
Sistema de control de accesos para cumplir restricciones por COVID-19

Access control system to comply with restrictions due to COVID-19



UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE INFORMÁTICA

CURSO 2020/21

Autora

Julia Miguélez Fernández Villacañas

Director

José Ignacio Hidalgo Pérez

Trabajo Fin de Grado en Ingeniería Informática

6 de septiembre de 2021

“El agua goteante ahueca la piedra, no a través de la fuerza sino a través de la persistencia”

- Ovidio,

Resumen

Este proyecto consistirá en un control de acceso para un centro educativo. El propósito del mismo es llevar un registro en una base de datos de los alumnos que han asistido a un determinado aula en una franja horaria concreta para, poder controlar posibles brotes ocurridos y evitar su propagación. Así como negar el acceso a los individuos que no están autorizados, ya sea por no encontrarse en el subgrupo correspondiente o por suponer un riesgo para los demás, por haber estado en contacto con un positivo o estar contagiados.

Está implementado con dos dispositivos Arduino, una pantalla TFT, un lector de huellas y una base de datos.

Palabras clave

Arduino, reconocimiento de huella, control de acceso, base de datos, pantalla TFT, covid-19, coronavirus, biometría, Arduino Wifi

Abstract

This project will consist of an access control for an educational center. The purpose of it is to keep a record in a database of the students who have attended a certain classroom in a specific time slot in order to control possible outbreaks that have occurred and prevent their spread. As well as denying access to individuals who are not authorized, either because they are not in the corresponding subgroup or because they pose a risk to others, because they have been in contact with a positive or are infected.

It is implemented with two Arduino devices, a TFT screen, a fingerprint reader and a database.

Keywords

Arduino, fingerprint recognition, access control, data base, TFT screen, covid-19, coronavirus, biometrics, Arduino Wifi

Dedicatoria

Le dedico este proyecto tanto a mi familia, que me ha apoyado durante todo este proceso, como a mis amigos que han dejado una huella muy positiva en mí y han conseguido que con mucho tesón y perseverancia haya llegado hasta aquí. Gracias a sus ánimos he podido continuar y ver la luz al final del túnel.

Esto va por todos vosotros y espero que estéis muy orgullosos de mí.

Agradecimientos

A mi tutor Iñaki, que gracias a su entrega, insistencia y cariño ha conseguido que este proyecto salga adelante y que saque fuerzas de dónde no creía que tenía.

Muchísimas gracias.

Índice

| | | |
|----------|---|-----------|
| 1 | Introducción | 11 |
| 1.1 | Antecedentes | 11 |
| 1.2 | Motivación | 11 |
| 1.3 | Objetivos | 11 |
| 1.4 | Organización | 12 |
| 2 | Introduction | 13 |
| 2.1 | Background | 13 |
| 2.2 | Motivation | 13 |
| 2.3 | Objectives | 13 |
| 2.4 | Organization | 14 |
| 3 | Herramientas de desarrollo y Plan de Trabajo | 15 |
| 3.1 | Metodología de trabajo | 15 |
| 3.2 | Herramientas de desarrollo | 17 |
| 4 | Dispositivos | 18 |
| 4.1 | Arduino Uno | 18 |
| 4.2 | Arduino Uno Wifi Rev2 | 21 |
| 4.3 | Lector de huellas Adafruit | 23 |
| 4.4 | Pantalla Seeed 2.8 inch TFT Touch Shield v2.0 | 23 |
| 4.5 | Batería externa con dos puertos | 24 |
| 4.6 | Cable A/B de tipo preiférico | 25 |
| 5 | Diseño de la base de datos | 26 |
| 6 | Implementación | 29 |
| 6.1 | Programas y lenguajes | 29 |
| 6.2 | Librerías | 29 |
| 6.3 | Montaje | 29 |
| 6.4 | Manual de uso | 30 |
| 6.4.1 | Descarga de los programas y el repositorio | 30 |
| 6.4.2 | Adición de las librerías | 31 |
| 6.4.3 | Base de datos | 31 |
| 6.4.4 | Colocación de las carpetas que contienen el código HTML y PHP | 32 |
| 6.5 | Impedimentos | 33 |
| 6.5.1 | Cambio de versión del PHP e incompatibilidad de funciones | 33 |
| 6.5.2 | Malfuncionamiento del XAMPP en mi ordenador | 33 |
| 6.5.3 | Interpretación de tildes y otros caracteres especiales por el HTML y el envío de los mismos | 33 |

| | | |
|-----------|---|-----------|
| 7 | Ejemplo de uso | 34 |
| 7.1 | Registro de alumno en el sistema | 34 |
| 7.1.1 | Registro de huella en el Arduino | 34 |
| 7.1.2 | Registro del alumno en el formulario | 37 |
| 7.2 | Acceso al aula | 39 |
| 7.3 | Asignación de grupos a aulas | 45 |
| 7.4 | Notificación de contagio o contacto con un positivo | 46 |
| 7.5 | Verificación de contagio | 47 |
| 8 | Resultados | 48 |
| 8.1 | Memoria de los Arduinos | 48 |
| 9 | Conclusión | 49 |
| 10 | Conclusion | 49 |

Índice de figuras

| | | |
|----|---|----|
| 1 | Diagrama de flujo del proyecto | 15 |
| 2 | Arduino Uno | 19 |
| 3 | Mapeo de Pines del Arduino UNO Rev3 | 20 |
| 4 | Arduino Uno Wifi Rev2 | 21 |
| 5 | Mapeo de Pines del Arduino UNO Wifi Rev2 | 22 |
| 6 | Sensor de huellas Adafruit | 23 |
| 7 | Pantalla Seeed TFT de 2.8 pulgadas | 24 |
| 8 | Batería externa | 25 |
| 9 | Cable A/B Macho/Macho de tipo periférico | 26 |
| 10 | Modelo E/R de la base de datos | 28 |
| 11 | Montaje y cableado | 30 |
| 12 | Iniciar Apache y MySQL | 31 |
| 13 | Importar Base de Datos en PHPMyAdmin | 32 |
| 14 | Comprobación de Base de Datos importada | 32 |
| 15 | Coloca el dedo | 35 |
| 16 | Retira el dedo | 35 |
| 17 | Vuelve a colocar el dedo | 36 |
| 18 | Huella guardada | 36 |
| 19 | Las huellas no coinciden | 37 |
| 20 | Pantalla de inicio de los formularios | 38 |
| 21 | Instrucciones para el registro del alumnado | 38 |
| 22 | Formulario de registro de los alumnos en la base de datos | 39 |
| 23 | Nuevo alumno registrado en la tabla Alumnos | 39 |
| 24 | Mensaje de bienvenida | 40 |
| 25 | Petición de huella | 40 |
| 26 | Alumno no registrado | 41 |
| 27 | Datos referentes a la asignación de grupo a esa aula | 42 |
| 28 | Diríjase a la clase que le corresponde | 42 |
| 29 | Clase habilitada para el otro color | 43 |
| 30 | Acceso bloqueado | 44 |
| 31 | Aforo completo | 44 |
| 32 | Acceso permitido | 45 |
| 33 | Formulario asignación aulas | 45 |
| 34 | Formulario de registro de contagio o contacto de un alumno | 46 |
| 35 | Calendario del formulario de contagio | 47 |
| 36 | Registro del alumno en la tabla cuarentena | 47 |
| 37 | Verificación del contagio o contacto estrecho con un positivo | 48 |

1 Introducción

En estos dos años que llevamos viviendo bajo una epidemia, hemos visto como las herramientas digitales pueden ayudar a sobrellevar el momento tan complicado e incluso proporcionar recursos para acotar los brotes y llegar en un futuro a controlar la enfermedad. Disponemos de múltiples ejemplos en las noticias de aplicaciones informáticas que se han utilizado para notificar los casos positivos e informar a los contactos cercanos. En España, incluso hubo una aplicación lanzada por el Gobierno estatal con este objetivo [1].

1.1 Antecedentes

El principal antecedente de este trabajo ha sido el realizado por Adrián Navazo Escudero y por mí misma, titulado **Sistema de control de acceso con huella dactilar y Arduino**[2]. Fue realizado en el curso 2018/19 en la Facultad de Informática y cuyos tutores fueron José Ignacio Hidalgo Pérez y Juan Lanchares Dávila. Por tanto este proyecto es ampliación del citado e incluye partes del mismo.

1.2 Motivación

Como está expuesto en el punto anterior, teníamos un sistema de control de acceso del anterior TFG y se nos ocurrió partir de ahí adecuarlo a la situación actual. Creo que la informática debe ponerse a disposición de la ciudadanía y, con la relevancia social que ha tenido la crisis del Coronavirus y el impacto que ha tenido en la facultad, era necesario orientar el proyecto en ese sentido.

A nivel personal, la situación ha tocado de cerca y a nivel profesional, he tenido la suerte de estar involucrada en dos aplicaciones relacionadas con el Covid-19 para dos consejerías autonómicas. Por tanto, existen motivos suficientes para dedicar este proyecto final de grado a la gestión del Covid-19 e intentar aportar una solución para mitigar el efecto del mismo.

1.3 Objetivos

Dotar de una herramienta para controlar el acceso de individuos a un recinto utilizando datos reales del aforo del mismo y permitir el paso solo a los que lo tengan permitido en ese día. También proporcionar un registro de los alumnos que están en cuarentena e impedir su acceso.

Con estas herramientas se pretende otorgar conocimiento sobre la situación epidemiológica en un centro educativo, viendo los registros para ayudar en la toma de decisiones en el centro sobre la crisis del Coronavirus. Además es un elemento físico visual a la entrada de cada aula que puede evitar despistes de aulas y organiza los espacios.

1.4 Organización

En los siguientes epígrafes iré desgranando los siguientes contenidos:

- **3. Herramientas de trabajo y plan de trabajo.** En este primer punto, se explicarán las herramientas que he usado para la comunicación con el tutor y para la planificación del proyecto.
- **4. Dispositivos.** En el cuarto punto, se expondrán los dispositivos utilizados y sus características técnicas.
- **5. Base de datos.** En el punto cinco, se explicará la estructura de las tablas dentro de la base de datos y el gestor de base de datos empleado.
- **6. Implementación.** En el sexto apartado, se describirá el manual de uso, las librerías de Arduino usadas, la implementación y las dificultades encontradas.
- **7. Ejemplo de uso.** En este capítulo se hará una demostración de todas las funcionalidades implementadas en este trabajo.
- **8. Resultados.** En la octava sección se tratarán los resultados obtenidos, el volumen de memoria usada y los datos técnicos.
- **9. Conclusión.** Para clausurar, se comentarán las conclusiones del proyecto y últimas aclaraciones.

2 Introduction

In these two years that we have been living under an epidemic, we have seen how digital tools can help us cope with such a difficult time and even provide resources to limit outbreaks and control the disease in the future. We have multiple examples in the news of computer applications that have been used to report positive cases and inform close contacts. In Spain, there was even an application launched by the state government for this purpose [1].

2.1 Background

The main antecedent of this work has been the one carried out by Adrián Navazo Escudero and myself, entitled **Access control system with fingerprint and Arduino** [2]. It was carried out in the 2018/19 academic year at the Faculty of Informatics and whose tutors were José Ignacio Hidalgo Pérez and Juan Lanchares Dávila. Therefore this project is an extension of the mentioned and includes parts of it.

2.2 Motivation

As stated in the previous point, we had an access control system from the previous TFG and it occurred to us to start from there and adapt it to the current situation. I believe that information technology should be made available to the public and, with the social relevance that the Coronavirus crisis has had and the impact it has had on the faculty, it was necessary to orient our project in that sense.

