno y Administrador
Precondiciones	Ninguna
Escenario	El usuario introduce la url de la pagina web. El sistema redirige al actor a la pagina de iniciar sesión. El sistema muestra la vista de inicio de sesión
Postcondiciones	El usuario accede a la página de inicio de sesión

Cuadro 8: CU-001: Acceder a la web

CU-002	Iniciar sesión
Actores	Alumno y Administrador
Precondiciones	El usuario debe estar registrado en el sistema
Escenario	El usuario introduce las credenciales de inicio de sesión. El sistema valida si las credenciales tienen el formato correcto El sistema comprueba si existe un usuario con esas credenciales de acceso El sistema inicia una sesión con las credenciales de acceso para el usuario El sistema redirige al usuario a la vista asignada para su rol
Postcondiciones	El usuario ha iniciado sesión en el sistema
Escenarios alternativos	Formato incorrecto: ■ Las credenciales introducidas no siguen el formato correcto para los datos ■ El sistema muestra los errores de formato en las credenciales de acceso Credenciales de acceso incorrectas: ■ El sistema no encuentra ningún usuario que coincida con esas credenciales de acceso ■ El sistema indica al usuario que no existe ningún usuario con esas credenciales de acceso

Cuadro 9: CU-002: Iniciar sesión

CU-003	Recordar contraseña
Actores	Alumno y Administrador
Precondiciones	El usuario debe estar registrado en el sistema
Escenario	El usuario hace click en el enlace para restablecer contraseña El sistema carga la vista asignada al proceso de restablecer contraseña El usuario introduce las credenciales de acceso solicitadas El sistema busca un usuario con esas credenciales de acceso El sistema manda un correo electrónico para verificar el restablecimiento de la contraseña El usuario hace click en el enlace que se le envía por el correo El sistema carga la vista asignada para que el usuario introduzca la nueva contraseña El usuario introduce una nueva contraseña para su cuenta El sistema comprueba que las contraseñas sean las mismas El sistema modifica la contraseña del usuario en la base de datos El sistema inicia la sesión del usuario El sistema redirige al usuario a la vista asignada para su rol
Postcondiciones	El usuario ha modificado la contraseña de su cuenta
Escenarios alternativos	Formato incorrecto: ■ Las credenciales introducidas no siguen el formato correcto para los datos ■ El sistema muestra los errores de formato en las credenciales de acceso Credenciales de acceso incorrectas: ■ El sistema no encuentra ningún usuario que coincida con esas credenciales de acceso ■ El sistema indica al usuario que no existe ningún usuario con esas credenciales de acceso

Cuadro 10: CU-003: Restablecer contraseña

CU-004	Modificar email
Actores	Alumno y Administrador
Precondiciones	El usuario debe haber iniciado sesión en el sistema y encontrarse en la vista del panel de usuario
Escenario	El usuario selecciona la sección de modificar email del panel de usuario El usuario introduce el nuevo email que desea para su cuenta El sistema comprueba los datos introducidos por el usuario El sistema comprueba si el email del usuario esta verificado El sistema envía un correo electrónico para que el usuario confirme el cambio de email El usuario hace click en el enlace para confirmar el cambio de email El sistema modifica el email del usuario en la base de datos y lo notifica al usuario
Postcondiciones	El usuario ha modificado el email asociado a su cuenta
Escenarios alternativos	Formato incorrecto: ■ Las credenciales introducidas no siguen el formato correcto para los datos ■ El sistema muestra los errores de formato en las credenciales de acceso Email sin confirmar: ■ Si el email de la cuenta no se encuentra verificado entonces el sistema directamente cambiara el mismo en la base de datos sin necesidad de confirmar la acción de parte del usuario

Cuadro 11: CU-004: Modificar email

CU-005	Modificar contraseña
Actores	Alumno y Administrador
Precondiciones	El usuario debe haber iniciado sesión en el sistema y encontrarse en la vista del panel de usuario
Escenario	El usuario selecciona la sección de modificar email del panel de usuario El usuario introduce la contraseña de la cuenta y la nueva contraseña que desea para su cuenta El sistema comprueba los datos introducidos por el usuario El sistema envía un correo electrónico para que el usuario confirme el cambio de email El usuario hace click en el enlace para confirmar el cambio de contraseña El sistema modifica la contraseña del usuario en la base de datos y lo notifica al usuario
Postcondiciones	El usuario ha modificado la contraseña asociado a su cuenta
Escenarios alternativos	Contraseña antigua incorrecta: La contraseña actual introducida no es la misma que la que tiene la cuenta registrada en la base de datos y lo notifica al usuario

Cuadro 12: CU-005: Modificar contraseña

CU-006	Cerrar sesión
Actores	Alumno y Administrador
Precondiciones	El usuario ha iniciado sesión en el sistema
Escenario	El usuario hace click en la sección de la barra de navegación para poder cerrar sesión El sistema cierra la sesión del usuario El sistema redirige al usuario a la pagina de inicio de sesión
Postcondiciones	El usuario cierra su sesión de acceso

Cuadro 13: CU-006: Cerrar sesión

CU-007	Registrar usuario
Actores	Alumno
Precondiciones	El usuario no tiene cuenta creada en el sistema
Escenario	El usuario hace click en el enlace para registrarse en el sistema El sistema redirige al usuario a la vista de registrar nuevo usuario El usuario introduce los datos de la nueva cuenta El sistema comprueba que las credenciales sean validas El sistema comprueba si no hay usuarios que compartan esas credenciales de acceso El sistema crea el nuevo usuario, manda un correo para verificar el email asociado a la nueva cuenta y redirige al usuario a la vista de placas
Postcondiciones	El sistema crea un nuevo usuario e inicia la sesión del mismo
Escenarios alternativos	Formato incorrecto: ■ Las credenciales introducidas no siguen el formato correcto para los datos ■ El sistema muestra los errores de formato en las credenciales de acceso Ya existe un usuario con esas credenciales: ■ Si las credenciales introducidas por el usuario ya coinciden con otro usuario del sistema no se creara la cuenta y se notificara por la interfaz web

Cuadro 14: CU-007: Registrar usuario

CU-008	Verificar email
Actores	Alumno y Administrador
Precondiciones	El usuario se ha registrado en el sistema
Escenario	El sistema manda un correo de verificación de email al usuario El usuario hace click en el enlace contenido en el email El sistema redirige al usuario a la pagina de verificación de email El sistema verifica el email y lo registra en la base de datos
Postcondiciones	El usuario ha verificado su email

Cuadro 15: CU-008: Verificar email

CU-009	Dar/Quitar rol de Administrador
Actores	Administrador
Precondiciones	El usuario se encuentra en la vista que lista todos los usuarios
Escenario	El usuario hace click en el icono respectivo para modificar el rol del usuario El sistema comprueba si el usuario al que se le van a modificar los permisos es administrador o no El sistema cambia el rol del usuario seleccionado a Administrado o se lo quita dependiendo de si ya lo es o no
Postcondiciones	El usuario ha verificado su email
Escenarios alternativos	El usuario al que se desea cambiar los permisos es el mismo que realiza la acción: ▪ El sistema indica que un usuario no puede modificar su propio rol El usuario al que se desea cambiar los permisos es el administrador principal: ▪ El sistema indica que ese usuario no se le puede modificar el rol

Cuadro 16: CU-009: Dar/Quitar rol de Administrador

CU-010	Crear usuario
Actores	Administrador
Precondiciones	El Administrador se encuentra en la vista que lista todos los usuarios
Escenario	El usuario hace click en el icono para crear un nuevo usuario en el sistema El sistema redirige al usuario a la vista de crear nuevo usuario El usuario introduce los datos de la nueva cuenta El sistema comprueba que las credenciales sean validas El sistema comprueba si no hay usuarios que compartan esas credenciales de acceso El sistema crea el nuevo usuario, manda un correo para verificar el email asociado a la nueva cuenta y redirige al usuario a la vista de placas
Postcondiciones	El sistema crea un nuevo usuario
Escenarios alternativos	Formato incorrecto: ■ Las credenciales introducidas no siguen el formato correcto para los datos ■ El sistema muestra los errores de formato en las credenciales de acceso Ya existe un usuario con esas credenciales: ■ Si las credenciales introducidas por el usuario ya coinciden con otro usuario del sistema no se creara la cuenta y se notificara por la interfaz web

Cuadro 17: CU-010: Crear usuario

CU-011	Eliminar usuario
Actores	Administrador
Precondiciones	El Administrador se encuentra en la vista que lista todos los usuarios
Escenario	El administrador hace click en el icono para eliminar un usuario del sistema El sistema redirige al administrador a la vista de confirmación de acción El administrador confirma que desea eliminar el usuario del sistema El sistema comprueba que se haya confirmado correctamente la acción El sistema elimina al usuario del sistema y todos los registros asociados al usuario del sistema también
Postcondiciones	El sistema elimina un usuario
Escenarios alternativos	Formato incorrecto: ■ La confirmación ha sido errónea No se puede eliminar el usuario activo: ■ EL sistema impide que un usuario pueda eliminar su propio usuario y lo notifica mediante la interfaz web No se puede eliminar el administrador central: ■ EL sistema impide que un usuario pueda eliminar el administrador central y lo notifica mediante la interfaz web

Cuadro 18: CU-011: Eliminar usuario

5.2.2. Casos de uso respecto de gestión

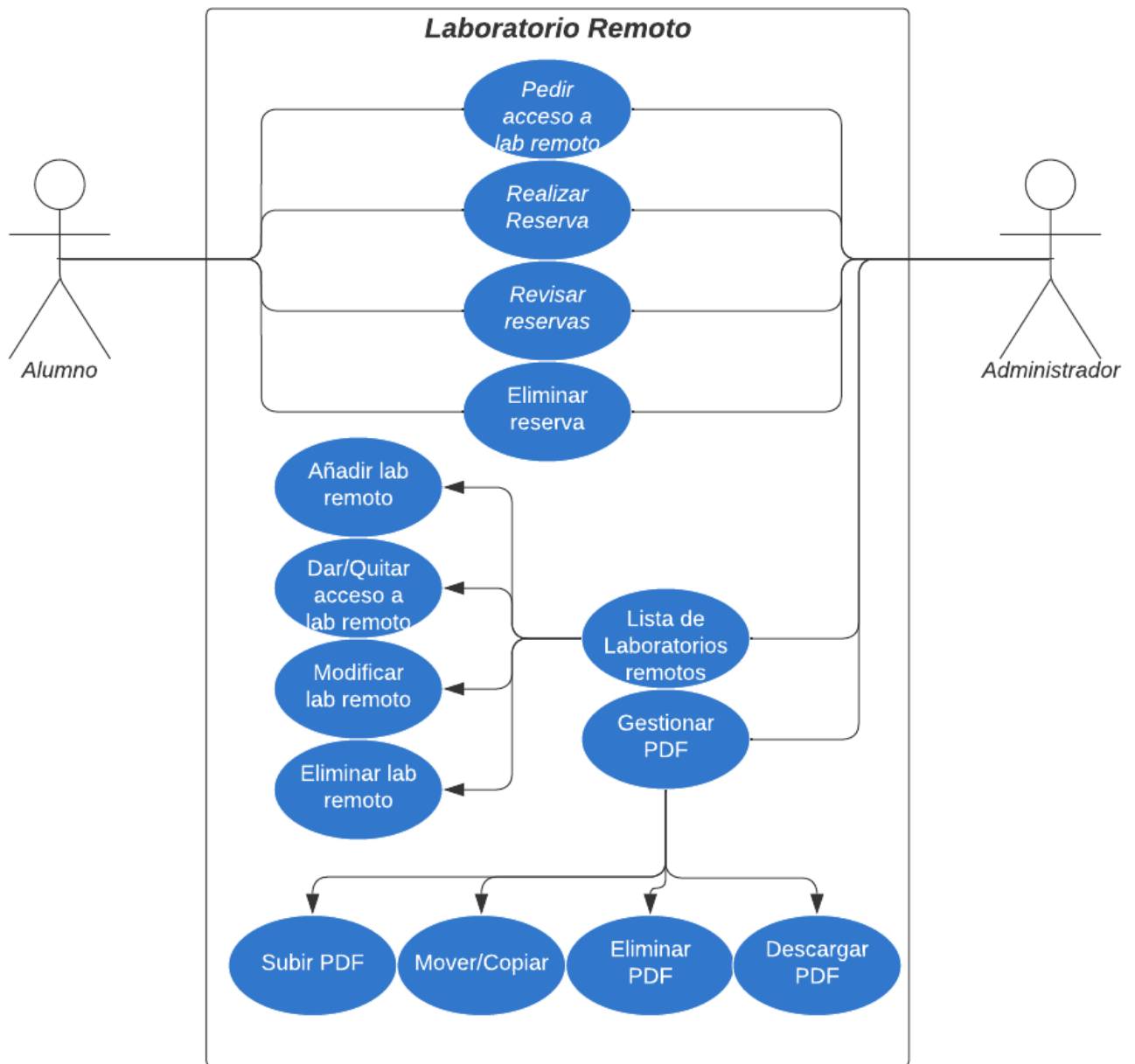


Figura 36: Casos de uso de gestión

CU-012	Pedir acceso a laboratorio remoto
Actores	Alumno
Precondiciones	El usuario ha iniciado sesión y se encuentra en la vista que lista todos los laboratorios remotos disponibles
Escenario	El usuario hace click en el icono disponible para poder desbloquear el laboratorio remoto pidiendo acceso al mismo El sistema redirige al usuario a la vista de confirmación de acción El usuario confirma que desea pedir el acceso al laboratorio remoto El sistema almacena la petición de acceso El sistema redirige al usuario de nuevo a la lista de los laboratorios remotos
Postcondiciones	El usuario ha pedido acceso a un laboratorio remoto

Cuadro 19: CU-012: Pedir acceso a laboratorio remoto

CU-013	Realizar reserva
Actores	Alumno y Administrador
Precondiciones	El usuario ha iniciado sesión y se encuentra en la vista que lista todos los laboratorios remotos disponibles
Escenario	El usuario hace click en el icono disponible para poder acceder a la vista de reservas de ese laboratorio remoto El sistema redirige al usuario a la vista de las reservas del laboratorio remoto El selecciona la semana que desea la reserva El sistema carga la información de la semana seleccionada El selecciona día, hora de inicio y hora de fin de la reserva y le da al botón de reservar El sistema realiza almacena la reserva y actualiza la vista de las reservas con la nueva reserva y manda un correo electrónico al usuario para confirmar que la reserva ha sido realizada con éxito
Postcondiciones	El usuario reservado un lapso de tiempo del laboratorio remoto
Escenarios alternativos	Reserva incorrecta: ■ El usuario selecciona un día incorrecto para la reserva, o el rango de horas de la reserva que se pide es mayor a 3 horas ■ El sistema notifica el error de la petición de reserva al usuario Máximo de reservas alcanzado: ■ El usuario ya posee una reserva pendiente que tiene que pasar ■ El sistema notifica al usuario que solamente puede tener una reserva por laboratorio remoto activa

Cuadro 20: CU-013: Realizar reserva

CU-014	Revisar reserva
Actores	Alumno y Administrador
Precondiciones	El usuario ha iniciado sesión y se encuentra en la vista que lista todos los laboratorios remotos disponibles
Escenario	El usuario hace click en el icono disponible para poder acceder a la lista de reservas de ese laboratorio remoto El sistema redirige al usuario a la vista de la lista de reservas del laboratorio remoto
Postcondiciones	El usuario accede a la lista de reservas que tiene para ese laboratorio remoto
Escenarios alternativos	Reservas anteriores: ■ El usuario una vez dentro de la lista de reservas hace click en el botón de reservas anteriores ■ El sistema carga la vista con todas las reservas anteriores pasadas del usuario Reservas totales del laboratorio remoto: ■ Si el usuario es un administrador y ha accedido a la lista de reserva desde el panel de laboratorio remoto del administrador el sistema cargara todas las reservas pendientes del laboratorio remoto, no solo las del usuario activo ■ Si dentro de este panel el administrador hace click al botón de reservas anteriores el sistema mostrara todas las reservas pasadas del laboratorio remoto, no solo las del usuario activo

Cuadro 21: CU-014: Revisar reserva

CU-015	Eliminar reserva
Actores	Alumno y Administrador
Precondiciones	El usuario ha iniciado sesión y se encuentra en la vista que lista todas sus reservas de los laboratorios remotos disponibles
Escenario	El usuario hace click en el icono disponible para eliminar la reserva pendiente que tiene El sistema redirige al usuario a la vista de la confirmación de acción El usuario confirma la acción de eliminar reserva El sistema elimina la reserva de la base de datos y manda un correo electrónico al usuario para notificar que ha sido eliminada correctamente la reserva El sistema redirige al usuario de vuelta a la lista de reservas
Postcondiciones	El usuario elimina una reserva activa que tiene en un laboratorio remoto
Escenarios alternativos	Confirmación errónea: ■ El usuario no confirma correctamente la acción de eliminar una reserva ■ El sistema notifica al usuario que no se ha confirmado correctamente la acción y no se ha realizado la misma

Cuadro 22: CU-015: Eliminar reserva

CU-016	Añadir laboratorio remoto
Actores	Administrador
Precondiciones	El usuario ha iniciado sesión y se encuentra en la vista que lista todos los laboratorios remoto, que se accede desde el panel del administrador
Escenario	El usuario hace click en el icono disponible para añadir un nuevo laboratorio remoto El sistema redirige al usuario a la vista para añadir un nuevo laboratorio remoto al sistema El usuario introduce los datos del nuevo laboratorio remoto El sistema añade el nuevo laboratorio remoto a la base de datos El sistema redirige al usuario de nuevo a la vista que lista todos los laboratorios remotos
Postcondiciones	El sistema almacena un nuevo laboratorio remoto en la base de datos
Escenarios alternativos	Datos inválidos: ■ El usuario no introduce un fichero de extensión “.csv” o no introduce correctamente uno de los datos del laboratorio remoto nuevo ■ El sistema notifica al usuario que datos ha introducido incorrectamente Nuevo directorio de PDF: ■ Si se ha seleccionado la opción de nueva ruta de PDF y se ha introducido el nombre de la nueva ruta, entonces el sistema creara la nueva carpeta contenedora de PDF

