



SMX Sistemas Microinformáticos y Redes

Daniel Cobo
Zakariae Boukraa
2SMXD
2025-2026

01. Introducció.....	3
02. Contexto.....	3
03. Justificació.....	4
04. Objectivos.....	4
4.1 Objectivo general.....	4
4.2 Objectivos específics.....	5
5. Estrategia y planificación del proyecto.....	5
5.1 Metodología de trabajo.....	6
6. Estudio económico y presupuestario.....	6
6.2 Estudio económico Real de nuestro Proyecto.....	7
7. Descripción del proyecto.....	9
7.1 Análisis de requisitos.....	10
7.1.1 Requisitos funcionales.....	10
7.1.2 Requisitos no funcionales.....	11
8. Comparativa de les tecnologies valoradas.....	11
9. Arduino o Raspberry Pi.....	14
10. Explicación de cada componente.....	16
10.1 Teclado numérico de membrana.....	17
10.2 Motor servo ES08A.....	18
10.3 IIC Oled Module.....	18
10.4 Placa Arduino UNO.....	19
10.5 Buzzer (zumbador).....	20
10.6 LEDs.....	20
10.7 Cables, resistencias y protoboard.....	21
11. Creación de nuestro proyecto.....	22
11.1 Montaje de la Caja.....	22
11.2 Programación.....	25
11.3 Creación de aplicación.....	29
12. Conclusiones.....	40
12.1 Conclusiones generales del proyecto.....	40
12.2 Consecución de los objetivos.....	41
13. FINAL.....	41

01. Introducció

El nombre elegido para el proyecto es Keepy, el cual proviene del verbo en inglés “keep”, que significa guardar o mantener seguro. Este nombre ha sido seleccionado porque representa de manera directa la función principal del sistema: proteger y almacenar objetos de forma segura dentro de la caja fuerte.

Además, el uso de un nombre corto, moderno y fácil de recordar aporta un toque más atractivo al proyecto, especialmente de cara a su presentación

El presente proyecto tiene como finalidad el diseño y desarrollo de una caja fuerte electrónica interactiva, controlada mediante un sistema de autenticación por código numérico implementado con una placa Arduino UNO. Este sistema permitirá restringir el acceso a su interior, proporcionando seguridad a objetos personales de pequeño tamaño. La caja fuerte incorporará diversos componentes electrónicos como un teclado matricial 4x4 para la introducción del código, una pantalla LCD 16x2 para la interacción con el usuario, un servomotor SG90 encargado del mecanismo de apertura y cierre, así como elementos de señalización como LEDs y un buzzer para indicar diferentes estados del sistema (correcto, error, alerta, etc.).

Además, el proyecto no se limita únicamente al hardware, sino que también incluye el desarrollo de una aplicación móvil básica mediante App Inventor, que permitirá interactuar con la caja fuerte de forma remota o complementaria, añadiendo una capa extra de funcionalidad y modernización al sistema.

El desarrollo del proyecto combina conocimientos adquiridos durante el ciclo formativo, tales como programación, electrónica digital, diseño de sistemas interactivos y resolución de problemas, ofreciendo una solución práctica y aplicable a situaciones reales.

02. Contexto

Nosotros se nos planteó hacer un proyecto sobre todo lo que hemos aprendido en estos dos años de ciclo medio, nosotros creemos que una de las cosas que más hemos aprendido es en buscarnos la vida, por eso pensamos que podríamos hacer algo diferente de que no hayamos tocado mucho en el ciclo, y que a la vez sea funcional.

Por eso mismo pensamos en Arduino, ya que nos parece una gran herramienta, para poder hacer proyectos tecnológicos, después de pensar en Arduino buscamos en varios sitios diferentes como en foros o y en Youtube, también miramos varios proyectos de anteriores años, pero la mayoría de cosas no nos llamaban la atención, ya que nosotros desde el principio queríamos tener un proyecto físico y que sea muy útil para cualquier tipo de persona, no simplemente para personas que ya dominan la informática, (la mayoría de

cosas de años anteriores nos parecían poco útiles para la vida cotidiana y nos parecía proyectos para personas muy en concreto).
y mirando y mirando, encontramos un proyecto físico y que se podía expandir (con el termino expandir me refiero a que podemos romper el limite de el proyecto ya previamente creado).
Nos encontramos con una idea que nos gusto, se trataba de una caja fuerte.