On a personal level, the situation has touched closely and on a professional level, I have been fortunate to be involved in two applications related to Covid-19 for two regional councils. Therefore, there are sufficient reasons to dedicate this final degree project to the management of Covid-19 and try to provide a solution to mitigate its effect.

2.3 Objectives

Provide a tool to control the access of individuals to a venue using real data of the capacity of the same and allow the passage only to those who are allowed it on that day. Also provide a record of students who are quarantined and prevent their access.

These tools are intended to provide knowledge about the epidemiological situation in an educational center, viewing the records to help in decision-making in the center about the Coronavirus crisis. It is also a visual physical element at the entrance of each classroom that can avoid classroom confusion and organize spaces.

2.4 Organization

In the following sections I will go over the following contents:

- **3. Work tools and work plan.** In this first point, the tools that I have used for communication with the tutor and for planning the project will be explained.
- **4. Devices.** In the fourth point, the devices used and their technical characteristics will be exposed.
- **5. Database.** In point five, the structure of the tables within the database and the database manager used will be explained.
- **6. Implementation.** In the sixth section, the user manual, the Arduino libraries used, the implementation and the difficulties encountered will be described.
- **7. Use case.** In this chapter there will be a demonstration of all the functionalities implemented in this work.
- **8. Results.** The eighth section will deal with the results obtained, the volume of memory used and the technical data.
- **9. Conclusion.** For closing, the conclusions of the project and final clarifications will be discussed.

3 Herramientas de desarrollo y Plan de Trabajo

3.1 Metodología de trabajo

La metodología empleada en este proyecto se basa en Scrum ya que se han hecho diversos sprints por funcionalidades. Estos sprints consistieron en análisis, desarrollo y testing de los diferentes componentes de este proyecto. A continuación se explican las etapas:

- **Planificación de objetivos:** En la primera fase del proyecto, nos reunimos el tutor y yo para establecer los objetivos del mismo y plantear por dónde queríamos encaminar el proyecto. De estas reuniones, entre otras cosas, salió un diagrama de flujo. Esta figura, es el diagrama refinado:

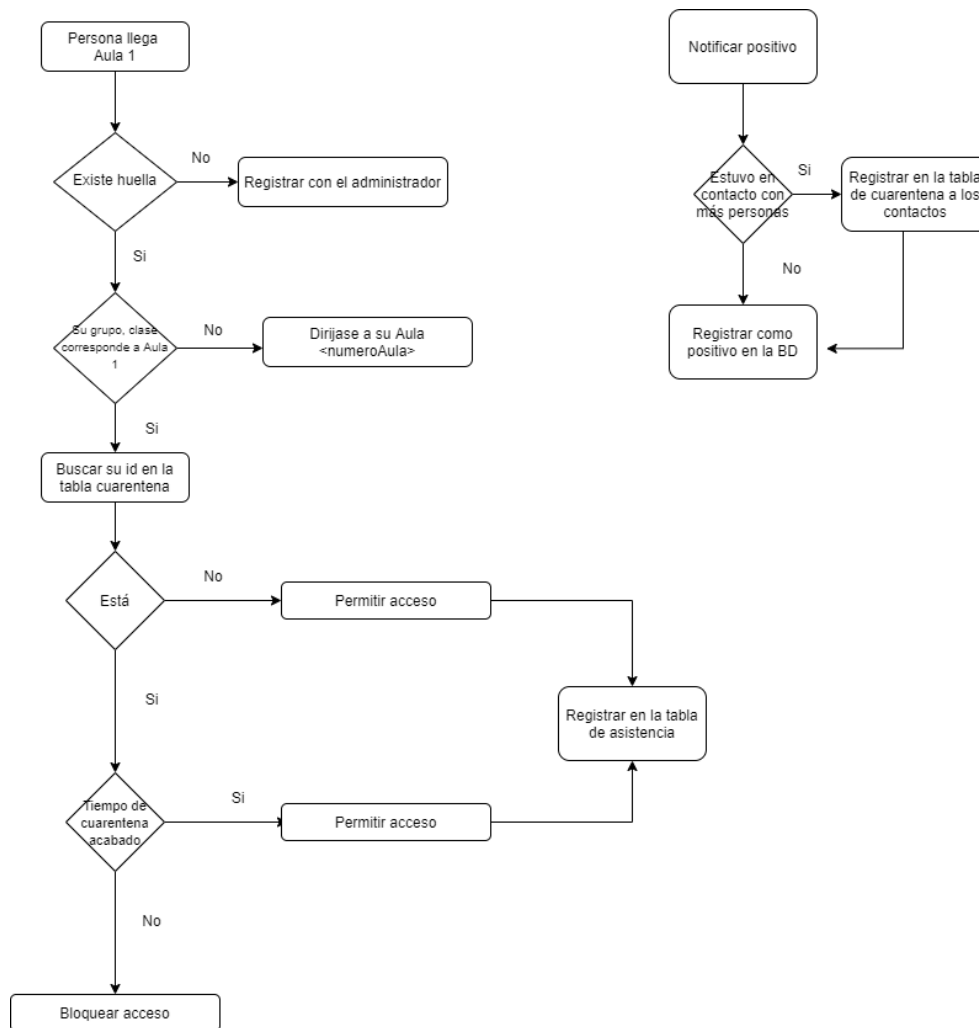


Figure 1: Diagrama de flujo del proyecto

- **Desarrollo de la Base de Datos:** Como la Base de Datos del proyecto anterior no encajaba con las necesidades del nuevo proyecto, se pasó por unas fases de planteamiento de las distintas tablas. Para empezar, se tenía de referencia la tabla alumnos pero hubo un cambio de atributos. Se añadieron a los básicos como nombre, apellido1 y apellido2 los referentes al grupo y al color.

Siguiendo con la propuesta de tablas, era imperiosa la creación de una tabla de asignación a aulas para poder distribuir a los alumnos. Con ello, se incorporó la tabla de aulas. Para acabar, se establecieron las últimas dos tablas: asistencia y cuarentena como modo de llevar un registro de los alumnos y los casos de Coronavirus, respectivamente.

- **Desarrollo de los formularios:** Habiendo completado el punto anterior, se tenían unas bases para diseñar los campos de los formularios. Se empezó por la parte de HTML del registro y el de cuarentena. Más tarde, se programó la parte de PHP y se enlazó todo. Se hicieron diversas pruebas para ver que el contenido estaba siendo registrado en la base de datos. En este punto ocurrieron algunos impedimentos explicados en el capítulo *Impedimentos* relacionados con la interpretación de archivos de extensión .php .
- **Desarrollo de la app de Arduino:** Este sprint fue el más importante ya que cerraba la funcionalidad más importante y con ello se completaba el proyecto (a espera de nuevas funcionalidades sugeridas para incorporar a las ya implementadas). Se comenzó por la parte de registro, revisando el código para ver si alguna función había quedado obsoleta. A continuación, se registraron huellas y se procedió a continuar con la parte principal. El sketch ¹ de Aula_Uno_Wifi.ino es el que más carga tenía ya que el de ID_Uno.ino solo contenía la gestión del lector de huella. Se reprodujo el diagrama mencionado en el punto de *Planificación de objetivos*, para ello se utilizaron las funciones auxiliares:

```
comprobarGrupoClaseColor(String grupoAula, int claseAula, String colorAula,
String grupoAlum, int claseAlum, String colorAlum, bool& prohibido)
```

```
void compruebaContagiado(bool hayContagio, int idAlumno, int numAula,
int& aforo, int capacidadAula)
```

```
int datosAula(int claseAula, String& grupoAula, String& colorAula,
int& capacidadAula, String& fechaHoy)
```

```
int datosAlumno(int identificador, String& grupoAlumno, String& colorAlumno,
bool& hayContagioAlumno, int& idAlumnoBD)
```

¹Nombre que se le da en Arduino a un programa [3]

```
void registrarAlumno(int idAlumno, int numAula)
```

La función *datosAula* recaba los atributos del aula de la tabla *asignacion_aula* y son previos a que el alumno pulse el lector (se actualizan diariamente), así como la capacidad de la clase en la tabla *aulas*. El parámetro de *prohibido* es creado como flag para impedir el acceso si ya se le ha denegado en la función anterior, de igual forma *hayContagio* marca si hay un registro vigente de contagio del alumno. En la función auxiliar *compruebaContagiado* se comprueba en la tabla *cuarentena*. Por último, La función *registrarAlumno* introduce los valores del alumno en la tabla de *asistencia* una vez que este tenga permitido el acceso. Las demás variables son obtenidas de la tabla *alumnos*.

- **Realización de la memoria:** La realización de la memoria se fue haciendo en paralelo. Aunque antes de la entrega del Borrador se priorizó su escritura y revisión. Fue realizada en L^AT_EX con la herramienta Overleaf. Para estructurar la misma se realizó un visionado de otros TFGs, una lectura exhaustiva de las normas en la página de la Facultad de Informática.
- **Mejoras en el código:** Se plantearon mejoras, como centralizar los formularios en una página índice que los enlazara. En la parte estética se incorporaron los colores amarillo y azul que representan a la facultad de Informática (se obtuvieron del tríptico del grado de Ingeniería Informática ofertado en la página web[4]) A su vez, se añadieron nuevos formularios como la asignación de aulas y el de verificación de contagio. También se modificaron los mensajes tanto de error como de información mostrados. Con respecto al Arduino, se optó por colorear ciertos mensajes en la pantalla para favorecer la comprensión de los mismos.
- **Últimos cambios de la memoria y entrega final:** Para terminar, se completaron algunos puntos en la memoria que tenían que ser actualizados por los cambios realizados en el código, los formularios y se revisó completamente la memoria para la entrega final.

3.2 Herramientas de desarrollo

Para la realización de este proyecto, se han utilizado herramientas tanto de comunicación (con el tutor), como de desarrollo de software y organización del tiempo:

- **Gmail:** Para concertar reuniones y como modo de comunicación utilizamos la cuenta institucional de la UCM, gestionada por Gmail.
- **Google Meet:** Tuvimos tanto reuniones presenciales como reuniones virtuales con la herramienta de Google Meet.

- **Google Drive:** Se ha utilizado como repositorio de documentos y temporalmente de código, además de backup de la información almacenada en local.
- **Diagrams.net:** Para la realización de los diagramas mencionados en el punto *Planificación de objetivos*, a parte de diseñarlo en papel se pasaron a esta herramienta.
- **Google Calendar:** Como organización propia de las fechas límite y gestión del tiempo se ha empleado este calendario.
- **Notepad++:** Para la edición del código en HTML y PHP y para la visualización del código SQL en local.
- **Google Chrome:** Se ha utilizado tanto para la opción trivial de búsqueda de información y explorador para poder acceder a otras herramientas aquí listadas, como para la ejecución de los formularios.
- **Overleaf:** Este editor de $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ online, ha sido útil para la redacción de la memoria.

