
Sistema de control de acceso con huella dactilar y Arduino



UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE INFORMÁTICA

CURSO 2018/19

Autores

Julia Miguélez Fernández Villacañas

Adrián Navazo Escudero

Directores

José Ignacio Hidalgo Pérez

Juan Lanchares Dávila

Trabajo Fin de Grado en Ingeniería Informática

20 de septiembre de 2019

“Nuestra mayor gloria no está en no caer nunca, sino en levantarnos cada vez que caemos.”

- Confucio,

Resumen

Este proyecto consiste en el diseño e implementación de un dispositivo de control de acceso para un comedor escolar. Se compone de dos Arduinos, un lector de huellas, una pantalla TFT LCD y una base de datos dónde se almacena la información. El objetivo del proyecto es facilitar el registro del alumnado y proporcionar datos relevantes del mismo al personal encargado de la supervisión de ese espacio. Se busca sustituir otros sistemas más arcaicos como el uso de ficheros físicos o el uso de códigos para cada alumno. Prestamos una atención especial a las alergias alimentarias y creemos, que este sistema puede ayudar a prevenir errores con consecuencias no deseadas.

Palabras clave

Arduino, reconocimiento de huella, control de acceso, base de datos, pantalla TFT, comedor escolar, alergias alimentarias, biometría

Abstract

This project consists of the design and implementation of an access control device for a school canteen. It consists of two Arduinos, a fingerprint reader, a TFT LCD screen and a database where the information is stored. The aim of the project is to facilitate the registration of the students and provide relevant data of the same to the staff responsible for supervising that space. It seeks to replace other more archaic systems such as the use of physical files or the use of codes for each student. We pay special attention to food allergies and believe that this system can help prevent errors with undesirable consequences.

Keywords

Arduino, fingerprint recognition, access control, data base, TFT screen, school canteen, food allergies, biometrics

Dedicatoria

Este proyecto se lo quiero dedicar principalmente a mis padres, Carmen y Juanma, y a mi hermano Diego, que siempre han estado ahí apoyándome durante toda la carrera, tanto en los buenos momentos como en los malos, y me han ayudado a seguir adelante.

A mi abuela Luisa, que reza por mí en cada examen y que tantas veces me ha preguntado de qué iba Ingeniería Informática, y yo intentaba explicárselo de la manera más fácil posible, aunque nunca le quedaba del todo claro.

A mi abuela María, que cada vez que me ve me pregunta qué tal me ha salido el examen aunque no lo haya hecho todavía.

A mi familia en general, por su preocupación, su apoyo y sus consejos.

A esos amigos que he conocido en la Facultad, Lidia, Bolaños y Carlos, con los que he pasado muy buenos momentos y a los que si no los hubiera conocido, mi vida tanto universitaria como fuera de la universidad habría sido muy distinta.

Y por supuesto, a mi compañera de TFG y amiga Julia, a la que conocí el primer año de carrera y que nunca me ha defraudado. Con la que tantas horas he pasado haciendo todas esas prácticas que al principio no salían, pero siempre acababan saliendo con nuestro esfuerzo.

Gracias a todos.

Adrián Navazo

Dedicatoria

Para empezar quiero agradecer a mi madre, Mercedes, porque ha estado siempre ahí y me ha animado cuando ya no me quedaba fuerza. A Luci por todo su apoyo y cariño durante muchos años. No tengo suficientes palabras de agradecimiento para ellas.

A mis hermanos Luisa y Andrés que, cada uno en su estilo, me han ayudado cuando lo he necesitado. A mis tíos y primos por proporcionarme momentos inolvidables. A mi abuelo Pepe porque me dijo que no me rindiera nunca y espero que esté orgulloso de mí. Te recordaré siempre.

A mis amigos Patricia, Lidia, Carlos y Bolaños que han llenado de experiencias y momentos felices esta década que llevamos conociéndonos y, además han sabido estar en los momentos difíciles. A Sara por alejar los nubarrones y hacer que confiara más en mí.

Y por último, pero no menos importante, a mi compañero y gran amigo Adrián. Ha estado ahí, constante siempre y con una paciencia increíble. No sé como habría podido acabar las prácticas sin tu ayuda.

Os quiero mucho

Julia Miguélez

Agradecimientos

Queremos agradecer a nuestro tutor Iñaki por todas sus ideas entusiastas y todo el apoyo y medios que nos ha proporcionado a lo largo del proyecto. También a Carlos Roa por todo el mantenimiento que le ha hecho a nuestros dispositivos y por recibirnos siempre de buen humor. A Lanchares por darnos las bases del proyecto del que hemos partido.

Por último queremos agradecer a todos los profesores que hemos tenido durante todos estos años y que gracias a sus enseñanzas hemos aprendido lo necesario para llegar a este punto y ser capaces de hacer este proyecto.

Índice

1	Introducción	12
1.1	Motivación	12
1.2	Objetivos	13
1.3	Organización	13
2	Introduction	15
2.1	Motivation	15
2.2	Objectives	15
2.3	Organization	16
3	Antecedentes	17
4	Dispositivos	17
4.1	Arduino Uno WiFi	17
4.2	Arduino Uno	19
4.3	Arduino Uno Wifi Rev2	20
4.3.1	Comparación de Arduino Uno WiFi y Arduino Uno WiFi Rev2 . .	21
4.4	Arduinos auxiliares	22
4.4.1	Arduino Mega ADK	22
4.4.2	Arduino Duemilanove	23
4.5	Lector de huellas 1	24
4.6	Lector de huellas Adafruit	24
4.7	Pantalla	25
4.8	Batería externa	26
4.9	Cable	27
5	Base de datos	28
5.1	XAMPP	28
5.1.1	MySQL	28
5.1.2	Apache	28
5.1.3	MariaDB	29
5.1.4	phpMyAdmin	29
5.2	Base de datos	29
6	Implementación	31
6.1	Programas y lenguajes	31
6.1.1	Arduino IDE	31
6.1.2	Processing	31
6.1.3	PHP	31
6.1.4	HTML	31
6.2	Librerías	31

6.2.1	Wire	32
6.2.2	Adafruit Fingerprint Sensor Library	32
6.2.3	TFT TouchShield V2, Touch Screen Driver	32
6.2.4	WiFiNINA	33
6.2.5	WifiClient	33
6.3	Montaje	33
6.4	Cableado	33
6.5	Manual de uso	34
6.5.1	Registro de huellas	37
6.5.2	Identificación de alumnos	38
6.5.3	Borrado de huellas	38
6.5.4	Copia de seguridad	40
6.5.5	Formulario de alta y baja en la base de datos	41
6.6	Impedimentos	44
6.6.1	Driver del lector de huellas 1	44
6.6.2	Avería de Arduino Uno Wifi	44
6.6.3	Incompatibilidad de lector de huellas Adafruit con Arduino Uno WiFi Rev 2	44
6.6.4	Tildes y caracteres especiales	45
7	Ejemplo de uso	45
7.1	Registro de huella	46
7.2	Identificación del alumno	48
8	Resultados	53
8.1	Tiempos de acceso a la red y a la base de datos	53
8.2	Memoria de los Arduinos	53
9	Conclusión	54
10	Conclusion	55

Índice de figuras

1	Arduino Uno WiFi	18
2	Arduino Uno	19
3	Arduino Uno Wifi Rev2	20
4	Arduino Mega ADK	22
5	Arduino Duemilanove	23
6	Sensor de huellas Adafruit	25
7	Pantalla Sseed TFT	26
8	Batería externa	27
9	Cable A/B Macho/Macho de tipo periférico	28
10	Modelo E/R de la base de datos	30
11	Montaje y cableado	34
12	Incluir librerías	35
13	Iniciar Apache y MySQL	35
14	Importar alergias.sql	36
15	Configuración de puertos	37
16	Buscar dirección ipv4	38
17	Borrar una huella	39
18	Borrar todas las huellas	40
19	Processing	41
20	Formulario de registro en la base de datos	43
21	Coloca el dedo	46
22	Retira el dedo	47
23	Vuelve a colocar el dedo	47
24	Huella guardada	48
25	Las huellas no coinciden	48
26	Introduzca huella	49
27	Datos del alumno	50
28	Alergias del alumno	50
29	Menú especial sin pescado y lácteos	51
30	Menú normal	51
31	Alumno no registrado	52
32	Alumno no dado de alta en la BD	52

1 Introducción

Desde la creación del primer modelo de Arduino en 2005 en el Instituto de Integración de Diseño de Ivrea (IDII) [1] situado en el norte de Italia, se ha conseguido un acercamiento a la electrónica a cientos de miles de usuarios. Esto ha sido debido a que los productos de esta marca son Hardware Libre, que consiste en que los planos son públicos e incluso cuando compras la placa de un distribuidor puedes modificarla y adaptarla a tus necesidades. A su vez, el entorno software de Arduino, Arduino IDE, está creado bajo licencia GPL (General Public License) que es software libre y permite libremente y gratuitamente ejecutar el código, modificarlo y transmitirlo.

Debido a lo expuesto, la plataforma Arduino ha acercado y economizado los microprocesadores a los usuarios no profesionales. Gracias a estos dispositivos con entornos tan amigables al usuario, como la placa Raspberry Pi, han permitido que se realicen infinidad de proyectos haciendo uso de sus sensores y de su modificabilidad, ya que es uno de los más importantes proyectos de hardware libre.

Se utiliza principalmente en docencia, pero gracias a Internet se han creado foros, como el foro oficial de Arduino [2] en los que muchas personas muestran al mundo los proyectos que han realizado e incluso comparten el código y hacen vídeo tutoriales de todo el proceso. Estos proyectos van desde relojes digitales a pianos que tocan solos, pasando por dispensadores automáticos de comida para gatos.

1.1 Motivación

Con el aumento del número de las personas afectadas por las diversas alergias e intolerancias alimentarias, se precisa de una herramienta que facilite el trabajo de proporcionar una dieta adecuada a la salud de cada individuo. Nos centramos especialmente en lugares dónde, debido a la cantidad de gente comiendo del mismo menú, (como los comedores) tienen que tener un cuidado excepcional con colectivos vulnerables (como los niños y los ancianos). Por tanto, nuestra audiencia objetivo son los comedores escolares. Deseamos que cada alumno sea identificado únicamente con respecto a sus compañeros de cara a los monitores y personal de cocina para atender a sus necesidades.

Esta labor es de vital importancia, ya que dar por error un alimento que tiene un alérgeno a una persona alérgica, ya de forma intrínseca al mismo o debido a la contaminación cruzada ¹, puede llegar, en los peores casos, a provocar un shock anafiláctico ²

¹proceso por el cuál un alimento que no contiene un alérgeno entra en contacto con una superficie o alimento que contiene el alérgeno y se contamina del mismo

²Cierre de las vías respiratorias por la reacción del sistema inmunológico al alérgeno

1.2 Objetivos

Proponer una forma visual de informar a los usuarios sobre los alimentos que pueden consumir dentro de los menús de los comedores escolares para poder evitar una respuesta alérgica de su cuerpo. La identificación de cada alumno se realiza mediante el reconocimiento de sus huellas dactilares, siendo una experiencia personalizada y sin tener que memorizar claves. Además los administradores podrán registrar a los alumnos de una manera rápida y sin esfuerzo, así como borrar sus huellas y hacer una copia de seguridad de todas las huellas. También tendrán la posibilidad de dar de alta a los alumnos en la base de datos mediante un formulario web de una manera rápida y sencilla.

Nuestro objetivo también consiste en utilizar un dispositivo asequible y polivalente, como es el Arduino que se pudiera modificar y ser escalable si fuera necesario con un coste reducido.

En un principio, nuestro objetivo era conseguir reconocer por huella a todos los usuarios que tuviéramos registrados en la base de datos y mostrar la información de los mismos en una pantalla. También se marcó como propósito tener la capacidad de poder registrar al conjunto de personas que desee el administrador. Conforme avanzó el proyecto, encontramos un uso práctico del sistema: los comedores escolares. En ese momento, añadimos el objetivo de combinarlo con los menús del día y marcar aquellos alérgenos que pudieran afectar a un usuario.

1.3 Organización

En los siguientes apartados expondremos los diferentes componentes hardware y software que hemos utilizado en la realización de este trabajo de fin de grado, así como la implementación, un ejemplo de uso y los resultados técnicos del mismo.

- **3. Antecedentes.** En el primer punto tras esta introducción, contaremos los antecedentes de proyectos de Arduino que nos sirvieron de guía o inspiración para la realización de este proyecto.
- **4. Dispositivos.** En el punto cuatro describiremos los dispositivos usados en el desarrollo del proyecto, sus características, modelos e información relevante.
- **5. Base de datos.** En este punto explicaremos la base de datos usada, junto con el gestor de bases de datos y la estructura de la misma: las diferentes tablas, la relación entre ellas y los atributos de las mismas.
- **6. Implementación.** En el punto seis comentaremos la implementación, junto con las librerías, un manual de uso y las diferentes dificultades y escollos encontrados.

- **7. Ejemplo de uso.** En el apartado siete mostraremos como se usan las funcionalidades de registrar una huella y de identificación de un alumno desde la pantalla inicial y con visualizaciones de cada paso.
- **8. Resultados.** El penúltimo punto, será el más técnico, donde haremos mediciones de nuestra aplicación y enseñaremos los resultados. Este punto incluye tiempos de acceso, memoria utilizada en ejecución, etcétera.
- **9. Conclusión.** Para finalizar realizaremos unas conclusiones sobre los objetivos conseguidos, las dificultades encontradas y el cierre del proyecto.

2 Introduction

Since the creation of the first Arduino model in 2005 at the Ivrea Design Integration Institute (IDII) [1] located in northern Italy, an approach to electronics has been achieved for hundreds of thousands of users. This has been possible because the products of this brand are Free Hardware, which means that the plans are public and even when you buy a board from a distributor you can modify it and adapt it to your needs. At the same time, the Arduino software environment, the Arduino IDE, is created under the GPL (General Public License) which is free software and allows you to freely execute the code without any expense, modify it and transmit it.

Due to the above, the Arduino platform has brought microprocessors closer and made them affordable to non-professional users. Thanks to these devices with user-friendly environments, such as the Raspberry Pi board, an infinite number of projects have been carried out using their sensors and their modifiability, since it is one of the most important free hardware projects.

It is mainly used in teaching, but thanks to the Internet, forums have been created, such as the official Arduino forum where many people show the projects they have done to the world and even share the code and make video tutorials of the whole process. These projects range from digital clocks to self-playing pianos to automatic cat food dispensers.

2.1 Motivation

With the increase in the number of people affected by the various allergies and food intolerances, we need a tool to facilitate the work of providing a proper diet for the health of each individual. We focus especially on places where, due to the large amount of people eating the same menu, (such as school canteens) must take an exceptional care with vulnerable collectives (like children and the elderly). Therefore, our target audience are school canteens. Our aim is that each student will be identified only with respect to their classmates for monitors and kitchen staff to fulfill their needs.