Cuadro 23: CU-016: Añadir laboratorio remoto

CU-017	Dar/Quitar acceso a laboratorio remoto
Actores	Administrador
Precondiciones	El usuario ha iniciado sesión y se encuentra en la vista que lista todos los laboratorios remoto, que se accede desde el panel del administrador
Escenario	El usuario hace click en el icono disponible para revisar los permisos de acceso de cada usuario al laboratorio remoto El sistema redirige al usuario a la vista que lista a todos los usuarios que han pedido permisos para poder acceder al laboratorio remoto y si tienen o no permisos para poder acceder, o esta pendiente de seleccionar El usuario hace click en el botón asignado para permitir o denegar el acceso al laboratorio remoto a otro usuario designado El sistema redirige al usuario a una vista para confirma su acción de modificación de permisos El usuario confirma la acción de modificar los permisos de acceso El sistema modifica el permiso de acceso del usuario designado y redirige al usuario activo de nuevo a la vista que lista a todos los usuarios que han pedido permisos para poder acceder al laboratorio remoto
Postcondiciones	El sistema modifica los permisos de acceso a un laboratorio remoto de otros usuarios
Escenarios alternativos	Usuario pendiente de asignar permisos por primera vez: ■ Si el usuario al que se le van a modificar los permisos de acceso no tiene de antemano ninguno, en vez de confirmar la acción el sistema realiza directamente la acción de modificar los permisos de acceso

Cuadro 24: CU-017: Dar/Quitar acceso a laboratorio remoto

CU-018	Modificar laboratorio remoto
Actores	Administrador
Precondiciones	El usuario ha iniciado sesión y se encuentra en la vista que lista todos los laboratorios remoto, que se accede desde el panel del administrador
Escenario	El usuario hace click en el icono disponible para editar los datos de un laboratorio remoto El sistema redirige al usuario a la vista que permite modificar los datos del laboratorio remoto El usuario modifica los datos del laboratorio remoto y registra los cambios El sistema modifica los datos del laboratorio remoto en la base de datos El sistema redirige al usuario a la vista que lista todos los laboratorios remoto y notifica que los cambios se al guardado correctamente
Postcondiciones	El sistema modifica los datos de un laboratorio remoto especifico en la base de datos
Escenarios alternativos	Laboratorio remoto con reservas pendientes: ■ Si el laboratorio remoto posee alguna reserva activa que tiene que consumirse no se podrá modificar sus datos, notificara al usuario y le impedirá acceder a la vista de modificación de datos Nuevo directorio de PDF: ■ Si se ha seleccionado la opción de nueva ruta de PDF y se ha introducido el nombre de la nueva ruta, entonces el sistema creara la nueva carpeta contenedora de PDF

Cuadro 25: CU-018: Modificar laboratorio remoto

CU-019	Eliminar laboratorio remoto
Actores	Administrador
Precondiciones	El usuario ha iniciado sesión y se encuentra en la vista que lista todos los laboratorios remoto, que se accede desde el panel del administrador
Escenario	El usuario hace click en el icono disponible eliminar laboratorio remoto del sistema El sistema redirige al usuario a la vista de confirmación de eliminación del laboratorio remoto El usuario modifica los datos del laboratorio remoto y registra los cambios El sistema modifica los datos del laboratorio remoto en la base de datos El sistema redirige al usuario a la vista que lista todos los laboratorios remoto y notifica que los cambios se al guardado correctamente
Postcondiciones	El sistema modifica los datos de un laboratorio remoto especifico en la base de datos
Escenarios alternativos	Laboratorio remoto con reservas pendientes: ■ Si el laboratorio remoto posee alguna reserva activa que tiene que consumirse no se podrá modificar sus datos, notificara al usuario y le impedirá acceder a la vista de modificación de datos Eliminar carpeta contenedora de PDF: ■ Si se ha seleccionado la opción de eliminar la carpeta contenedora de PDF se eliminara también la carpeta contenedora de PDF y los PDF correspondientes del sistema • Si existe otro laboratorio remoto que utilice esa carpeta contenedora de PDF entonces no se podrá eliminar la carpeta contenedora de PDF y se le notificará al usuario

Cuadro 26: CU-019: Eliminar laboratorio remoto

CU-020	Gestionar PDF
Actores	Administrador
Precondiciones	El usuario ha iniciado sesión y se encuentra en la vista del panel del administrador
Escenario	El usuario hace click en el botón de Gestionar PDF El sistema redirige al usuario a la vista que lista todos los laboratorios remotos El usuario hace click en el icono de acceso a un laboratorio remoto específico El sistema carga la vista de gestión de PDF del laboratorio remoto seleccionado por el usuario
Postcondiciones	El sistema modifica los datos de un laboratorio remoto específico en la base de datos

Cuadro 27: CU-020: Gestionar PDF

CU-021	Subir PDF
Actores	Administrador
Precondiciones	El usuario ha iniciado sesión y se encuentra en la vista de gestión de PDF de un laboratorio remoto en específico
Escenario	El usuario selecciona un archivo PDF para subirlo al sistema El usuario hace click en el botón de subir archivo El sistema guarda el PDF subido por el usuario en la carpeta contenedora de PDF del laboratorio remoto específico que se encuentra gestionando el usuario
Postcondiciones	El usuario sube un nuevo PDF para que se muestre en la vista de trabajo del laboratorio remoto específico que este gestionando sus documentos
Escenarios alternativos	Formato de archivo incorrecto: ■ El usuario intenta subir un archivo que no sea un PDF al sistema ■ El sistema notifica al usuario que solamente puede recibir archivos PDF y no guarda el archivo del usuario PDF ya existente: ■ El usuario sube un archivo con un nombre igual al de otro ya existente en el sistema ■ El sistema indica al usuario que ya existe un archivo con ese nombre y no guarda el archivo subido por el usuario

Cuadro 28: CU-021: Subir PDF

CU-022	Mover/Copiar PDF
Actores	Administrador
Precondiciones	El usuario ha iniciado sesión y se encuentra en la vista de gestión de PDF de un laboratorio remoto en específico
Escenario	El usuario hace click en el botón asignado para mover PDF a la carpeta de PDFs de otro laboratorio remoto, o en su defecto en el icono asignado para copiar PDF en la carpeta contenedora de PDF de otro laboratorio remoto El sistema carga la vista correspondiente a mover o a copiar PDF en la carpeta de otro laboratorio remoto El usuario selecciona el destino del PDF El sistema mueve el PDF a la carpeta destino o genera una copia del PDF en la carpeta destino
Postcondiciones	El usuario mueve o copia un PDF específico en la carpeta contenedora de PDF de otro laboratorio remoto
Escenarios alternativos	PDF ya existente: ■ El usuario sube un archivo con un nombre igual al de otro ya existente en el sistema ■ El sistema sobrescribe el archivo en la carpeta destino con el archivo seleccionado para mover/copiar

Cuadro 29: CU-022: Mover/Copiar PDF

CU-023	Eliminar PDF
Actores	Administrador
Precondiciones	El usuario ha iniciado sesión y se encuentra en la vista de gestión de PDF de un laboratorio remoto en específico
Escenario	El usuario hace click en el botón asignado para eliminar un PDF de la carpeta de PDFs del laboratorio remoto seleccionado El sistema carga la vista de confirmación de eliminación del PDF El usuario confirma que desea eliminar el PDF seleccionado El sistema elimina el PDF de la carpeta contenedora de PDF del laboratorio remoto El sistema redirige al usuario a la vista de gestión de PDF del laboratorio remoto específico
Postcondiciones	El usuario elimina del sistema un PDF de la carpeta contenedora de PDF de un laboratorio remoto específico
Escenarios alternativos	Confirmación errónea: ▪ El usuario no confirma correctamente la acción de eliminar una reserva ▪ El sistema notifica al usuario que no se ha confirmado correctamente la acción y no se ha realizado la misma

Cuadro 30: CU-023: Eliminar PDF

CU-024	Descargar PDF
Actores	Administrador
Precondiciones	El usuario ha iniciado sesión y se encuentra en la vista de gestión de PDF de un laboratorio remoto en específico
Escenario	El usuario hace click en el botón asignado para descargar un PDF específico de la carpeta de PDFs del laboratorio remoto seleccionado El sistema manda el archivo al usuario mediante la interfaz web El usuario descarga el documento PDF en su equipo
Postcondiciones	El usuario descarga un documento PDF del sistema

Cuadro 31: CU-024: Descargar PDF

5.2.3. Casos de uso de trabajo en el laboratorio remoto

Durante esta sección los casos de uso se comparten entre Alumno y administrados, así que se marcara solamente 1 Actor que sera el Usuario que sera equivalente ambos actores que se usaron anteriormente.

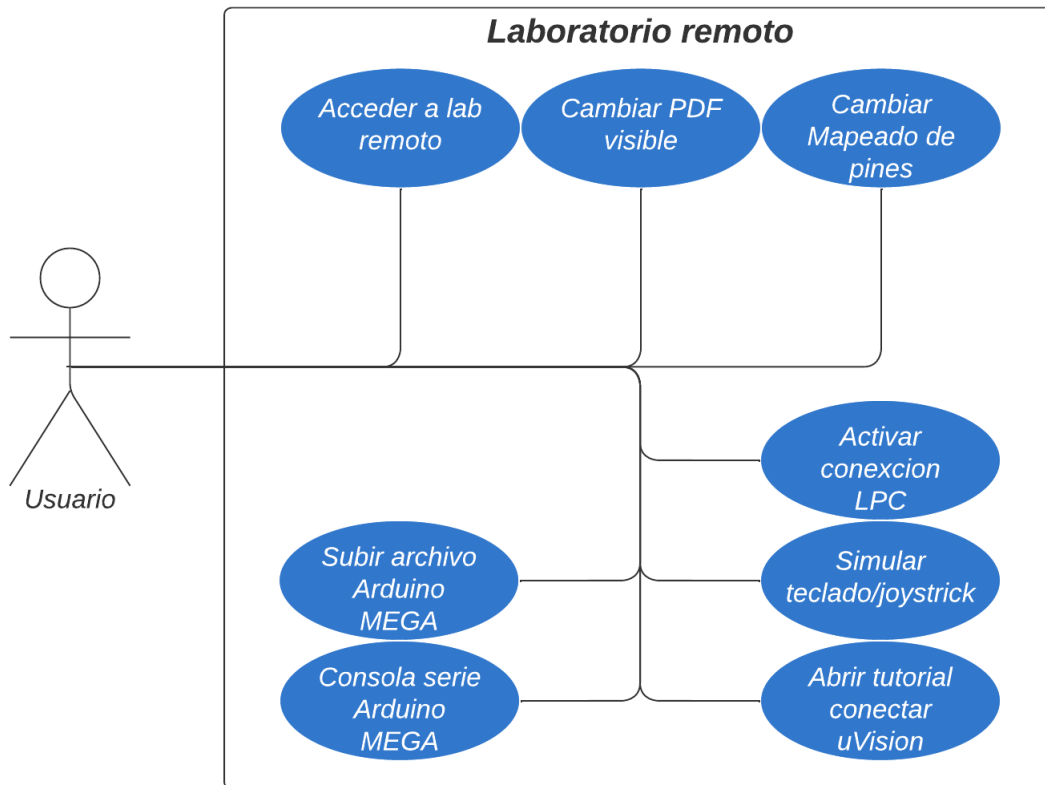


Figura 37: Casos de uso de trabajo en el laboratorio remoto

CU-025	Acceder a lab remoto
Actores	Usuario
Precondiciones	El usuario ha iniciado sesión y se encuentra en la vista que lista los laboratorios remotos
Escenario	El usuario hace click en el icono asignado para acceder al laboratorio remoto seleccionado El sistema comprueba si el usuario tiene una reserva pendiente de completarse el día de específico en el que intenta acceder y si además el momento en el que el usuario quiere acceder se encuentra dentro del lapso de tiempo de la reserva El sistema carga la vista correspondiente al laboratorio remoto seleccionado
Postcondiciones	El usuario accede a la vista para trabajar con un laboratorio remoto
Escenarios alternativos	No se posee reserva: ▪ Cuando el sistema comprueba si el usuario tiene una reserva determina que no tiene reserva o que el momento cuando el usuario intenta acceder no esta dentro de la reserva que tiene el usuario ▪ El sistema notifica al usuario no puede acceder en ese momento a la vista del laboratorio remoto Acceso de administrador: ▪ Si el usuario que intenta acceder tiene el rol de Administrador puede acceder directamente sin necesidad de una reserva

Cuadro 32: CU-025: Acceder a lab remoto

CU-026	Cambiar PDF visible
Actores	Usuario
Precondiciones	El usuario ha iniciado sesión y se encuentra en la vista que lista los laboratorios remotos
Escenario	El usuario selecciona el PDF que desea visualizar en la vista de trabajo El sistema carga el PDF seleccionado en el visualizador de PDFs de la vista de trabajo del laboratorio remoto
Postcondiciones	El usuario cambia el PDF que se está visualizado en el visor de PDF

Cuadro 33: CU-026: Cambiar PDF visible

CU-027	Cambiar mapeado de pines
Actores	Usuario
Precondiciones	El usuario ha iniciado sesión y se encuentra en la vista que lista los laboratorios remotos
Escenario	El usuario abre los distintos desplegables de pines pulsando botón correspondiente de la vista El usuario utilizando los selectores indica que pin de dispositivo desea conectar a que pin del microcontrolador LPC4088 o Arduino MEGA del laboratorio remoto El usuario hace click en el botón asignado para enviar el nuevo mapeado de pines al laboratorio remoto El sistema envía mediante comunicación SocketIO al laboratorio remoto el nuevo mapeado de pines como una cadena de texto El laboratorio remoto manda esa información al controlador del laboratorio remoto El sistema recibe una respuesta de que el laboratorio remoto ha recibido correctamente el nuevo mapeado de pines y lo notifica al usuario
Postcondiciones	El usuario envía el nuevo conexionado de pines al laboratorio remoto

Cuadro 34: CU-027: Cambiar mapeado de pines

CU-028	Subir archivo Arduino MEGA
Actores	Usuario
Precondiciones	El usuario ha iniciado sesión y se encuentra en la vista que lista los laboratorios remotos
Escenario	El usuario carga el archivo de extensión “.ino” en el cargador de archivos y le da al botón de envía “Conectar” que también actúa con el CU-027[34] El sistema recibe el archivo “.ino” y lo almacena El sistema envía mediante comunicación SocketIO al equipo conectado al laboratorio remoto El laboratorio remoto programa el Arduino MEGA con el archivo de código recibido por SocketIO El sistema recibe una respuesta de que el laboratorio remoto ha recibido correctamente el archivo y lo notifica al usuario
Postcondiciones	El usuario envía un archivo de código “.ino” y el laboratorio remoto programa el microcontrolador Arduino MEGA con dicho archivo

Cuadro 35: CU-028: Subir archivo Arduino MEGA

CU-029	Consola serie Arduino MEGA
Actores	Usuario
Precondiciones	El usuario ha iniciado sesión y se encuentra en la vista que lista los laboratorios remotos
Escenario	Recibir mensaje consola serie: ■ El sistema recibe un mensaje mediante el protocolo SocketIO del laboratorio remoto de Arduino MEGA ■ El sistema escribe en la caja de texto asignada el mensaje recibido para mostrárselo al usuario Enviar mensaje consola serie: ■ El usuario escribe un mensaje en la caja de texto ■ El usuario hace click en el botón de enviar ■ El sistema envía la cadena de texto escrita por el usuario al laboratorio remoto de Arduino MEGA mediante el protocolo SocketIO ■ El laboratorio remoto envía por el puerto serie el mensaje del usuario al Arduino MEGA
Postcondiciones	El usuario se comunica con el laboratorio remoto de Arduino MEGA mediante una consola serie emulada

Cuadro 36: CU-029: Consola serie Arduino MEGA

CU-030	Activar conexión LPC
Actores	Usuario
Precondiciones	El usuario ha iniciado sesión y se encuentra en la vista que lista los laboratorios remotos
Escenario	El sistema envía una petición para saber el estado de la conexión del laboratorio remoto de la LPC4088 El laboratorio remoto responde con el estado de la conexión El sistema muestra un botón dependiendo del estado de la conexión (botón de encender o de apagar la conexión) El usuario hace click en el botón de Encender/Apagar conexión LPC El sistema envía la orden correspondiente al laboratorio remoto de la LPC4088 El laboratorio remoto activa el protocolo de conexión remota del uVision con la LPC4088 El sistema informa al usuario de que se ha activado la conexión
Postcondiciones	El usuario se comunica con el laboratorio remoto de la LPC4088 para activar o desactivar la conexión remota

Cuadro 37: CU-030: Activar conexión LPC

CU-031	Simular teclado/joystick
Actores	Usuario
Precondiciones	El usuario ha iniciado sesión y se encuentra en la vista que lista los laboratorios remotos
Escenario	El usuario abre el panel correspondiente al teclado o al joystick pulsando el botón diseñado para ello El sistema despliega el menú del teclado o joystick El usuario introduce una entrada de datos pulsando los botones del menú desplegado simulando el teclado o el joystick El sistema envía esta entrada de datos al laboratorio remoto de la LPC4088 usando el protocolo SocketIO El laboratorio remoto envía esta información al controlador del laboratorio remoto
Postcondiciones	El usuario utiliza el dispositivo de teclado o joystick desde la interfaz web

Cuadro 38: CU-031: Simular teclado/joystick

CU-032	Abrir tutorial conectar uVision
Actores	Usuario
Precondiciones	El usuario ha iniciado sesión y se encuentra en la vista que lista los laboratorios remotos
Escenario	El usuario hace click en el botón diseñado para abrir el tutorial de como conectar el uVision al laboratorio remoto El sistema abre una ventana en el equipo del cliente donde se muestra un tutorial de como conectar el uVision al laboratorio remoto
Postcondiciones	El usuario abre el tutorial de como conectar el uVision al laboratorio remoto

Cuadro 39: CU-032: Abrir tutorial conectar uVision

6. Diseño

Una vez realizado el estudio previo de los requisitos funcionales, no funcionales y de los distintos casos de uso se empezó a plantear el diseño de la arquitectura que tendrá que seguir el producto final para poder ofrecer toda la funcionalidad posible de la mejor forma posible.

6.1. Decisiones de diseño

Para este punto se tomaron de referencia los distintos laboratorios remotos que se pueden encontrar desarrollados por el mundo. Por ende se decidió implementar un servidor principal que será aquel que aloje toda la lógica implementada en la interfaz web, que estará en la red local de la universidad y que se conectará a través de un protocolo bidireccional de comunicación llamado SocketIO con los equipos que se encuentran conectados a los distintos laboratorios remotos para poder transmitirles la información que se envía el usuario, ya sea en forma de comando, de información del puerto serie para la placa Arduino MEGA, alguna entrada simulada de algunos dispositivos físicos, como serían el teclado y el joystick, o también datos como el archivo tipo “.ino” para programar el Arduino MEGA o el mapeo de pines para interconectar los dispositivos a pines específicos del Arduino MEGA o de la LPC4088.