4 Dispositivos

Por la parte del hardware de este proyecto han intervenido dos tipos de Arduinos: Arduino Wifi Rev2 y Arduino Uno, un lector de huellas Adafruit, una pantalla TFT y una batería externa. A continuación se exponen sus características técnicas:

4.1 Arduino Uno

Se reutilizó el montaje del TFG anterior con lo cuál el modelo de Master-Slave se mantuvo. El Arduino Uno toma el rol de Master y se encarga de la gestión del lector de huellas y mandar la información al Slave (Arduino Uno Wifi Rev2). En la figura 2 se puede apreciar este dispositivo:

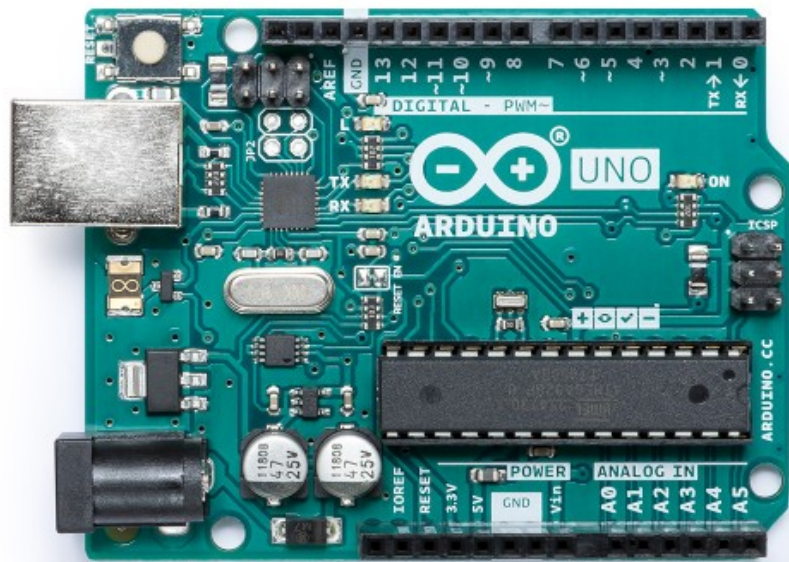


Figura 2: Arduino Uno

Características técnicas:

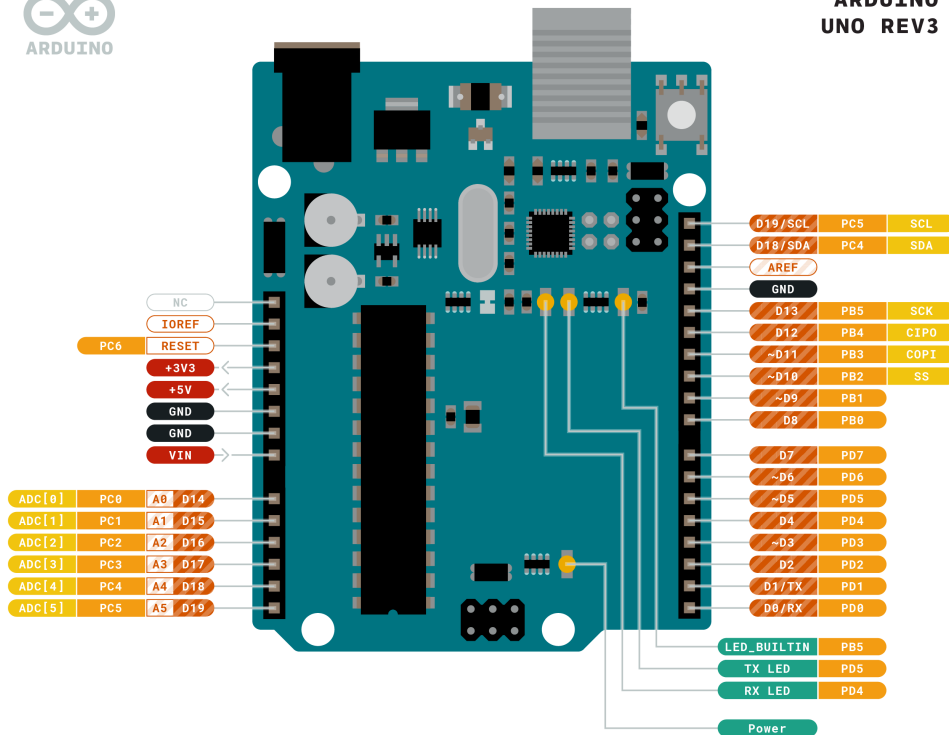
| | |
|---|---|
| Microcontrolador | ATmega328P |
| Pins de E/S (Entrada/ Salida) digitales | 14 |
| Salidas PWM ² | 6 |
| Pins de entrada analógicos | 6 |
| Voltaje operativo | 5 V |
| Memoria flash | 32 KB (0.5 KB usados por el bootloader) |
| SRAM | 2 KB |
| EEPROM | 1 KB |
| Velocidad del reloj | 16 MHz |

Mapeo de los pines:³

³imagen extraída de la web de Arduino: <https://store.arduino.cc/products/arduino-uno-rev3>



ARDUINO UNO REV3



| | | | |
|--------|--------------|-------------|------------------------|
| Ground | Internal Pin | Digital Pin | Microcontroller's Port |
| Power | SWD Pin | Analog Pin | |
| LED | Other Pin | Default | |

ARDUINO . CC

This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0/> or send a letter to Creative Commons, PO Box 1888, Mountain View, CA 94042, USA.

Figura 3: Mapeo de Pines del Arduino UNO Rev3

4.2 Arduino Uno Wifi Rev2

Según la web de Arduino es el componente básico para los proyectos de IoT. [5] En el proyecto tiene el rol de Slave y se monta la pantalla encima sobre los pines. Figura 4

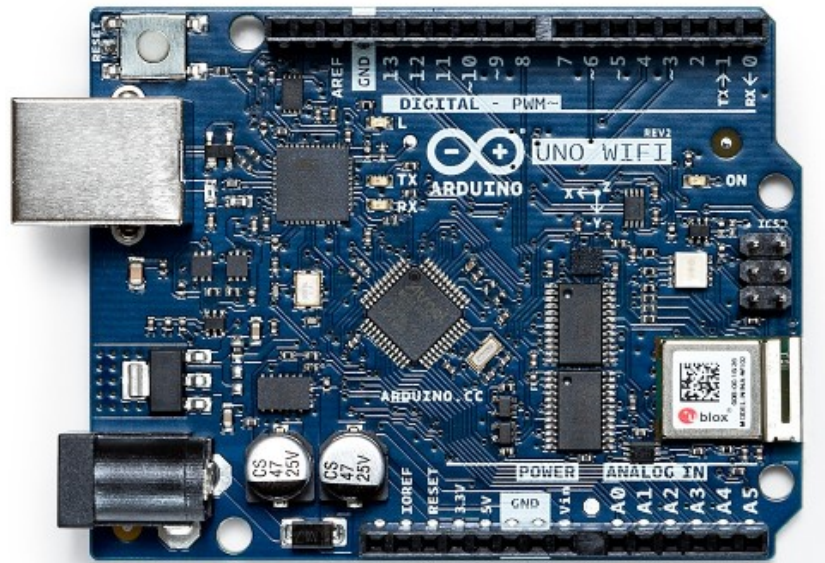


Figura 4: Arduino Uno Wifi Rev2

Características técnicas:

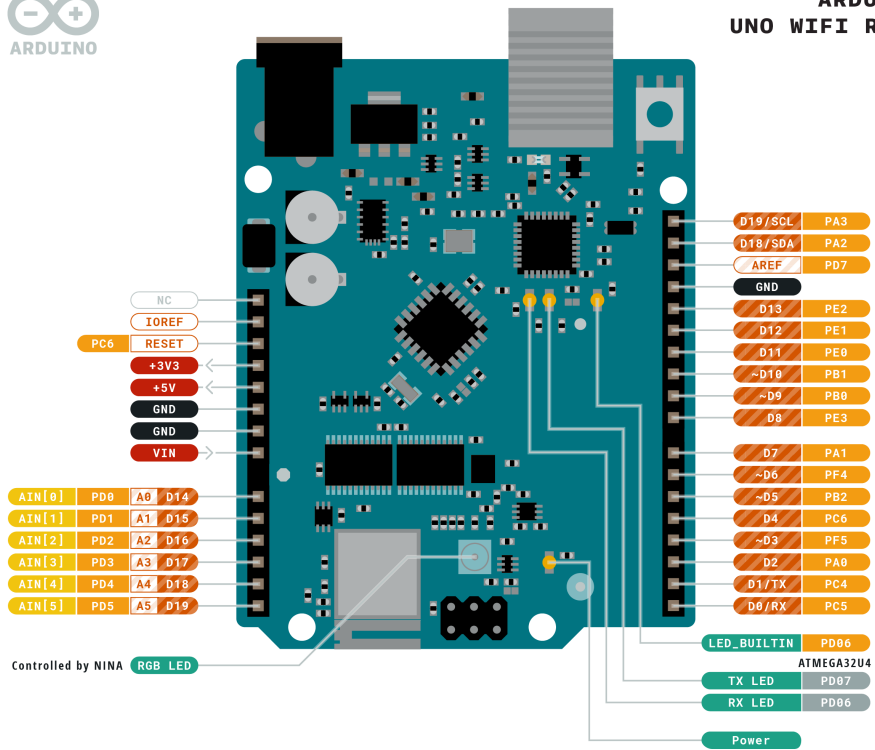
| | |
|---|----------------------|
| Microcontrolador | ATmega4809 |
| Pins de E/S (Entrada/ Salida) digitales | 14 |
| Salidas PWM ⁴ | 5 |
| Pins de entrada analógicos | 6 |
| Voltaje operativo | 5 V |
| Memoria flash | 48 KB |
| SRAM | 6,144 Bytes |
| EEPROM | 256 Bytes |
| Velocidad del reloj | 16 MHz |
| WiFi | 802.11 b/g/n 2.4 GHz |

Mapeo de los pines:⁵

⁵imagen extraída de la web de Arduino: <https://store.arduino.cc/arduino-uno-wifi-rev2>



ARDUINO UNO WIFI REV2



| | | | |
|----------|----------------|---------------|--------------------------|
| ■ Ground | ■ Internal Pin | ■ Digital Pin | ■ Microcontroller's Port |
| ■ Power | ■ SWD Pin | ■ Analog Pin | |
| ■ LED | ■ Other Pin | ■ Default | |

ARDUINO . CC

This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0/> or send a letter to Creative Commons, PO Box 1888, Mountain View, CA 94042, USA.

Figura 5: Mapeo de Pines del Arduino UNO Wifi Rev2

4.3 Lector de huellas Adafruit

De acuerdo con la página oficial de Adafruit [6], este lector puede registrar hasta un máximo de 162 huellas en su memoria FLASH. Es compatible con Arduino e incluso tiene una librería dedicada para ponerlo en funcionamiento (utilizada en este proyecto). Su temperatura de trabajo es de -20 grados Celsius hasta +50, con lo que cumple los requisitos para estar operativo dentro de un centro educativo. Entre otras características está el bajo índice de falsos positivos (menor de un 0.001 %) y falsos negativos (<1 %), el voltaje proporcionado tiene que ser entre 3,6 y 6 V DC, compatible con los pines del Arduino. Trabaja con tasas de baudios de 9600, 19200, 28800, 38400, 57600 (esta última por defecto) y tarda en tomar las imágenes de las huellas dactilares menos de 1 segundo.