03. Justificació

Este proyecto se justifica porque combina múltiples áreas de aprendizaje, primero permite integrar diferentes áreas de conocimiento en un único sistema funcional: electrónica (circuitos y componentes), programación (lógica de control), diseño (interfaz de usuario) y desarrollo de aplicaciones móviles.

Segundo se trata de un proyecto con una utilidad clara: la protección de objetos personales mediante un sistema de seguridad sencillo pero eficaz. Esto lo convierte en un producto aplicable a la vida real, a diferencia de otros proyectos más teóricos.

Además, el uso de Arduino facilita el aprendizaje y la experimentación, permitiendo modificar y ampliar el sistema fácilmente. Esto fomenta la creatividad y el pensamiento crítico.

Por último, el desarrollo de una aplicación móvil complementaria añade valor al proyecto, acercándolo a soluciones tecnológicas actuales y mejorando la interacción con el usuario.

04. Objetivos

4.1 Objetivo general

Diseñar y desarrollar una caja fuerte electrónica basada en Arduino, capaz de gestionar el acceso mediante un código numérico, incorporando elementos interactivos y una aplicación móvil básica para su control de forma inalámbrica.

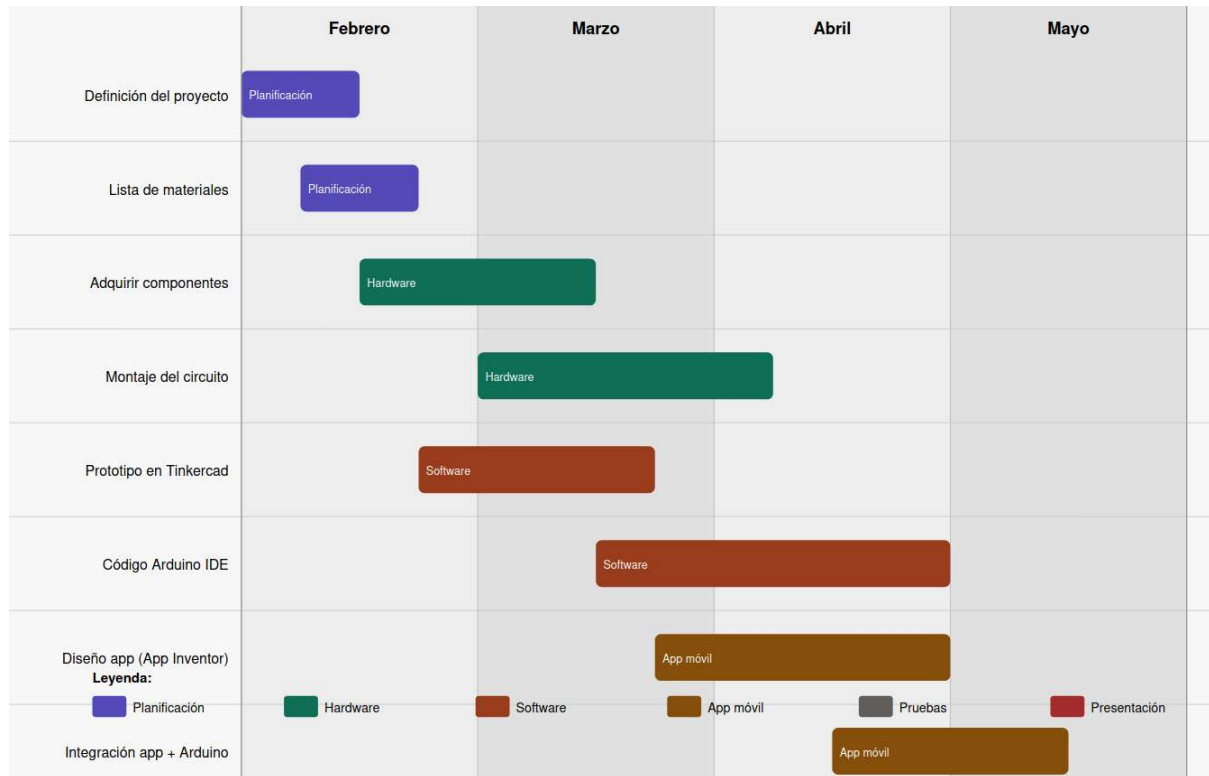
4.2 Objetivos específicos

- Diseñar el circuito electrónico completo del sistema
- Integrar todos los componentes de hardware correctamente
- Programar la lógica de control en Arduino IDE
- Implementar un sistema de validación de contraseña seguro
- Mostrar información al usuario mediante una pantalla LCD
- Utilizar LEDs y buzzer como indicadores visuales y sonoros
- Controlar el mecanismo de apertura mediante un servomotor
- Simular el circuito en Tinkercad antes del montaje físico
- Desarrollar una aplicación móvil sencilla con App Inventor
- Realizar pruebas de funcionamiento y detectar errores
- Documentar todo el proceso de desarrollo

5. Estrategia y planificación del proyecto