Además por temas de la infraestructura necesaria para poder emplear el uVision de forma remota para poder programar y depurar la LPC4088 se empleará el software de SEGGER que permite utilizar un dispositivo como servidor remoto al que conectar el IDE de Keil uVision y poder trabajar así sin estar conectado físicamente al dispositivo JLink y al microcontrolador LPC4088, pero por desgracia el software de servidor remoto, para poder conectar a través de internet crea una conexión túnel la cual no se genera como forma de IP, sino mas bien como forma de URL, pero el IDE de uVision solamente admite IP para poder conectarse de forma remota, lo cual limita a utilizando en una red LAN.

Además de esto, la utilización del IDE de uVisión necesita del empleo de una licencia de uso para poder compilar proyectos que ocupen mas de un mínimo de 4098KB de peso total.

La solución de ambos problemas se ha decidido que es la utilización de la VPN de la UCA[17] para poder emplear la licencia flexible que posee la UCA para el IDE uVision, que permite que hasta 10 entornos de desarrollo se encuentren compilando de forma simultánea, y que mientras que no se esta compilando no esta siendo ocupado ese espacio de la licencia, además de esta forma se puede establecer una conexión en forma local aunque ambos dispositivos se encuentre en redes distintas, gracias a que ambos dispositivos se encuentran en la misma red VPN.

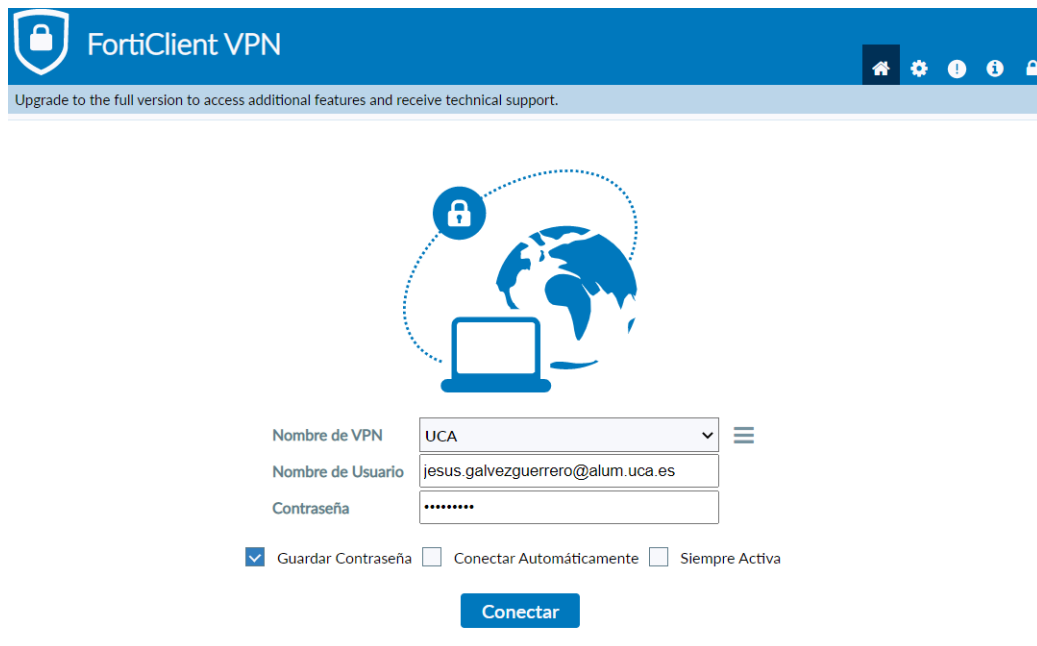


Figura 38: Interfaz del cliente Forti Client para conectarse a la VPN de la UCA

6.2. Diseño de la arquitectura

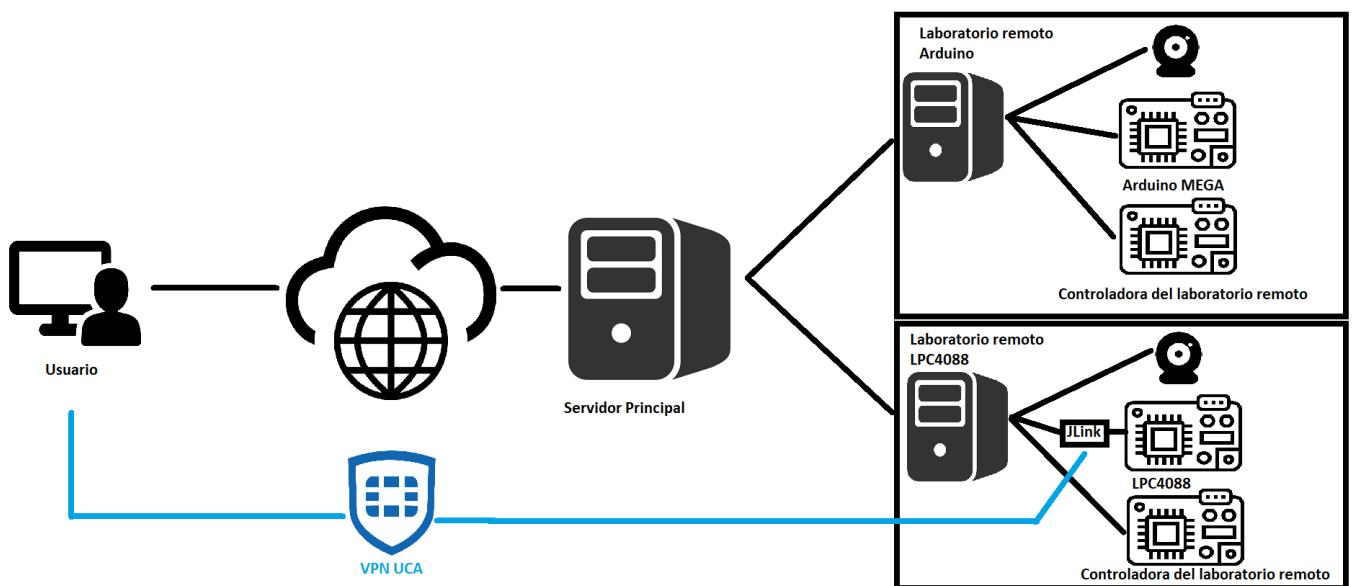


Figura 39: Arquitectura física del proyecto

La arquitectura bajo la que está ideada este trabajo se basa en el hecho de utilizar internet para poder conectarte al servidor principal a través de internet, siendo necesario que los laboratorios remotos se encuentren conectados al servidor principal en la misma red, mientras que además se tiene una conexión aparte para poder conectar de forma “local” usando la VPN el entorno de desarrollo de uVision con el JLink conectado a la LPC4088.

Esa conexión no se encuentra abierta todo el rato, sino que esta controlado por el equipo conectado al JLink y esta a su vez de enciende o se apaga dependiendo de lo que el usuario indique en la interfaz web, que esta información es notificada al laboratorio remoto desde el servidor principal con el protocolo de comunicación bi-direccional de baja latencia de SocketIO.

Además de la arquitectura que se ve en el diagrama puesto anteriormente también se remarca que tanto la controladora del laboratorio remoto como el microcontrolador del laboratorio remoto (ya sea el Arduino MEGA o la LPC4088) se encuentran conectados a distintos Sensores/Actuadores, a término del diagrama de la arquitectura han sido tratados como el mismo microcontrolador/controladora.

6.3. Diseño de la base de datos

La base de datos del trabajo se ha realizado utilizando la tecnología PostgreSQL que ofrece una gran compatibilidad base con las herramientas de Python y con SQLAlchemy, permitiendo gracias a eso que el desarrollador no tenga que preocuparse demasiado el tipo de formato específico que devuelve la base de datos o que recibe de la misma, y simplemente teniendo que tener en cuenta el tipo de dato que recibe y como quiere trabajar con el específicamente.

Y para gestionar esta base de datos se ha empleado el software de PGAdmin4 que es una plataforma de administración y desarrollo Open Source para PostgreSQL y sus sistemas de administración de bases de datos relacionados. PGAdmin esta programada con Python y jQuery. Se puede utilizar pgAdmin para tareas que van desde redactar consultas de SQL básicas hasta monitorizar sus bases de datos y configurar arquitecturas de bases de datos avanzadas.

La base de datos es sencilla puesto que principalmente se necesita solamente almacenar los datos de los distintos usuarios, de los distintos laboratorios remotos y ya además de eso los permisos de acceso de cada usuario para los distintos laboratorios remotos y las reservas que hayan podido hacer esos mismos usuarios en los laboratorios remotos a los que tengan acceso, por ello se presenta un diagrama sencillo de las clases empleadas en la base de datos.

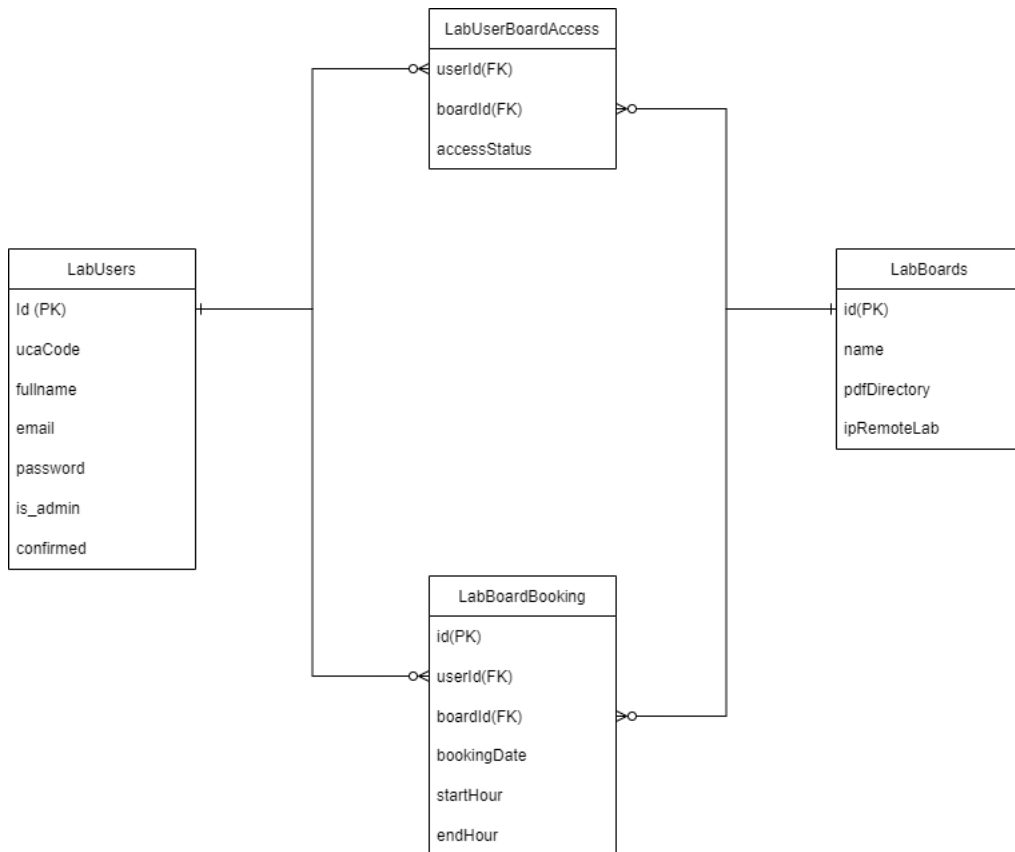


Figura 40: Diagrama de clases de la base de datos

En este diagrama podemos ver como la base de datos se divide en 4 tablas las cuales son:

- **LabUsers:** Tabla que almacenará la información de los usuarios, y cuyos campos son:
 - Id - Clave primaria, generada de forma automática, para poder identificar internamente a cada usuario con un numero único y que se irá incrementando conforme se vayan creando usuarios.
 - ucaCode - Cadena de texto que tendrá un valor único para cada usuario y que debe seguir un formato tal que uXXXXXXXXX
 - fullname - Cadena de texto que almacenara el nombre completo y los apellidos del usuario.
 - email - Cadena de texto que almacenara el correo electrónico del usuario para poder enviarle las notificaciones o confirmaciones pertinentes.
 - password - Cadena de texto que almacenara la contraseña de acceso del usuario y que sera encriptada por el servidor antes de guardar la información.
 - is_admin - Campo booleano que indicara si el usuario tiene rol de administrador o no.
 - confirmed - Campo booleano que indicara si el usuario ha confirmado su dirección de correo electrónico o no.

- **LabBoards:** Tabla que almacenará la información de los laboratorios remotos, y cuyos campos son:
 - Id - Clave primaria, generada de forma automática, para poder identificar internamente a cada laboratorio remoto con un numero único y que se ira incrementando conforme se vayan añadiendo nuevos laboratorios remotos.
 - name - Cadena de texto que almacenará el nombre que se le dará al laboratorio remoto.
 - pdfDirectory - Cadena de texto que almacenará la ruta de la carpeta contenedora de los PDF de prácticas que podrán ser visualizados en la interfaz web cuando se trabaje en ese laboratorio remoto
 - ipRemoteLab - Cadena de texto que almacenará la IP del laboratorio remoto para poder así establecer la comunicación entre el mismo y el servidor principal mediante el protocolo SocketIO
- **LabUserBoardAccess:** Tabla que almacenará los permisos de acceso de cada usuario a cada laboratorio remoto y cuyos campos son:
 - userId - Clave Foránea, que coincidirá con el Id del usuario a que hace referencia esa entrada de la base de datos.
 - boardId - Clave Foránea, que coincidirá con el Id del laboratorio remoto a que hace referencia esa entrada de la base de datos.
 - accessStatus - Indicará que tipo de acceso puede tener el usuario con id igual a userId para la placa con id igual a boardId, y este campo puede tomar 3 posibles valores que serian “PERMITIDO”(el usuario puede acceder al laboratorio remoto), “DENEGADO”(el usuario no puede acceder al laboratorio remoto) y “PENDIENTE”(el usuario ha pedido el acceso al laboratorio remoto y esta pendiente de que un administrador le de acceso o se lo deniegue).
- **LabBoardBooking:** Tabla que almacenará las reservas de los usuarios en los laboratorios remotos y cuyos campos son:
 - Id - Clave primaria, generada de forma automática, para poder identificar internamente a cada reserva del laboratorio remoto con un numero único y que se ira incrementando conforme se vayan añadiendo nuevos reservas.
 - userId - Clave Foránea, que coincidirá con el Id del usuario a que hace referencia esa entrada de la base de datos.
 - boardId - Clave Foránea, que coincidirá con el Id del laboratorio remoto a que hace referencia esa entrada de la base de datos.
 - bookingDate - Campo de tipo Date que almacenará el día, mes y año de la reserva.
 - startHour - Campo de tipo Time que almacenará la hora a la que empieza la reserva.

- endHour - Campo de tipo Time que almacenará la hora a la que termina la reserva.

6.4. Diagrama de flujo

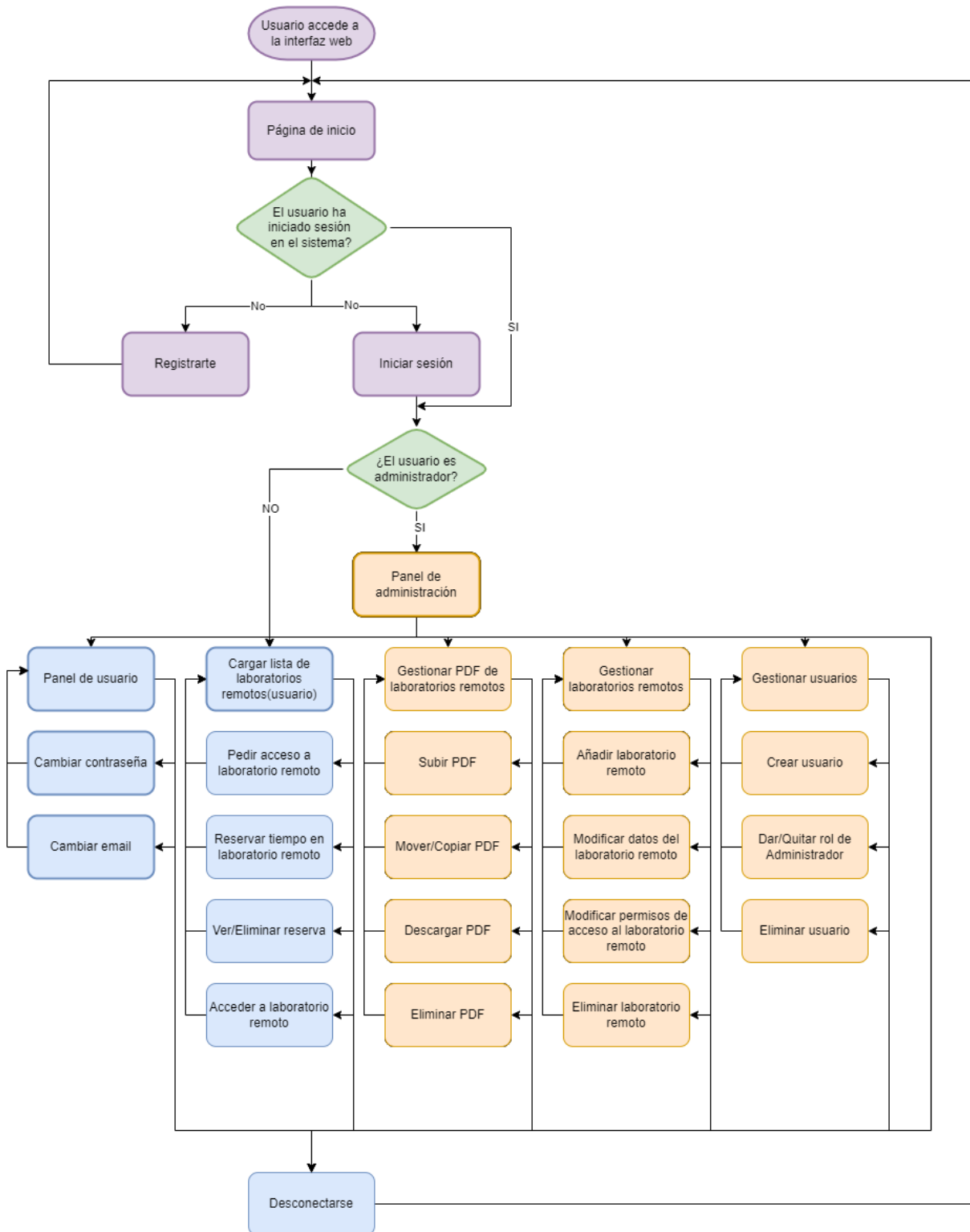


Figura 41: Diagrama de flujo de la aplicación web

Para comprender completamente el diagrama de flujo se aportan unas pequeñas aclaraciones sobre el tema de los colores:

- **Morado**: Puede ser accedido por cualquier usuario.
- **Verde**: Son comprobaciones que hace el sistema para redirigir correctamente a los usuarios.
- **Azul**: Puede ser accedido por cualquier usuario que haya iniciado sesión.
- **Naranja**: Puede ser accedido solamente por los usuarios que hayan iniciado sesión y tengan el rol de Administrador.