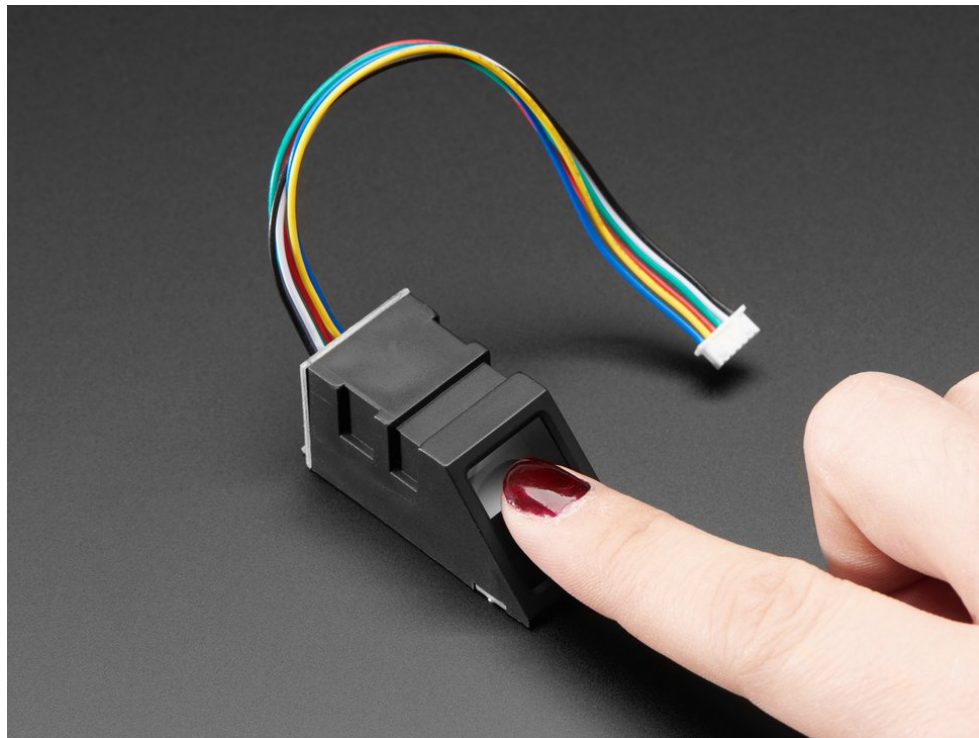


Figura 6: Sensor de huellas Adafruit

4.4 Pantalla Seeed 2.8 inch TFT Touch Shield v2.0

Esta pantalla es una pantalla táctil (aunque no se utilice esta funcionalidad en este proyecto) compatible con los Arduinos Mega, Uno, Zero y Leonardo. Como chip tiene el driver ILI9341. Su voltaje operativo es de 5 voltios. Posee una resolución de 320x240 y dispone de 65 mil colores. Además tiene una ranura para tarjeta SD.



Figura 7: Pantalla Seeed TFT de 2.8 pulgadas

4.5 Batería externa con dos puertos

Esta batería se emplea en este trabajo tanto por la capacidad 6600 mAh, como por la disponibilidad de dos puertos USB. Para la recreación del proyecto, valdría también con dos baterías externas y conectando las dos tomas de tierra (GND) de los dos Arduinos.

La extensa capacidad otorga una ventana de tiempo de carga de los Arduinos bastante importante. Esta característica es muy útil para un proyecto en el que se establecen los dispositivos y tienen que mantenerse muchas horas en activo. Su voltaje operativo es 5V. La batería es de marca poss comprada en el hipermercado Carrefour y la imagen que se ve a continuación es de su web.



Figura 8: Batería externa

4.6 Cable A/B de tipo periférico

Este cable se llama comúnmente de tipo periférico porque es que han utilizado los dispositivos periféricos. [7] La parte USB ha servido para la conexión con el ordenador y la batería externa, la parte B conecta con el Arduino.



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

5 Diseño de la base de datos

Accesocovid es el nombre de la base de datos y se compone por cinco tablas:

- **alumnos:** está tabla contiene los datos personales del alumnado, tiene los siguientes atributos:
 - **idalumno:** la clave primaria de la tabla, por tanto es única y representa de forma unívoca a cada alumno en la base de datos. Es de tipo int, con un máximo de cuatro cifras para albergar a 10000 alumnos (0 a 9999). En caso de saturar, se puede ampliar el número de cifras permitidas o también se puede proceder a eliminar a los alumnos que ya no esten matriculados.
 - **codigo_alumno:** el identificador obtenido por el lector de huellas y es único. Es de tipo int(4), análogamente al idalumno.
 - **nombre:** el nombre del alumno. Es de tipo varchar y permite hasta 20 caracteres.
 - **apellido1:** el primer apellido del alumno. De tipo varchar(20) también.
 - **apellido2:** el segundo apellido del alumno. Igual que el nombre y el primer apellido, es de tipo varchar(20). Permite una entrada nula, a diferencia de los anteriores atributos, para adecuarse a alumnos que puedan no tener segundo apellido.

- **grupo:** refleja el grupo del alumno. Se pondría de la forma número de grupo seguido de letra (por ejemplo, 4A). De tipo varchar(2).
- **color:** es la mitad de la clase, nombrada con un color(en el caso de la facultad, amarillo o azul). Esta distinción se realiza para reducir el número de alumnos en una clase y así cumplir con la normativa de aforo de espacios cerrados por el Coronavirus. Se produce una alternancia entre los dos colores para asistir a un aula. Es de tipo enumerado y contiene los dos colores antes citados (amarillo y azul).
- **asignacion_aulas:** esta tabla asocia que grupo con qué color tiene permitido un día el acceso a un aula determinada. Está compuesta por las siguientes columnas:
 - **numero:** representa el número de la clase. Es una de las claves primarias junto con el día. Es de tipo int(3).
 - **grupo:** es el grupo con la letra que tiene permitido el acceso al aula. Su tipo es varchar(2).
 - **color:** De igual modo que en la tabla alumnos, consiste en un subconjunto del atributo grupo. Es de tipo enumerado y contiene los dos colores antes citados (amarillo y azul).
 - **día:** de tipo date, recoge las diferentes fechas para las que se permite la entrada a los grupos en esas aulas. El formato de la misma es Año-Mes-Día.
- **asistencia:** esta tabla registra la entrada de los alumnos en las aulas. Se compone de:
 - **aula:** número de aula a la que accede el alumno. De tipo int(4).
 - **idalumno:** el identificador del alumno que tiene este en la base de datos, no confundir con el codigo_alumno, que se dedica solo al identificador de huella. De tipo int(3).
 - **día:** el día en el que se produce la asistencia al aula. Es de tipo date, formato Año-Mes-Día como en el caso de asignacion_aulas. Tiene por defecto, current_timestamp(). Es decir, en caso de no introducir fecha, se pondrá la del momento en el que se registre en la base de datos.
- **aulas:** esta tabla tiene la lista de aulas con estos tres atributos:
 - **numero:** identificador del aula. Es la clave primaria. Su tipo es int(3), por tanto tendría de margen hasta 1000 aulas diferentes.
 - **piso:** indica el piso en el que se encuentra el aula. 0 representa la planta principal y a partir de ahí las sucesivas (primera planta ->1, etc.). También admite números de piso negativos para indicar los subterráneos.

- **capacidad:** recoge el número de personas que pueden entrar en una clase. Este número está ya acotado para cumplir con la normativa sobre aforo debido a la epidemia de Covid-19. Este número es modificable, en caso de que las previsiones sean mejores y las restricciones se rebajen, se puede cambiar en la base de datos.
- **cuarentena:** tiene un registro de todos los alumnos que han notificado que han sido contagiados o han estado en contacto con un positivo. Tiene cuatro columnas:
- **idalumno:** es una de las claves primaria de la tabla. Es el identificador único de los alumnos. Su tipo es int(4).
- **contagiado:** indica si el alumno es positivo en la enfermedad o no. Es un tipo tinyint(1) ya que solo se necesitan dos valores para reflejar el mismo. Para la gestión con el Arduino es mejor que sea un tinyint que un boolean.
- **dia_contagio:** registra el día en el que el alumno se contagió o estuvo en contacto con un positivo. Es de tipo date y el formato es Año-Mes-Día. Este valor es la segunda clave primaria de la tabla. Con ello se consigue que se puedan tener registrados varios episodios de contagio o contacto de un mismo alumno. Ya que puede tener varios brotes una misma persona.
- **dia_salida:** es un valor calculado que suma 10 al día del contagio para estimar el día que acabaría su cuarentena. El valor de 10 días de cuarentena es el que están estipulando las autoridades sanitarias a septiembre de 2021. De igual forma que el anterior, es de tipo date y el formato es Año-Mes-Día.

| Tabla | Columnas |
|------------------------------|--|
| accesocovid cuarentena | idalumno : int(4) contagiado : tinyint(1) dia_contagio : date dia_salida : date |
| accesocovid aulas | numero : int(3) piso : int(2) capacidad : int(3) |
| accesocovid asistencia | idalumno : int(4) dia : date aula : int(3) |
| accesocovid alumnos | idalumno : int(4) nombre : varchar(20) apellido1 : varchar(20) apellido2 : varchar(20) codigo_alumno : int(4) grupo : varchar(2) color : enum('amarillo','azul') |
| accesocovid asignacion_aulas | numero : int(3) grupo : varchar(2) color : enum('amarillo','azul') dia : date |

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

6 Implementación

En este capítulo se comentan tanto los programas de desarrollo de software

6.1 Programas y lenguajes

- **XAMPP:** Distribución de Apache que incluye Maria DB,PHP y Pearl.
- **Arduino IDE:** Entorno de programación para el Arduino.
- **PHP:** Lenguaje para el desarrollo web. Versión 8.0.10
- **HTML:** Lenguaje de front para el desarrollo de páginas web.

6.2 Librerías

- **Wire:** Encargada de la transmisión de datos entre el Master-Slave
- **Adafruit Fingerprint Sensor Library:** Encargada de gestionar el lector de huellas.
- **TFT TouchShield V2, Touch Screen Driver:** Contiene el driver de la pantalla.
- **WiFiNINA y WifiClient:** Encargadas de la conexión a la red WiFi.

6.3 Montaje

Para proceder al montaje del conjunto, se colocan los dos Arduinos ligeramente elevados con respecto a la madera (esto evita el recalentamiento, ya que proporciona un espacio de liberación del calor) y el lector de huellas. En una versión final se podría imprimir en una impresora 3D una carcasa que solo dejase ver el lector y la pantalla, pero que fuese desmontable para acceder a los componentes en caso de avería o actualización. La conexión de cables tiene que ser tal y cómo aparece en la imagen. Los dos cables que no se aprecia su inserción (debajo de la pantalla) se situarían el blanco en el pin SCL y el negro en el SDA del Arduino Wifi Rev2. Los cables periféricos irían a los puertos de la batería externa una vez que se haya cargado el sketch en los mismos.

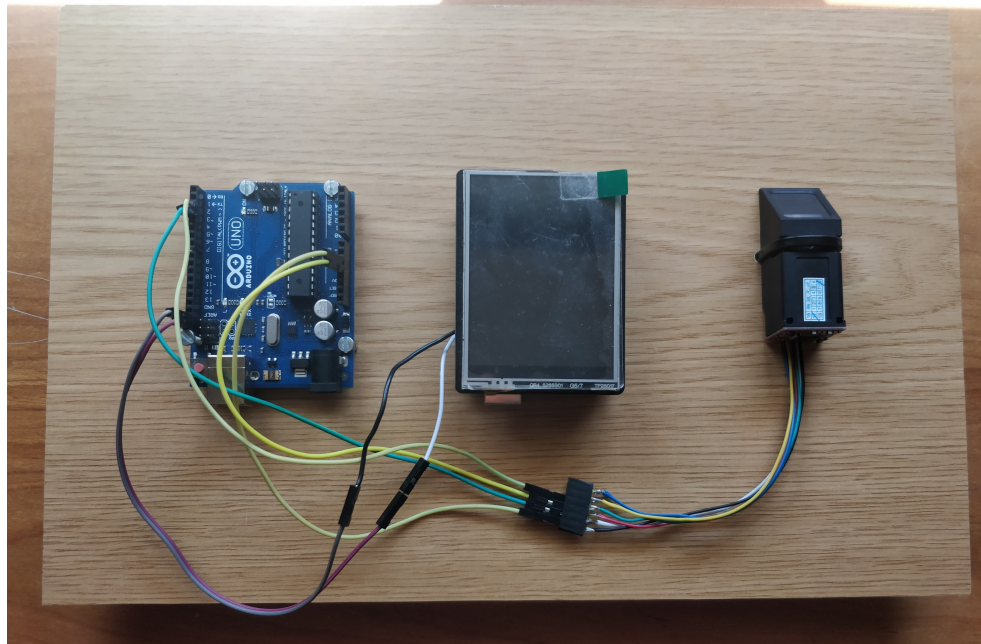


Figura 11: Montaje y cableado

6.4 Manual de uso

Para poder replicar este proyecto y ponerlo en marcha, se deben seguir una serie de pasos.