La planificación del proyecto la vamos a hacer usando un diagrama de Gantt, que básicamente es una herramienta que nos ayuda a organizar todas las tareas y verlas de forma más clara. Con este diagrama podemos ver cuánto tiempo va a durar cada parte del trabajo, en qué orden se tienen que hacer las cosas y cómo se van relacionando entre ellas. Gracias a esto, es más fácil controlar si estamos cumpliendo los plazos que hemos marcado desde el principio. Así, si vemos que alguna tarea se está retrasando, podemos darnos cuenta rápido y solucionarlo antes de que afecte al resto del proyecto.

Además, el diagrama de Gantt nos ayuda a trabajar de una forma más ordenada, ya que permite tener una visión general de todo lo que hay que hacer. También facilita el reparto de tareas y hace que sea más sencillo seguir el progreso del proyecto. En resumen, es una herramienta muy útil para organizarnos mejor y no perdernos con el trabajo.



5.1 Metodología de trabajo

La metodología que hemos elegido para hacer el proyecto es el modelo ágil, más concretamente Scrum. Este método consiste en ir trabajando el proyecto por partes pequeñas, llamadas iteraciones o sprints, en vez de hacerlo todo seguido de una sola vez. Hemos escogido esta metodología porque creemos que es más flexible y nos permite ir mejorando el proyecto poco a poco. Así, si hay algún error o algo que no funciona bien, lo podemos cambiar sin tener que rehacer todo el trabajo desde el principio.

Además, el modelo ágil encaja bien con el diagrama de Gantt, ya que podemos organizar las tareas por etapas y revisarlas a medida que avanzamos. Esto nos ayuda a tener un mejor control del proyecto y a ir adaptándonos si surgen cambios o imprevistos.

En resumen, pensamos que Scrum es una buena opción porque nos permite trabajar de forma más organizada, dinámica y con mejores resultados finales.

6. Estudio económico y presupuestario



El estudio económico que se ha realizado lo hemos dividido en 2 partes: El estudio económico realista y el estudio económico que se ha usado en nuestro proyecto.


6.1 Estudio económico caso real


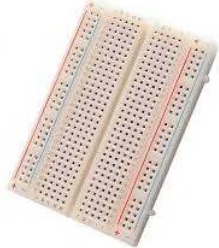


Para hacer el estudio económico se ha tenido en cuenta el conjunto de materiales y el desplazamiento del técnico más la mano de obra.