This work is of vital importance, since giving food mistakenly with an allergen to an allergic person, either intrinsically or due to cross-contamination ³, can lead to an anaphylactic shock ⁴ in the worst cases.

2.2 Objectives

To propose a visual way to inform users about the food they can consume within the canteen menus in order to avoid an allergic response of their body. The student's identification

³process by which a food that does not contain an allergen comes into contact with a surface or food that contains the allergen and is contaminated with it

⁴closure of the airways due to the reaction of the immune system to the allergen

is carried out by means of checking their fingerprints and in this way it is a personalized experience avoiding the memorizing of keys. Moreover, administrators will be able to register their students efficiently as well as to erase and make a backup of their fingerprints. They may also enter their students in a database by filling in a web form quickly and easily.

Our objective also consists of using an affordable and versatile device, such as the Arduino that could be modified and be scaled if necessary at a reduced cost.

At the beginning, our goal was to recognize all users we had registered in our database by their fingerprints and display their information on a screen. Our aim was also to be able to register sets of people as the administrator wished. As the project progressed, we found a practical use for our system: school canteens. At that moment, we added the objective of combining it with their daily menus and highlighting the allergens the user was sensitive to in the different menus.

2.3 Organization

In the following sections we will expose the different hardware and software components that we have used in the completion of this degree final project, as well as the implementation, and an example of usage and its technical outcomes.

- **3. Background.** In the first part after this introduction, we will tell the background of Arduino projects that served as a guide or inspiration for the realization of this project.
- **4. Devices.** In the fourth part we will describe the devices used in the development of the project, their features, models and relevant information.
- **5. Database.** In this part we will explain the database used, together with the database manager and its structure: the different tables, the relationship between them and their attributes.
- **6. Implementation.** In part six we will discuss the implementation, with the libraries, an user's manual and the different difficulties and drawbacks encountered.
- **7. Use case.** In section seven we will show how the application runs, fingerprint register and student identification, from the initial screen with screenshots for each step.
- **8. Outcomes.** The last but one section will be the most technical, where we will obtain measurements in our application and show the outcomes. This part can include access times, memory used in execution, etc.
- **9. Conclusion.** Finally we will come to our conclusions about the objectives reached, the difficulties found and the end of the project.

3 Antecedentes

En esta sección describiremos los trabajos previos que se han hecho tanto de control de acceso como de reconocimiento de huellas y proyectos de Arduino.

Por la relevancia de Arduino en el campo de la educación, solamente en la Facultad de Informática de la Universidad Complutense encontramos dos proyectos en los que se emplea este dispositivo.

- **Una red de sensores para las Smart Cities.** [3] En este trabajo de fin de Máster se utiliza Arduino y sus sensores para medir los niveles de ruido en diferentes puntos de una ciudad.
- **Conducción asistida por dron con soporte hardware para la visión.** [4] En este caso, este trabajo de fin de grado utiliza un Arduino dentro del automóvil como controlador.

Estos dos trabajos son sólo la punta del iceberg de la cantidad de proyectos en los que se emplea esta placa. Basta una búsqueda básica con la palabra Arduino en un buscador de Internet para encontrar un conjunto enorme de proyectos con una variedad extraordinaria.

4 Dispositivos

Hemos utilizado tres placas de Arduino y dos lectores de huellas diferentes. A continuación les presentaremos sus características, sus ventajas e inconvenientes de cada uno de ellos con respecto al otro.

Debido a una avería en la placa Arduino Uno WiFi que utilizábamos como principal, se aprovechó para comprar una placa con mayor capacidad de procesado.

4.1 Arduino Uno WiFi

El original, con el que empezó el proyecto y sobre el que estaba montado el lector de huellas 1. Se puede ver en la Figura 1

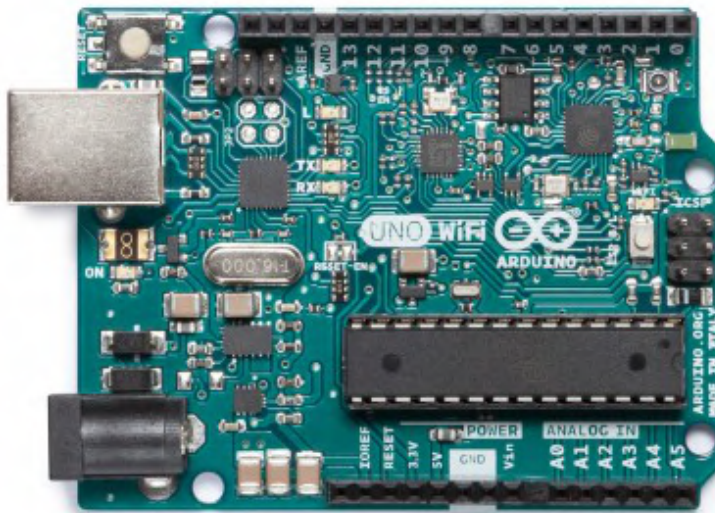


Figura 1: Arduino Uno WiFi

Sus características técnicas son:

- Pins de E/S (Entrada/ Salida) digitales: 20
- Salidas PWM [5] (Modulación de Ancho de Pulsos): 5
- Consumo de energía: 93 mA
- Microcontrolador: ATmega328
- Voltaje al que funciona: 5 V
- Memoria flash: 32 KB
- SRAM: 2 KB
- Velocidad del reloj: 16 MHz
- Pins de entrada analógicos: 6
- EEPROM: 1 KB
- Corriente continua por cada pin E/S : 40 mA
- Procesador WiFi : ESP8266

- **Voltaje al que funciona el microprocesador WiFi:** 3,3 V
- **Memoria flash del microprocesador WiFi:** 4 MB
- **RAM del microprocesador WiFi:** 8 MB instrucción, 12 MB de datos
- **WiFi:** 802.11 b/g/n 2.4 GHz
- **Tiempo de respuesta:** menor a 2 ms

4.2 Arduino Uno

Al principio del proyecto se utilizó como placa auxiliar que utilizamos para probar la pantalla y con el lector de huellas Adafruit pero, al no tener WiFi, no lo podíamos emplear para la transferencia de datos con la base de datos.

Cuando avanzó el proyecto y debido a una dificultad que se explicará en el apartado de Impedimentos dentro de implementación, se estableció como dispositivo maestro (Master). Las características técnicas son las mismas que las de la placa Arduino Uno WiFi a excepción del módulo ESP8266, que es el que le otorga conectividad WiFi al modelo explicado en el punto anterior. Se expone una fotografía del mismo a continuación.

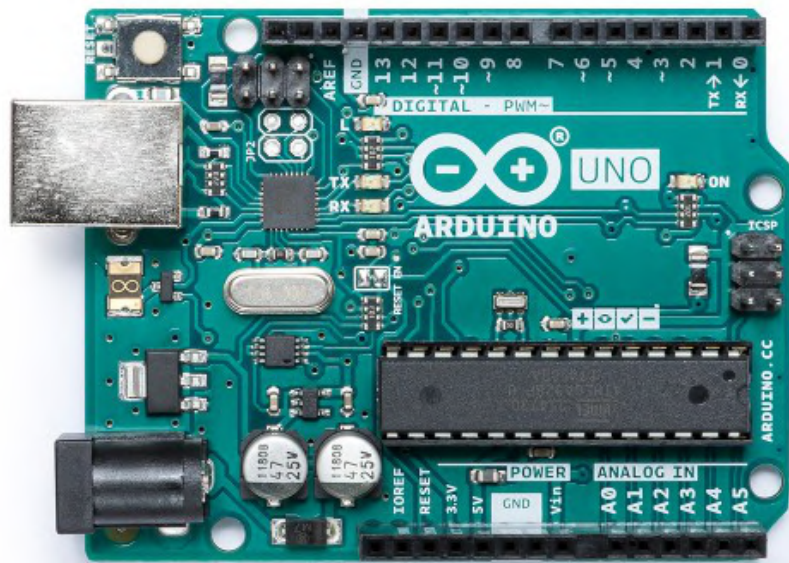


Figura 2: Arduino Uno

Sus características técnicas son:

- **Pins de E/S (Entrada/ Salida) digitales:** 14

- Salidas PWM (Modulación de Ancho de Pulsos): 6
- Consumo de energía: –
- Microcontrolador: ATmega328P
- Voltaje al que funciona: 5 V
- Memoria flash: 32 KB
- SRAM: 2 KB
- Velocidad del reloj: 16 MHz
- Pins de entrada analógicos: 6
- EEPROM: 1 KB
- Corriente continua por cada pin E/S : –

4.3 Arduino Uno Wifi Rev2

Durante el transcurso del proyecto se pensó en comprar este Arduino para disminuir los tiempos de acceso a la base de datos, pero su compra se vió precipitada por una avería del dispositivo del primer punto, como ya se ha mencionado. En el proyecto tiene el rol de Slave y se monta la pantalla encima sobre los pines. Figura 3

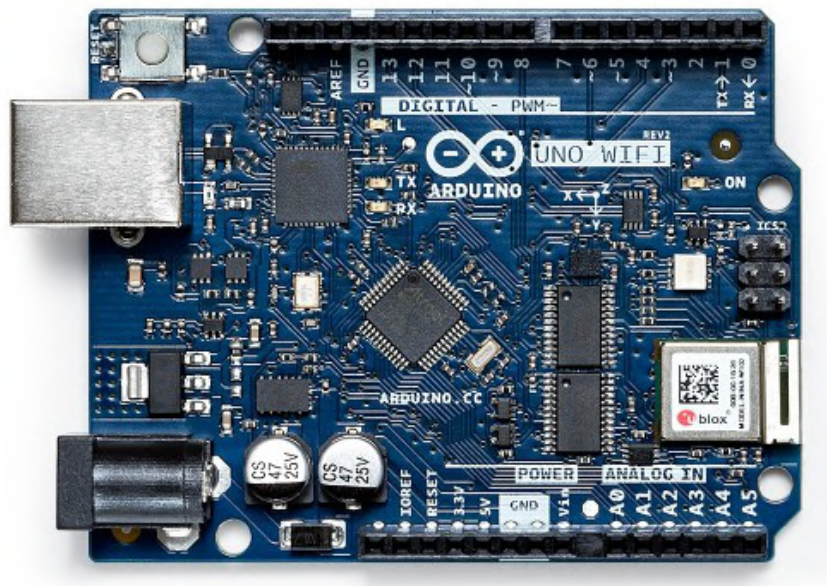


Figura 3: Arduino Uno Wifi Rev2

Sus características técnicas son:

- **Pins de E/S (Entrada/ Salida) digitales:** 14
- **Salidas PWM (Modulación de Ancho de Pulsos):** 5
- **Consumo de energía:** –
- **Microcontrolador:** ATmega4809
- **Voltaje al que funciona:** 5 V
- **Memoria flash:** 48 KB
- **SRAM:** 6,144 B
- **Velocidad del reloj:** 16 MHz
- **Pins de entrada analógicos:** 6
- **EEPROM:** 256 B
- **Corriente continua por cada pin E/S :** 20 mA
- **WiFi:** 802.11 b/g/n 2.4 GHz

4.3.1 Comparación de Arduino Uno WiFi y Arduino Uno WiFi Rev2

A continuación explicaremos el por qué de la elección de este modelo en concreto:

- **Memoria flash.** La diferencia entre la placa de Arduino Uno WiFi y esta es que la primera tiene 32 KB de memoria flash y esta tiene 48 KB. La memoria flash es dónde se almacena el código del programa o sketch [6] Al ser mayor, esta nos permitía escribir más funcionalidades, con sus consecuentes líneas de código, además de poner nombres de variables más significativos.
- **SRAM.** En este apartado, la diferencia entre el modelo de la placa del Arduino estropeado y el modelo que lo reemplazó en nuestro proyecto es de 2 KB frente a 8KB. La SRAM es la memoria RAM estática, este tipo de memoria es volátil (es decir, que cuando se apaga el Arduino, los datos que hubiera almacenados en esta memoria, se pierden).[6]
- **EEPROM.** En este punto sí hay una pérdida a favor del dispositivo anterior. Estamos hablando de 1 KB que tiene el Arduino Uno WiFi y 256 B que tiene el Arduino Uno WiFi Rev 2.

En este proyecto, la pérdida en este tipo de memoria no es significativa, ya que

no almacenamos ningún dato en el "disco duro" del Arduino.[6] Los modelos de huellas están almacenados en el propio lector de huellas y los demás datos se encuentran alojados en la base de datos.

- **Retirado.** En último lugar, la decisión de cambiar de modelo fue debido a que en la página de Arduino [7] el modelo del Arduino Uno WiFi aparecía como retirado. Por ello consideramos que podía estar obsoleto.

4.4 Arduinos auxiliares

En esta sección hablaremos sobre los dos modelos que empleamos auxiliariamente para la realización del proyecto en momentos puntuales.