6.4.1 Descarga de los programas y el repositorio

- **XAMPP:** Se requiere el programa XAMPP para montar la base de datos, ya que este programa es un gestor de base de datos. Por tanto se recomienda descargarlo de su página oficial [8] y proceder a la instalación del mismo con las opciones por defecto. Además es crucial descargar la misma versión explicada anteriormente. En el capítulo *Impedimentos* se explican algunos problemas de incompatibilidad por el uso de una versión diferente (ya que en la instalación viene incluida también la versión de PHP).
- **Arduino IDE:** Para compilar el código proporcionado de Arduino, necesitamos el entorno de programación de Arduino IDE. En la siguiente página web se puede descargar sin problemas [9]. En este caso, se han probado versiones diferentes y no hay riesgo por el momento. Puede resultar que en unos años alguna librería se vuelva obsoleta y haya que cambiarla por su versión actualizada. Aunque no se usa en este proyecto, la versión anterior de la librería TFT ya está marcada como retirada por Arduino.[10]
- **Repositorio:** Por último, se descarga “Acceso_Aulas_Covid” del directorio proporcionado. En este caso, ha sido una carpeta en Google Drive proporcionada por la facultad

de informática llamada 222 - Sistema de control de accesos para cumplir restricciones por COVID-19

6.4.2 Adición de las librerías

En el entorno de programación del Arduino IDE, en la pestaña Programas se clic en añadir Librería -> Librería ZIP. Con esta función, procedemos a incorporar todas las librerías citadas. Se proporcionan en una carpeta llamada *Librerias* para mayor facilidad.

6.4.3 Base de datos

Para tener disponible la base de datos, hay que ejecutar el XAMPP. Buscando en la barra de Windows se teclea XAMPP y se clic en el primer enlace. Se abrirá una ventana como la figura 12

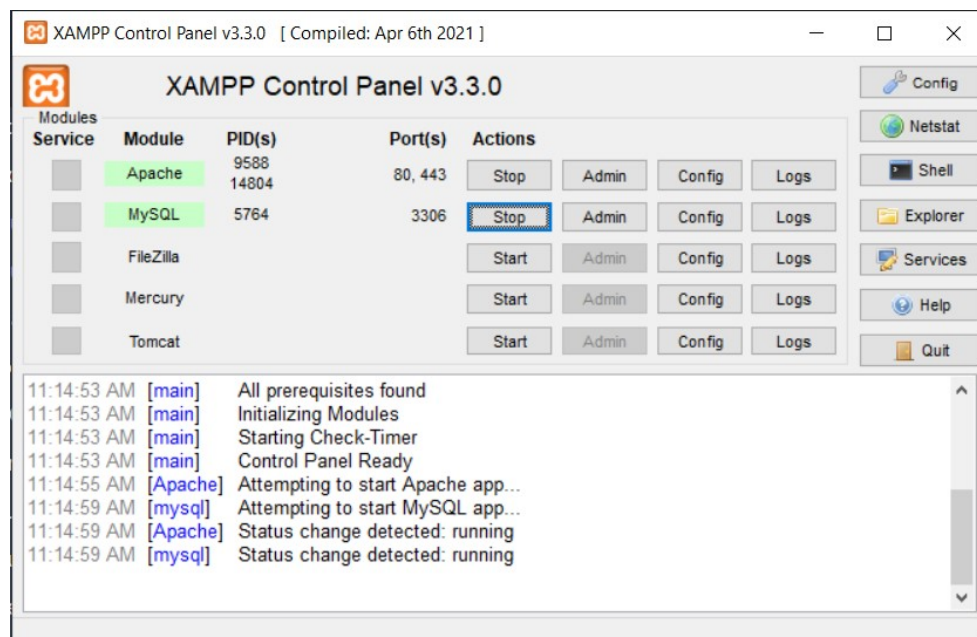


Figura 12: Iniciar Apache y MySQL

Se accionan los botones Start correspondientes a Apache y MySQL en ese orden. Cuando salgan en verde como en la imagen, se abre un explorador y se teclea `localhost/phpmyadmin`. A continuación, se presiona “Importar” o “Import” si se tiene en inglés. Se sitúa en la barra superior. En el botón “Browse” o “Seleccionar archivo” y se importa el archivo `accesocovid.sql`.

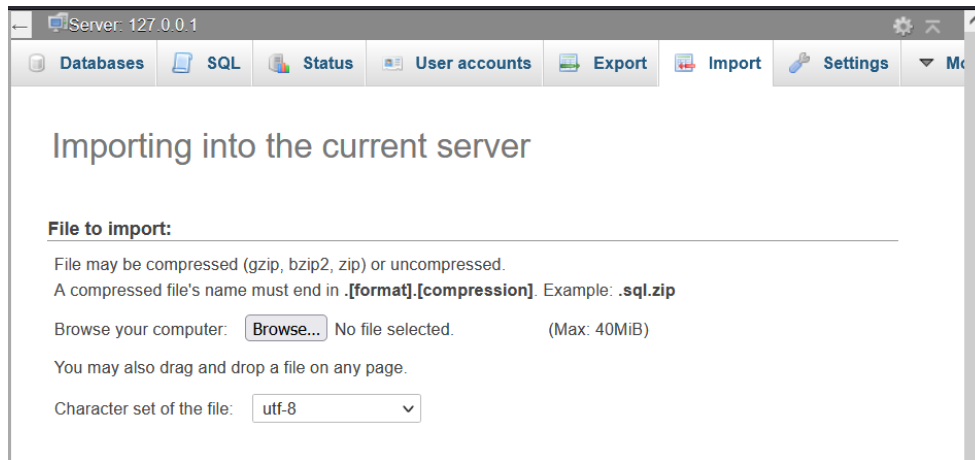


Figura 13: Importar Base de Datos en PHPMyAdmin

Se debe dar a continuar para que produzca el efecto deseado. Posteriormente, se puede ver listada esta base de datos en el panel izquierdo. Para desplegar y visionar las diferentes tablas, basta con pulsar el icono + que se encuentra a la izquierda del nombre de la base de datos (*acesocovid*).

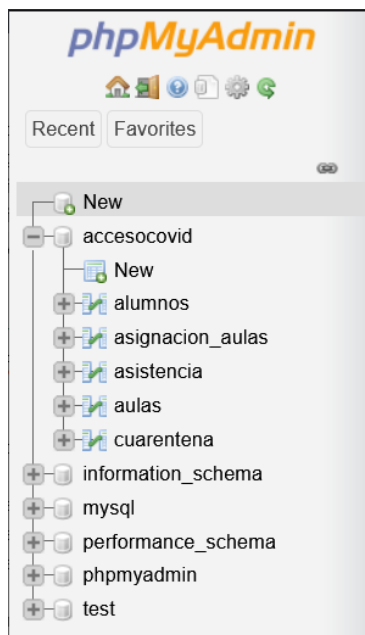


Figura 14: Comprobación de Base de Datos importada

6.4.4 Colocación de las carpetas que contienen el código HTML y PHP

Con el objetivo de poder usar los formularios, hay que desplazar las carpetas *formularios* y *bd* a la ruta del ordenador `C:/xampp/htdocs`. Una vez localizado en esta ubicación, ya

estarían operativos tanto ambos formularios y la función de conexión de la base de datos del Arduino Wifi Rev2. Ya que necesita que la carpeta bd tenga esa URL.

Tras estos pasos, estarían listos para usarse todos los componentes de la aplicación.

Se recomienda que se apague el firewall del ordenador dónde esté ubicada la base de datos mientras se haga uso de la misma a través del Arduino o, en su defecto, que se abra el puerto necesario para la conexión del mismo (80). Ya que esta situación impide el buen funcionamiento del programa.

6.5 Impedimentos

En el desarrollo del proyecto, ha habido algunos impedimentos que han retrasado la ejecución del mismo. En los siguientes puntos se explican estas dificultades y cuál es la solución que se ha encontrado.

6.5.1 Cambio de versión del PHP e incompatibilidad de funciones

Al recrear el proyecto anterior, la primera dificultad encontrada fue que al volver a instalar los programas y ejecutar los códigos, había funciones que estaban obsoletas y que no se recomendaba ya su uso. El ejemplo más evidente fue que la conexión en php a la base de datos antes se hacía con `mysqli_connect` y con la nueva versión del PHP (8.0.10) se marcaba como obsoleto [12]. Se investigó una forma de actualizar el código y la opción fue por PDO.⁶ Así que con argumentos [13] a favor de la migración, se procedió a la misma.

6.5.2 Malfuncionamiento del XAMPP en mi ordenador

Tras instalar en varias ocasiones el XAMPP y desinstalarlo por problemas con la interpretación del PHP por parte de los exploradores. Procedí a modificar la ruta en las Variables del sistema, modificar el config de Apache. Al final, la solución que opté, siguiendo la recomendación de la documentación [11] fue la de cambiar la ruta en la URL, por ejemplo *C:/xampp/htdocs/cuarentena/index.html* por *localhost/cuarentena/index.html*.

6.5.3 Interpretación de tildes y otros caracteres especiales por el HTML y el envío de los mismos

El envío a través del formulario no codificaba bien tanto los caracteres de las ñ como las tildes. La pantalla de TFT tiene el mismo problema. Con lo cuál, tanto los formularios como los nombres de los alumnos al mandarlos a través de los mismos tendrán que omitir estos caracteres.

⁶PHP Data Objects

7 Ejemplo de uso

Se identifican cinco casos de uso: Registro del alumnado en el sistema, Acceso al aula, Asignación de aulas, Notificación de Contagio y Verificación de contagio. En las siguientes subsecciones se explicará cada uno de ellos en profundidad:

7.1 Registro de alumno en el sistema

Este capítulo se desglosa en dos epígrafes: Registro de huella en el Arduino y Registro del alumno en el formulario. El orden de la ejecución por parte del administrador del sistema tiene que ser el siguiente, ya que se necesita el código de alumno (ID) obtenido en el Arduino para introducirlo en la base de datos. Sin el mismo, no se puede realizar la inserción del alumno, ya que los códigos de alumno (huella) tienen que estar asociados a un alumno y viceversa con una relación 1 a 1. En este sentido, se puede pactar que todos los estudiantes utilicen el dedo índice de la mano derecha (por ejemplo) en el lector con el objetivo de no tener varias huellas en el lector de una misma persona o que los alumnos tengan dificultades al entrar en el aula asignada.

7.1.1 Registro de huella en el Arduino

Esta funcionalidad se haría uso de ella en los primeros días del cuatrimestre, con el fin de registrar de forma escalonada a todos los alumnos. En este proceso, se registrará su huella y se volcarán los datos del alumno a la base de datos mediante un formulario alojado en `localhost/registro/index.html`.