Productos	Precios
Tecnico mas mano de obra	250€
Materiales	200€
Propinas	45€
TOTAL	495€

6.2 Estudio económico Real de nuestro Proyecto

Componentes	unidades	funcion	imagen
Arduino UNO	1	Placa principal que controla todo el sistema	
Servo motor (SG90)	1	Permite abrir y cerrar la caja automáticamente	

Teclado numérico 4x4	1	Para introducir la contraseña	
Buzzer (zumbador)	1	Muestra mensajes al usuario (código, estado, errores, etc.)	
LED verde	1	Emite sonidos para indicar acciones (correcto/incorrecto)	
LED rojo	1	Indica acceso correcto	
Resistencias (220Ω aprox.)	4	Indica acceso incorrecto	 220 OHM

Botón (reset)	1	Protegen los LEDs y otros componentes	
Protoboard	1	Base para montar el circuito sin soldar	
Cables (jumpers)	varios	Conectan todos los componentes	
Pantalla LCD I2C	1	Pantalla principal para saber el estado de la caja fuerte Abierto/Cerrado	

7. Descripción del proyecto

El proyecto Keepy se basa en el desarrollo de una caja fuerte electrónica automatizada controlada mediante una placa Arduino UNO, cuyo objetivo principal es permitir el acceso seguro a su interior a través de un sistema de autenticación por contraseña.

El sistema dispone de dos formas principales de interacción. La primera consiste en el uso de un teclado numérico físico 4x4, mediante el cual el usuario introduce una contraseña que será validada por el sistema. En caso de que la contraseña sea correcta, el sistema activará un servomotor (SG90) que abrirá la caja automáticamente. Si la contraseña es incorrecta, se activarán señales de aviso mediante un LED rojo y un buzzer, indicando el error.

La segunda forma de interacción es mediante una aplicación móvil desarrollada con App Inventor, que permite complementar el sistema físico, añadiendo funcionalidades como la simulación de control o futuras ampliaciones como conexión remota.

Además, el sistema cuenta con una pantalla LCD 16x2, que muestra información en tiempo real al usuario, como instrucciones, estado del sistema o mensajes de error, mejorando así la experiencia de uso.

En conjunto, Keepy es un sistema que combina electrónica, programación y diseño interactivo para crear una solución funcional y aplicable a la vida real.

7.1 Análisis de requisitos

Para poder desarrollar correctamente el proyecto, ha sido necesario implementar múltiples sistemas que trabajan de forma conjunta. Cada uno de estos elementos ha requerido tiempo de desarrollo, pruebas y correcciones.

Entre los aspectos más importantes del proyecto se encuentran:

- Configurar correctamente la placa Arduino para que todos los componentes (teclado, LCD, servo, LEDs, buzzer, etc.) funcionen de manera coordinada, permitiendo una interacción fluida con el usuario.
- Desarrollar el programa en Arduino IDE que gestione la lógica del sistema, incluyendo la lectura del teclado, validación de la contraseña, control del servomotor y activación de los sistemas de aviso.
- Diseñar y desarrollar una aplicación móvil sencilla con App Inventor, que permita complementar el sistema físico y abrir la posibilidad de futuras ampliaciones.
- Simular el sistema previamente en Tinkercad para detectar errores antes del montaje físico y optimizar el funcionamiento del circuito.
- Realizar el montaje físico del sistema, asegurando una correcta conexión de todos los componentes y solucionando problemas reales derivados del hardware.

7.1.1 Requisitos funcionales

El sistema Keepy debe cumplir con las siguientes funcionalidades:

Permitir la introducción de una contraseña mediante el teclado numérico 4x4.

Validar la contraseña introducida comparándola con una previamente programada en el sistema.

Abrir la caja fuerte automáticamente mediante el servomotor cuando la contraseña sea correcta.

Mostrar mensajes informativos en la pantalla LCD (por ejemplo: "Introduce código", "Acceso correcto" o "Error").

Activar un LED verde cuando el acceso sea correcto.

Activar un LED rojo y un buzzer cuando la contraseña sea incorrecta.