4.4.1 Arduino Mega ADK

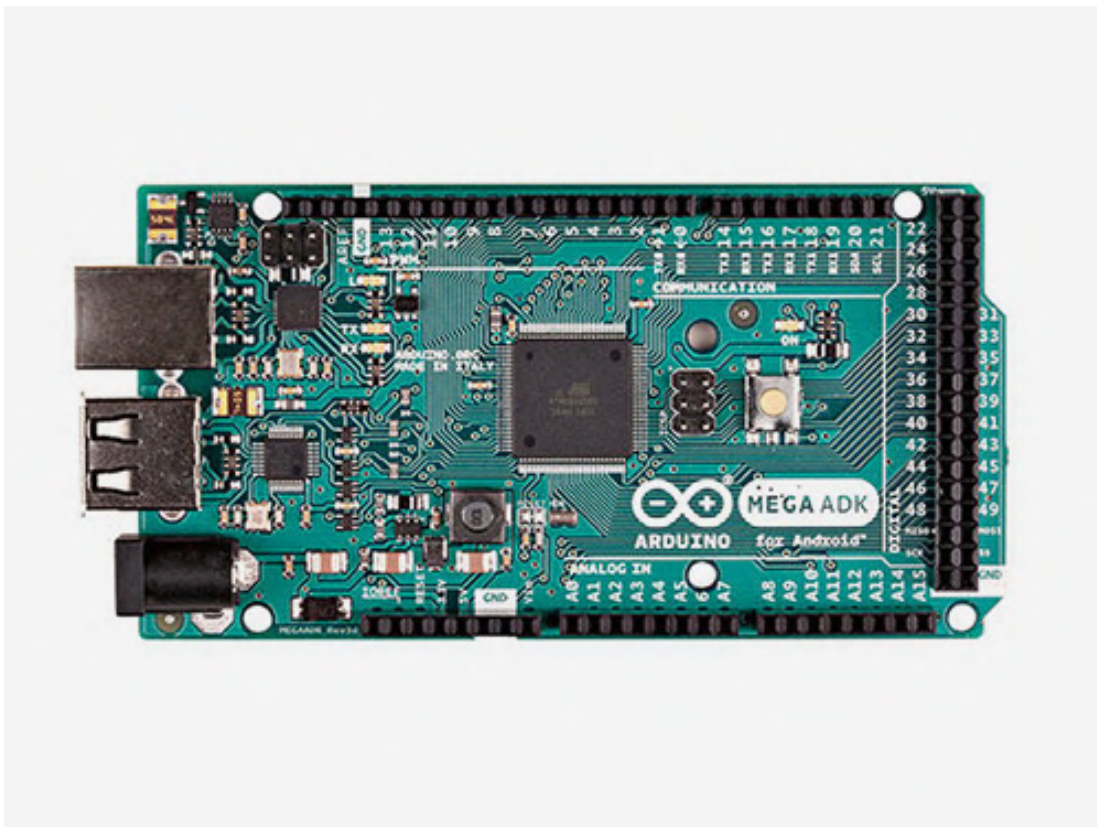


Figura 4: Arduino Mega ADK

Sus características técnicas son:

- **Pins de E/S (Entrada/ Salida) digitales:** 54

- Salidas PWM (Modulación de Ancho de Pulsos): 15
- Consumo de energía: –
- Microcontrolador: ATmega2560
- Voltaje al que funciona: 5 V
- Memoria flash: 256 KB, de los cuales 8 KB dedicados al bootloader. [8]
- SRAM: 8 KB
- Velocidad del reloj: 16 MHz
- Pins de entrada analógicos: 16
- EEPROM: 4 KB
- Corriente continua por cada pin E/S : 40 mA

4.4.2 Arduino Duemilanove

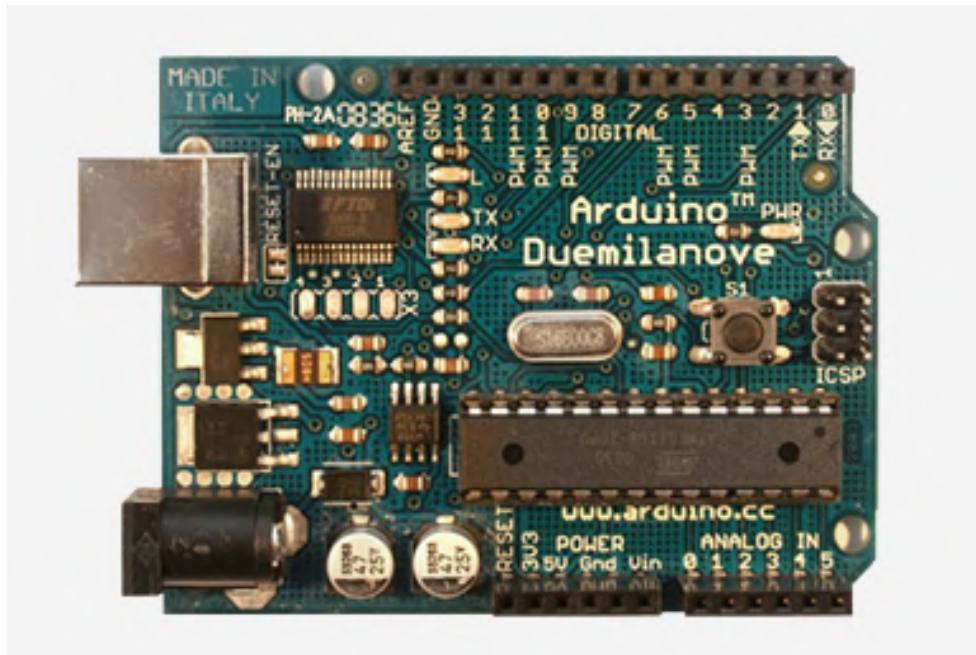


Figura 5: Arduino Duemilanove

Debido a su escasa SRAM, al cargar ejemplos del lector de huellas en el Arduino IDE nos salía el aviso de : "Poca memoria disponible, pueden aparecer problemas de estabilidad".

Este tipo de memoria es el que tiene que almacenar las variables globales. La memoria flash es la que se encarga de almacenar el sketch (programa) y con los mismos ejemplos rondaba una ocupación de entre el 60 % y el 80 % de la memoria del Arduino. Por ese motivo, en cuanto nuestro sketch empezó a tener una longitud mayor para contener todo el código específico de la pantalla, el lector y la comunicación I2C, la memoria de este Arduino se quedó insuficiente y no podía cargar el programa.

Sus características técnicas son:

- **Pins de E/S (Entrada/ Salida) digitales:** 14
- **Salidas PWM (Modulación de Ancho de Pulsos):** 6
- **Consumo de energía:** –
- **Microcontrolador:** ATmega168
- **Voltaje al que funciona:** 5 V
- **Memoria flash:** 16 KB, de los cuales 2 KB dedicados al bootloader. [8]
- **SRAM:** 1 KB
- **Velocidad del reloj:** 16 MHz
- **Pins de entrada analógicos:** 6
- **EEPROM:** 512 B
- **Corriente continua por cada pin E/S :** 40 mA

4.5 Lector de huellas 1

El primer lector de huellas, es uno que iba en una estructura preparada por los directores junto con el Arduino Uno Wifi. Consiste en un lector de huellas con capacidad mayor que el Lector de huellas Adafruit y que tiene un sistema de lector de tarjetas RFID incorporado. Tuvimos problemas con las librerías de los drivers y la transferencia de datos al Arduino, por tanto, aunque la capacidad de huellas era mayor en este dispositivo, nos acabamos decantando por el segundo lector de huellas.

4.6 Lector de huellas Adafruit

El segundo lector de huellas que usamos, es de la marca Adafruit (hardware libre). Este dispositivo tiene una disponibilidad de guardar hasta 127 huellas diferentes, por tanto, una capacidad menor que el anterior. Es un periférico que ofrece librerías para su uso con Arduino y por tanto la transferencia de datos entre el Arduino y el reconocedor de huellas estaba contenida en funciones.

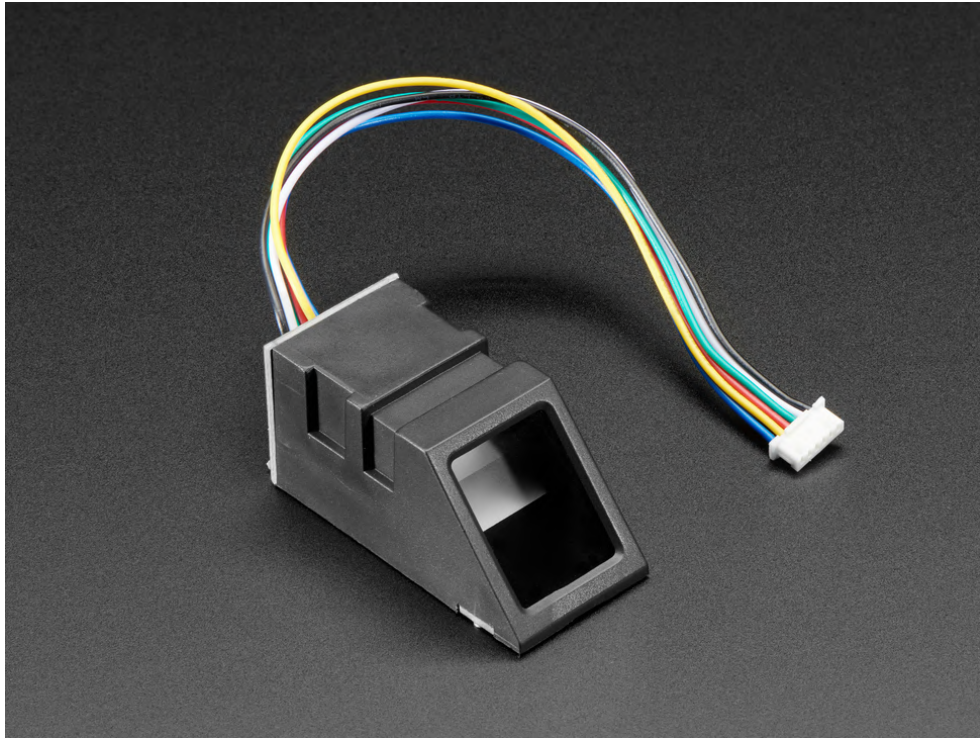


Figura 6: Sensor de huellas Adafruit

4.7 Pantalla

La pantalla es una pantalla táctil TFT de la marca Seeed y es el modelo 2.8 TFT Touch Shield V2.0. Su driver es ILI9341. Es compatible con los modelos de Arduino: Uno, Mega y Leonardo. Figura 7



Figura 7: Pantalla Seeed TFT

4.8 Batería externa

Para conseguir que nuestro proyecto fuera autónomo, sin necesidad de conectarlo a un ordenador, subimos el código a los Arduino y luego los alimentamos con una batería externa de las que se usan para cargar los teléfonos móviles. En este caso, al tener que dar energía a dos Arduinos, necesitamos usar una batería externa con dos USBs. En caso de no tener una batería externa con doble puerto USB, se puede utilizar una batería externa con un solo puerto y conectar el pin de tierra de un Arduino al pin de tierra (GND, ground en inglés) del otro para que funcione y, a su vez conectar los pines de 5V de ambos mediante un cable. Sus características técnicas son 6600 mAh (miliAmperios) de capacidad y dos puertos de 5 voltios, uno de ellos a 2.1 Amperios y el otro a 1 Amperio.



Figura 8: Batería externa

4.9 Cable

El cable utilizado lo contenía el Arduino: Cable A/B Macho/Macho de tipo periférico. (Su nombre es debido a que es ampliamente usado en periféricos como impresoras). [9] Al tener un puerto USB se utiliza tanto en la conexión con el ordenador como con la batería externa.



Figura 9: Cable A/B Macho/Macho de tipo periférico

5 Base de datos

Para almacenar toda la información sobre los usuarios y sus alergias se ha utilizado un sistema de gestión de bases de datos MySQL alojada en un servidor web Apache con la distribución XAMPP. fotos de la base de datos, e/r y relacional

5.1 XAMPP

XAMPP es una distribución de Apache completamente gratuita y fácil de instalar que contiene MariaDB, PHP y Perl. De este paquete de software libre utilizamos el gestor de bases de datos MySQL, el servidor web Apache y el intérprete para PHP.

5.1.1 MySQL

MySQL es un gestor de bases de datos de código abierto basado en SQL (Structured Query Language, por sus siglas en inglés)[10]

5.1.2 Apache

Apache es un servidor web HTTP de código abierto desarrollado y mantenido por la comunidad de usuarios. Versión Apache/2.4.38 (Win32) OpenSSL/1.1.1a

5.1.3 MariaDB

MariaDB es un servidor de base de datos contenido en XAMPP. La versión que hemos usado es 10.1.38-MariaDB.

5.1.4 phpMyAdmin

phpMyAdmin es una herramienta de software gratuita y libre escrita en PHP, cuyo objetivo es manejar la administración de MySQL en la Web.

PhpMyAdmin soporta un gran rango de operaciones de MySQL y MariaDB. Es usado para realizar múltiples operaciones (manejar bases de datos, tablas, columnas, relaciones, índices, usuarios, permisos, etc). Se puede ejecutar mediante una interfaz de uso, pero también permite ejecutar directamente cualquier sentencia SQL[11] . Versión 4.8.5

5.2 Base de datos

Para crear y gestionar la base de datos utilizamos phpMyAdmin, una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web, utilizando un navegador web.

La base de datos, llamada alergias, está formada por cuatro tablas:

- **alumnos:** una tabla con la información personal de cada alumno, compuesta por seis columnas:
 - **idAlumno:** un número de identificación único para cada alumno. Es la clave primaria de la tabla.
 - **codigo_alumno:** el número de identificación que cada alumno tiene asignado en el lector de huellas. El identificador es único para cada alumno.
 - **Nombre:** el nombre del alumno.
 - **Apellido1:** el primer apellido del alumno.
 - **Apellido2:** el segundo apellido del alumno.
 - **Pago:** un atributo que indica con un 1 si se ha realizado el pago y con un 0 si no.
- **alergenosen:** representa los alérgenos que puedan tener las comidas que se sirven en el comedor escolar. Está compuesta por las siguientes columnas:
 - **idAlergias:** un identificador único para cada alérgeno. Es la primary key de la tabla.
 - **Descripción:** el nombre de cada alérgeno o la palabra "NINGUNA" para el caso de que un alumno no tenga alergias. Los 14 alérgenos son: pescado, frutos secos, lácteos, moluscos, gluten, crustáceos, huevos, cacahuetes, soja, apio, mostaza, sésamo, altramuiz y sulfatos.

- **tiene_alergia:** relaciona cada alumno con ninguna, una, o más alergias. Tiene dos columnas:
 - **idTiene_alergia:** es la clave primaria de esta tabla, por tanto identifica únicamente a cada tipo de alergia alimentaria que padece un alumno.
 - **codigo_alumno:** referencia al número de identificación del alumno en el lector de huellas.
 - **codigo_alergia:** referencia al identificador de la alergia que tiene ese alumno, o el identificador 15 en caso de que no tenga ninguna alergia.
- **alergia_diaria:** indica para una fecha cuáles son las alérgenos que tiene el menú del día. Consiste en tres columnas:
 - **id_fecha:** es la clave primaria de la tabla,
 - **fecha:** una fecha en formato Año-Mes-Día.
 - **alergia_diaria:** referencia al identificador de la alergia que tiene el menú de ese día. Si un día no hay ninguna alergia en el menú, esa fecha no se registra en la base de datos.

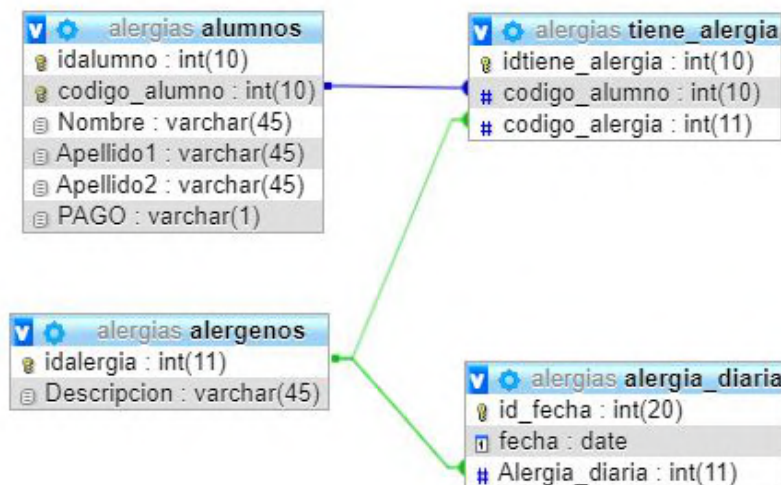


Figura 10: Modelo E/R de la base de datos

6 Implementación

En este apartado se exponen los diferentes programas, librerías y tecnologías usados para llevar a cabo el proyecto. Además se expone una guía para el montaje y el cableado y un manual de uso para el software.