En el diagrama de flujo podemos ver que tal y como accede un usuario a la aplicación web el mismo revisa si tenía una sesión iniciada de alguna vez anterior que accedió a la aplicación, en caso de que no sea así el usuario tiene 2 opciones, o registrarse, lo cual tras terminar el proceso es redirigido a la página principal (ya con la sesión iniciada, con lo cual pasaría a la comprobación de si es administrador), o iniciar sesión con un usuario ya registrado en el sistema.

Una vez pasamos al interior de la aplicación con una sesión iniciada el sistema comprueba si somos administradores o no, en caso de no ser así, se nos cargará automáticamente la lista de laboratorios remotos asignada para los usuarios no administradores, desde la cual podremos proceder a pedir acceso a un laboratorio remoto específico, realizar una reserva en los laboratorios remotos que tengamos acceso, ver y eliminar las reservas que tengamos realizadas y acceder a los laboratorios remotos que tengamos reservados.

Además para todos los usuarios también pueden acceder desde cualquier momento a un panel de usuario que le mostrará sus datos y les permitirá modificar sus correos electrónicos y contraseñas.

Si el usuario que es por el contrario un usuario con el rol de Administrador se cargará el panel de administrador, desde el cual, además de poder acceder a las mismas vistas que un usuario normal, porque acceder a la vista de gestión de PDFs de los laboratorios remotos, que le permitirá subir nuevos PDF para ser vistos desde un laboratorio remoto, mover o copiar un PDF de la carpeta contenedora de PDF de un laboratorio remoto a la carpeta contenedora de PDF de otro laboratorio remoto, descargar un PDF que se encuentre en la carpeta contenedora de PDF del laboratorio remoto o eliminar un PDF de esta misma carpeta.

Los Administradores también tienen acceso a un panel de gestión de laboratorios remotos, que les permitirá añadir nuevos laboratorios remotos, modificar los datos de los mismos que se encuentren almacenados en el sistema, modificar los permisos de acceso de los distintos usuarios a cada laboratorio remoto y eliminar un laboratorio remoto del sistema.

Por último los administradores tendrán acceso a un panel de gestión de usuarios, que les permitirá añadir o eliminar usuarios del sistema, como también poder dar o quitar roles de administrador a esos usuarios, por seguridad habrá un usuario Administrador que no podrá ser eliminado ni se le podrá quitar el rol de administrador.

Desde cualquier parte dentro del sistema, una vez iniciada la sesión se podrá acceder a la pagina principal, que rediría al usuario al panel correspondiente dependiendo de si posee el rol de Administrador o no, o también podrá cerrar la sesión activa del usuario y consecuentemente sera redirigido a la pagina de inicio.

6.5. Diseño de la web

En este apartado se mostrarán las distintas vistas que componen la interfaz web:

6.5.1. Iniciar sesión/Registrar usuario

Primero tendremos la ventana de inicio de sesión que contiene un campo para introducir el usuario UCA, la contraseña del usuario, un botón para acceder a la sesión y dos enlaces para poder acceder a la ventana de restablecer contraseña o la de registrar cuenta.



The image shows a login window titled "Acceso". It contains two input fields: "Usuario uca" and "Contraseña". Below the password field is a checkbox labeled "Recuérdame". A blue button labeled "Acceso" is positioned below the checkbox. At the bottom of the window, there are two links: "¿No tienes cuenta? [Regístrate](#)" and "¿Olvidaste tu contraseña? [Restablecer contraseña](#)".

Figura 42: Ventana de inicio de sesión

Si el usuario hace click en en enlace de Registrarse entonces el usuario accederá al panel de registrarse en el sistema el cual se ve de la siguiente forma.

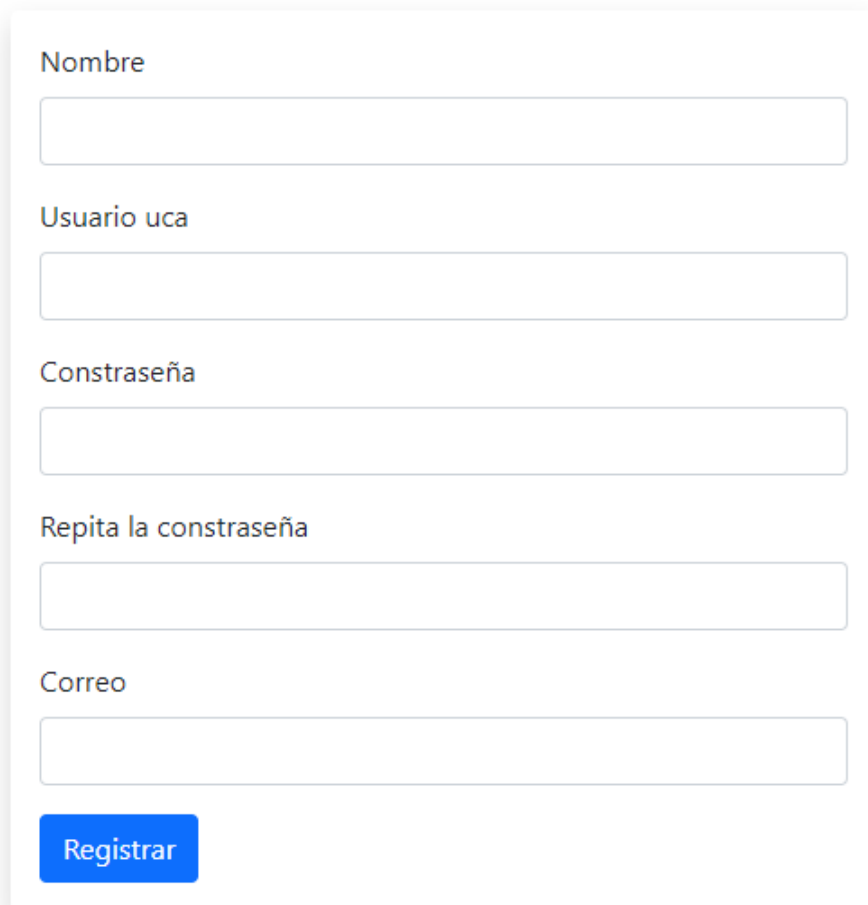
El formulario de registro de usuario está contenido en un recuadro blanco con una sombra suave. Incluye cinco campos de entrada de texto, cada uno precedido por un label: 'Nombre', 'Usuario uca', 'Contraseña', 'Repita la contraseña' y 'Correo'. Los campos de texto son rectangulares con bordes finos. Al final del formulario, hay un botón rectangular de color azul con el texto 'Registrar' en blanco.

Figura 43: Ventana de registrar usuario

En este panel el usuario necesitaría indicar su nombre completo, su usuario UCA (el cual será de la forma uXXXXXXXX, siendo XXXXXXXX los números de su DNI), la contraseña que usará en su cuenta, que será introducida 2 veces para asegurar que el usuario no se equivoca escribiendo la contraseña y por último se deberá introducir el correo que desea utilizar el usuario para recibir las notificaciones y las confirmaciones de ciertas acciones que podrá hacer en un futuro, como sería cambiar la contraseña o el correo electrónico.

6.5.2. Panel de administrador

Si cuando el usuario inicia sesión dependiendo de si el usuario tiene el rol de Administrador o no, el sistema, cargará el panel de administrador o directamente la lista de laboratorios remotos que esta destinada a los usuarios no administradores.

Primero veremos lo que ocurre cuando un Administrador inicia sesión, que es cargar el panel de administrador, el cual posee 4 botones para acceder a los tres distintos paneles exclusivos para un Administrador y el panel de usuario (que un usuario normal puede acceder haciendo click en la barra de navegación, en un enlace que se ve con el nombre del usuario, mientras que hacer eso teniendo el rol de Administrador redirigirá al usuario al panel de administrador)

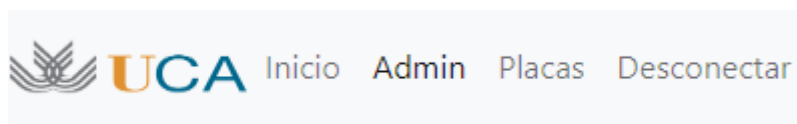


Figura 44: Barra de navegación

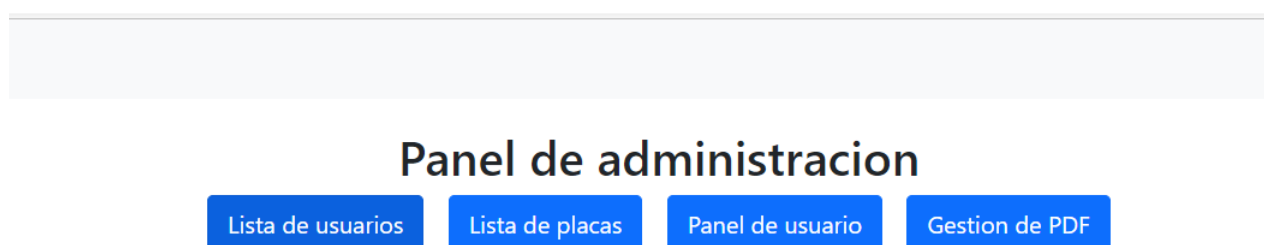


Figura 45: Panel de administrador

Desde este panel se puede acceder a 4 diferentes secciones de la aplicación web.

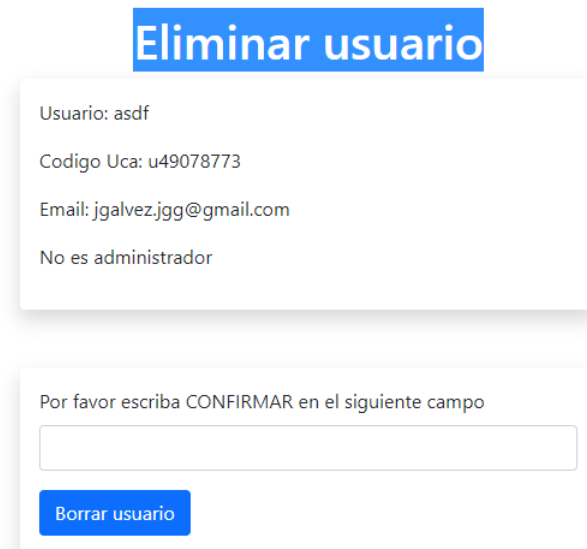
6.5.3. Lista de usuarios

Desconectar						
#	Codigo uca	Nombre	Email	Email confirmado	Admin	Acciones
8	u00000000	Admin	admin@admin.com	Si	Si	
10	u11111111	NoAdmin	noadmin@noadmin.com	Si	No	
12	u22222222	Prueba	pruebas@pruebas.es	Si	No	
19	u49078773	asdf	jgalvez.jgg@gmail.com	Si	No	
20	u44444444	Hola Amigo	neogameresp@gmail.com	Si	No	

Figura 46: Lista de usuarios

Este menú muestra a todos los usuarios que están registrados en el sistema, junto con su identificador único, su código UCA, su nombre, correo electrónico, si tienen confirmada el correo electrónico y si son

administradores. También posee 2 botones que permiten cambiarle el rol del usuario a Administrador o quitarle ese rol (el icono de la llave) y eliminar cualquier usuario(que no sea uno mismo o el administrador central(u00000000)).



Eliminar usuario

Usuario: asdf

Codigo Uca: u49078773

Email: jgalvez.jgg@gmail.com

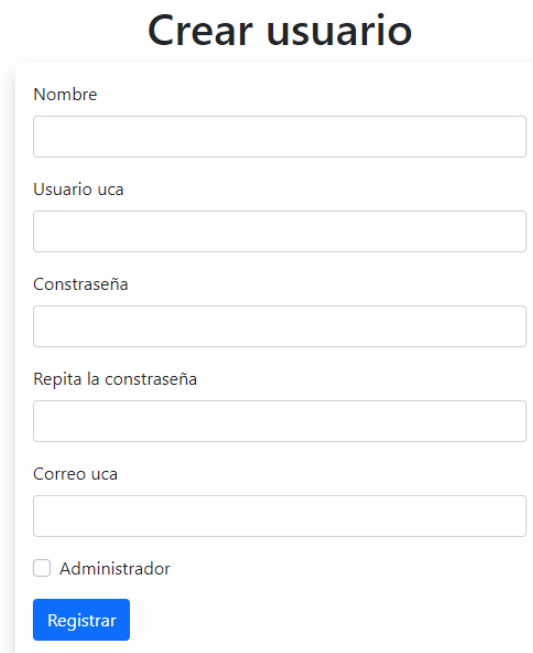
No es administrador

Por favor escriba CONFIRMAR en el siguiente campo

Borrar usuario

Figura 47: Panel de eliminar usuario

Por ultimo, arriba de los icono de acción de cada usuario podemos ver un icono de un + el cual permite añadir un nuevo usuario al sistema, sin que el mismo se registre.



Crear usuario

Nombre

Usuario uca

Contraseña

Repita la contraseña

Correo uca

☐ Administrador

Registrar

Figura 48: Panel de crear nuevo usuario

La única diferencia con el panel de registrar usuario, es que en este

panel también hay una checkbox que permite que el usuario que se va a crear sea creado como administrador directamente.

6.5.4. Lista de placas(Administrador)

Desconectar

#	Nombre de la Placa	Ubicacion de los pdf	Ip	Acciones
3	Casa	placas/test3part	192.168.1.129	    
4	placa-2	placas/placa-2	192.168.1.128	    
7	LPC3	placas/LPC3	192.168.1.127	    
9	SubirArchivoa	placas/placa-2	localhost	    
12	Prueba doc	placas/prueba doc	pruebaDocs	    

Figura 49: Panel que lista todos los laboratorios remotos

En este panel podremos ver listados todos los laboratorios remotos registrados en el sistema, junto con sus Id únicos, el nombre que se les da a los laboratorios remotos, la ruta de la carpeta contenedora de PDF y la IP de los equipos conectados a los laboratorios remotos.

La primera de las opciones que tiene el administrador en este panel esta la creación de nuevos laboratorios remotos en el sistema, que se accede pulsando el icono de un + que se encuentra en la cabecera de la columna de **Acciones**.

Añadir laboratorio remoto

Nombre

Ubicacion PDFs

csv

Nombre nueva ruta

Solo necesario si se selecciona Nueva ruta

Ip

Mapeado de pines

Elegir archivo

No se ha seleccionado ningún archivo

Añadir placa

Figura 50: Panel que creación de laboratorios remoto

A la hora de la creación de un nuevo laboratorio remoto se deberá introducir el nombre que se le dará al laboratorio remoto, seleccionar la

ruta de la carpeta contenedora de PDF, si desea compartir la carpeta con otro laboratorio remoto, si se selecciona que se creará una nueva carpeta, entonces se deberá introducir el nombre de esa nueva carpeta a crear, también se deberá introducir la IP del equipo que esta conectado al laboratorio remoto y también subir un archivo “.csv” que contendrá las posibilidades de cambio de pines de los sensores/actuadores con el microcontrolador LPC4088 o Arduino MEGA.

La tabla que se ha usado para el mapeado de los pines de la LPC4088 es la siguiente:

	A	B	C	D	E	F	G	H	I	J
1	Pin 1	Joystick JP3-1	P0.2	P0.12	P0.19	P4.22	P2.22	P3.16	P3.22	P3.29
2	Pin 2	Teclado JP1-1	P0.2	P0.12	P0.19	P4.22	P2.22	P3.16	P3.22	P3.29
3	Pin 3	Teclado JP2-4	P0.2	P0.12	P0.19	P4.22	P2.22	P3.16	P3.22	P3.29
4	Pin 4	nombre 4	P0.2	P0.12	P0.19	P4.22	P2.22	P3.16	P3.22	P3.29
5	Pin 5	nombre 5	P0.3	P0.13	P0.20	P5.1	P2.23	P3.17	P3.30	P3.31
6	Pin 6	EEPROM - MISO	P0.3	P0.13	P0.20	P5.1	P2.23	P3.17	P3.30	P3.31
7	Pin 7	Joystick JP3-3	P0.3	P0.13	P0.20	P5.1	P2.23	P3.17	P3.30	P3.31
8	Pin 8	Teclado JP2-3	P0.3	P0.13	P0.20	P5.1	P2.23	P3.17	P3.30	P3.31
9	Pin 9	Sensor de Temperatura	P0.4	P0.23	P0.27	P1.5	P1.16	P1.18	P3.23	P5.0
10	Pin 10	EEPROM - MOSI	P0.4	P0.23	P0.27	P1.5	P1.16	P1.18	P3.23	P5.0
11	Pin 11	Teclado JP2-2	P0.4	P0.23	P0.27	P1.5	P1.16	P1.18	P3.23	P5.0
12	Pin 12	Led RGB - Azul	P0.4	P0.23	P0.27	P1.5	P1.16	P1.18	P3.23	P5.0
13	Pin 13	Fotoresistor	P0.5	P0.24	P0.28	P1.6	P1.19	P3.24		
14	Pin 14	Teclado JP2-1	P0.5	P0.24	P0.28	P1.6	P1.19	P3.24		
15	Pin 15	nombre 15	P0.5	P0.24	P0.28	P1.6	P1.19	P3.24		
16	Pin 16	nombre 16	P0.5	P0.24	P0.28	P1.6	P1.19	P3.24		
17	Pin 17	I2C - SDA	P0.6	P0.25	P1.7	P2.4	P2.26	P3.18	P3.25	P5.2
18	Pin 18	EEPROM - SCK	P0.6	P0.25	P1.7	P2.4	P2.26	P3.18	P3.25	P5.2
19	Pin 19	Joystick JP3-2	P0.6	P0.25	P1.7	P2.4	P2.26	P3.18	P3.25	P5.2
20	Pin 20	nombre 20	P0.6	P0.25	P1.7	P2.4	P2.26	P3.18	P3.25	P5.2
21	Pin 21	I2C - SCL	P0.7	P0.26	P1.11	P4.23	P2.27	P3.19	P3.26	P5.3
22	Pin 22	Joystick JP3-4	P0.7	P0.26	P1.11	P4.23	P2.27	P3.19	P3.26	P5.3
23	Pin 23	Teclado JP1-2	P0.7	P0.26	P1.11	P4.23	P2.27	P3.19	P3.26	P5.3
24	Pin 24	nombre 24	P0.7	P0.26	P1.11	P4.23	P2.27	P3.19	P3.26	P5.3
25	Pin 25	Led RGB - Rojo	P0.0	P1.2	P1.12	P1.30	P2.30	P3.20	P3.27	P5.4
26	Pin 26	Teclado JP1-3	P0.0	P1.2	P1.12	P1.30	P2.30	P3.20	P3.27	P5.4
27	Pin 27	nombre 27	P0.0	P1.2	P1.12	P1.30	P2.30	P3.20	P3.27	P5.4
28	Pin 28	nombre 28	P0.0	P1.2	P1.12	P1.30	P2.30	P3.20	P3.27	P5.4
29	Pin 29	nombre 29	P0.1	P1.3	P1.31	P3.28	P2.31	P3.21		
30	Pin 30	EEPROM - CS	P0.1	P1.3	P1.31	P3.28	P2.31	P3.21		
31	Pin 31	Teclado JP1-4	P0.1	P1.3	P1.31	P3.28	P2.31	P3.21		
32	Pin 32	Led RGB - Verde	P0.1	P1.3	P1.31	P3.28	P2.31	P3.21		