El proceso de registro de la huella en el Arduino y obtención del código del alumno es el mismo que el TFG anterior. Al principio, se exige colocar el dedo en el lector mediante el mensaje “Coloca el dedo” (Figura 15). A continuación se procede a retirar el dedo, esto también lo indica la pantalla (Figura 16). Para grabar correctamente la huella se pide volverlo a colocar (Figura 17). Para finalizar, la pantalla muestra un mensaje de “Huella guardada” (Figura 18) y ese ID mostrado es el que se precisa para el formulario siguiente. Si hay algún error porque no se coloca el mismo dedo, se proyectará en la pantalla un mensaje de no coincidencia (Figura 19). El procedimiento en este caso sería volver a poner el segundo dedo, ya que la memoria del Arduino se queda con la última huella. En la situación que se quisiera poner el primer dedo, habría que volver a realizar el proceso completo.⁷

⁷Las siguientes imágenes pertenecen al TFG Sistema de control de acceso con huella dactilar y Arduino

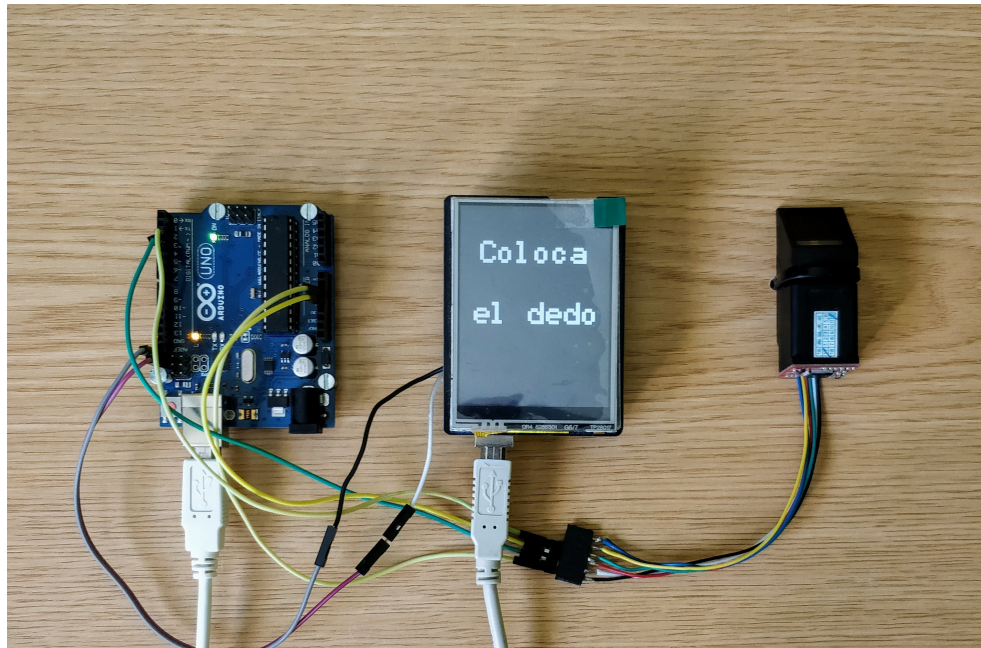


Figura 15: Coloca el dedo



Figura 16: Retira el dedo

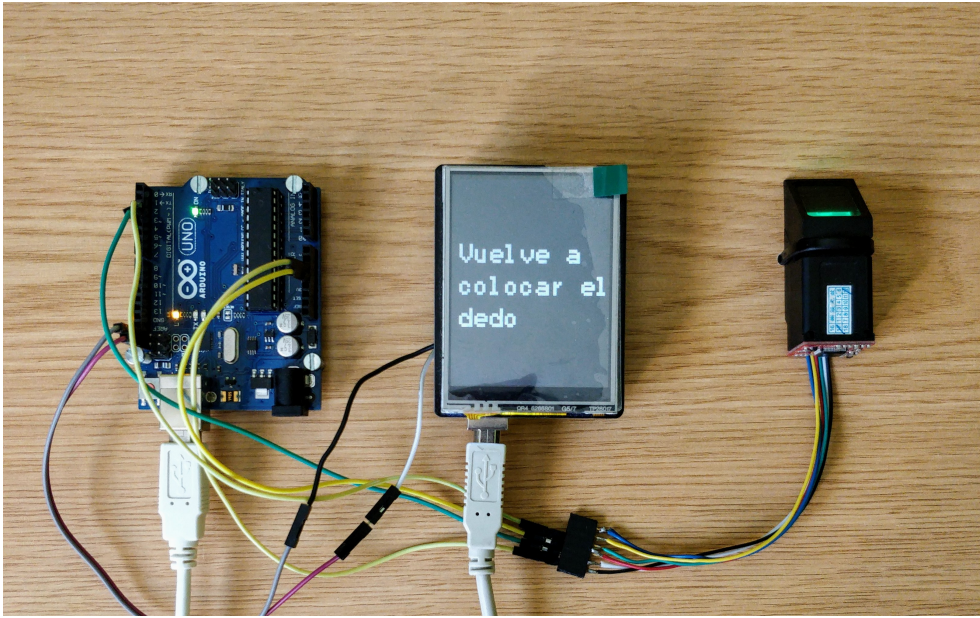


Figura 17: Vuelve a colocar el dedo

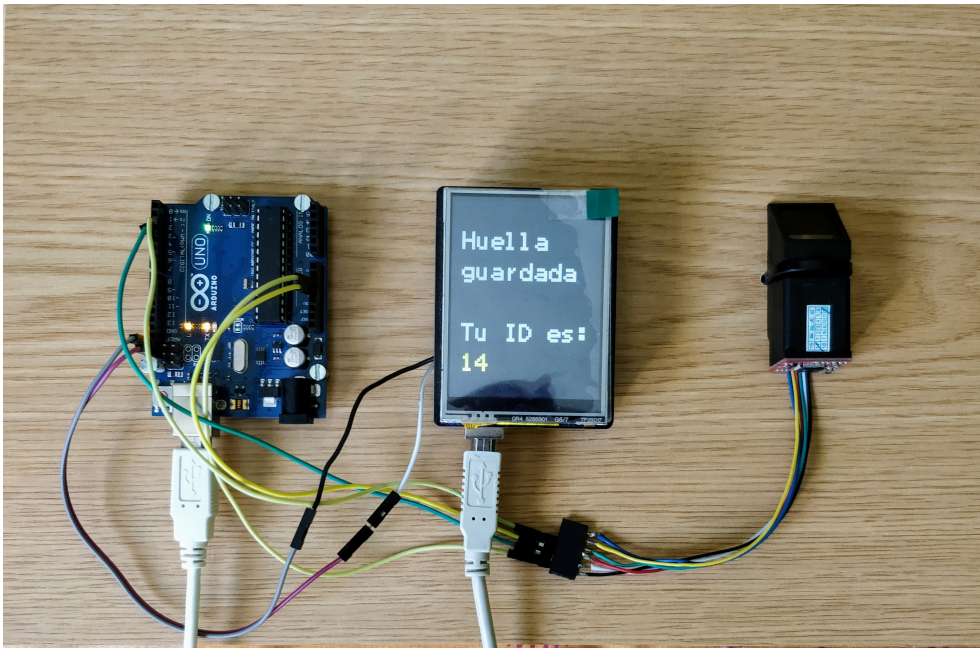


Figura 18: Huella guardada

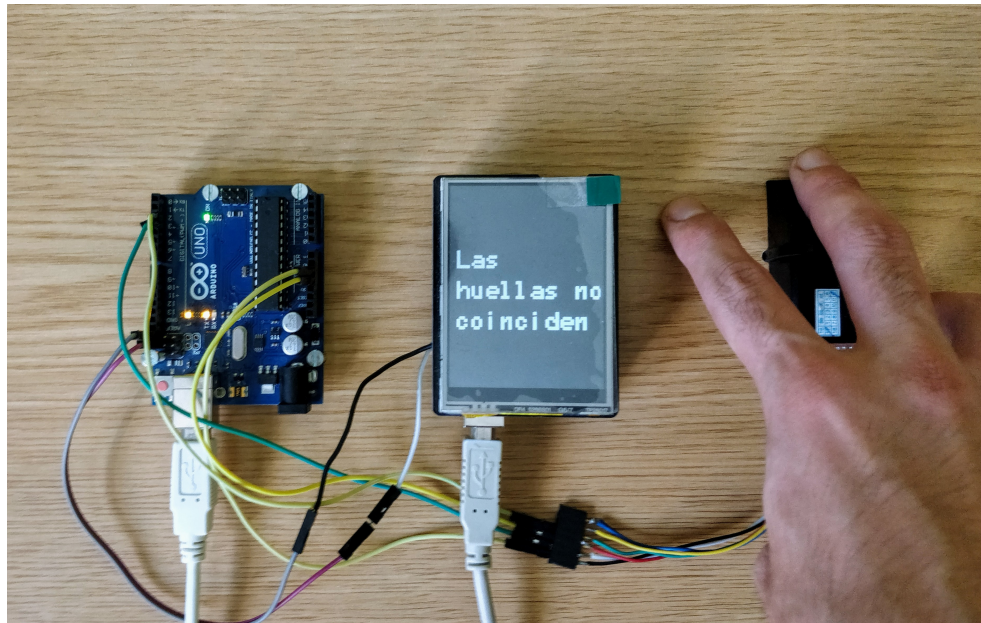


Figura 19: Las huellas no coinciden

7.1.2 Registro del alumno en el formulario

En este segundo apartado, se procede a registrar al alumno en la base de datos.

Mientras está el XAMPP en ejecución con el Apache y el MySQL iniciados, se escribe la URL *localhost/formularios/inicio.html* o se va a la carpeta formularios (descrita anteriormente, que se ha movido a la carpeta *htdocs* en el directorio dónde está ubicado el XAMPP) y se clicla el icono *inicio.html*.



Figura 20: Pantalla de inicio de los formularios

En los siguientes casos de uso, a excepción del Acceso al aula, se pasará por esta pantalla de inicio.

En este caso, se pulsa en el botón de “Registro alumnos” para acceder al formulario correspondiente. En la siguiente figura, se puede observar el resultado de esa ejecución. El usuario, se encuentra una pantalla de instrucciones para verificar que ha realizado el paso anterior, crucial para el registro como se ha expuesto anteriormente.

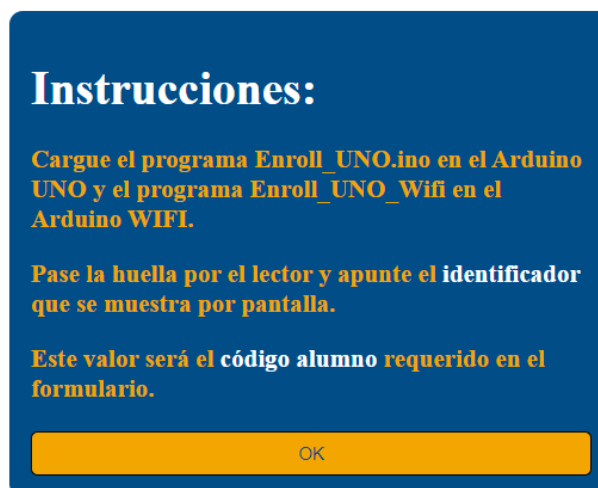


Figura 21: Instrucciones para el registro del alumnado

Una vez que se hayan completado las instrucciones, se da a “OK” y ya aparecerá el formulario que se muestra en la Figura 22. El administrador del sistema procederá a rellenar

los datos de cada alumno y escribirá en el campo Código de alumno, el ID obtenido del Arduino en el capítulo anterior.

Cuando esté completo, se da a Submit y si está correctos todos los datos, se mostrará “Alumno registrado con éxito”. En el caso de que el alumno ya esté registrado, saldrá “Este alumno ya está presente en la Base de Datos”.