6.1 Programas y lenguajes

6.1.1 Arduino IDE

El entorno open-source de Arduino, permite escribir código y subirlo a cualquier placa Arduino. No solo permite programarlo, sino que con el bootloader inteligente, se puede dejar corriendo el programa que le has cargado sin conectarlo al ordenador. Con una batería externa es suficiente. La versión usada es la 1.8.8.

6.1.2 Processing

Processing es un entorno que hemos utilizado para crear un script que corre en el ordenador para recoger la salida del puerto serie del Arduino. Era necesario para crear nuestra copia de seguridad, ya que el IDE de Arduino solo permite guardar datos de su puerto serie si tiene un módulo de tarjeta SD. El programa se puede descargar en la página <https://processing.org/download/>.

6.1.3 PHP

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML [12]. Lo usamos como intermediario entre el Arduino y la base de datos para hacer las consultas SQL que permiten identificar a los alumnos. También se usará en el formulario web para dar de alta a los alumnos en la base de datos. Versión de PHP: 7.3.2

6.1.4 HTML

HTML, siglas en inglés de HyperText Markup Language (Lenguaje de Marcas de Hipertexto), es un lenguaje de marcado que se utiliza para el desarrollo de páginas web. Nosotros usamos la última versión, HTML5, para realizar un formulario donde rellenar los datos del alumno, y que mediante PHP se conecte con la base de datos.

6.2 Librerías

Estas librerías han sido utilizadas en este proyecto. A continuación se procederá al desglose de las mismas, su función dentro de este trabajo y su origen.

6.2.1 Wire

Esta librería sirve para usar la conectividad I2C entre los dos Arduinos. Se establecen dos roles: Master y Slave, o Maestro y Esclavo en español. Estos roles sirven para tomar el control del bus por el cual se mandan los datos. En la sección Cableado ya se hablará de cómo se conectan el cable de datos y el cable que lleva la señal de reloj. El cable que conecta los pines SDA transporta los datos y el cable que conecta los pines SCL de cada uno manda la señal de reloj para sincronizar la señal. En nuestro proyecto utilizamos la librería Wire para poder mandar el identificador de la huella del Arduino Uno (Master) al Arduino UNO WiFi Rev2 (Slave). Al principio del sketch, empezamos con la función `Wire.begin()` que une el bus entre el Master y el Slave. Luego funciona por eventos con las funciones `Wire.onReceive()` y para escribir en el bus, se realiza con `Wire.write()` y para leer del mismo lo hacemos con `Wire.read()`. En el bus tiene preferencia el Master y el Slave pregunta por la disponibilidad del mismo. Este modelo se puede replicar teniendo más Arduinos conectados al bus pero, siempre tiene que haber un Master, los demás tienen que ser Slaves. En caso de que haya más de dos Arduinos conectados, se tiene que nombrarlos al iniciar con el `Wire.begin()`.

6.2.2 Adafruit Fingerprint Sensor Library

Esta librería es la oficial de Adafruit y se encarga de todo lo relacionado con el sensor de huellas. Es el driver del lector de huellas. En nuestro proyecto la usamos para registrar las huellas, para reconocer huellas, para hacer una copia de seguridad de todas las huellas, y para eliminar huellas.

Indagando en el código de la librería vemos que guarda los modelos de las huellas como tipo `uint-8`. El lector de huellas realiza una cuadrícula (se puede ver exactamente en el pdf que colgamos en el GitHub como parte de la documentación) para mapear las huellas y con ello crea un modelo que es un conjunto de caracteres alfanumérico con que identifica únicamente a cada huella. Cuando introduces un dedo en el sensor, realiza la misma operación pero esta vez compara con los modelos que tiene almacenados en ese momento. Si hay un código que corresponde al código que acaba de calcular te reconocerá la huella y te dará la posición en su memoria interna en la que está almacenado (dispone de un array de modelos del 1 al 128). En caso contrario, te dirá que no está registrado. Estas funciones nosotros las procesamos en el Arduino UNO y las mandamos a través del bus mencionado en el punto anterior hasta el otro Arduino (UNO Wifi Rev2).

6.2.3 TFT TouchShield V2, Touch Screen Driver

Estas librerías se usan para la pantalla TFT que va montada sobre el Arduino UNO WiFi Rev2. Nosotros la utilizamos para mostrar texto en la pantalla. Se inicializa con `TFT.TFTinit()` en el código y a partir de ahí se puede empezar a usar. La función `drawString()` que utilizamos abundantemente tiene 5 argumentos: el primero es para meter la cadena de caracteres que quieres mostrar por pantalla, el segundo es la coordenada X dentro de la pantalla en la que quieres escribir, el tercero la coordenada Y, el cuarto es el

tamaño y por último le introduces en inglés el color en que va a escribir. Esta librería tiene un problema con los caracteres que no tengan una codificación ASCII como se explica en el punto **6.6.4 Tildes y caracteres especiales** de esta memoria.

6.2.4 WiFinINA

Librería para conectar con el WiFi, ésta es nueva para el Arduino UNO WiFi Rev2. Soporta conexiones con seguridad WEP y WPA2. Anteriormente, cuando aún usábamos el Arduino UNO WiFi, la librería usada era WiFiLink. Esta librería permite trabajar con errores derivados de una mala conexión a la red WiFi y en su función de inicialización se introduce el SSID de la conexión WiFi (el nombre de la misma) y la contraseña de acceso.

6.2.5 WifiClient

La librería para el cliente de WiFi. Con esta librería nos comunicamos con la base de datos mediante un script PHP. Usamos la función `HttpRequest()` para comunicarnos con la base de datos y le mandamos un identificador de la huella que hemos reconocido. Esto procederá a hacer una consulta en la base de datos y nos devolverá los datos de la persona registrada con esa huella.

6.3 Montaje

Con el fin de que los dos Arduinos y el lector de huellas estuviesen sujetos y los cables no se quitasen de los pines, ya que tuvimos que volver a soldarlos en numerosas ocasiones, decidimos anclar los Arduinos y el lector de huellas a una tabla de madera de 31x20.5 cm. Los Arduinos van sujetos con cuatro tornillos cada uno en cada esquina, junto a cuatro elevadores que los separan medio centímetro de la tabla para que no se sobrecalienten. El lector de huellas se sujeta con una brida. El montaje una vez finalizado puede verse en la Figura 11.

6.4 Cableado

Para que el sistema realice correctamente todas las funciones es fundamental que los cables que conectan el lector de huellas con el Arduino y los dos Arduinos entre sí se encuentren bien conectados.

El cable verde del lector de huellas se conecta al pin 2 del Arduino UNO, y el cable blanco al pin 3. Para que le llegue energía al lector de huellas hay que conectar el cable rojo al pin 5V y el cable negro a GND.

La pantalla TFT se inserta directamente sobre el Arduino UNO WiFi Rev 2.

Para poder transmitir el número de identificación del alumno desde el Arduino UNO al Arduino UNO WiFi Rev 2 es necesario conectar los pines SCL y SDA del Arduino UNO con los pines SCL y SDA del Arduino UNO WiFi Rev 2 respectivamente. La Figura 11 presenta el sistema con todo el cableado.

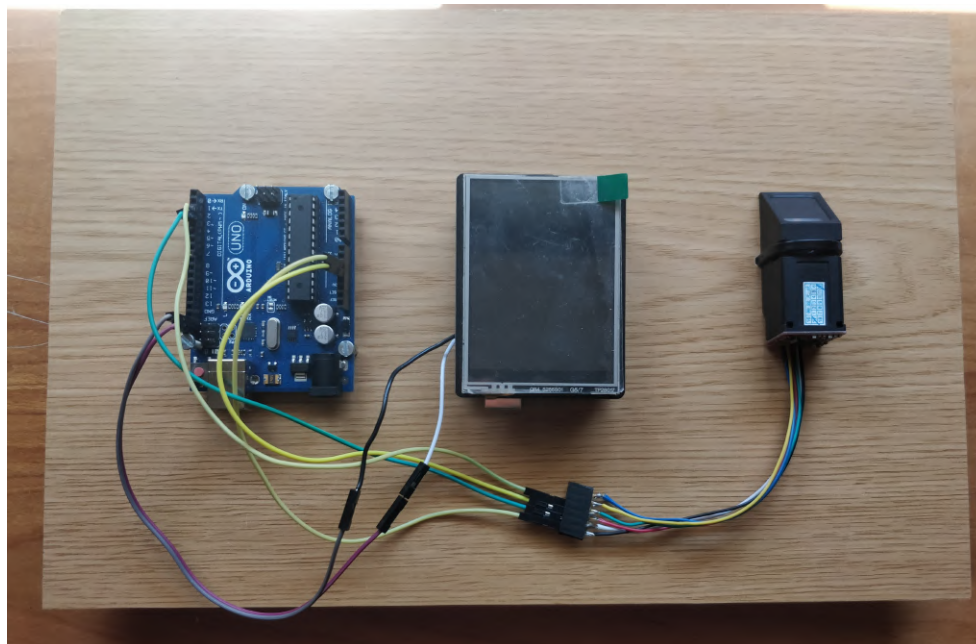


Figura 11: Montaje y cableado

6.5 Manual de uso

Lo primero que hay que hacer para poder hacer uso del sistema desarrollado es descargar los programas necesarios. El principal es el Arduino IDE, que se puede descargar de su página web [13]. Está disponible para Windows, Linux y Mac OS X Este programa sirve para cargar y editar el código en formato “ino”.

También habrá que descargar e instalar la distribución XAMPP [14]. XAMPP es una distribución de Apache completamente gratuita y fácil de instalar que contiene MariaDB, PHP y Perl. De este paquete de software libre utilizamos el gestor de bases de datos MySQL, el servidor web Apache y el intérprete para PHP.

Otro programa requerido es el Processing. En este caso solo se utilizará para llevar a cabo la copia de seguridad de las huellas.

También será necesario el uso de un navegador de Internet para usar el formulario web para dar de alta a los alumnos en la base de datos.

El siguiente paso será descargar el repositorio de Github “TFG2019_HUELLAS” [15], el cual contiene todos los archivos requeridos para que el sistema funcione, a parte de toda la documentación necesaria.

Una vez descargada la carpeta “TFG2019_HUELLAS” e instalados todos los programas, se debe añadir al Arduino IDE todas las librerías que necesitamos. Se tienen que descargar las siguientes librerías: WiFinINA [16], TFT Touch Shield V2 [17], Touch Screen Driver [18] y Adafruit Fingerprint Sensor Library [19]. Cada una de ellas se explican en el apartado **6.2 Librerías**.

Para incluirlas en el entorno, abrimos el Arduino IDE y en la barra de menú hacemos click en Programa - Incluir librería - Añadir librería .ZIP (Figura 12).

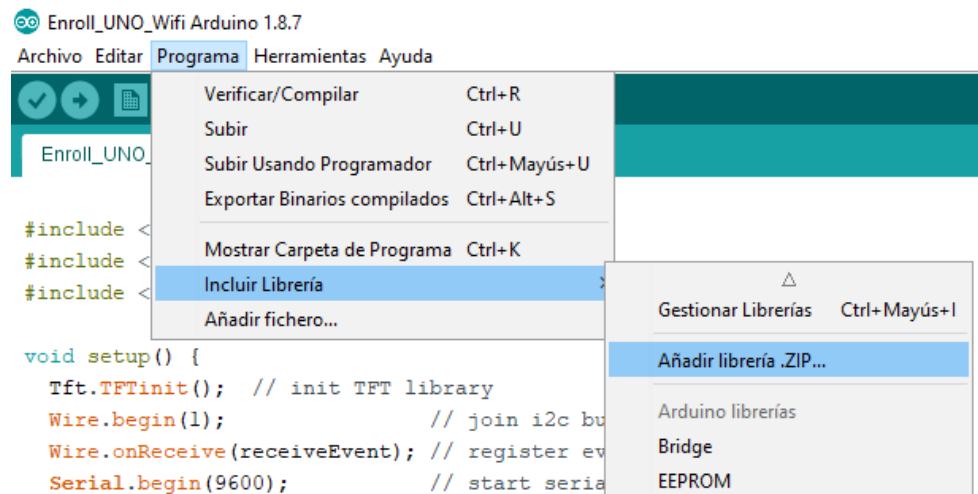


Figura 12: Incluir librerías

Se abrirá una ventana donde hay que seleccionar las librerías y pulsar en Abrir.

A continuación vamos a importar la base de datos con la información de los alumnos. Para ello, abrimos el programa XAMPP Control Panel e iniciamos los módulos Apache y MySQL (Figura 13).

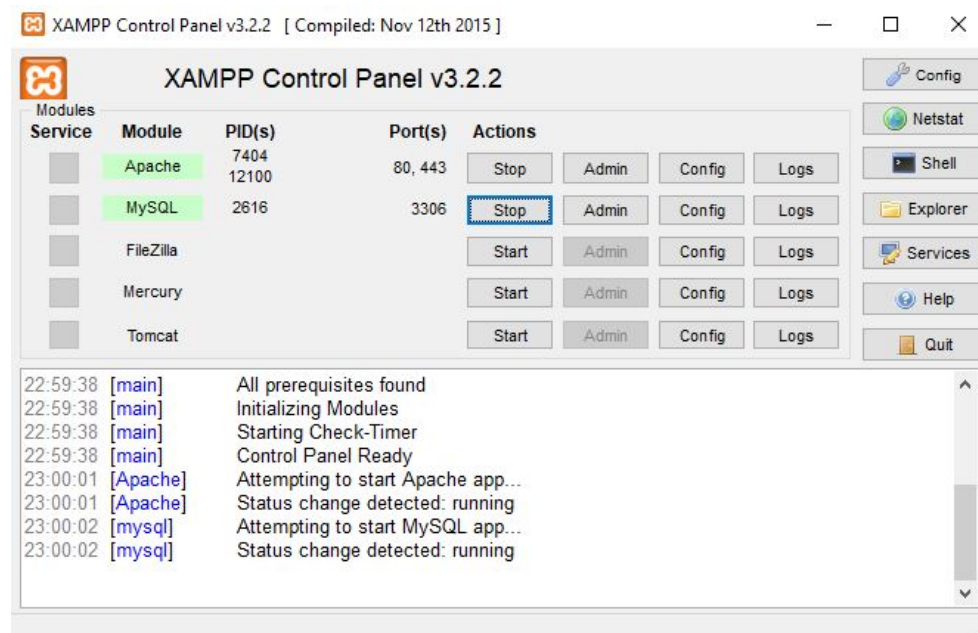


Figura 13: Iniciar Apache y MySQL

Una vez iniciados estos dos módulos, ya se puede abrir el navegador e introducir la dirección `http://localhost/phpmyadmin`. En la barra superior, hacemos click en Importar. En la pantalla que se abre hay que pulsar en Seleccionar archivo y abrir el archivo “alergias.sql”, que se encuentra en `TFG2019_HUELLAS/Proyecto_Julia_Adrian/src/`. Para terminar de importar la base de datos solo queda darle a Continuar en la parte inferior de la página (Figura 14).