Figura 51: Tabla de mapeado de pines de la LPC4088

En esta tabla podemos ver 3 distintas secciones, la columna A, que se encuentra en la sección [1] la cual debe permanecer siempre igual, independientemente de para que laboratorio remoto sea la tabla de mapeado de pines, la columna B, que es la sección [2] y en donde se almacena el nombre que tendrá cada desplegable, de acuerdo a como estén montados los sensores/actuadores del laboratorio remoto, por ejemplo podemos ver como en la fila 1 tenemos el desplegable del Pin 1 de la placa multiplexora se llamara “Joystick JP3-1”, y las columnas a la derecha de la columna B, que es la sección [3] son las opciones de la LPC4088 a los que se puede conectar el Pin X, siendo X el Pin que se encuentra en la columna A de esa misma fila, por ejemplo el Pin 1 puede conectarse a los pines P0.2, P0.12, P0.19, P4.22, P2.22, P3.16, P3.22 y P3.29 de la LPC4088, y esto se extrapola a todas las filas del documento “.csv”, en el Anexo [A]

Para el mapeado de pines del laboratorio remoto del Arduino MEGA se ha usado la siguiente tabla, que sigue la misma lógica que la tabla del laboratorio remoto de la LPC4088:

	A	B	C	D	E	F	G	H	I	J
1	Pin 1	nombre 1	D33	D31	D29	D27	D25	D23	D21	D20
2	Pin 2	nombre 2	D33	D31	D29	D27	D25	D23	D21	D20
3	Pin 3	nombre 3	D33	D31	D29	D27	D25	D23	D21	D20
4	Pin 4	nombre 4	D33	D31	D29	D27	D25	D23	D21	D20
5	Pin 5	nombre 5	D19	D18	D17	D16	D15	D14	D0	D1
6	Pin 6	nombre 6	D19	D18	D17	D16	D15	D14	D0	D1
7	Pin 7	nombre 7	D19	D18	D17	D16	D15	D14	D0	D1
8	Pin 8	nombre 8	D19	D18	D17	D16	D15	D14	D0	D1
9	Pin 9	nombre 9	D2	D3	D4	D5	D6	D7	D8	D9
10	Pin 10	nombre 10	D2	D3	D4	D5	D6	D7	D8	D9
11	Pin 11	nombre 11	D2	D3	D4	D5	D6	D7	D8	D9
12	Pin 12	nombre 12	D2	D3	D4	D5	D6	D7	D8	D9
13	Pin 13	nombre 13	D10	D11	D12	D13	GND			
14	Pin 14	nombre 14	D10	D11	D12	D13	GND			
15	Pin 15	nombre 15	D10	D11	D12	D13	GND			
16	Pin 16	nombre 16	D10	D11	D12	D13	GND			
17	Pin 17	nombre 17	D48	D46	D44	D42	D40	D38	D36	D34
18	Pin 18	nombre 18	D48	D46	D44	D42	D40	D38	D36	D34
19	Pin 19	nombre 19	D48	D46	D44	D42	D40	D38	D36	D34
20	Pin 20	nombre 20	D48	D46	D44	D42	D40	D38	D36	D34
21	Pin 21	nombre 21	D32	D30	D28	D26	D24	D22	A15	A14
22	Pin 22	nombre 22	D32	D30	D28	D26	D24	D22	A15	A14
23	Pin 23	nombre 23	D32	D30	D28	D26	D24	D22	A15	A14
24	Pin 24	nombre 24	D32	D30	D28	D26	D24	D22	A15	A14
25	Pin 25	nombre 25	A13	A12	A11	A10	A9	A8	A7	A6
26	Pin 26	nombre 26	A13	A12	A11	A10	A9	A8	A7	A6
27	Pin 27	nombre 27	A13	A12	A11	A10	A9	A8	A7	A6
28	Pin 28	nombre 28	A13	A12	A11	A10	A9	A8	A7	A6
29	Pin 29	nombre 29	A5	A4	A3	A2	A1	A0		
30	Pin 30	nombre 30	A5	A4	A3	A2	A1	A0		
31	Pin 31	nombre 31	A5	A4	A3	A2	A1	A0		
32	Pin 32	nombre 32	A5	A4	A3	A2	A1	A0		

Figura 52: Tabla de mapeado de pines de la Arduino MEGA

Las siguientes opciones ya están dentro de la fila de cada laboratorio remoto y serán aplicadas al laboratorio remoto en cuestión de esa misma fila. La primera de esas opciones permite ver la lista de usuarios que pueden o no pueden acceder al laboratorio remoto, también aquellos que han pedido el acceso, y modificar esos permisos, con los dos botones a la derecha de cada usuario, siendo el primero para permitir y el segundo para denegar el acceso.

Desconectar			
Arduino MEGA			
Codigo Uca	Nombre	Tipo de acceso	Actions
u11111111	NoAdmin	PERMITIDO	✓ ✕
u49078773	asdf	PERMITIDO	✓ ✕
u00000000	Admin	PERMITIDO	✓ ✕

Figura 53: Panel que lista los permisos de acceso a los laboratorios remotos

La segunda de las opciones que ofrece la lista de laboratorios remoto es el poder acceder al proceso de reserva de ese laboratorio remoto, donde se verá una tabla que contendrá la semana actual y quienes tienen reservado cada momento del laboratorio remoto, poder reservar algún espacio de tiempo, indicando que día, hora de inicio y hora de fin (máximo de tiempo 3 horas) de la reserva, o en su defecto cambiar de semana con las

flechas a los lados del nombre del laboratorio remoto.

UCA Inicio Admin Placas Desconectar

Arduino MEGA

Hora	Lunes 27/6	Martes 28/6	Miercoles 29/6	Jueves 30/6	Viernes 1/7	Sabado 2/7	Domingo 3/7
00:00 - 01:00	Libre	Libre	Libre	Libre	Libre	Libre	Libre
01:00 - 02:00	Libre	Libre	Libre	Libre	Libre	Libre	Libre
02:00 - 03:00	Libre	Libre	Libre	Libre	Libre	Libre	Libre
03:00 - 04:00	Libre	Libre	Libre	Libre	Libre	Libre	Libre
04:00 - 05:00	Libre	Libre	Libre	Libre	Libre	Libre	Libre
05:00 - 06:00	Libre	Libre	Libre	Libre	Libre	Libre	Libre
06:00 - 07:00	Libre	Libre	Libre	Libre	Libre	Libre	Libre
07:00 - 08:00	Libre	Libre	Libre	Libre	Libre	Libre	Libre
08:00 - 09:00	Libre	Libre	Libre	Libre	Libre	Libre	Libre
09:00 - 10:00	Libre	Libre	Libre	Libre	Libre	Libre	Libre
10:00 - 11:00	Libre	Libre	Libre	asdf	Libre	Libre	Libre
11:00 - 12:00	Libre	Libre	Libre	Libre	Libre	Libre	Libre
12:00 - 13:00	Libre	Libre	Libre	Libre	Libre	Libre	Libre
13:00 - 14:00	Libre	Libre	Libre	Libre	Libre	Libre	Libre
14:00 - 15:00	Libre	Libre	Libre	Libre	Libre	Libre	Libre

Dia

Lunes

Hora de inicio de la reserva

00:00

Hora de fin de la reserva

01:00

Reservar

Figura 54: Panel para reservar un espacio de tiempo del laboratorio remoto

La tercera de las opciones que ofrece la lista de laboratorios remoto es el poder acceder a la lista de reservas que tiene ese laboratorio remoto, las cuales mostraran el nombre del usuario que tiene la reserva, la fecha de cuando, la hora de inicio, la hora de fin y un botón para eliminar la reserva, en caso de que entráramos en el panel de reservas desde la lista de laboratorios remoto del usuario, solamente se verían las reservas propias. Además tiene un botón para poder ver las reservas pasadas, que se mostrarán de la misma forma que las actuales, solamente que no se permitirá eliminar la reserva, porque esta ya fue completada.

Arduino MEGA

Reservas anteriores

Usuario	Dia	Hora inicio	Hora fin	
Admin	2022-07-04	00:00:00	01:00:00	
NoAdmin	2022-07-07	00:00:00	01:00:00	

Figura 55: Panel que lista las reservas de los laboratorios remotos

La cuarta de las opciones que ofrece la lista de laboratorios remoto

permite modificar los datos de algún laboratorio específico y permite modificar tanto el nombre del mismo, la ruta de la carpeta contenedora de PDFs como la IP del dispositivo. A la hora de cambiar la carpeta contenedora de PDF permite crear una nueva marcando la opción de **Nueva ruta** y poniendo en el campo inferior el nombre de la misma, y si se desea eliminar la carpeta contenedora de PDFs y ningún otro laboratorio remoto utiliza la carpeta actual solamente se tiene que marcar la opción en la checkbox que se encuentra abajo de todos los campos.

Modificar laboratorio remoto

Id: 3

Placa: Casa

Ruta PDFs: placas/test3part

La placa NO esta siendo utilizada ahora mismo

Nombre

Ubicacion PDFs

Nombre nueva ruta

Ip

☐ ¿Desea eliminar tambien la antigua carpeta contenedora de pdf?

[Modificar placa](#)

Figura 56: Panel de modificación de laboratorios remoto

La quinta de las opciones que ofrece la lista de laboratorios remoto permite eliminar un laboratorio remoto, que no tenga ninguna reserva pendiente el día que se va a eliminar, y que para eliminarlo tienes la opción de eliminar también la carpeta contenedora de PDFs, siempre y cuando no la comparta con otro laboratorio remoto.

Eliminar laboratorio remoto

Id: 3

Placa: Casa

Ruta PDFs: placas/test3part

La placa NO esta siendo utilizada ahora mismo

Por favor escriba CONFIRMAR en el siguiente campo

☐ ¿Desea eliminar tambien su carpeta contenedora de pdf?

Borrar placa

Figura 57: Panel de eliminación de laboratorios remoto

La última de las opciones que ofrece la lista de laboratorios remoto permite acceder al panel del laboratorio remoto en específico, si se tiene una reserva hecha para ese momento si se accede desde el panel de laboratorios remoto de usuarios (comparten el mismo icono) o si se es un administrador, y estos paneles los veremos mas adelante de forma individual.

6.5.5. Panel de usuario

 **UCA** Inicio Admin Placas Desconectar

Panel de usuario

Información Contraseña Correo

Información

Nombre: Admin

Codigo uca: u00000000

Correo: admin@admin.com

Figura 58: Panel de usuarios

En el panel de usuarios se puede ver la información principal del usuario, como puede ser el nombre, el código UCA y el correo electrónico y se divide en 3 pestañas, siendo la primera la que permite ver la información, en la segunda pestaña se vera un pequeño formulario que permite modificar la contraseña y para hacerlo necesitas tener el correo confirmado, escribir la contraseña actual del usuario y la nueva contraseña, que

será introducida 2 veces para asegurar de que no se introduce de forma errónea. Y en la tercera pestaña se podrá modificar el correo simplemente añadiendo el nuevo correo, de la misma forma que con la contraseña introduciéndolo 2 veces para minimizar errores de escritura.

6.5.6. Panel de gestión de PDFs

Desconectar	
Nombre carpeta	
csv	→
LPC3	→
placa-2	→
prueba doc	→
test3part	→

Figura 59: Panel de gestión de PDF

Cuando se accede al panel de gestión de PDF se vera una tabla que comprende todas las carpetas contenedoras de PDF y donde podrá acceder a la carpeta contenedora de PDF específica que desea.

Desconectar	
Subir PDF	
Elegir archivo	No se ha seleccionado ningún archivo
Subir Archivo	
Nombre PDF	
prueba.pdf	🗑️ ➡️ ⬇️
prueba2.1.pdf	🗑️ ➡️ ⬇️

Figura 60: Panel de gestión de PDF de una carpeta específica

Una vez seleccionada que carpeta específica se quiere gestionar nos encontraremos que tenemos 5 opciones, la primera de todas permite subir nuevos archivos del tipo “.pdf” a la carpeta, las siguientes opciones ya son específicas a cada archivo PDF y serán expresadas en la fila específica de ese PDF y de esas opciones se puede:

- Eliminar el PDF de la carpeta.
- Copiar el PDF a otra carpeta.
- Mover el PDF a otra carpeta.
- Descargar el PDF al equipo del usuario.

6.5.7. Lista de laboratorio remoto (Usuario)

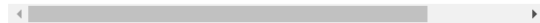
Desconectar

Placas disponibles	
Placa	Acciones
Casa	  →
placa-2	  →
LPC3	  →
SubirArchivoa	  →
Prueba doc	

Figura 61: Panel que lista todos los laboratorios remotos (vista de usuario)

Cuando un usuario no administrador inicia sesión se le moverá automáticamente a la ventana que lista todos los laboratorios remotos, pero la versión para usuario no administrador, la cual comparte varias opciones con las presentadas en la misma vista pero accedida al panel de administrador[6.5.4], como serían la opción para reservar un espacio de tiempo de un laboratorio remoto, poder ver las reservas pendientes y anteriores (en este caso serían solamente del usuario actual) y la de poder acceder al laboratorio remoto. La única función nueva que se incluye en este panel se puede ver en el laboratorio remoto llamado como “Prueba doc” el cual tiene un icono de una tarjeta con un candado y es que aun no se ha pedido acceso a ese laboratorio y mandará una petición para poder recibir los permisos de acceso al laboratorio remoto específico.

6.5.8. Laboratorio remoto Arduino MEGA



Enter your message here

Enviar

◀ ▶

Tiempo restante de la reserva:

00:00:00

Pines:

prueba.pdf

1-8-1

9-16

17-24

25-32

Archivo

Seleccionar archivo Ninguno archivo selec.

Conectar

Figura 62: Panel del laboratorio remoto de Arduino MEGA

En este panel podemos ver como en la parte inferior izquierda se encuentra una simulación de la consola serie de Arduino que sera usada para poder recibir información escrita en el canal serie por el Arduino MEGA y se podrá enviar comandos, usando la caja de texto que se encuentra debajo de donde se escribirá la información del canal serie, que serán enviadas desde el cliente al servidor principal y del mismo al equipo conectado al laboratorio remoto de Arduino MEGA.

En la parte derecha de la interfaz podemos ver un visor de PDF que mostrara el pdf que el usuario escoja en el selector que se encuentra justamente abajo del visor, a la izquierda de este selector se encuentra un temporizador que indicara cuanto queda para que la reserva actual se acabe, que se vera de color negro, pero que cuando se acerque a los últimos minutos de la reserva (antes de que eche al usuario del panel) se mostrará en rojo para indicar que se termina el tiempo, este selector ha sido adaptado del código escrito por “helloflask”[18].

Inferior al timer y al selector de PDF se encuentran los distintos botones que permitirán interconectar los sensores/actuadores a los distintos pines del Arduino MEGA (y en el panel del laboratorio remoto de la LPC4088) escogidos por el usuario, cada botón desplegará una tabla con 8 selectores, simplemente se despliega la tabla deseada, se selecciona que pin, del 1 al 32, se desea conectar al pin destino de Arduino MEGA y se pulsa el botón de **Conectar** que enviará esta selección de mapeo de componentes al servidor principal, del mismo al equipo conectado laboratorio remoto en cuestión y por último se mandará esta información a la placa controladora del laboratorio remoto que se encargará de realizar el mapeo correspondiente.

El mensaje enviado desde el servidor al equipo conectado a la placa controladora del laboratorio remoto se envía con el siguiente formato: **<Valor_primera_columna>-<Elección_Pin_Microcontrolador**, y si se selecciona más de un pin se verán separados por el carácter “|”, un ejemplo sería seleccionar en el desplegable del Pin 1 el Pin D23 de Arduino MEGA lo cual enviará como mensaje a la controladora del laboratorio remoto “**Pin 1-D23**” y si además seleccionamos por ejemplo el Pin 28 al Pin A13 del Arduino MEGA el servidor mandará el mensaje “**Pin 1-D23|Pin 28-A3**”.

El botón de **Conectar** también sirve para poder enviar un archivo “.ino” al laboratorio remoto del Arduino MEGA que sera usado para programar el microcontrolador, solamente se tiene que seleccionar un archivo que tenga el usuario con esa extensión y pulsar el botón y entonces el usuario mandara el archivo al servidor principal, este se lo reenviará al equipo conectado al laboratorio remoto del Arduino MEGA y este equipo programara el microcontrolador con el código de ese archivo.

6.5.9. Laboratorio remoto LPC4088

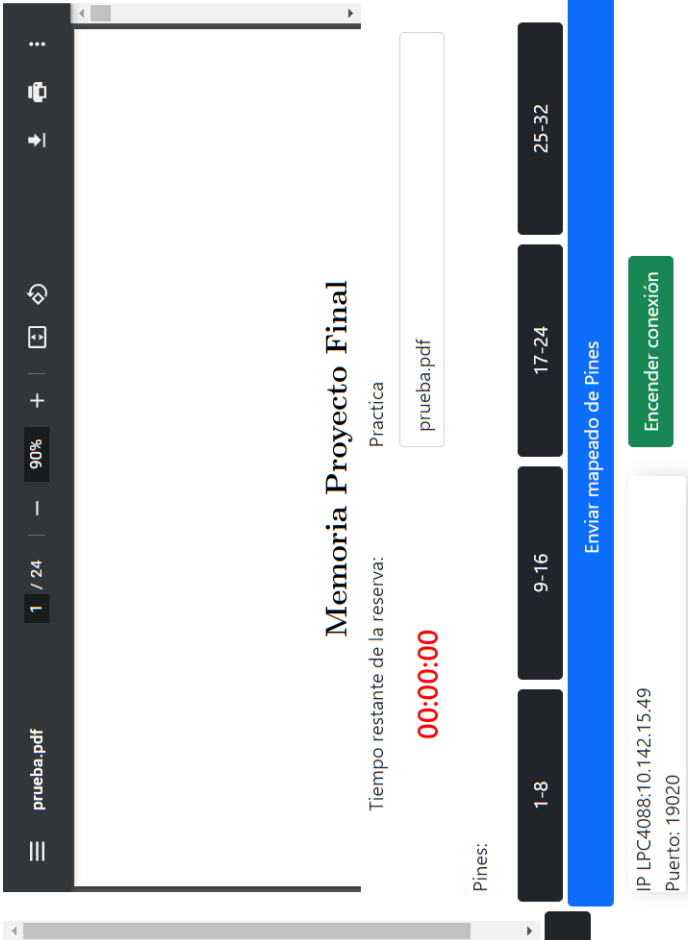


Figura 63: Panel del laboratorio remoto de la LPC4088

Este panel comparte muchas similitudes con el panel anterior, que era del laboratorio remoto de Arduino MEGA, siendo las únicas diferencias que en vez de tener una sección para subir un archivo “.ino” se tiene un botón que mandará la señal correspondiente para que se active o se desactive la conexión remota de la LPC y se pueda conectar con un uVision de manera remota y que en vez de poseer la simulación del terminal serie, se posee una serie de desplegables que actuarán como simuladores de dispositivos que requieren de un input físico como sería un teclado numérico o un joystick (que actualmente se encuentra abierto) y además de eso un botón que abrirá un pequeño tutorial de como conectar de forma remota el IDE de uVision.