Permitir el reinicio del sistema mediante un botón de reset.

Interactuar con la aplicación móvil desarrollada en App Inventor para complementar el funcionamiento del sistema.

7.1.2 Requisitos no funcionales

Además de las funcionalidades principales, el sistema debe cumplir ciertos requisitos de calidad:

- El sistema debe ser fácil de usar, con una interfaz clara mediante la pantalla LCD.
- Debe ser fiable, evitando errores en la lectura del teclado o en el funcionamiento del servo
- El consumo energético debe ser reducido, ya que funciona con una fuente de alimentación tipo pila.
- El diseño debe ser compacto y organizado, facilitando el montaje y mantenimiento.
- El sistema debe ser escalable, permitiendo futuras mejoras como conexión Bluetooth o integración con otras tecnologías.
- La aplicación móvil debe ser simple e intuitiva, enfocada a complementar el sistema sin complicar su uso.

8. Comparativa de las tecnologías valoradas

Durante el desarrollo del proyecto se han valorado distintas opciones para la creación de la aplicación móvil encargada de controlar la caja fuerte con Arduino. Principalmente, se han analizado dos plataformas:



vs



En un primer momento, se consideró utilizar Kodular, ya que ofrece una amplia variedad de

componentes avanzados y opciones de personalización. Esta herramienta permite desarrollar aplicaciones con una apariencia más profesional y con funcionalidades más completas sin necesidad de programar en lenguajes complejos. Sin embargo, se observó que su interfaz puede resultar más cargada y menos intuitiva para usuarios con poca experiencia como nosotros en este caso, lo que dificultaba el proceso de aprendizaje y desarrollo de nuestro proyecto.

Por otro lado, App Inventor se presenta como una plataforma más sencilla y accesible. Su interfaz gráfica basada en bloques facilita la comprensión de la lógica de programación, especialmente para usuarios principiantes como nosotros. Además, su estructura es más clara y organizada, lo que permite desarrollar la aplicación de forma más rápida y con menos errores.

Después de comparar ambas opciones, se decidió utilizar App Inventor como tecnología principal para el desarrollo de la aplicación. Esta decisión se basa principalmente en su facilidad de uso, la mejor comprensión del entorno por nuestra parte y su adecuación a los requisitos de nuestro proyecto.

8.1 Porque hemos elegido App Inventor?

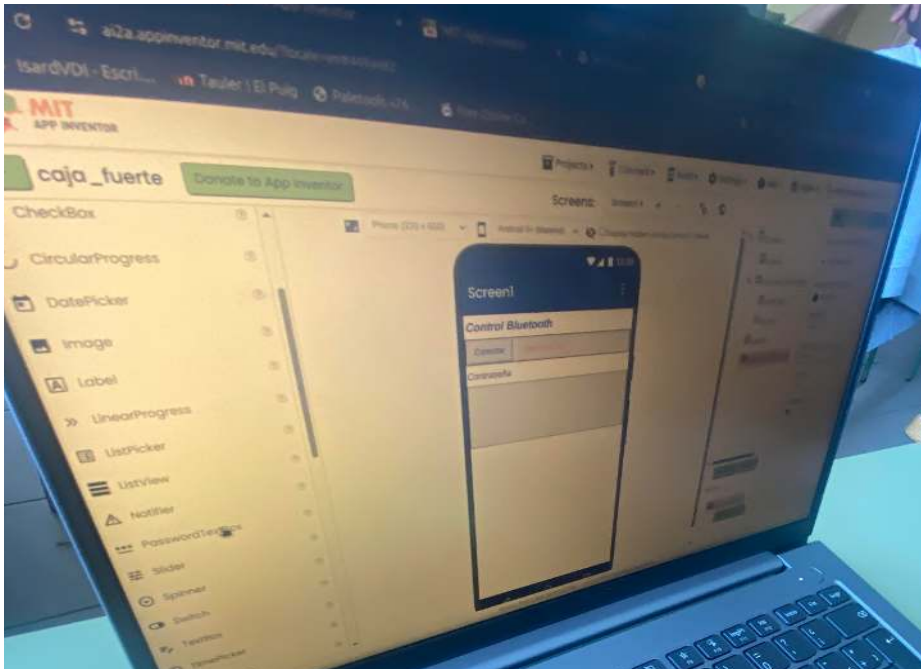
Aunque Kodular ofrece más funcionalidades avanzadas, App Inventor ha resultado ser la opción más adecuada gracias a su simplicidad, claridad y facilidad de aprendizaje, factores clave para garantizar el correcto desarrollo de la aplicación de la caja fuerte.