Figura 14: Importar alergias.sql

Una vez importada la base de datos, únicamente queda copiar la carpeta “comedor” disponible en `TFG2019_HUELLAS/Proyecto_Julia_Adrian/src/` a la carpeta “htdocs” de XAMPP, que se encontrará en la ruta del PC donde se haya instalado el XAMPP. Esta carpeta contiene los archivos html y php necesarios para poder llevar a cabo el registro de los alumnos en la base de datos mediante un formulario HTML y para consultar la base de datos con el fin de identificar a los alumnos. En los archivos `select.php` y `registro.php` se deben actualizar las variables `server`, `dbusername` y `dbpassword`, con el nombre del servidor, el nombre de usuario y la contraseña respectivamente.

Tras realizar estos primeros pasos ya se puede hacer uso de las distintas funcionalidades que ofrece el sistema, cuyo funcionamiento se detalla a continuación.

6.5.1 Registro de huellas

Los archivos que llevan a cabo el registro de huellas son `Enroll_UNO.ino` y `Enroll_UNO_Wifi.ino`, que se localizan en la carpeta `/src`. Tras abrir estos dos archivos, conectamos los dos Arduinos a los puertos USB del ordenador. Ahora hay que decirle al Arduino IDE en qué placa y en qué puerto funcionan cada uno. El `Enroll_UNO.ino` funciona en el Arduino UNO. Es el encargado de registrar las huellas y mandar la información al Arduino UNO WiFi Rev2. Con el `Enroll_UNO` abierto hacemos click en Herramientas - Placa: y seleccionamos Arduino/Genuino UNO. Para seleccionar el puerto, pulsamos en Herramientas - Puerto y elegimos el puerto que tenga escrito al lado Arduino UNO, como se muestra en la Figura 15.

Se debe hacer lo mismo en el `Enroll_UNO_Wifi`, pero en este caso seleccionamos placa Arduino UNO WiFi Rev2 y el puerto en el que ponga al lado Arduino UNO WiFi Rev2. Este Arduino se encarga de mostrar la información por la pantalla.

Es importante tener en cuenta estas dos configuraciones cada vez que se use el programa ya que sino el código no se sube correctamente a los Arduinos y obviamente no funciona.

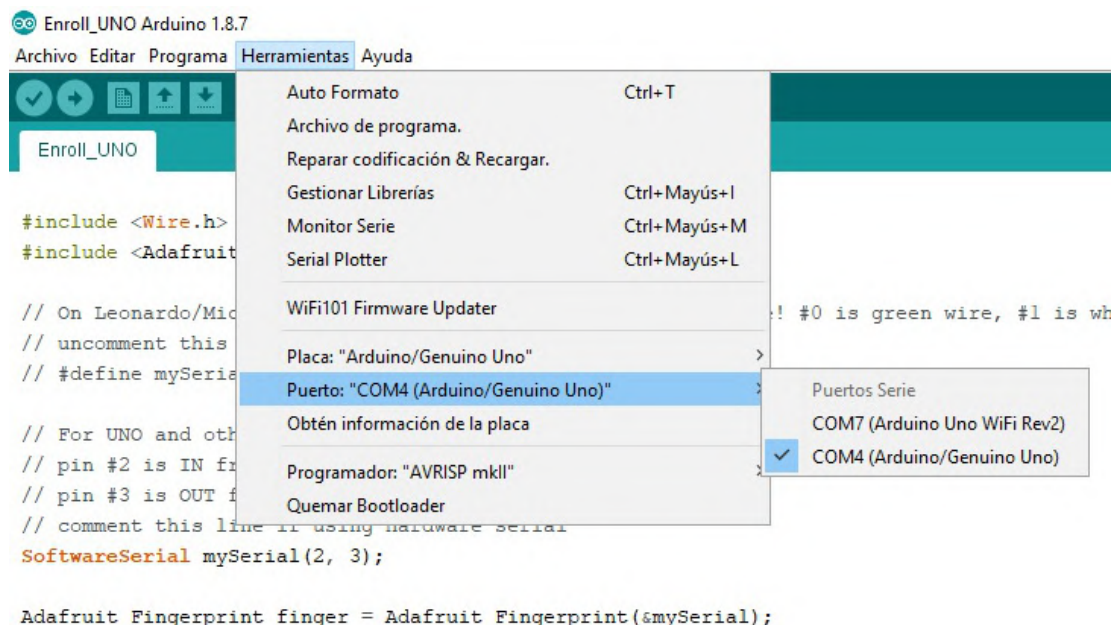


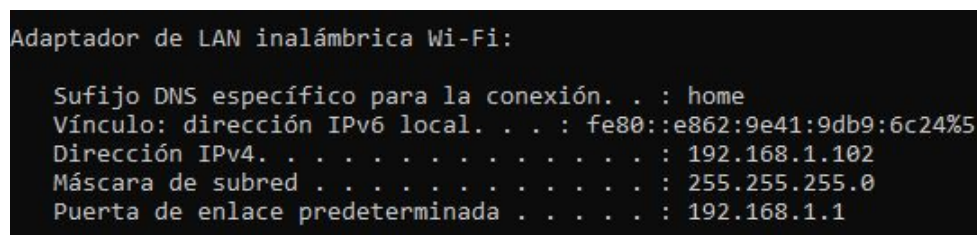
Figura 15: Configuración de puertos

Una vez configurado todo correctamente, solo queda pulsar en el segundo icono empezando por la izquierda (Subir). Hacemos esto para ambos códigos. Tras unos segundos, primero se compila el código y luego se sube a los Arduinos, el lector de huellas empezará a parpadear y el usuario debe seguir las instrucciones de la pantalla. Al final del proceso se le asigna un ID al alumno, que es el que se le debe poner al darle de alta en la base de datos. Este ID será el siguiente disponible que no esté siendo utilizado por el lector.

El procedimiento para registrar una huella dactilar se detalla más profundamente en la sección **7. Ejemplo de uso**.

6.5.2 Identificación de alumnos

Para identificar a los estudiantes y mostrar la información por la pantalla se deben cargar los archivos “ID_UNO.ino” e “ID_UNO_WiFi.ino”. Ambos se encuentran en la carpeta /src. Lo primero, tras conectar los Arduinos al PC si no se ha hecho ya, es seleccionar la placa y el puerto para cada archivo. “ID_UNO” se conecta al Arduino UNO y “ID_UNO_WiFi” al Arduino UNO WiFi Rev2. El código de “ID_UNO” se encarga de leer la huella del usuario, mientras que el código de “ID_UNO_WiFi” realiza la conexión con la base de datos y presenta los datos en la pantalla. Precisamente, con el fin de conectarse a la base de datos mediante WiFi, se debe modificar la variable SSID, referida al SSID de la conexión WiFi (el nombre de la misma); la variable pass, referida a la contraseña del router; y la variable server, siendo ésta la dirección IP del ordenador. Podemos encontrar nuestra dirección IP escribiendo el comando ipconfig en la consola de comandos y buscando el valor de la dirección IPv4, como se muestra en Figura 16.



```
Adaptador de LAN inalámbrica Wi-Fi:

Sufijo DNS específico para la conexión. . . : home
Vínculo: dirección IPv6 local. . . : fe80::e862:9e41:9db9:6c24%5
Dirección IPv4. . . . . : 192.168.1.102
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . . . : 192.168.1.1
```

Figura 16: Buscar dirección ipv4

Después de escribir el nombre del punto de acceso, la contraseña y la dirección IP, es preciso asegurarse de que los módulos Apache y MySQL del Panel de Control de XAMPP están iniciados, ya que sino lo están no se realizará la conexión con la base de datos. Una vez hecho esto, ya se pueden subir los códigos a los Arduinos. Después de unos segundos en los cuales se compila y se sube el código ya se podrá llevar a cabo la identificación de los alumnos. Esta funcionalidad se explica más detalladamente en la sección **7. Ejemplo de uso**.

6.5.3 Borrado de huellas

Las huellas de los alumnos se pueden eliminar de dos maneras, o bien borrando de una en una o bien eliminando todas a la vez. En ambos casos solo se necesitará conectar al puerto USB del ordenador el Arduino UNO.

Si se quiere borrar únicamente una huella, se debe abrir el archivo “deleteFingerprint.ino”, dentro de la carpeta /src. Después de seleccionar la placa y el puerto correcto (aquel dónde ponga Arduino UNO) y de subir el código al Arduino hay que abrir el Monitor

Serie haciendo click en Herramientas - Monitor Serie. A continuación se han de seguir las instrucciones que aparecen el Monitor Serie, las cuales consisten en introducir el ID del alumno cuya huella se quiere eliminar y pulsar ENTER. Nos pedirá que si estamos seguros pulsemos la tecla 'Y'. Tras ello, la huella se habrá eliminado.

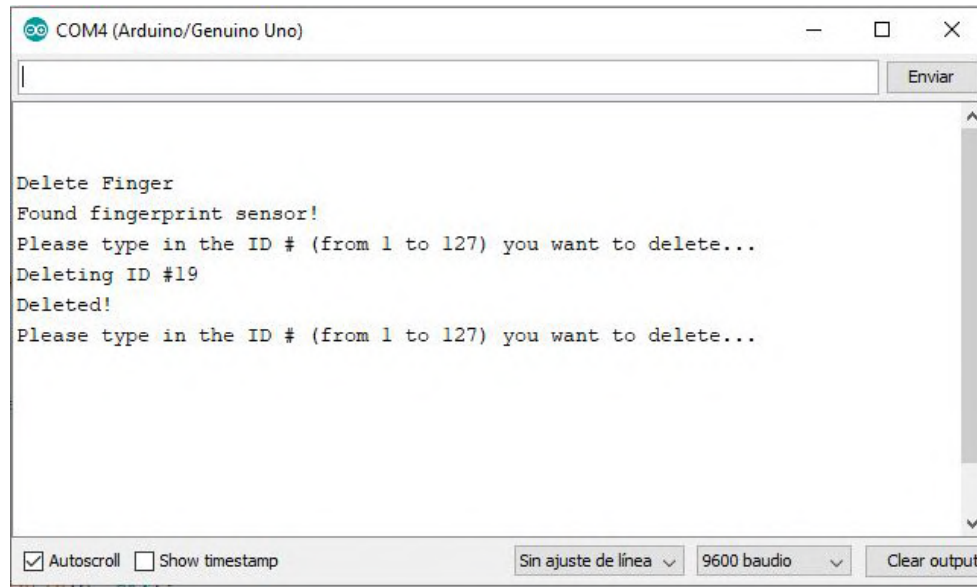


Figura 17: Borrar una huella

En caso de que se requiera la eliminación de todas las huellas reconocidas hay que cargar el archivo “emptyDatabase.ino”, que como todo el código se puede encontrar en la carpeta /src del proyecto. También se tiene que seleccionar la placa Arduino UNO y el puerto que corresponda a esta placa. Tras subir el código al Arduino se debe abrir el Monitor Serie y seguir las instrucciones, que en este caso son tan simples como escribir la letra 'Y' y pulsar ENTER. Una vez hecho esto se habrán borrado todas las huellas de los alumnos.

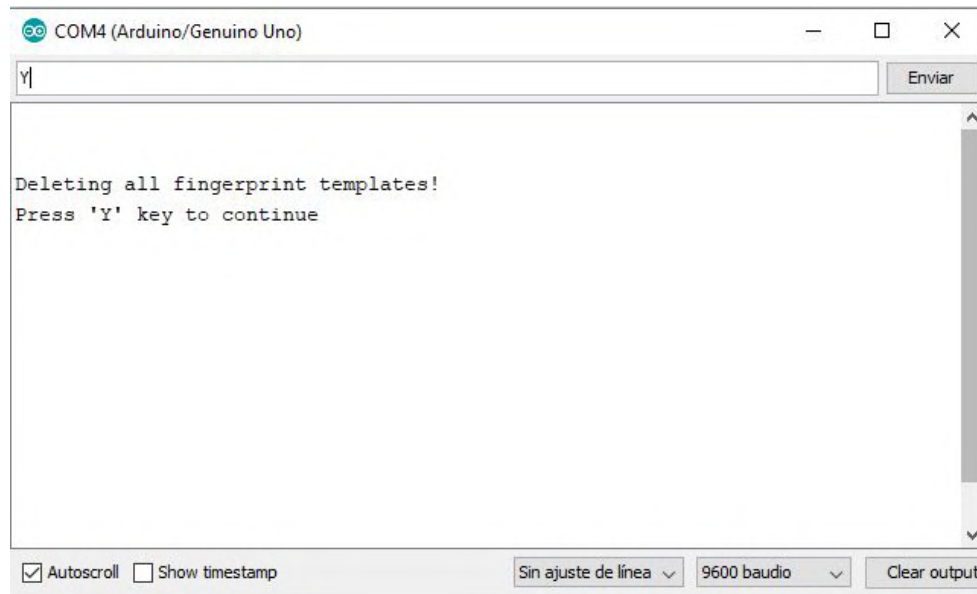


Figura 18: Borrar todas las huellas

6.5.4 Copia de seguridad

Para hacer un backup de las huellas guardadas en el lector hacen falta los archivos “backup.ino” y “backupProcessing.pde”, disponibles la carpeta /src del proyecto. Además hará falta el programa Processing (Figura 19), el cual podemos descargar la versión para nuestro sistema operativo de su página web [20]. Cuando tengamos la carpeta descargada, la descomprimos y ejecutamos la aplicación.

En el Arduino IDE con backup.ino abierto seleccionamos la placa Arduino UNO y el puerto que corresponda al Arduino UNO. Después subimos el código.

En el entorno de Processing ejecutamos el script backup.pde. Y en unos segundos se creará un archivo de texto con los códigos alfanuméricos que corresponden a cada huella. El código de Processing utilizado para hacer una copia de seguridad pertenece a Mohan Chandra Pradhan y se ha extraído de la página web [21]. Nosotros simplemente hemos cambiado el nombre del archivo de texto.