Registro de alumnos

Nombre

Primer Apellido

Segundo Apellido

Código del alumno

Número de grupo y letra

Color Amarillo Azul

Figura 22: Formulario de registro de los alumnos en la base de datos

Se puede comprobar que el alumno ha sido incorporado a la base de datos, accediendo a la misma con el XAMPP tecleando localhost/phpmyadmin/ y navegando hasta la base de datos *acesocovid* y de ahí a la tabla *alumnos* y en la pestaña de Browse. El alumno que se ingrese en ese momento aparecerá el último en esta tabla.

| | idalumno | nombre | apellido1 | apellido2 | codigo_alumno | grupo | color |
|---|----------|---------|-----------|-----------|---------------|-------|----------|
| <input type="checkbox"/> Edit Copy Delete | 1 | Sandra | Sanchez | Jaime | 5 | 3C | amarillo |
| <input type="checkbox"/> Edit Copy Delete | 2 | Alberto | Gines | Lopez | 18 | 1F | amarillo |
| <input type="checkbox"/> Edit Copy Delete | 4 | Julia | Miguel | Fernandez | 1 | 4A | azul |
| <input type="checkbox"/> Edit Copy Delete | 11 | Ignacio | Hidalgo | Perez | 3 | 4A | amarillo |

Figura 23: Nuevo alumno registrado en la tabla Alumnos

7.2 Acceso al aula

Como principal funcionalidad, está el Acceso al aula. El procedimiento es el siguiente:

Cada conjunto de los dos Arduinos y el lector estaría cada uno en una clase diferente.

Por lo tanto se deja establecido en el Arduino el número de Aula. En el proyecto se asigna al aula 1.

Al principio, se muestra un mensaje de Bienvenida, recordando que está en el Aula 1.



Figura 24: Mensaje de bienvenida

El alumno se acerca al lector de huellas y pone el dedo en el mismo cuando se pida que ponga la huella (Figura 25). En el caso que no esté registrado se muestra el mensaje “Regístrese con el Administrador” (Figura 26)



Figura 25: Petición de huella

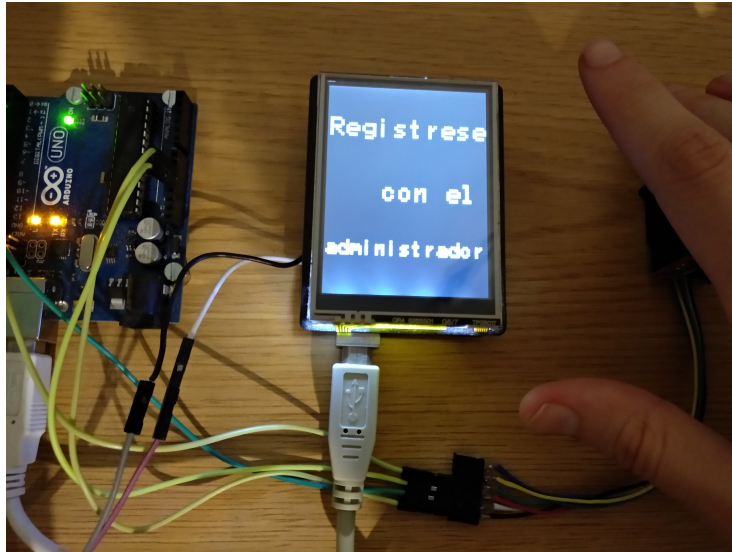


Figura 26: Alumno no registrado

Antes de consultar la huella del alumno, el Arduino Wifi consulta en la base de datos el grupo y el color que le corresponden ese día a esa clase, así como la capacidad del aula para el control del aforo. Como se puede apreciar en la siguiente imagen, esta información es mostrada en la pantalla para informar al alumno.

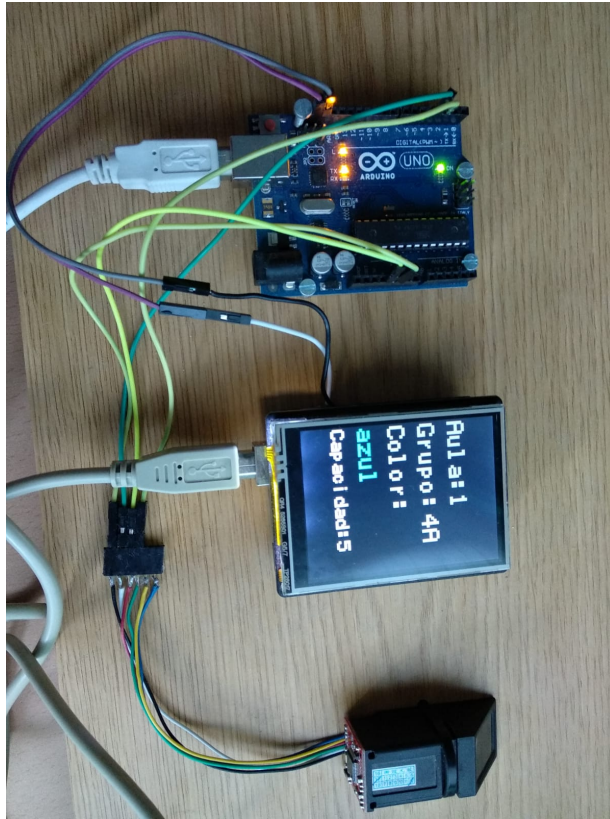


Figura 27: Datos referentes a la asignación de grupo a esa aula

El Arduino consulta con la base de datos los datos del alumno. Si el grupo no corresponde al aula 1, se muestra el mensaje “Esta clase no le corresponde. Diríjase a su aula”.



Figura 28: Diríjase a la clase que le corresponde

Después, si el grupo es el que tiene permitido el acceso ese día, verifica el color. Si no es el adecuado, la pantalla pondrá “Hoy no le corresponde clase presencial”.



Figura 29: Clase habilitada para el otro color

Pasado este filtro, se revisa si el estudiante está registrado como contagiado o que ha tenido contacto cercano con un positivo. Si es así y el período de cuarentena no ha finalizado aún se bloquea el acceso.

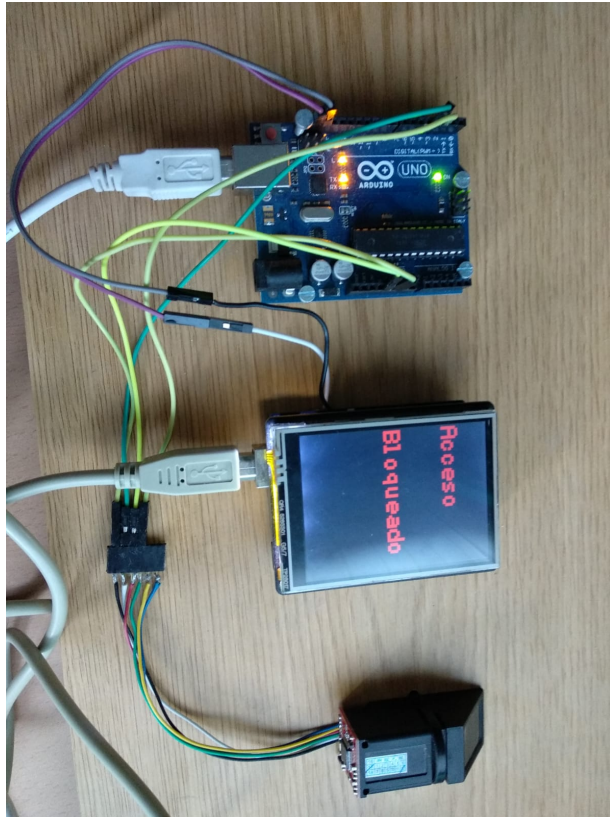


Figura 30: Acceso bloqueado

En caso contrario, se comprueba si el aforo esté completo o haya disponibilidad. En el primer caso, se muestra “Aforo completo”.

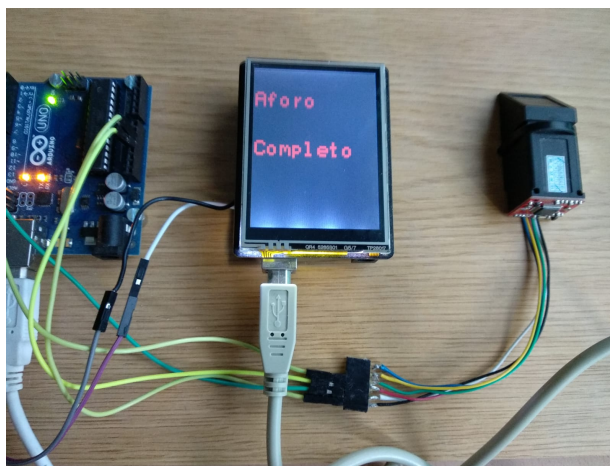


Figura 31: Aforo completo

Si ha pasado todos los filtros, se permite su paso y se registra su asistencia.

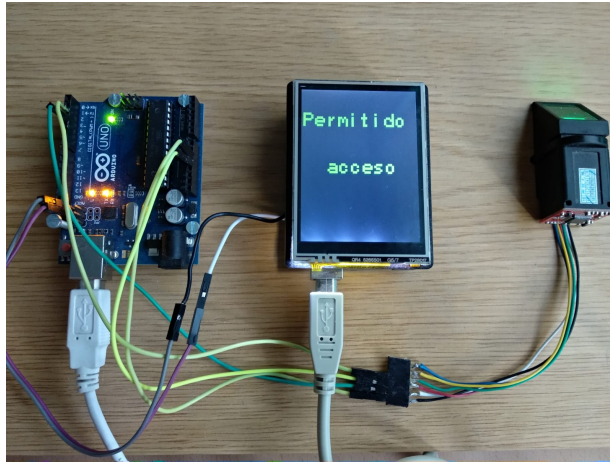


Figura 32: Acceso permitido

7.3 Asignación de grupos a aulas

Este caso de uso es necesario para hacer una planificación de las aulas y los grupos.

Para empezar, se clica en *inicio.html* , como se ha explicado anteriormente. Se ve la pantalla de inicio (Figura 20). Se accede a la “Asignacion de grupos a aulas”. Se rellena el formulario con los datos correspondientes y se da a “Añadir”. Si todo es correcto, se muestra “Registro completado”. Si el aula que se intenta registrar no existe en la tabla de aulas pondrá “Número de aula no registrada en la Base de Datos”.

Figura 33: Formulario asignación aulas

7.4 Notificación de contagio o contacto con un positivo

En esta funcionalidad, se encuentra la notificación de los contagios y contactos con positivos. Este caso de uso se vió como punto muy importante en el planteamiento del proyecto. Se necesitaba un mecanismo para registrar las personas que estaban en cuarentena para evitar su acceso a las aulas y revocarle el acceso cuando acabara la misma sin tener que volver a avisar. Con esta finalidad, se creó un formulario para que los alumnos o el administrador pudieran alertar de la situación y que se almacenara en la base de datos.

De forma similar al registro del alumno en el sistema, se presenta el siguiente formulario:



El formulario, con un fondo azul oscuro, tiene el título "Registro de contacto/contagio" en blanco. Incluye los siguientes campos:

- Código del alumno:** un campo de texto blanco.
- Nombre:** un campo de texto blanco.
- Primer Apellido:** un campo de texto blanco.
- Segundo Apellido:** un campo de texto blanco.
- Contagiado:** dos botones de radio, "Sí" y "No", con un punto negro a la izquierda de cada uno.
- Día del contagio/contacto:** un campo de texto con el formato "mm/dd/yyyy" y un icono de calendario a la derecha.

En la parte inferior hay un botón naranja con el texto "AÑADIR" en blanco.

Figura 34: Formulario de registro de contagio o contacto de un alumno

El acceso al formulario también se hace a partir de la pantalla de inicio.