7. Implementación

7.1. Estructura del código

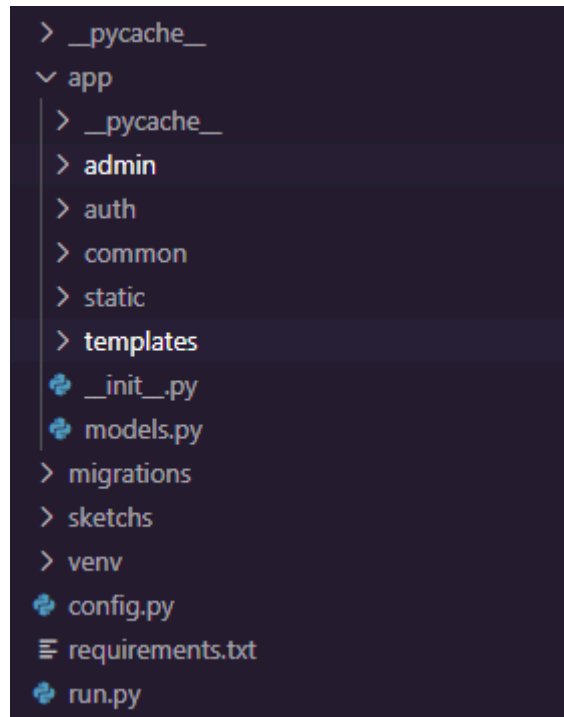


Figura 64: Estructura del código del servidor principal

El código del proyecto se ha organizado en una estructura de carpetas tal que en la carpeta raíz encontramos lo siguiente:

- **config.py:** Este archivo contendrá las distintas claves de configuración, como serían la clave secreta, la dirección del correo que enviara los emails por gmail, las credenciales de acceso a la base de datos, y se hace así para que no estén escritas directamente en el código ejecutable del software.
- **run.py:** Este archivo es el que inicia el servidor principal.
- **requirements.txt:** Archivo de texto plano que contiene todos las librerías de Python utilizados para poder instalar de forma fácil en cualquier otro equipo.
- **venv:** Carpeta que contiene el entorno virtual donde se ejecuta el software, que se hace para poder mantener un mayor control de que librerías funcionan en el proyecto, sin que estas librerías puedan tener incompatibilidades de versiones o incompatibilidades con otras librerías.
- **migrations:** Carpeta que contendrá los criptas de creación y modificación a la base de datos, para que se pueda hacer de forma sencilla y cómoda modificando el código Python, y todo esto gracias al módulo de flask-migrations.

- **sketch:** Carpeta que almacenará de forma temporal los sketches de Arduino que son recibidos por el usuario para ser enviados al laboratorio remoto de Arduino MEGA.
- **app:** Carpeta que contiene todo el código de la aplicación web.

La carpeta app como podemos ver contiene los siguientes archivos y carpetas:

- **__init__.py:** Este archivo es llamado en el momento en el que se inicia el programa y se encarga de incluir los distintos módulos de los que se compone el proyecto.
- **models.py:** Este archivo contiene el código de las distintas tablas de la base de datos y las consultas a la base de datos que serán empleadas en el código de la aplicación web.
- **templates:** Esta carpeta contiene las plantillas básicas y genéricas de las distintos archivos HTML que serán usadas en los distintos archivos HTML.
- **static:** Esta carpeta contiene todos los archivos estáticos que serán empleados para el Frontend, como serían los distintos archivos css de bootstrap, los archivos JavaScript empleados en la aplicación web, los archivos PDF que se pueden visualizar en los paneles de los laboratorios remotos y los archivos que contienen las opciones de mapeo de sensores/actuadores en los laboratorios remotos.
- **admin:** Esta carpeta el módulo que implementa las funciones de administrador de la aplicación web.
- **auth:** Esta carpeta el módulo que implementa las funciones de usuarios, como creación, verificación, cambio de permisos,... de la aplicación web.
- **common:** Esta carpeta el módulo que implementa las funciones de comunes, que pueden usar todos los usuarios, de la aplicación web.

Las carpetas que contiene los módulos de la aplicación web se estructuran de la siguiente forma:

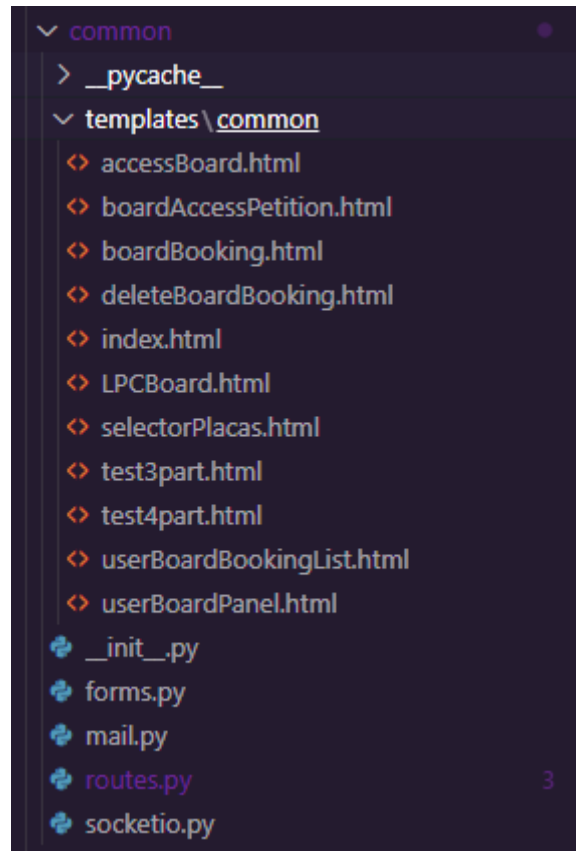


Figura 65: Estructura de la carpeta de los módulos de la aplicación web

- **templates\nombre_modulo:** Carpeta que contendrá los distintos archivos HTML respectivas a las vistas del modulo en específico.
- **__init__.py:** Este archivo es llamado en el momento en el que se inicia el programa y se encarga de incluir todas las funciones repartidas en los distintos archivos del módulos cuando se inicia el servidor, este archivo es obligatorio en todos los módulos del proyecto.
- **forms.py:** En este archivo se codifican los distintos formularios, por los que los usuarios introducirán datos al sistema, en las vistas respectivas al módulo, y se escriben utilizando la librería de Flask.WTForm, no todos los módulos tiene que tener este archivo.
- **mail.py:** En este archivo se implementa la lógica necesaria para que el módulo en específico pueda realizar la generación de correos electrónicos y los envíe al usuario correcto, no todos los módulos tiene que tener este archivo.
- **routes.py:** En este archivo se implementa la lógica detrás de todas las vistas del módulo, desde todas la comprobaciones necesarias cuando se carga la página, que formularios utilizará, que archivo HTML debe de cargar el cliente, procesar los datos del formulario

recibidos..., este archivo es obligatorio en todos los módulos del proyecto.

- **socketio.py**: En este archivo se implementa la lógica detrás del protocolo de comunicación SocketIO dentro de las vistas asociadas a este modulo, tanto la recepción de mensajes, las respuestas que envía, como hasta la gestión de los usuarios en lo que Flask-SocketIO llama “rooms” para poder separar los mensajes de distintos usuarios a pesar de usar canales con el mismo nombre, no todos los módulos tiene que tener este archivo.

7.1.1. Modularidad de la aplicación web

Para la inclusión de los distintos módulos en la aplicación web se ha empleado el sistema de Blueprints que ofrece Flask, que permite realizar componentes de la aplicación para separar las diferentes secciones que serán individuales de cada módulo y permite que los problemas de un blueprint no se expandan a otro blueprint de forma sencilla, permitiendo un mejor mantenimiento del código a futuro en aplicaciones de tamaño medio o grande.

Para implementar los blueprints se hace de la siguiente manera:

```
1  #En los archivos __init__.py de cada modulo se debe
   incluir el siguiente codigo
2
3  #Se importa la logica de los Blueprint de flask
4  from flask import Blueprint
5
6  #Aqui nombramos el blueprint y lo asignamos en una
   variable
7  commonBp = Blueprint('common', __name__, template_folder=
   'templates')
8
9  #Se importan los diferentes archivos que componen el
   modulo
10 from . import routes , socketio , mail
11
12
13 #####
14
15 #En el archivo __init__.py que se encuentra en la carpeta
   app se incluye el siguiente codigo
16 #Registro de los Blueprints
17 #Importamos la variable que almacena el blueprint
18 from .auth import authBp
19 #Registramos el blueprint en la app
20 app.register_blueprint(authBp)
21
22 from .common import commonBp
23 app.register_blueprint(commonBp)
24
25 from .admin import adminBp
26 app.register_blueprint(adminBp)
```

7.2. Enrutamiento de las vistas

En esta sección mostraremos el código que contiene el enrutamiento a todos las vistas de la aplicación web

```
1  #Enrutamiento common
2  app.add_url_rule('/', methods=['GET'], endpoint='
index')
3
4  app.add_url_rule('/lista-placas', methods=['GET', 'POST']
, endpoint='common.userBoardPanel')
5
6  app.add_url_rule('/peticion-acceso-placa/<int:id>',
methods=['GET', 'POST'], endpoint='common.
boardAccessPetition')
7
8  app.add_url_rule('/reservas/<int:id>/<int:week>', methods
=['GET', 'POST'], endpoint='common.boardBooking')
9
10 app.add_url_rule('/acceso-placa/<int:id>', methods=['GET'
, 'POST'], endpoint='common.accessBoard')
11
12 app.add_url_rule('/eliminar-reserva/<int:id>', methods=['
GET', 'POST'], endpoint='common.deleteBoardBooking')
13
14
15 #Enrutamiento auth
16 app.add_url_rule('/signup', methods=['GET', 'POST'],
endpoint='auth.signup')
17
18 app.add_url_rule('/confirmar-email', methods=['GET'],
endpoint='auth.emailConfirmation')
19
20 app.add_url_rule('/confirm/<string:token>', methods=['GET'
, 'POST'], endpoint='auth.confirmMail')
21
22 app.add_url_rule('/restablecer-contrasena', methods=['GET'
, 'POST'], endpoint='auth.passwordReset')
23
24 app.add_url_rule('/restablecer-contrasena/<string:token>'
, methods=['GET', 'POST'], endpoint='auth.
resetChangePassword')
25
26 app.add_url_rule('/login', methods=['GET', 'POST'],
endpoint='auth.login')
27
28 app.add_url_rule('/logout', methods=['GET', 'POST'],
endpoint='auth.logout')
29
30 app.add_url_rule('/panel-usuario', methods=['GET', 'POST'
], endpoint='auth.userPanel')
31
32 app.add_url_rule('/lista-reservas/<int:id>', methods=['
GET', 'POST'], endpoint='auth.userBoardBookingList')
33
34 app.add_url_rule('/lista-reservas-antiguas/<int:id>',
methods=['GET', 'POST'], endpoint='auth.
pastUserBoardBookingList')
35
```

```

36
37 #Entrutamiento admin
38 app.add_url_rule('/admin/panel', methods=['GET'],
39                 endpoint='admin.adminPanel')
40
41 app.add_url_rule('/admin/lista-usuarios', methods=['GET',
42                 'POST'], endpoint='admin.userList')
43
44 app.add_url_rule('/admin/crear-usuario', methods=['GET',
45                 'POST'], endpoint='admin.createUser')
46
47 app.add_url_rule('/admin/borrar-usuario/<int:id>',
48                 methods=['GET', 'POST'], endpoint='admin.deleteUser')
49
50 app.add_url_rule('/admin/cambiar-usuario/<int:id>',
51                 methods=['GET', 'POST'], endpoint='admin.updateUser')
52
53 app.add_url_rule('/admin/lista-placas', methods=['GET', '
54                 POST'], endpoint='admin.boardList')
55
56 app.add_url_rule('/admin/anadir-placa', methods=['GET', '
57                 POST'], endpoint='admin.addBoard')
58
59 app.add_url_rule('/admin/eliminar-placa/<int:id>',
60                 methods=['GET', 'POST'], endpoint='admin.deleteBoard')
61
62 app.add_url_rule('/admin/modificar-placa/<int:id>',
63                 methods=['GET', 'POST'], endpoint='admin.editBoard')
64
65 app.add_url_rule('/admin/acceso-placas/<int:id>', methods
66                 =['GET', 'POST'], endpoint='admin.boardAccessList')
67
68 app.add_url_rule('/admin/permitir-acceso-placa/<int:
69                 boardId>/<int:userId>', methods=['GET', 'POST'], endpoint
70                 ='admin.allowBoardAccess')
71
72 app.add_url_rule('/admin/denegar-acceso-placa/<int:
73                 boardId>/<int:userId>', methods=['GET', 'POST'],
74                 endpoint='admin.denyBoardAccess')
75
76 app.add_url_rule('/admin/lista-reservas/<int:id>',
77                 methods=['GET', 'POST'], endpoint='admin.boardBookingList'
78                 )
79
80 app.add_url_rule('/admin/lista-reservas-antiguas/<int:id>
81                 ', methods=['GET', 'POST'], endpoint='admin.
82                 pastBoardBookingList')
83
84 app.add_url_rule('/admin/lista-carpetas-pdf', methods=['
85                 GET'], endpoint='admin.pdfFolderList')
86
87 app.add_url_rule('/admin/lista-pdf/<string:folder>',
88                 methods=['GET', 'POST'], endpoint='admin.pdfList')
89
90 app.add_url_rule('/admin/borrar-pdf/<string:folder>/<
91                 string:pdf>', methods=['GET', 'POST'], endpoint='admin.
92                 deletePDF')

```

```

72 app.add_url_rule('/admin/copiar-pdf/<string:folder>/<string:pdf>', methods=['GET', 'POST'], endpoint='admin.  
copyPDF')
73
74 app.add_url_rule('/admin/mover-pdf/<string:folder>/<string:pdf>', methods=['GET', 'POST'], endpoint='admin.  
movePDF')
75
76 app.add_url_rule('/admin/descargar-pdf/<string:folder>/<string:pdf>', methods=['GET'], endpoint='admin.  
downloadPDF')

```

Para la utilización de estas rutas cada función asociada a una vista debe tener el decorador “@nombre_blueprint.endpoint('nombre_endpoint)’”

8. Pruebas del software

A la hora de realizar las pruebas del software se ha realizado utilizando a varios usuarios que han ido probando las distintas partes tanto por separado como en su conjunto, para asegurar de encontrar todos los errores posibles en la aplicación web.

Para los distintos formularios se han intentado poner más y menos caracteres que el rango establecido o escribirlos con formatos erróneos, o hasta incluir comandos HTML y SQL para revisar que no se realice ningún comando sin control en el software de la aplicación. Además, también se han probado todas las interacciones de la misma aplicación consigo misma, con la base de datos y con el sistema de gestión de archivos como serían los PDF, de distintas formas para asegurar que no generara un crasheo inesperado de ninguna forma.

Para la comunicación con los laboratorios remotos se ha simulado tener varios equipos que actuarían de laboratorio remoto enviando y recibiendo mensajes para asegurar que no se colara ningún mensaje de un laboratorio remoto en una interfaz que no le corresponda y esos mensajes se han ido comprobando de automáticamente para asegurar que no se pierde ningún carácter en mitad de la comunicación.

A la hora de la integración con los laboratorios remotos, además de la comunicación, las pruebas se hicieron con un proyecto de uVision que encendía y apagaba 4 leds de forma periódica y mientras el JLink y el microcontrolador LPC4088 estaban conectados a un equipo, otro ordenador actuando de cliente, conectado a una red móvil para asegurarnos que de no se conecta de forma local, sino a través de la VPN de la UCA, usaba el uVision para controlar el microcontrolador, tanto subiendo archivos como usando el modo de Debug.

Para la interacción con el microcontrolador Arduino MEGA, se colocaron 2 Arduinos MEGA conectados para asegurarnos de que solamente se programara el Arduino MEGA que nos interesaba y no modificar por

accidente el programa que se encuentra en el controlador del laboratorio remoto, y además se mandaron distintos archivos “.ino” para asegurar que se cargaban correctamente en el microcontrolador de Arduino MEGA.

Parte III

Epilogo

9. Conclusiones

Para este Trabajo de Fin de Grado se ha desarrollado una aplicación web para poder conectar distintos laboratorios remotos que se construirán en la Escuela Superior de Ingeniería del Campus de Puerto Real de la Universidad de Cádiz. Dicha aplicación permitirá que los alumnos puedan realizar sus asignaciones de practicas de forma sencilla desde fuera de los laboratorios, permitiendo que puedan trabajar desde cualquier lugar en el que se encuentren, en cualquier franja de tiempo.

El trabajo ha sido supervisado por Arturo Morgado Estévez y Eduardo Alejandro Romero Bruzón, ambos miembros del Departamento Ingeniería en Automática, Electrónica, Arquitectura y Redes de Computadores, y se les ha ido enseñando los distintos prototipos para poder realizar los cambios pertinentes al trabajo de la forma mas eficiente posible.

Ahora se van a revisar los distintos requisitos funcionales, que se propusieron al inicio del proyecto[6] en la siguiente tabla [40] en la cual hemos marcado como se han completado todos los requisitos funcionales que se propusieron al iniciar el proyecto.