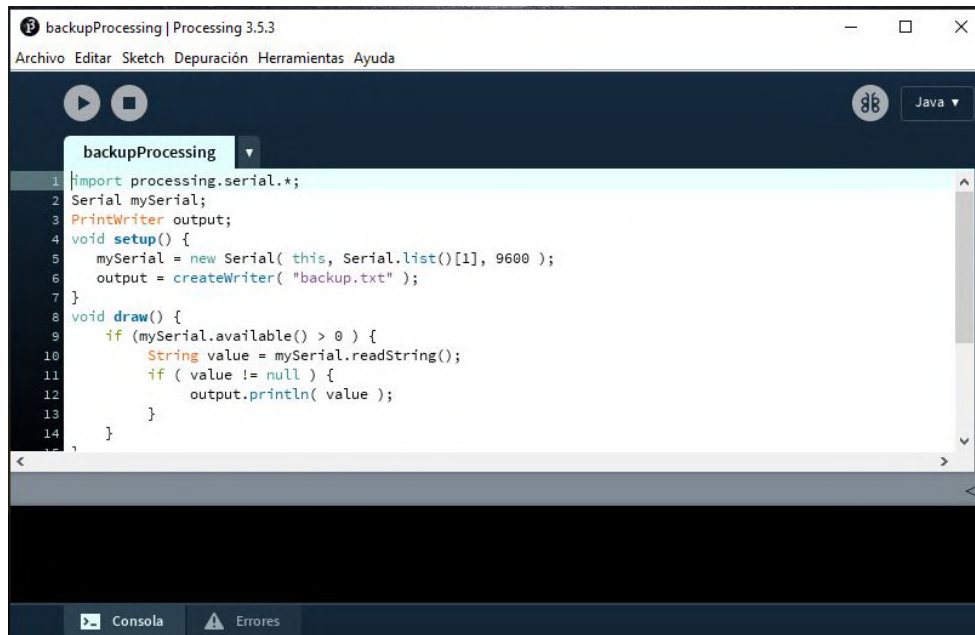


Figura 19: Processing

6.5.5 Formulario de alta y baja en la base de datos

El registro de los alumnos en la base de datos se hace desde un navegador de Internet accediendo a la dirección <http://localhost/comedor/bd.html>. Para que la página se cargue es necesario abrir el Panel de Control de XAMPP e iniciar los módulos Apache y MySQL. Tanto la tarea de añadir como de eliminar a un estudiante se hace de una manera muy simple a través de un formulario (Figura 20). La parte superior es para añadir y la zona inferior para eliminar. Tan solo hay que completar los campos requeridos:

- **ID del alumno:** se trata del numero de identificación que se le asigna al alumno al registrar su huella.
- **Nombre:** el nombre del alumno. Debido a que la pantalla TFT no muestra las tildes ni las ñ se debe evitar usar estos caracteres.
- **Primer apellido:** el primer apellido del alumno. Al igual que con el nombre no se pueden usar tildes ni la letra ñ.
- **Segundo apellido:** el segundo apellido del alumno. Este campo no es obligatorio ya que puede haber alumnos de países donde únicamente tengan un apellido.
- **Pago:** se debe marcar SÍ, si el alumno ha pagado el servicio de comedor, o NO si por el contrario no lo ha pagado.

- **Alergias:** Si el alumno en cuestión no tiene alergias se tiene que marcar la casilla Ninguna. En caso de que sí que tenga se deben marcar todos aquellas alérgenos a los que sea alérgico.

Después de rellenar todos los campos hay que hacer click sobre Añadir o Eliminar. Se abrirá una ventana y si el alta o la baja del alumno se ha realizado de una manera satisfactoria se informará mediante un mensaje. En caso contrario se mostrara un mensaje informando del error. Para volver al formulario hay que hacer click en Volver.

Registro para el comedor

ID del alumno

Nombre

Primer apellido

Segundo apellido

Pago

☐ Sí ☐ No

Alergias

- | | | | |
|-------------------------------------|-----------------------------------|---------------------------------------|----------------------------------|
| <input type="checkbox"/> Ninguna | <input type="checkbox"/> Pescado | <input type="checkbox"/> Frutos secos | <input type="checkbox"/> Lácteos |
| <input type="checkbox"/> Moluscos | <input type="checkbox"/> Gluten | <input type="checkbox"/> Crustáceos | <input type="checkbox"/> Huevos |
| <input type="checkbox"/> Cacahuetes | <input type="checkbox"/> Soja | <input type="checkbox"/> Apio | <input type="checkbox"/> Mostaza |
| <input type="checkbox"/> Sésamo | <input type="checkbox"/> Altramuz | <input type="checkbox"/> Sulfatos | |

AÑADIR

ID del alumno

ELIMINAR

Figura 20: Formulario de registro en la base de datos

6.6 Impedimentos

Durante el desarrollo del proyecto han ido surgiendo una serie de problemas e impedimentos que no nos han dejado más remedio que buscar soluciones y cambiar el rumbo del trabajo. Estos impedimentos se explican en detalle a continuación.

6.6.1 Driver del lector de huellas 1

Al principio del proyecto cuando estábamos juntando los diferentes dispositivos e investigando sobre las librerías que teníamos que emplear, nos encontramos con un escollo importante. El lector de huellas que tenía más capacidad y estaba preparado como lector de tarjetas RFID, no conseguíamos una librería que gestionase las señales eléctricas del cable y lo convirtiese en datos que el Arduino entendiese. Buscamos el modelo del lector de huellas 1 y no encontramos el modelo exacto y en las instrucciones no se mencionaba ninguna marca. Por esas razones, nos inclinamos por el segundo lector de huellas y centramos nuestros esfuerzos en la comunicación de este último con los Arduinos. Del segundo lector disponíamos de manuales y especificaciones técnicas (que se encuentran en el GitHub del proyecto) y hallamos la librería. Esta librería era oficial de Adafruit.

6.6.2 Avería de Arduino Uno Wifi

En el mes de marzo tuvimos una avería en el Arduino Uno WiFi. Un suceso que hizo que parasesmos con la evolución del proyecto y empezásemos a buscar un sustituto. Como ya se cuenta en el apartado de dispositivos, existía una intención clara de utilizar un Arduino mejor para que los tiempos de latencia fueran menores, pero, este accidente paró el proyecto y no dio tiempo ni a medir los tiempos de latencia de acceso a la base de datos del Arduino estropeado. Suponemos que hubo un cortocircuito en la placa, ya que al conectarlo se calentó demasiado y se apagó a los cinco segundos. Las veces siguientes que se intentó encenderlo, no fue posible.

Después de este percance, buscamos un Arduino que contuviera unas características básicas y a su vez mejorase con respecto al anterior para que tuviera más capacidad y ejecutase el programa a mayor velocidad con lo que redujese los tiempos de espera al usuario. Este Arduino fue el Arduino WiFi Rev2. Las características básicas que buscábamos eran: una distribución de los pines y tamaño en el que encajase la pantalla, una memoria suficiente, tenía que tener conectividad WiFi, etc. El nuevo Arduino era la siguiente versión del Arduino con el que habíamos trabajado. Al ser una nueva versión las características técnicas eran superiores.

6.6.3 Incompatibilidad de lector de huellas Adafruit con Arduino Uno WiFi Rev 2

Este evento sucedió después del evento del punto anterior 6.3.2. Nos dimos cuenta tras enfocar el desarrollo en el registro de las huellas mediante el Arduino. En ese momento

fuimos conscientes de que la librería de Adafruit (Adafruit Fingerprint Sensor) dejaba que usásemos las huellas almacenadas y reconocerlas si poníamos el mismo dedo en el lector, descargar los modelos de las huellas y visualizarlas por Serial (en el ordenador) pero no podíamos registrar nuevas huellas con el Arduino Uno WiFi Rev2.

Como una parte de nuestro proyecto consiste en registrar a nuevos alumnos, era un requerimiento esencial que teníamos que satisfacer. Investigamos en foros de Arduino y era un asunto que no estaba resuelto debido a la novedad del dispositivo (salió al mercado en enero de 2019). La solución que planteamos es que como el registro funcionaba en otros modelos de Arduino, podíamos encargar al Arduino Uno que realizase las operaciones con el lector de huellas y el Arduino Uno WiFi Rev2 que gestionara todas las demás funcionalidades: la pantalla y la conexión WiFi con la base de datos. Ya que, solo teníamos problema con el driver del lector de huellas. Para comunicarse entre ellos utilizamos el protocolo I2C que está contenido en el Arduino y que dispone de la librería Wire para gestionar la conexión mediante cable entre los Arduinos. Procedimos a extender un cable desde el pin SCL desde el Arduino Uno al mismo pin del Arduino Uno Wifi Rev2 y otro cable entre los pines SDA de los mismos Arduinos. El pin SCL comunica la señal de control del reloj y los ‘pines SDA realizan la transferencia de datos. Por tanto, el modelo que planteamos es un “Master and Slave”. El Arduino Uno es el Master, por tanto empieza la comunicación y toma el control del bus y el Arduino Uno WiFi Rev2 es el Slave y espera a que el Master deje el bus de datos para ocuparlo él.

6.6.4 Tildes y caracteres especiales

Al mostrar por la pantalla del Arduino la consulta de la base de datos, tuvimos problemas con las tildes y los caracteres que no estuviesen contenidos en la codificación ASCII. En los resultados de las consultas en HTML y PHP estos caracteres aparecían, sin embargo, en la pantalla no lo hacían. Hablamos de tildes y de la ñ por ejemplo.

La librería de la pantalla no los contenía, por tanto, nuestra solución fue prescindir de estos caracteres y ceñirnos a los que estuviesen incluidos. Se especificará en el manual de usuario, para que el administrador de la base de datos la haga pensando en esta limitación. Pensamos que, con las futuras actualizaciones de las librerías de los dispositivos, se podrá solucionar este condicionamiento que limita a usuarios de otros países no angloparlantes.

7 Ejemplo de uso

A continuación se expone un ejemplo de uso completo para las funcionalidades de registrar una huella y de identificar a un alumno.

7.1 Registro de huella

Para que un alumno pueda hacer uso del sistema de identificación primero ha de registrar su huella en el lector de huellas.

La pantalla del Arduino mostrará el mensaje “Coloca el dedo” (Figura 21). El alumno coloca el dedo en el sensor hasta que la pantalla indica que puede retirar el dedo (Figura 22). Se le pide que vuelva a colocar el mismo dedo una segunda vez (Figura 23). Enseguida se verá en la pantalla el mensaje “Huella guardada” y debajo de éste se le indica qué ID se le ha asignado (Figura 24). Este ID es el que debe utilizar el administrador al darle de alta en la base de datos.

En caso de que el estudiante coloque un dedo distinto la segunda vez, se le informará por la pantalla de que las huellas no coinciden (Figura 25) y tendrá que volver a ponerlo. El lector se queda con la última huella, por lo que el dedo que se debe volver a poner es el segundo. Si se pone el primero dirá que no coincide (Figura 25) y habrá que colocarlo una vez más.