Como se puede observar, el único obligatorio es el primer apellido, aunque se recomienda poner los dos en caso de que los haya.

Se rellenan los campos indicando si el estudiante está contagiado o no y en el campo de *Día de contagio* se desplegará un calendario ⁸ como el de la Figura 35. En el mismo podremos

⁸El calendario y la fecha que muestra está en inglés porque refleja el idioma en el que tienes tu sistema operativo. En un ordenador con idioma español, la fecha saldrá en formato español y tanto los nombres de los meses como de los días estarán en este idioma.

seleccionar el día que se produjo el contacto o cuando se obtuvo el positivo en la PCR⁹. Al acabar de completar el formulario, se da en “Añadir”. Si todo es correcto, se mostrará “Registro completado”. Si no está registrado ese alumno, aparecerá el mensaje “Alumno no encontrado”. Por último, si no concuerda el identificador con el alumno, pondrá “El identificador no concuerda con los datos del alumno introducidos.”

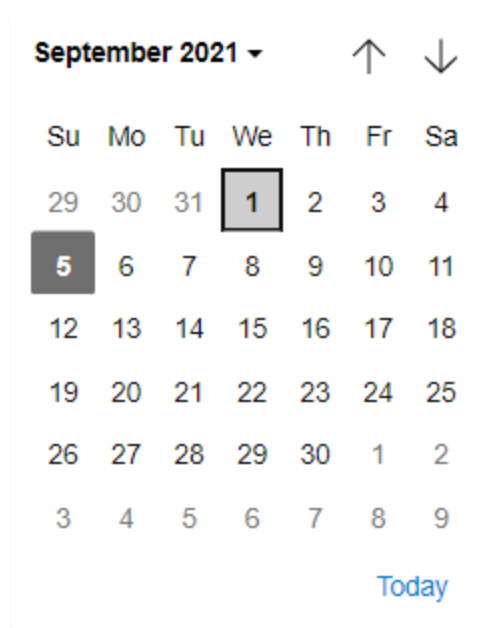


Figura 35: Calendario del formulario de contagio

Se puede comprobar su registro en la tabla **cuarentena** (siguiendo los pasos mencionados en el *Registro del alumno en el formulario*). Se ve reflejado el nuevo registro que ha sido añadido.

| | idalumno | contagiado | dia_contagio | dia_salida |
|--|----------|------------|--------------|------------|
| <input type="checkbox"/> ✎ Edit 📄 Copy 🗑 Delete | 4 | 1 | 2021-09-01 | 2021-09-11 |

Figura 36: Registro del alumno en la tabla cuarentena

7.5 Verificación de contagio

Como última funcionalidad, está la verificación de contagio. Cuando un alumno no ha podido acceder a un aula, puede tener dudas si tendría que seguir en cuarentena por un contagio o un contacto estrecho con un infectado por Covid-19. Por tanto, se puede rellenar el siguiente formulario para ver si está registrado y hasta cuándo tendría que estar en cuarentena. Para averiguarlo, se rellenan los campos del formulario y hay tres posibles

⁹Prueba que permite detectar el ARN del virus (Covid-19) y ver si está latente en el cuerpo.

respuestas. La primera sería “Alumno no registrado en la lista de contagios” por no tener ningún registro de ese alumno en la tabla de cuarentena. El segundo mensaje pondría “Registrado contacto estrecho del alumno con un positivo” y se añadiría “Se recomienda cuarentena hasta el día ”. Este día sería lo equivalente a 10 días desde el registro del contacto. De forma equivalente, el mensaje del contagio pondría “Registrado contagio por Covid-19. Se recomienda cuarentena hasta el día”. Con estos mensajes quedaría claro para el alumno hasta cuándo tiene que guardar cuarentena y no asistir al centro porque su acceso será bloqueado.



Verificar el contacto/contagio de un alumno

Código del alumno

Nombre

Primer Apellido

Segundo Apellido

Verificar

Figura 37: Verificación del contagio o contacto estrecho con un positivo

8 Resultados

8.1 Memoria de los Arduinos

El programa del Arduino Wifi Rev2 usa 24606 bytes, lo que supondría la mitad del espacio dedicado a la memoria. El máximo posible para un programa son 48640 bytes. Las variables globales usan 778 bytes (12 %) de la memoria dinámica. Se dispondrían de 5366 bytes para variables locales. Con respecto a ID_Uno.ino, el programa cargado en el Arduino UNO, se usan 7034 bytes (21 %) de la memoria. El Arduino UNO tendría menor capacidad y sólo permitiría un máximo de 32256 bytes. En este caso, la memoria dinámica dedicada a las variables globales es un 35 % (733 bytes) y a las locales se quedaría en 1315 bytes.

A pesar de la longitud de la longitud del sketch de Aula_Uno_Wifi.ino que consiste en un total de 586 líneas de código, se puede apreciar que todavía quedaría la mitad de la memo-

ria. Con estas estimaciones, se podría contar con un máximo de 1000 líneas de código para el Arduino Wifi Rev 2. Con esta extensión se podrían añadir todavía más funcionalidades al código sin llegar a sobrecargarlo.

9 Conclusión

La elección de la huella dactilar como método de acceso puede ser un poco comprometido en estos tiempos de distancia social y no compartición de objetos. Sin embargo, dotando de gel hidroalcohólico y limpiando el lector tras cada uso, da un resultado mucho más fiable, ya que es un identificador único de cada persona y no es transferible. En el caso estudiado en este proyecto es de vital importancia saber de primera mano quién ha estado realmente en la sala.

Las herramientas otorgadas en este proyecto suponen una base para la gestión de accesos de un centro educativo o, incluso una empresa donde se tengan usuarios registrados que comparten un espacio durante un período extenso de tiempo. Además la principal ventaja es que es un producto que no se necesita un conocimiento de programación para ser usado, con el que el público objetivo puede ser mayor. Se han expuesto momentos en los que se puede consultar la base de datos para verificar, pero este comportamiento no es necesario para la utilización del sistema. Si se quiere hacer una actualización del sistema para modificar la capacidad de las aulas o, para agregar nuevas, se necesitaría un administrador que sepa controlar el XAMPP pero la interfaz de este programa es más amigable.

Por tanto, considero que se han cumplido los objetivos expuestos.

Para finalizar, he descubierto que el Trabajo de Fin de Grado realizado por mi compañero Adrián Navazo y yo tenía mucha potencialidad y se podía orientar y ampliar hacia otros casos de uso. Personalmente he desempolvado y extendido mis conocimientos sobre Arduino, HTML, PHP,... Además creo que es muy gratificante poder aplicar el conocimiento obtenido en la Facultad de Informática para un bien que pueda ayudar a la facultad o a otros centros similares.

Estamos viviendo una situación convulsa, inesperada e impactante pero poco a poco podremos salir de ella con más recursos y más aprendizaje.

10 Conclusion

The choice of the fingerprint as an access method can be a bit compromising in these times of social distance and no sharing of objects. However, adding hydroalcoholic gel and cleaning the reader after each use, gives a much more reliable result, since it is a unique identifier for each person and is not transferable. In the case studied in this project, it is vitally important to know first-hand who has actually been in the room.

The tools provided in this project are a basis for managing access to an educational center or even a company where there are registered users who share a space for an extended period of time. In addition, the main advantage is that it is a product that does not need programming knowledge to be used, with which the target audience can be greater. There have been times when the database can be queried for verification, but this behavior is not required for system utilization. If you want to make a system update to modify the capacity of the classrooms or to add new ones, you would need an administrator who knows how to use XAMPP, but the interface of this program is more user-friendly. Therefore, I consider that the stated objectives have been met.

To finish, I have discovered that the Final Degree Project carried out by my colleague Adrián Navazo and I had a lot of potential and could be oriented and expanded towards other use cases. Personally I have dust off and extended my knowledge about Arduino, HTML, PHP, ... I also think that it is very gratifying to be able to apply the knowledge obtained in the Faculty of Informatics for a good that can help the faculty or other similar centers.

We are experiencing a convulsive, unexpected and shocking situation but little by little we will be able to get out of it with more resources and more learning.

Referencias

- [1] <https://www.mineco.gob.es/portal/site/mineco/menuitem.ac30f9268750bd56a0b0240e026041a0/?vgnextoid=359f50a4c34b3710VgnVCM1000001d04140aRCRD&vgnnextchannel=864e154527515310VgnVCM1000001d04140aRCRD>
- [2] <https://eprints.ucm.es/id/eprint/65221/>
- [3] <https://www.arduino.cc/en/tutorial/sketch>
- [4] <https://www.ucm.es/data/cont/docs/titulaciones/1909.pdf>
- [5] <https://store.arduino.cc/products/arduino-uno-wifi-rev2>
- [6] <https://www.adafruit.com/product/751>
- [7] <https://hardzone.es/2018/06/30/conectores-usb-tipos/>
- [8] <https://www.apachefriends.org/index.html>
- [9] <https://www.arduino.cc/en/software>
- [10] <https://www.arduinolibraries.info/types/retired>
- [11] https://www.apachefriends.org/es/faq_windows.html Pregunta: ¿Dónde debo colocar mi contenido web?
- [12] <https://www.php.net/manual/en/ref.mysqli.php>
- [13] <https://stackoverflow.com/questions/18326753/pdo-replacing-mysqli>

Bibliografía

- <https://php-legacy-docs.zend.com/manual/php5/en/ref.mysql>
- <https://www.php.net/manual/es/mysql.quickstart.connections.php>
- <https://github.com/Anaya-Multimedia/curso-de-php-8-y-mysql-8>
- <https://dev.mysql.com/doc/apis-php/en/>
- <https://store.arduino.cc/arduino-uno-wifi>
- <https://store.arduino.cc/arduino-uno-wifi-rev2>
- https://github.com/Tariqu/database_connection_php
- https://www.youtube.com/watch?v=2HVKizgcfjo&list=PLtYTVIzWW8XE_f0yoaLwDNQOMY3JNfPrj
- <https://www.youtube.com/watch?v=fsQoA7NMhsU>
- <https://medium.com/@chongfai13/fixed-www-example-com-443-0-418087630e94>
- <https://www.php.net/manual/en/pdostatement.execute.php>
- <https://coursesweb.net/php-mysql/pdo-prepare-execute>
- <https://www.php.net/manual/es/pdo.prepare.php>
- https://www.w3schools.com/php/func_date_date_add.asp
- <https://arduino.stackexchange.com/questions/1013/how-do-i-split-an-incoming-string>
- <https://www.webdeveloper.com/d/396203-how-to-replace-mysql-code-with-pdo-code/>
3
- <https://www.rinconingenieril.es/arduino-ethernet-y-mysql/>
- <https://forum.arduino.cc/t/arduino-base-de-datos/471060>
- <https://espaciotecnologico.co/sensor-de-temperatura-data-logger-arduino/>
- <https://stackoverflow.com/questions/16562577/how-can-i-make-a-button-redirect-my-page-to-anot>
- <http://www.chinanordson.com/standalone-fingerprint-access-control/fingerprint-access-control.html>
- <https://es.stackoverflow.com/questions/436293/mysql-connect-esta-obsoluto-en-php-8-para-conectarse-a-la-base-de-datos>

Imágenes

- <https://store.arduino.cc/products/arduino-uno-rev3>
- <https://store.arduino.cc/arduino-uno-wifi-rev2>
- <https://adafruit.com/product/751>
- https://wiki.seeedstudio.com/2.8inch_TFT_Touch_Shield_v2.0/
- <https://www.carrefour.es/bateria-externa-poss-ppbli10400-negro/VC4A-3133845/p>
- <https://store.arduino.cc/usb-2-0-cable-type-a-b>