8.2 Experiencia inicial con App Inventor

El primer día trabajando con App Inventor estuvo centrado en familiarizarnos con la plataforma y entender su funcionamiento básico. Al tratarse de una herramienta nueva para nosotros, dedicamos tiempo a explorar su interfaz y a comprender cómo se organizan los distintos elementos, tanto en la parte de diseño como en la programación por bloques.

Durante el primer día utilizando la aplicación, nos enfocamos principalmente en desarrollar la parte visual de nuestra aplicación. Nuestro objetivo fue crear un primer prototipo de cómo se vería la app que controlará la caja fuerte. Para ello, añadimos componentes como botones, etiquetas y campos de texto, hemos puesto un apartado para poner la contraseña, para conectarse o desconectarse con el funcionamiento de bluetooth, hemos intentado que la interfaz fuera clara, sencilla y fácil de usar para el usuario.

Aunque al principio tuvimos algunas dudas sobre la colocación de los elementos y su configuración, poco a poco fuimos entendiendo mejor el entorno ya que gracias a miles de videos que hay en youtube explicando el funcionamiento de app inventor. Esta primera toma de contacto nos permitió establecer una base visual sobre la que seguir trabajando en los siguientes días, además de darnos una idea más clara de cómo queremos que sea el resultado final de la aplicación.



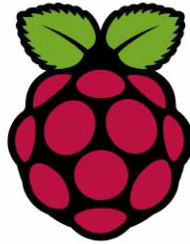
9. Arduino o Raspberry Pi

En el desarrollo de nuestro proyecto se han considerado dos tipos de dispositivos electrónicos muy utilizados: Arduino y Raspberry Pi. Aunque ambos pueden emplearse para crear sistemas interactivos, presentan diferencias importantes en su funcionamiento y en su uso.

Por un lado, **Arduino** es una placa microcontroladora diseñada para ejecutar tareas específicas de forma directa. Es ideal para controlar componentes electrónicos como sensores, motores o sistemas de apertura y cierre, ya que funciona en tiempo real y responde de manera inmediata a las señales que recibe. Además, su programación es relativamente sencilla y está especialmente orientada a proyectos de electrónica.



Por otro lado, **Raspberry Pi** es un ordenador de tamaño reducido (microordenador) que funciona con un sistema operativo completo, similar a un ordenador convencional. Esto le permite realizar tareas más complejas, como ejecutar programas avanzados, gestionar interfaces gráficas completas o conectarse a internet de forma más versátil. Sin embargo, no está tan enfocado al control directo de hardware en tiempo real como Arduino.



En nuestro caso, hemos elegido **Arduino** para el proyecto de la caja fuerte porque se adapta mejor a las necesidades del sistema y porque ya tenemos algo de experiencia en la eso. Necesitamos un dispositivo que controle de forma rápida y eficiente elementos físicos como el mecanismo de apertura, y Arduino cumple perfectamente con esta función. Además, su simplicidad y facilidad de integración con sensores y actuadores lo convierten en la opción más adecuada frente a Raspberry Pi, que resultaría más compleja y menos necesaria para los objetivos del proyecto, y porque también el Arduino aquí en el Puig Castellar lo utilizan mucho más y porque es más barato.

10. Descripción de cada componente

10.1 Teclado numérico de membrana

Esto es un teclado matricial 4x4, un componente electrónico usado para introducir datos o comandos en un circuito o microcontrolador. Tiene 16 teclas organizadas en 4 filas y 4 columnas.