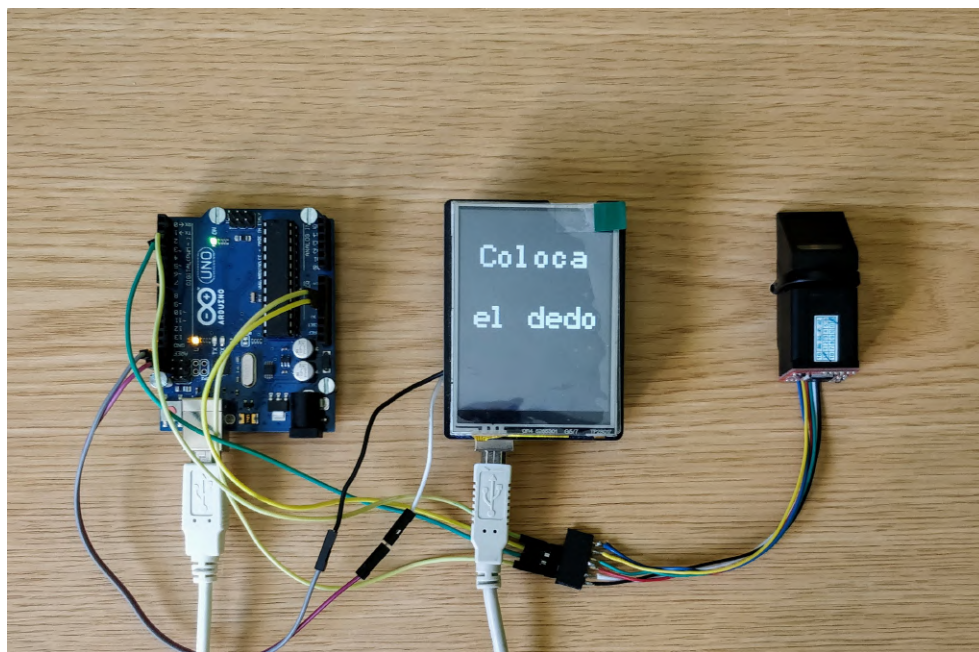


Figura 21: Coloca el dedo

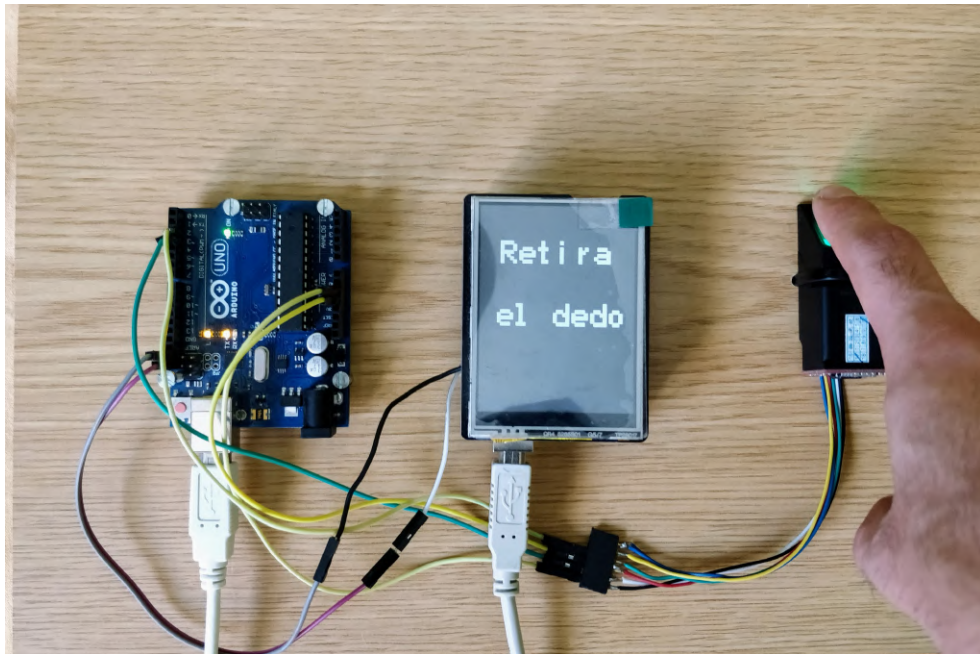


Figura 22: Retira el dedo

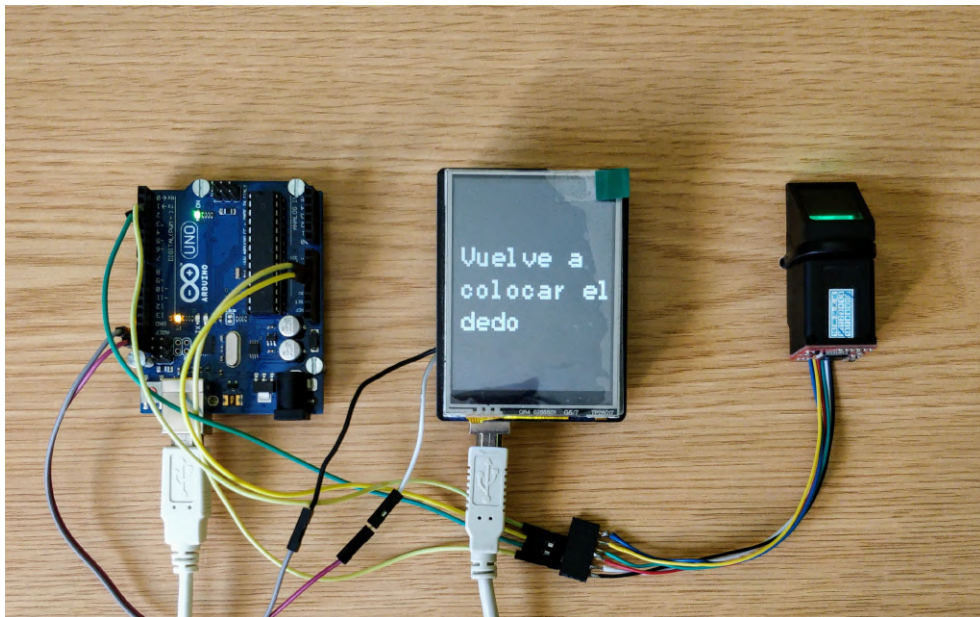


Figura 23: Vuelve a colocar el dedo

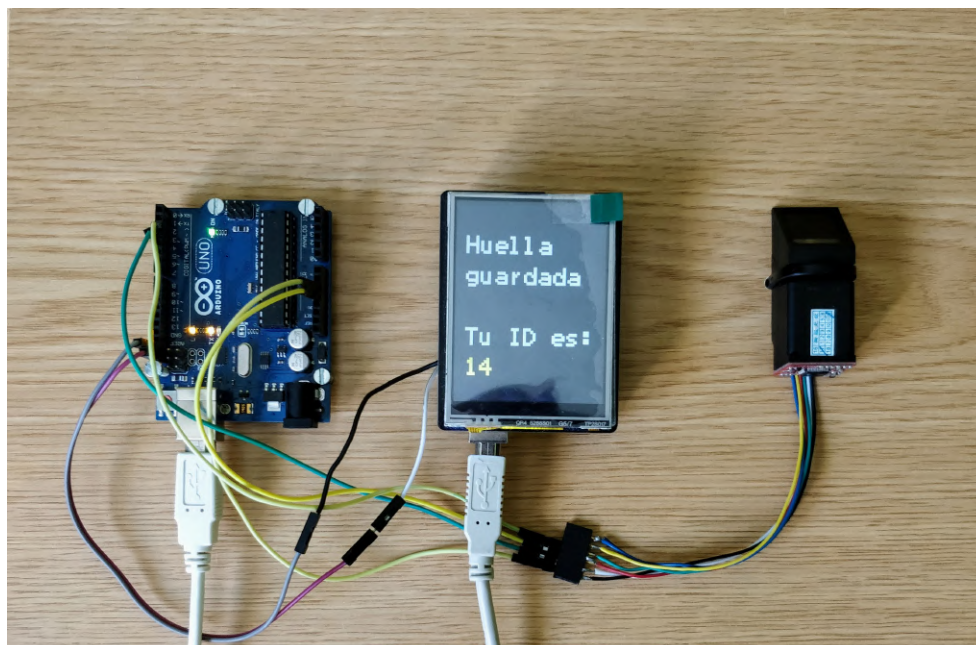


Figura 24: Huella guardada



Figura 25: Las huellas no coinciden

7.2 Identificación del alumno

Un alumno llega al comedor escolar. En la pantalla del Arduino se le indica que introduzca su huella dactilar (Figura 26). El estudiante coloca su dedo en el lector de huellas. Si se

reconoce su huella y está dado de alta en la base de datos, se muestra en la pantalla su nombre, apellidos y si ha realizado el pago o no (Figura 27). Después se muestran en la pantalla las alergias que tiene (Figura 28). En caso de que el alumno no tenga ninguna alergia se muestra en la pantalla la palabra “NINGUNA”. A continuación, si el menú del día contiene algún alérgeno que el alumno no pueda consumir, se muestra en la pantalla el mensaje “Menú especial sin:” y debajo estos alérgenos (Figura 29).

Si el alumno no tiene ninguna alergia o en el menú del día no hay ningún alérgeno al que el estudiante sea alérgico se verá en la pantalla el mensaje “MENU NORMAL” en verde (Figura 30).

En caso de que la huella del alumno no haya sido registrada, se muestra en la pantalla “Alumno no registrado” (Figura 31) y se vuelve al mensaje “Introduce huella” para que otro alumno se identifique.

Si el alumno sí que se ha registrado con anterioridad, pero aún no se le ha dado de alta en la base de datos, se le indica en la pantalla con el mensaje “Alumno no dado de alta en la BD” (Figura 32).

Y en caso de que el alumno no haya realizado el pago del comedor no se muestran sus alergias ni se le dice si puede comer el menú normal o si tiene que comer el menú especial. El sistema queda en espera de reconocer la huella de otro alumno.