RF-01 - Permitir crear usuarios	✓
RF-02 - Poder verificar el correo de los usuarios creados	✓
RF-03 - Permitir modificar contraseña y email del usuario	✓
RF-04 - Poder modificar la contraseña si el email del usuario esta verificado	✓
RF-05 - Permitir añadir nuevos laboratorios remotos	✓
RF-06 - Mostrar PDF de practicas cuando se trabaje con laboratorios remotos	✓
RF-07 - Gestionar que PDF de practicas se mostraran en cada laboratorio remoto	✓
RF-08 - Sistema de reservas individual para cada laboratorio remoto	✓
RF-09 - Gestionar el acceso a cada laboratorio remoto	✓
RF-10 - Comunicar el servidor principal con los laboratorios remotos	✓
RF-11 - Visualizar en la interfaz web la webcam del laboratorio remoto	✓
RF-12 - Panel de comunicación entre la interfaz web y la placa de control del laboratorio remoto	✓
RF-13 - Subir archivos de código “.ino” al laboratorio remoto del microcontrolador Aduino MEGA	✓
RF-14 - Comunicar la interfaz web con el Arduino MEGA actuando como consola Serie para entrada y salida	✓
RF-15 - Permitir activar el protocolo de servidor remoto JLink del laboratorio remoto del microcontrolador LPC4088	✓
RF-16 - Diseñar panel de simulación de teclado y joystick	✓

Cuadro 40: Revisión de requisitos funcionales

10. Experiencia

Este proyecto ha sido el proyecto de mayor complejidad y tamaño en general que se ha trabajado por el alumno, y además de esto se ha tenido que realizar en su mayoría de forma solitaria, con la puntual ayuda de los tutores y de algún compañero del laboratorio D12 el cual es el **Laboratorio de Robótica Aplicada**, pero en su completa mayoría en solitario, lo cual complico bastante las fases del proyecto que no se centraban tanto en la programación como tal, sobretodo a la hora del análisis previo del proyecto y la redacción de un documento correcto para poder entregar junto al trabajo.

Además de esto, el alumno ha cursado la especialidad de **Ingeniería en Computadores** lo cual esta bastante alejado al como se trabaja a la hora de programar aplicaciones web, cosa que complico los primeros pasos del desarrollo de la misma, junto a que se debieron realizar varios cambios a las mismas cosas por no estar desarrolladas de la mejor forma, todo fruto de la inexperiencia del alumno en el ámbito del desarrollo web, aunque al menos se pudo aprovechar el conocimiento previo en Python empleando un microframework del mismo llamado Flask, que al ser un “micro”Framework lo que implica es que incluye la funcionalidad mínima para que el software funcione como una aplicación web, pero la mayoría de cosas tendrán que ser importadas de librerías adicionales e implementadas por el alumno, lo cual sirvió para poder aprender mas sobre la programación en Python y de aplicaciones web.

11. Trabajo futuro

El software de la aplicación puede ser mejorado en un futuro de varias formas:

- La primera de todas sería de forma estética, como el desarrollador del mismo es un alumno el cual pertenece a la especialidad de Computadores del Grado en Ingeniería Informática ha tenido bajo o nulo acercamiento al entorno web, lo cual hizo que se le complicara de mas las cosas mas referentes al tema del diseño e implementación del Frontend aunque esto ha sido mitigado un poco gracias a la utilización de Jinja2 para integrar los datos recibidos del Backend en el Frontend.
- Añadir interactividad con JavaScript, aunque el software funciona de forma correcta siempre se puede añadir algo mas de interactividad con JavaScript para poder ofrecer una mejor calidad de servicio al usuario, como seria filtrando un poco algunas tablas o añadiendo algunas automatizaciones que simplifiquen la tarea del usuario.
- Mejorar la integración con las webcams, porque con lo complicado que ha sido conseguir que los usuarios puedan conectarse de forma

relativamente cómoda al JLink para poder emplear el uVision para poder programar y depurar a distancia desde sus hogares se decidió utilizar una aplicación de apoyo para la utilización de la webcam, y en un futuro se podrá implementar la lógica para no necesitar un programa de apoyo para visualizar las webcams.

- Añadir integración con moodle para poder subir al campus virtual las practicas desde la interfaz web, para que directamente el alumno pueda trabajar con el laboratorio remoto y desde el mismo subir las practicas al moodle.
- Incluir las mejoras de usabilidad que se detecten en un futuro.

12. Manual de configuración uVision para poder conectar de forma remota al laboratorio remoto de la LPC4088

Lo primero necesario es tener instalada la VPN de la UCA siguiendo el tutorial que ofrece la pagina web de la UCA [17].

Una vez tengas la interfaz de la VPN de la UCA de la siguiente forma, que es como se ve cuando estas conectado a la VPN:

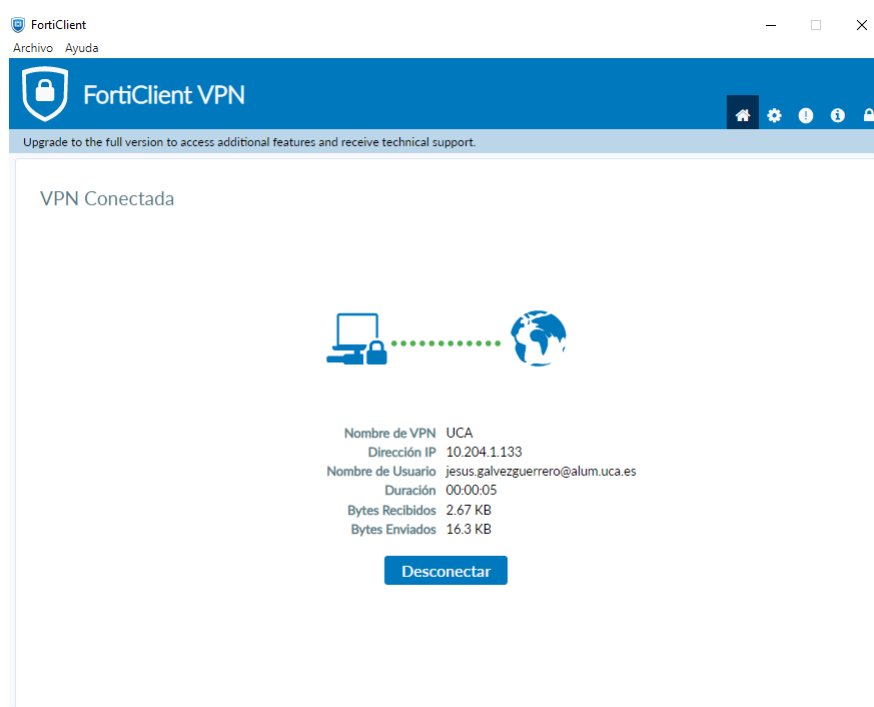


Figura 66: VPN de la UCA cuando esta conectada

Una vez esto se ha cumplido procederemos a configurar el proyecto de uVision para poder conectar de manera remota al laboratorio remoto de la LPC4088, y estas acciones se deberán hacer para cada proyecto con el que el alumno desee trabajar.

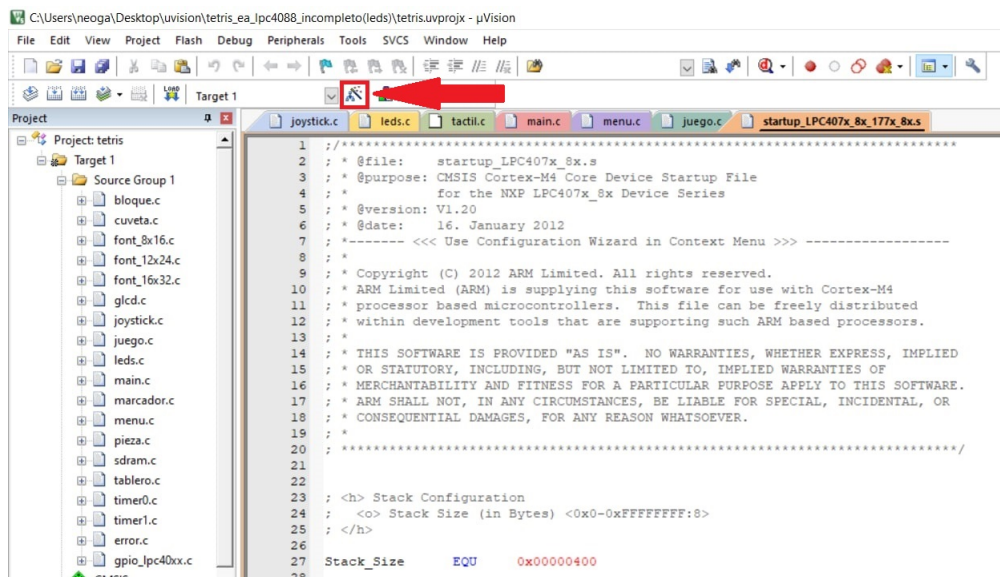


Figura 67: uVision: Abrir configuración del proyecto

Primero abriremos el proyecto con el que se desea trabajar de forma remota y pulsamos en el botón indicado para abrir el panel de configuración del proyecto, este panel también puede ser abierto pulsando la combinación de teclas **Alt+F7** o accediendo al menú **Project->Options for target "Nombre del proyecto"**.

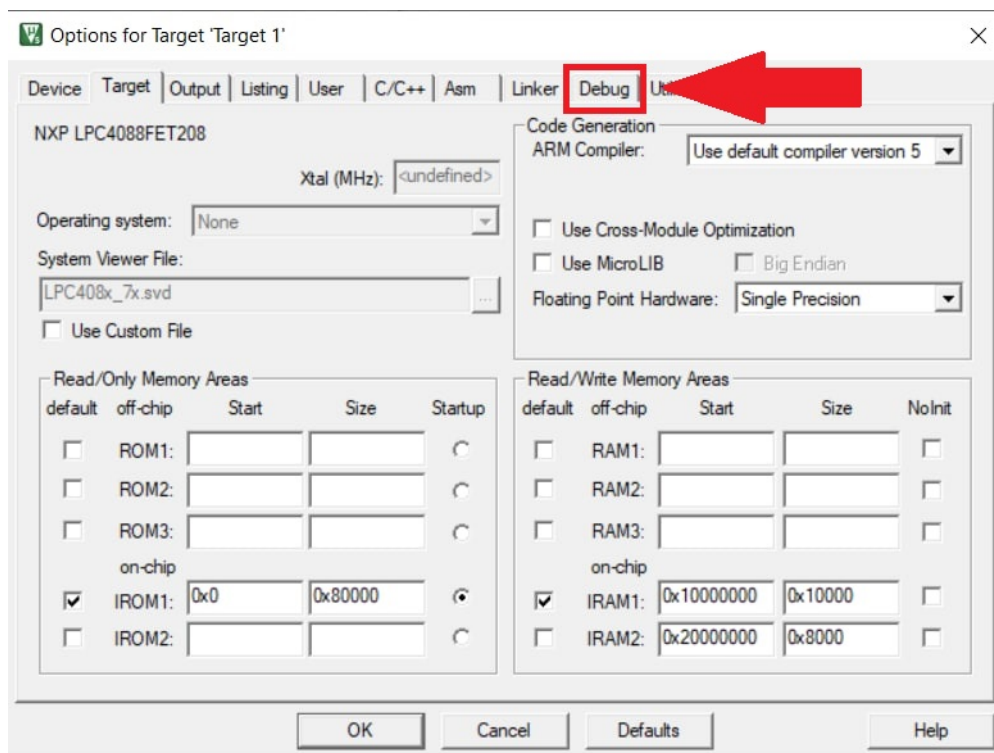


Figura 68: uVision: Acceder pestaña Debug de la configuración del proyecto

Una vez abierto el panel de configuración del proyecto se procederá a

acceder a la pestaña llamada **Debug** del panel.

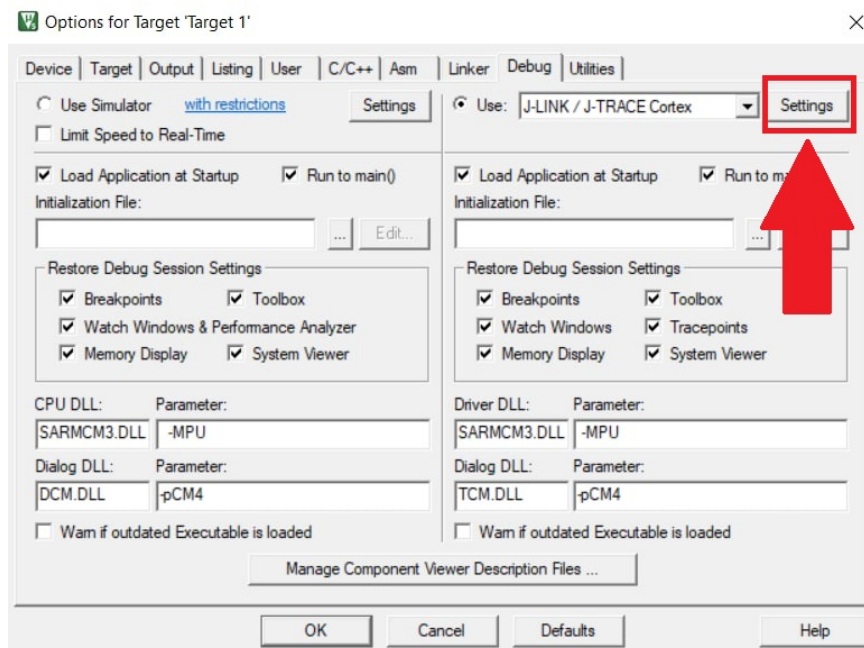


Figura 69: uVision: Acceder a la configuración del Debugger JLink

En la pestaña **Debug** de la configuración del proyecto accederemos a la configuración del dispositivo de debug pulsando el botón de **Settings** marcado en la imagen[69]

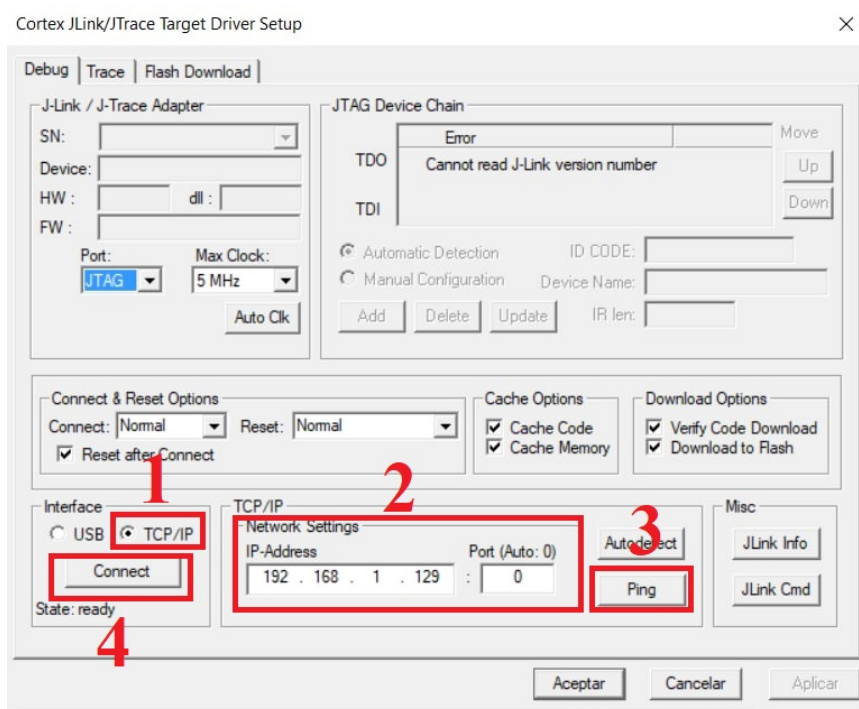


Figura 70: uVision: Acceder a la configuración del Debugger JLink

Ya en el panel de configuración del debugger JLink procederemos a marcar la opción de interfaz **TCP/IP**[1], posteriormente en la sec-

ción 2 se introducirá la dirección IP del laboratorio remoto que tiene la LPC4088, que es **10.142.15.49**, y el puerto de la conexión, que es **19020**. Posteriormente se pulsará el botón de Ping[3] para poder asegurar que hay conectividad con el laboratorio remoto, y cuando se confirme la conectividad se pulsará el botón **Connect**[4] para conectar el uVision con el laboratorio remoto de la LPC4088.

13. Manual de instalación

Para la instalación debemos tener en cuenta 2 factores, el primero es la instalación del software necesario para comunicar el laboratorio remoto con el servidor principal, el cual es bastante simple, y tenemos 2 scripts ya preparados para ello, el script para el laboratorio remoto de Arduino MEGA y el script para el laboratorio remoto de la LPC4088.

Primero de todo, instalaremos Python 3 en el servidor principal y los equipos conectados a los laboratorios remotos, Python 3 se descarga directamente de la página principal de Python[19]. Y una vez descargado el instalador de Python3 se procederá a seguir los pasos que el mismo instalador va indicando completar la instalación.

Python incluye de base una librería llamada **venv** que permite generar entornos de trabajo virtuales con los cual un usuario puede descargar librerías sin miedo a que estas interfieran con otras versiones de la misma librería, o con otras librerías, de otros proyectos, para poder genera un entorno virtual usaremos el comando **python -m venv nombre_entorno_virtual** o si tenemos instala más de una versión de python, como sería Python 2 y el 3(necesario para este proyecto) usaremos el siguiente comando **python3 -m venv nombre_entorno_virtual**, nosotros crearemos el entorno de trabajo en la carpeta **RemoteLab** incluida en los entregables de este TFG y lo llamaremos venv.

Python incluye un sistema que permite instalar de forma cómoda y sencilla sus distintas librerías, el cual se llama **pip** el cual usaremos junto al archivo requirements.txt incluido en la carpeta **RemoteLab** del software, para instalar todas las librerías necesarias para ejecutar el software con un solo comando, pero antes de hacer el comando de instalación procederemos a activar el entorno virtual, situándonos en la carpeta **RemoteLab** e introduciendo el siguiente comando **venv\Scripts\activate**.

Tras ejecutar el comando veremos como el prompt de nuestra consola de comandos cambiara a incluir el nombre del entorno virtual entre paréntesis al principio del mismo, esto es lo que indicara que hemos activado correctamente el entorno virtual.

Una vez realizado esto procederemos a instalar todas las librerías necesarias para ejecutar el software usando el comando **pip install -r**

```
Microsoft Windows [Versión 10.0.19044.1766]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\neoga\Desktop\Remota lab\RemoteLab>venv\Scripts\activate

(venv) C:\Users\neoga\Desktop\Remota lab\RemoteLab>
```

Figura 71: Activar entorno virtual

requirements.txt lo cual instalara todas las librerías que estarán escritas en el archivo **requirements.txt**.