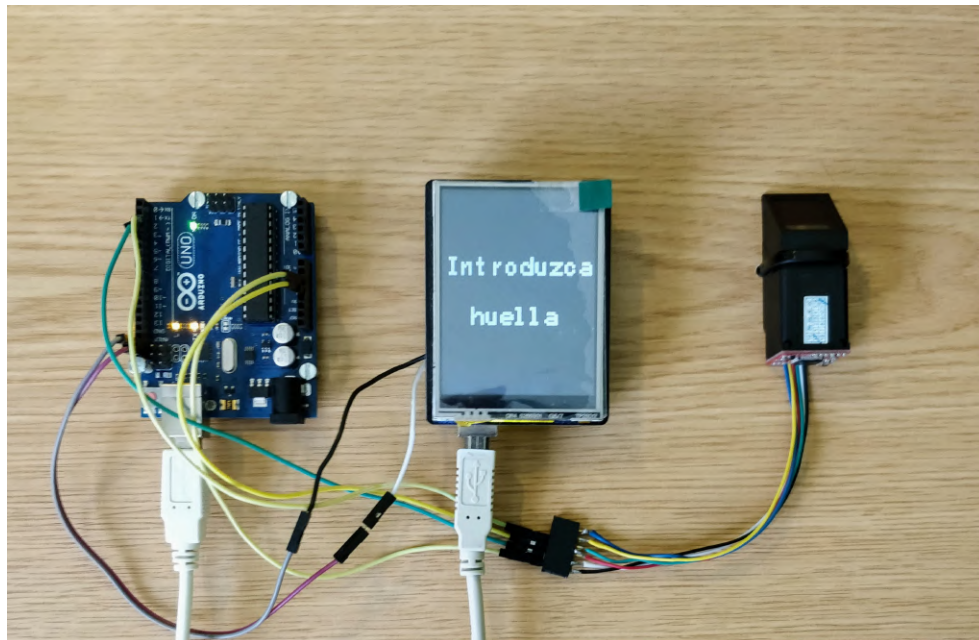


Figura 26: Introduzca huella

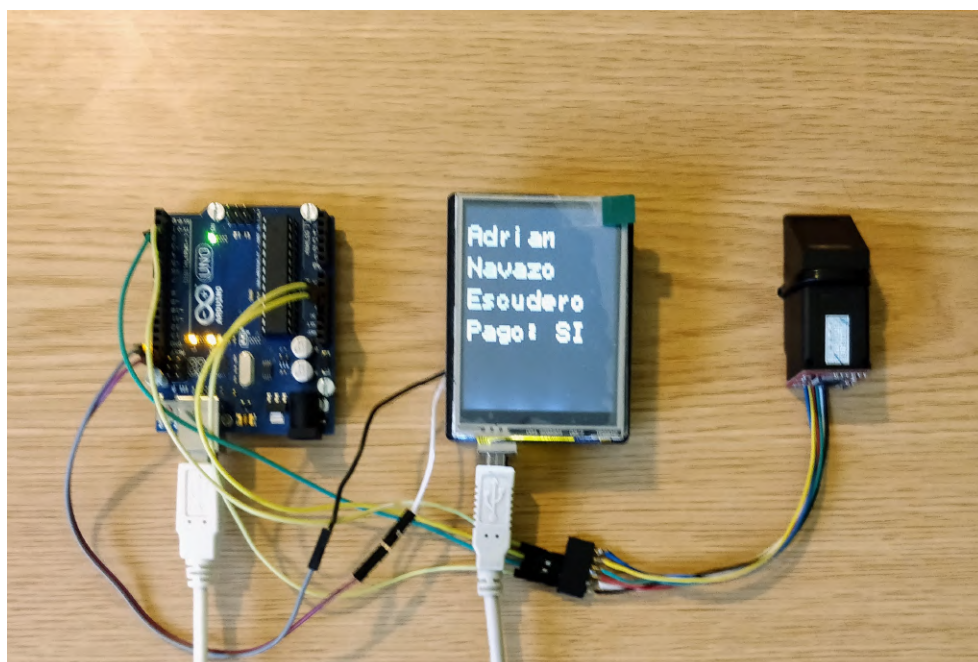


Figura 27: Datos del alumno

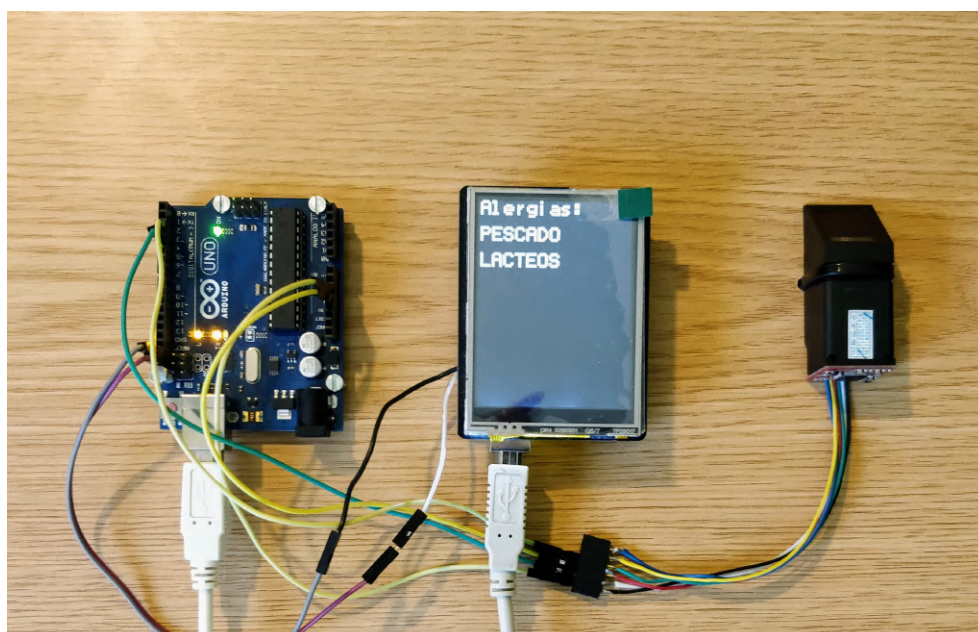


Figura 28: Alergias del alumno

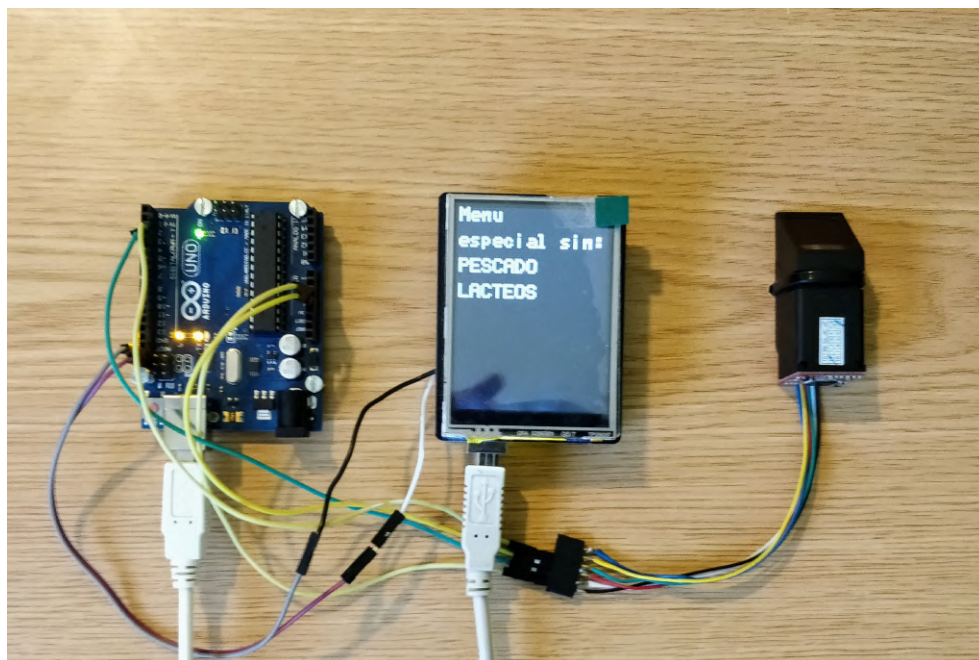


Figura 29: Menú especial sin pescado y lácteos

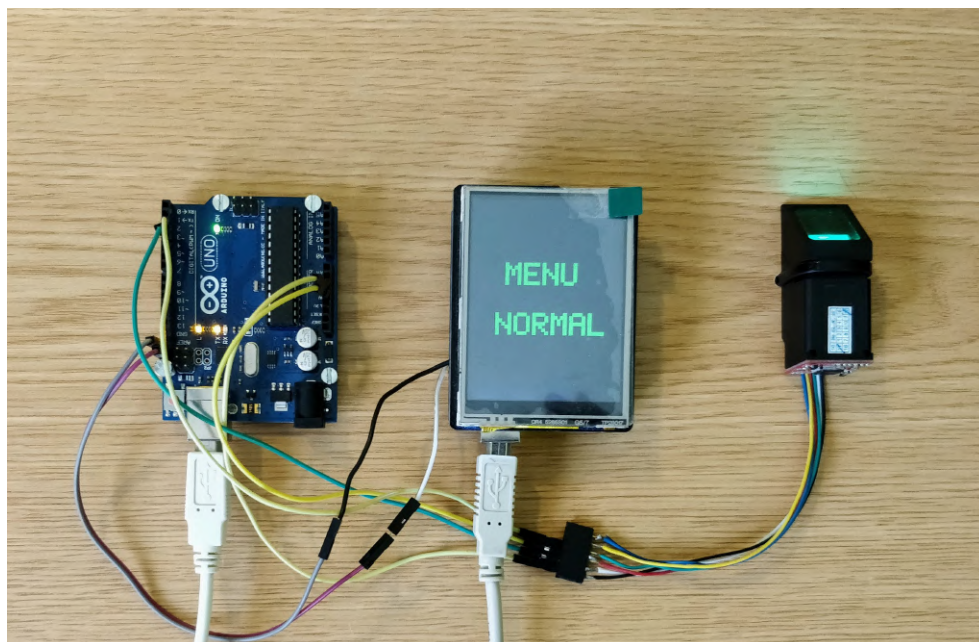


Figura 30: Menú normal

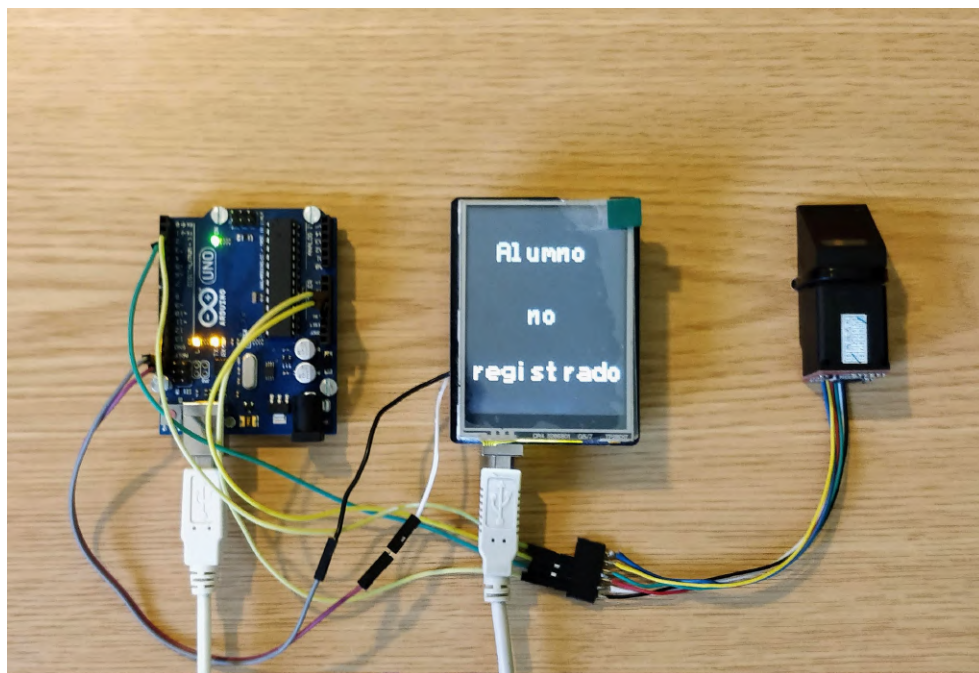


Figura 31: Alumno no registrado

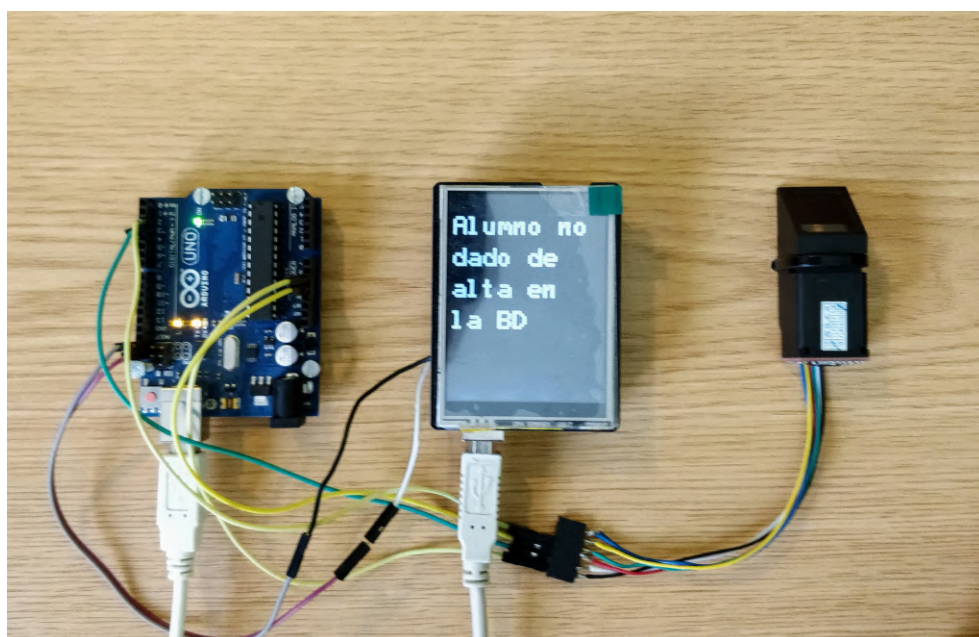


Figura 32: Alumno no dado de alta en la BD

8 Resultados

8.1 Tiempos de acceso a la red y a la base de datos

El Arduino UNO WiFi Rev2 se conecta a la red WiFi en unos 2 segundos. El Arduino UNO WiFi que se usó al principio tardaba algo más, pero se averió antes de poder hacer mediciones.

El intervalo de tiempo desde que se hace la petición a la base de datos hasta que se muestran los datos en el Monitor Serie y en la pantalla ha mejorado enormemente cuando se ha usado el Arduino UNO WiFi Rev2 con respecto a las pruebas con el Arduino UNO WiFi, ya que en el primero se produce casi de forma inmediata y en el segundo caso la espera era notable.

8.2 Memoria de los Arduinos

En este apartado se especifica el espacio de almacenamiento que usa cada código y cuanto ocupan las variables globales.

Enroll_UNO

El Sketch usa 7390 bytes (22 %) del espacio de almacenamiento de programa. El máximo es 32256 bytes. Las variables Globales usan 962 bytes (46 %) de la memoria dinámica, dejando 1086 bytes para las variables locales. El máximo es 2048 bytes.

Enroll_UNO_WIFI

El Sketch usa 8796 bytes (17 %) del espacio de almacenamiento de programa. El máximo es 49152 bytes. Las variables Globales usan 544 bytes (8 %) de la memoria dinámica, dejando 5600 bytes para las variables locales. El máximo es 6144 bytes.

ID_UNO

El Sketch usa 6610 bytes (20 %) del espacio de almacenamiento de programa. El máximo es 32256 bytes. Las variables Globales usan 711 bytes (34 %) de la memoria dinámica, dejando 1337 bytes para las variables locales. El máximo es 2048 bytes.

ID_UNO_WiFi

El Sketch usa 19906 bytes (40 %) del espacio de almacenamiento de programa. El máximo es 49152 bytes. Las variables Globales usan 860 bytes (13 %) de la memoria dinámica, dejando 5284 bytes para las variables locales. El máximo es 6144 bytes.

DeleteFingerprint

El Sketch usa 5344 bytes (16 %) del espacio de almacenamiento de programa. El máximo es 32256 bytes. Las variables Globales usan 601 bytes (29 %) de la memoria dinámica, dejando 1447 bytes para las variables locales. El máximo es 2048 bytes.

EmptyDatabase

El Sketch usa 4526 bytes (14 %) del espacio de almacenamiento de programa. El máximo es 32256 bytes. Las variables Globales usan 495 bytes (24 %) de la memoria dinámica, dejando 1553 bytes para las variables locales. El máximo es 2048 bytes.

Backup

El Sketch usa 6370 bytes (19 %) del espacio de almacenamiento de programa. El máximo es 32256 bytes. Las variables Globales usan 351 bytes (17 %) de la memoria dinámica, dejando 1697 bytes para las variables locales. El máximo es 2048 bytes.

9 Conclusión

A raíz de los resultados obtenidos en el apartado anterior, creemos que el cambio del Arduino UNO WiFi al Arduino UNO WiFi Rev2 nos ha beneficiado, ya que éste último se conecta mucho más rápido a la red y a la base de datos aunque tuviéramos que solventar lo del driver del lector de huellas.

Por otra parte, la elección de separar el código en dos Arduinos, debido a la imposibilidad de usar el lector de huellas con el Arduino UNO WiFi Rev2, ha sido una buena idea. Se reparte la carga de trabajo y el sistema es más robusto, aparte de que es más fácil detectar errores. Ningún código llega a ocupar el 50 % del almacenamiento disponible. El único que se acerca es ID_UNO_WiFi.ino con un 40 %.

En cuanto al proyecto en general, se han cumplido los objetivos declarados al principio. Los responsables del comedor pueden saber de una manera rápida y sin errores que alimentos pueden comer los alumnos, y los alumnos no se tienen que preocupar de nada. Además los administradores pueden dar de alta a los alumnos sin apenas esfuerzo. Ya que les brindamos las herramientas y el tutorial para que no tengan ni que saber programar.

En lo personal, gracias a este Trabajo de Fin de Grado hemos aprendido mucho. Hemos aprendido a programar para Arduino y qué tipos de Arduinos existen. También hemos ampliado nuestros conocimientos en HTML y PHP. y además hemos visto aplicados los conocimientos de bastantes asignaturas de la carrera, por citar algunas: Bases de Datos, Robótica, Fundamentos de Programación, Aplicaciones Web, ... También hemos visto un interés de las empresas en las entrevistas de trabajo sobre nuestro proyecto. Pero sobre todo nos sentimos muy orgullosos de haber hecho una aplicación que puede usarse para mejorar la vida de las personas con lo de las alergias alimentarias. Es un tema que nos pilla muy de cerca, ya que tenemos familiares y amigos con alergias alimentarias severas.

10 Conclusion

As a result of the outcomes obtained in the previous section, we believe that the change from the Arduino UNO WiFi to the Arduino UNO WiFi Rev2 has been to our advantage, since the latter connects much faster to the network and database even if we had to fix the driver of the fingerprint reader.

On the other hand, the choice of separating the code into two Arduinos, due to the impossibility of using the fingerprint reader with the Arduino UNO WiFi Rev2, has been a good idea. The workload is distributed and the system is more robust, and it is easier to detect errors. No code takes up to 50 % of the available storage. The only one that comes close is `ID_UNO_WiFi.ino` with 40 %.

Regarding the project generally, the objectives stated at the outset have been met. The people in charge of the canteen can know quickly and unmistakably what food the pupils can eat, and the pupils do not have to worry about anything. In addition, the administrators can register the students with hardly any effort. We provide them with the tools and the tutorial so that they don't even have to know how to program.

On a personal note, thanks to this Degree Final Project we have learned a lot. We have learned how to program for Arduino and what types of Arduinos exist. We have also expanded our knowledge in HTML and PHP. We have also applied the knowledge acquired in quite a few subjects of our degree, to name a few: Databases, Robotics, Programming Fundamentals, Web Applications, ... We have also perceived an interest from companies in job interviews about our project. But above all we are very proud to have made an application that can be used to improve the lives of people with food allergies. It's a topic that affects us personally, as some of our family and friends suffer from severe food allergies.

Referencias

- [1] <https://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino>
- [2] <https://forum.arduino.cc/>
- [3] <https://eprints.ucm.es/34796/>
- [4] <https://eprints.ucm.es/45230/>
- [5] <https://www.arduino.cc/en/Tutorial/PWM>
- [6] <https://www.arduino.cc/en/tutorial/memory>
- [7] <https://store.arduino.cc/arduino-uno-wifi>
- [8] <https://www.arduino.cc/en/Hacking/Bootloader?from=Tutorial.Bootloader>
- [9] <https://www.sparkfun.com/products/512>
- [10] <https://searchoracle.techtarget.com/definition/MySQL>
- [11] <https://www.phpmyadmin.net/>
- [12] <http://php.net/manual/es/intro-what-is.php>
- [13] <https://www.arduino.cc/en/Main/Software>
- [14] <https://www.apachefriends.org/es/index.html>
- [15] https://github.com/ABSysGroup/TFG2019_HUELLAS
- [16] <https://github.com/arduino-libraries/WiFiNINA>
- [17] https://github.com/Seeed-Studio/TFT_Touch_Shield_V2
- [18] https://github.com/Seeed-Studio/Touch_Screen_Driver
- [19] <https://github.com/adafruit/Adafruit-Fingerprint-Sensor-Library>
- [20] <https://processing.org/download/>
- [21] https://www.researchgate.net/post/How_can_I_save_data_directly_in_the_pc_using_an_Arduino_UNO

Bibliografía

- <http://forum.arduino.cc/index.php?topic=578515.15>
- <https://store.arduino.cc/arduino-uno-wifi>
- <https://store.arduino.cc/arduino-uno-wifi-rev2>
- <https://aprendiendoarduino.wordpress.com/2016/09/25/que-es-arduino/>
- <https://create.arduino.cc/projecthub>
- <https://www.arduino.cc/en/Guide/ArduinoUno#toc4>
- <https://www.arduino.cc/en/Guide/BoardAnatomy>
- <https://www.youtube.com/watch?v=KQiVLEhzzV0>
- <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-optical-fingerprint-sensor.pdf>
- <https://www.seeedstudio.com/2-8-TFT-Touch-Shield-V2-0-p-1286.html>
- <http://blog.elcacharreo.com/2016/09/30/probando-la-arduino-uno-wifi/>
- <https://www.arduino.cc/en/Guide/ArduinoUnoWiFi#toc7>
- <https://www.rinconingenieril.es/de-mysql-a-arduino/>
- https://www.youtube.com/watch?v=xajy_fMjTpY
- <http://www.sinaptec.alomar.com.ar/2017/06/tutorial-6-esp8266-mysql-php-en.html?m=1>
- <https://youtu.be/Mp81ekoXV4Yh>
- <https://randomnerdtutorials.com/fingerprint-sensor-module-with-arduino/>
- <https://www.youtube.com/watch?v=AZpKqEQS4e4>
- <http://www.forosdelweb.com/f18/solucionado-problemas-con-tildes-enes-php-mysql-498511/>
- <https://arduino.stackexchange.com/questions/1013/how-do-i-split-an-incoming-string>
- <http://www.chinanordson.com/standalone-fingerprint-access-control/fingerprint-access-control.html>

Imágenes

- <https://store.arduino.cc/arduino-uno-wifi>
- <https://store.arduino.cc/arduino-uno-rev3-with-long-pins>
- <https://store.arduino.cc/arduino-uno-wifi-rev2>
- <https://www.arduino.cc/en/Main/ArduinoBoardMegaADK?from=Main.ArduinoBoardADK>
- <https://www.arduino.cc/en/Main/arduinoBoardDuemilanove>
- <https://www.adafruit.com/product/751?gclid=CjwKEAj...>
- <https://www.seeedstudio.com/2-8-TFT-Touch-Shield-V2-0-p-1286.html>
- <https://www.carrefour.es/bateria-externa-poss-pspbli10400-negro/VC4A-3133845/p>
- <https://store.arduino.cc/usb-2-0-cable-type-a-b>