Para este momento también tendremos que tener instalada la base de datos de PostgreSQL, la cual se puede descargar desde su página oficial[20] y simplemente tendremos que seguir las instrucciones que nos indica el instalador de la misma, solo que en algún momento se nos pedirá generar un superusuario para administrar la base de datos, nos aseguraremos de guardar bien estas credenciales para no olvidarlas.

Con todo esto instalado procederemos a introducir los datos necesarios en el archivo **config.py** para que la aplicación web pueda utilizar la cuenta de correo que le asignemos, tenga acceso a la base de datos y todo lo necesario.

Con todo esto ya preparado del lado del servidor principal, procederemos a usar los siguientes comandos:

- **flask db init** : El cual creara una carpeta en el servidor principal que servirá para guardar los scripts que actualizaran la base de datos cuando el administrador lo desee(estructuralmente, no a nivel de datos).
- **flask db migrate -m “Nombre de la modificación”** : Creara un script de migración que servirá para actualizar la estructura de tablas de la base de datos, en este caso, como seria el primer script de actualización también creara la base de datos.
- **flask db upgrad** : Se encargará de ejecutar el script que actualizará la base de datos, tras este paso tendremos creada la base de datos.

Con esto la parte del servidor principal ha sido instalada y para ejecutarlo procederemos a utilizar el comando **flask run --host=0.0.0.0**

13.1. Instalación del software necesario para conectar los laboratorios remotos al servidor principal

Para ambos laboratorios remotos usaremos Yawcam[21] para poder grabar la imagen de webcam y poder visualizarla en el cliente, para eso

simplemente se descarga la aplicación desde su página principal y se siguen los pasos del instalador para instalarla, luego de esto se abre el programa y simplemente se hace click en el botón de **Enable** que está a la derecha de **Stream** y de **HTTP**, para que se active círculo rojo y se cambie el botón a disable.

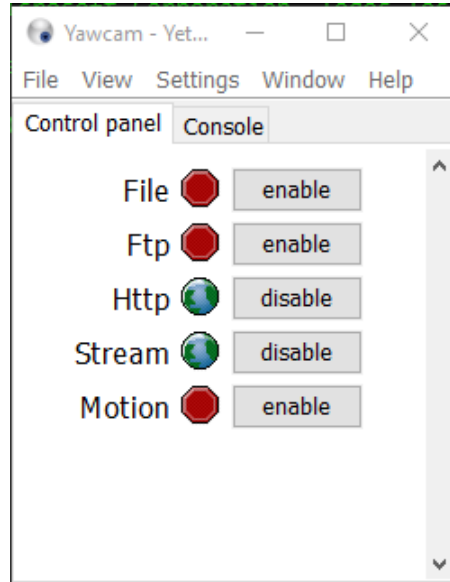


Figura 72: Activar Yawcam

Para el equipo conectado al laboratorio remoto de Arduino MEGA se instalará Python y se creará un entorno virtual para instalar las librerías igual que con el servidor Principal, creando un entorno virtual e instalando las librerías necesarias utilizando el archivo **requirements.txt**, solamente que aquí solamente necesitaremos indicar dentro del código cuál es la IP del servidor y cuáles son los puertos **COM** al que estarán conectados el Arduino MEGA y la controladora del laboratorio remoto. También se necesitará descargar y poner en la misma carpeta que el script de Python el ejecutable de Arduino-CLI[22]

Con el Arduino CLI instalado se deberá poner la siguiente lista de comandos:

- **arduino-cli core update-index** : Para iniciar el reconocimiento de placas Arduino en una instalación limpia.
- **arduino-cli board list** : Para revisar que placas hay conectadas y que core utilizan para funcionar.
- **arduino-cli core install arduino:avr:mega** : Para instalar el core necesario para poder programar el Arduino MEGA.

Para el equipo conectado al laboratorio remoto de la LPC4088 se seguirán los mismos pasos que para el laboratorio remoto de Arduino MEGA, solamente que aquí no se necesitara indicar el puerto serie del Arduino MEGA ni descargar el ejecutable del Arduino-CLI, en vez de eso se tendrá que descargar el software de servidor remoto de SEGGER[23] e

indicarlo dentro del código del archivo Python para el laboratorio remoto.

Ambos scripts están en sus propias carpetas llamadas **Laboratorio Remoto Arduino MEGA** y **Laboratorio Remoto LPC4088** situadas en el archivo “.zip” entregado junto a esta documentación.

Anexos

A. Tabla de ejemplo del mapeado de interconexión de pines para los laboratorios remotos

A continuación se mostrará una tabla de ejemplo para usar de plantilla a la hora de crear los nuevos documentos “.csv” para el mapeado de la interconexión de pines de los nuevos laboratorios remotos.

Pin 1	Nombre 1	PDM	PDM	PDM	PDM	PDM	PDM	PDM	PDM
Pin 2	Nombre 2	PDM	PDM	PDM	PDM	PDM	PDM	PDM	PDM
Pin 3	Nombre 3	PDM	PDM	PDM	PDM	PDM	PDM	PDM	PDM
Pin 4	Nombre 4	PDM	PDM	PDM	PDM	PDM	PDM	PDM	PDM
Pin 5	Nombre 5	PDM	PDM	PDM	PDM	PDM	PDM	PDM	PDM
Pin 6	Nombre 6	PDM	PDM	PDM	PDM	PDM	PDM	PDM	PDM
Pin 7	Nombre 7	PDM	PDM	PDM	PDM	PDM	PDM	PDM	PDM
Pin 8	Nombre 8	PDM	PDM	PDM	PDM	PDM	PDM	PDM	PDM
Pin 9	Nombre 9	PDM	PDM	PDM	PDM	PDM	PDM	PDM	PDM
Pin 10	Nombre 10	PDM	PDM	PDM	PDM	PDM	PDM	PDM	PDM
Pin 11	Nombre 11	PDM	PDM	PDM	PDM	PDM	PDM	PDM	PDM
Pin 12	Nombre 12	PDM	PDM	PDM	PDM	PDM	PDM	PDM	PDM
Pin 13	Nombre 13	PDM	PDM	PDM	PDM	PDM	PDM	PDM	PDM
Pin 14	Nombre 14	PDM	PDM	PDM	PDM	PDM	PDM	PDM	PDM
Pin 15	Nombre 15	PDM	PDM	PDM	PDM	PDM	PDM	PDM	PDM
Pin 16	Nombre 16	PDM	PDM	PDM	PDM	PDM	PDM	PDM	PDM
Pin 17	Nombre 17	PDM	PDM	PDM	PDM	PDM	PDM	PDM	PDM
Pin 18	Nombre 18	PDM	PDM	PDM	PDM	PDM	PDM	PDM	PDM
Pin 19	Nombre 19	PDM	PDM	PDM	PDM	PDM	PDM	PDM	PDM
Pin 20	Nombre 20	PDM	PDM	PDM	PDM	PDM	PDM	PDM	PDM
Pin 21	Nombre 21	PDM	PDM	PDM	PDM	PDM	PDM	PDM	PDM
Pin 22	Nombre 22	PDM	PDM	PDM	PDM	PDM	PDM	PDM	PDM
Pin 23	Nombre 23	PDM	PDM	PDM	PDM	PDM	PDM	PDM	PDM
Pin 24	Nombre 24	PDM	PDM	PDM	PDM	PDM	PDM	PDM	PDM
Pin 25	Nombre 25	PDM	PDM	PDM	PDM	PDM	PDM	PDM	PDM
Pin 26	Nombre 26	PDM	PDM	PDM	PDM	PDM	PDM	PDM	PDM
Pin 27	Nombre 27	PDM	PDM	PDM	PDM	PDM	PDM	PDM	PDM
Pin 28	Nombre 28	PDM	PDM	PDM	PDM	PDM	PDM	PDM	PDM
Pin 29	Nombre 29	PDM	PDM	PDM	PDM	PDM	PDM	PDM	PDM
Pin 30	Nombre 30	PDM	PDM	PDM	PDM	PDM	PDM	PDM	PDM
Pin 31	Nombre 31	PDM	PDM	PDM	PDM	PDM	PDM	PDM	PDM
Pin 32	Nombre 32	PDM	PDM	PDM	PDM	PDM	PDM	PDM	PDM

Los colores de esta tabla indican lo siguiente:

- **Azul:** Representa aquella columna que se mantendrá invariable en todos los documentos “.csv” puesto que sirve de identificador interno para el software.
- **Verde:** La columna en este color representa el nombre que se le quiere dar al pin en la interfaz web, es decir, el desplegable que contendrá las opciones de conexión del microcontrolador LPC4088 o Arduino MEGA.
- **Amarillo:** En esta sección se colocarán los Pines Destino del Microcontrolador, abreviado a PDM en la tabla, es decir, las opciones de conexión que tendrá el alumno a la hora de conectar el Pin X perteneciente a la misma fila y que tendrá el nombre representado en la segunda columna del documento “.csv”.

Un ejemplo se podría ver en la tabla[51] donde podemos ver que al Pin 1 se le identificara en la interfaz web como **Joystick JP3-1** y tendrá como opciones de conexión del microcontrolador LPC4088 los pines **P0.2, P0.12, P0.19, P4.22, P2.22, P3.16, P3.22 y P3.29**.

Referencias

- [1] “Pagina principal de WebLab-Deusto,” <https://weblab.deusto.es/website/>, *Ultima vez consultado el 20/04/2022.*
- [2] “Documentación oficial de la librería Flask_login,” <https://flask-login.readthedocs.io/en/latest/>, *Ultima vez consultado el 20/02/2022.*
- [3] “Documentación oficial de la librería Flask_sqlalchemy,” <https://flask-sqlalchemy.palletsprojects.com/en/2.x/>, *Ultima vez consultado el 10/05/2022.*
- [4] “Documentación oficial de la librería Flask_sqlalchemy,” <https://flask-migrate.readthedocs.io/en/latest/>, *Ultima vez consultado el 10/05/2022.*
- [5] “Documentación oficial de la librería Flask_sqlalchemy,” <https://flask-wtf.readthedocs.io/en/1.0.x/>, *Ultima vez consultado el 10/05/2022.*
- [6] “Documentación oficial de la WTForm,” <https://wtforms.readthedocs.io/en/3.0.x/>, *Ultima vez consultado el 01/06/2022.*
- [7] “Documentación oficial de la librería Bootstrap_Flask,” <https://bootstrap-flask.readthedocs.io/en/stable/>, *Ultima vez consultado el 24/06/2022.*
- [8] “Documentación oficial de la librería Flask_JSGlue,” <https://stewartpark.github.io/Flask-JSGlue/>, *Ultima vez consultado el 22/05/2022.*
- [9] “Documentación oficial de la librería de Werkzeug,” <https://werkzeug.palletsprojects.com/en/2.0.x/>, *Ultima vez consultado el 16/03/2022.*
- [10] “Documentación oficial de la librería de Functools,” <https://docs.python.org/3/library/functools.html>, *Ultima vez consultado el 16/03/2022.*
- [11] “Documentación oficial de la librería de Flask_SocketIO,” <https://flask-socketio.readthedocs.io/en/latest/>, *Ultima vez consultado el 24/06/2022.*
- [12] “Documentación oficial de la librería de AIOHTTP,” <https://docs.aiohttp.org/en/stable/>, *Ultima vez consultado el 24/06/2022.*
- [13] “Documentación oficial de la librería de Flask-Mail,” <https://pythonhosted.org/Flask-Mail/>, *Ultima vez consultado el 24/06/2022.*

- [14] “Documentación oficial de la librería de PySerial_Asyncio,” <https://python-socketio.readthedocs.io/en/latest/intro.html>, *Ultima vez consultado el 22/06/2022.*
- [15] “Documentación oficial de la librería de Python_Asyncio,” <https://docs.python.org/3/library/asyncio.html>, *Ultima vez consultado el 10/06/2022.*
- [16] “Documentación oficial de la librería de PySerial_Asyncio,” <https://pyserial-asyncio.readthedocs.io/en/latest/shortintro.html#protocol-example>, *Ultima vez consultado el 10/06/2022.*
- [17] “Página de la VPN de la UCA,” <https://informatica.uca.es/vpn-uca/>, *Ultima vez consultado el 30/06/2022.*
- [18] “Contador de tiempo para JS,” <https://github.com/helloflask/timer>, *Ultima vez consultado el 10/06/2022.*
- [19] “Página de descarga de Python3,” <https://www.python.org/downloads/>, *Ultima vez consultado el 30/06/2022.*
- [20] “Página oficial de PostgreSQL,” <https://www.postgresql.org/>, *Ultima vez consultado el 10/05/2022.*
- [21] “Pagina de descarga de Yawcam,” <https://www.yawcam.com/download.php>, *Ultima vez consultado el 30/06/2022.*
- [22] “Pagina de descarga de Arduino CLI,” <https://arduino.github.io/arduino-cli/0.22/installation/#download>, *Ultima vez consultado el 30/06/2022.*
- [23] “Segger: Pagina sobre Jlink Remote Server,” <https://www.segger.com/products/debug-probes/j-link/tools/j-link-remote-server/>, *Ultima vez consultado el 26/06/2022.*
- [24] “Laboratorios Remotos y Virtuales de I. Calvo, E. Zulueta, U. Gangoiiti y J. M. López,” https://www.ehu.eus/ikastorratza/3_alea/laboratorios.pdf, 2009, *Ultima vez consultado el 10/04/2022.*
- [25] “State of the Art About Remote Laboratories Paradigms de la Université Jean Monnet,” <https://hal-ujm.archives-ouvertes.fr/ujm-00319612/document>, 2008, *Ultima vez consultado el 10/04/2022.*
- [26] “Implementation of an Arduino Remote Laboratorywith Raspberry Pi,” https://www.researchgate.net/publication/333496778_Implementation_of_an_Arduino_Remote_Laboratory_with_Raspberry_Pi, 2019, *Ultima vez consultado el 10/04/2022.*
- [27] “Pagina oficial del proyecto iLab del MIT,” <http://icampus.mit.edu/projects/ilabs>, 2012, *Ultima vez consultado el 04/04/2022.*

- [28] “RECOLAB: Laboratorio remoto de control utilizando Matlab y Simulink de L. M. Jiménez, R. Puerto, Ó. Reinoso, C. Fernández y R. Neco,” https://www.researchgate.net/publication/28141884_RECOLAB_Laboratorio_remoto_de_control_utilizando_Matlab_y_Simulink, 2005, *Ultima vez consultado el 10/04/2022*.
- [29] “Remotely controlled laboratory de la Universidad Palacký,” http://ictphysics.upol.cz/remotelab/index_en.html, 2005, *Ultima vez consultado el 05/04/2022*.
- [30] “Pagina oficial de la Wikipedia sobre los laboratorios remotos,” https://en.wikipedia.org/wiki/Remote_laboratory, *Ultima vez consultado el 01/04/2022*.
- [31] “LCP4088 Mouser,” <https://www.mouser.es/new/embedded-artists/embeddedartists-LPC40884357DB/>, *Ultima vez consultado el 15/04/2022*.
- [32] “embeddedartists, «LPC4088 DEVELOPER’S KIT»,” <https://www.embeddedartists.com/products/lpc4088-developers-kit/>, *Ultima vez consultado el 15/04/2022*.
- [33] “Guía de usuario de ARM para JLink,” <https://developer.arm.com/documentation/101453/0100>, *Ultima vez consultado el 15/04/2022*.
- [34] “Pagina principal de uVision,” <https://www2.keil.com/mdk5/uvision/>, *Ultima vez consultado el 15/04/2022*.
- [35] “Pagina principal de labs land,” <https://labsland.com/en>, *Ultima vez consultado el 20/04/2022*.
- [36] “Metodologia cascada o ágil,” <https://www2.deloitte.com/es/es/pages/technology/articles/waterfall-vs-agile.html>, *Ultima vez consultado el 21/04/2022*.
- [37] “¿Qué es SCRUM?” <https://proyectosagiles.org/que-es-scrum/>, *Ultima vez consultado el 21/04/2022*.
- [38] “Subreddit de Flask,” <https://www.reddit.com/r/flask/>, *Ultima vez consultado el 24/06/2022*.
- [39] “Subreddit de Python,” <https://www.reddit.com/r/python>, *Ultima vez consultado el 24/06/2022*.
- [40] “Stackoverflow,” <https://stackoverflow.com/>, *Ultima vez consultado el 24/06/2022*.
- [41] “ARDUINO(1) Manual Page,” <https://github.com/arduino/Arduino/blob/master/build/shared/manpage.adoc>, *Ultima vez consultado el 20/06/2022*.
- [42] “Getting Started with arduino-cli,” <https://create.arduino.cc/projecthub/B45i/getting-started-with-arduino-cli-7652a5>, *Ultima vez consultado el 20/06/2022*.

- [43] “Tutorial de Flask en español: Desarrollando una aplicación web en Python,” <https://j2logo.com/tutorial-flask-espanol/>, *Ultima vez consultado el 01/06/2022.*
- [44] “Arduino-cli: compile, upload and manage libraries, cores, and boards,” https://www.pcbway.com/blog/Activities/Arduino_cli_compile_upload_and_manage_libraries_cores_and_boards.html, *Ultima vez consultado el 20/06/2022.*
- [45] “Documentacion de la libreria Subprocess de Python,” <https://docs.python.org/3/library/subprocess.html>, *Ultima vez consultado el 21/06/2022.*
- [46] “Pagina de la documentación de uVision para conexión remota,” <https://developer.arm.com/documentation/ka003921/latest>, *Ultima vez consultado el 24/06/2022.*
- [47] “Documentacion de SQLAlchemy 14,” <https://docs.sqlalchemy.org/en/14/index.html>, *Ultima vez consultado el 15/06/2022.*
- [48] “Documentación del modulo de bootstrap para Flask,” https://bootstrap-flask.readthedocs.io/_/downloads/en/1.4.x/pdf/, *Ultima vez consultado el 24/06/2022.*
- [49] “Página oficial de la documentación de overleaf sobre LaTeX,” https://es.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes#What_is_LaTeX.3F, *Ultima vez consultado el 30/06/2022.*
- [50] “Página oficial de la documentacion de Bootstrasp 5,” <https://getbootstrap.com/docs/5.0/getting-started/introduction/>, *Ultima vez consultado el 24/06/2022.*
- [51] “Página oficial de la documentación de Python,” <https://docs.python.org/>, *Ultima vez consultado el 24/06/2022.*
- [52] “Página oficial de la documentación de Flask,” <https://flask.palletsprojects.com/en/2.1.x>, *Ultima vez consultado el 24/06/2022.*
- [53] “Página oficial en ingles de la Wikipedia sobre Jinja,” [https://en.wikipedia.org/wiki/Jinja_\(template_engine\)](https://en.wikipedia.org/wiki/Jinja_(template_engine)), *Ultima vez consultado el 10/05/2022.*