Funciona detectando qué fila y qué columna se conectan cuando se presiona una tecla, permitiendo usar muchas teclas con pocos cables. El cable flexible sirve para enviar las señales eléctricas y el conector negro facilita conectarlo a una placa o protoboard.

Se utiliza mucho en proyectos con Arduino, ESP32 y otros microcontroladores para contraseñas, cerraduras electrónicas, alarmas, menús LCD y sistemas de control. Las teclas A, B, C y D normalmente se usan para funciones especiales programadas, en nuestro caso lo programaremos para utilizarlo para borrar algún número si te has equivocado y otro para confirmar.



Especificaciones Teclado numérico

- Resistencia de contacto: 10 a 500 ohmios.
- Resistencia de aislamiento: 100 megaohmios a 100 voltios.
- Fuerza operativa de la tecla: 150 a 200 Newtons.
- Tiempo de rebote: 1 milisegundo
- Vida útil: 100 millones de veces.
- Temperatura de funcionamiento: -40 a +80 grados Celsius.

10.2 Motor servo ES08A

El servo motor ES08A es un pequeño motor de precisión muy utilizado en electrónica y robótica para controlar posiciones exactas. A diferencia de un motor normal que gira continuamente, este servo puede moverse a un ángulo específico, normalmente entre 0° y 180°.

Funciona recibiendo señales PWM desde un microcontrolador como Arduino o ESP32. Según la señal enviada, el servo gira hasta la posición indicada y la mantiene fija gracias a su sistema interno de control.

Tiene 3 cables:

- Alimentación positiva (VCC)
- Tierra (GND)
- Señal de control

Se usa mucho en robots, brazos mecánicos, puertas automáticas, dirección de carros RC, mecanismos móviles y proyectos donde se necesite movimiento preciso y controlado. Es pequeño, ligero y consume poca energía, por eso es muy popular en proyectos educativos y de automatización.



10.3 IIC Oled Module

El módulo IIC OLED para Arduino es una pequeña pantalla electrónica usada para mostrar texto, números, símbolos e imágenes en proyectos de electrónica. "OLED" significa que cada píxel emite su propia luz, por



lo que la pantalla tiene muy buen contraste, bajo consumo y se ve claramente incluso sin luz externa.

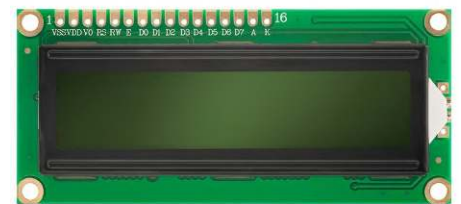
La parte "IIC" o "I2C" indica el tipo de comunicación que utiliza. Gracias a este sistema, la pantalla solo necesita 4 conexiones: VCC, GND, SDA y SCL, lo que simplifica mucho el cableado. El microcontrolador envía los datos por el bus I2C y la pantalla los muestra automáticamente.

Estas pantallas suelen ser de 128×64 píxeles.

Eso significa que tienen 8192 puntos individuales para formar letras e imágenes.

Se usan mucho para mostrar menús, sensores, temperatura, mensajes, relojes, voltajes, datos de robots y sistemas de monitoreo. Son muy populares porque son pequeñas, rápidas y fáciles de programar con librerías de Arduino.

Aclaremos de que nosotros no íbamos a utilizar esta pantalla teníamos mas que claro utilizar una pantalla **LCD 16x2**, pero por problemas que hemos tenido con el tiempo, nos compramos una pantalla LCD 16x2 por aliexpress pero nos ponía que llegara 2 meses tarde, así que con la ayuda del profesor **Jaime Morcillo**, nos ha dado la oportunidad de dejarnos un par de objetos necesarios para trabajar y seguir con nuestro proyecto y por eso tenemos la pantalla IIC Oled Module en vez de la LCD 16x2.

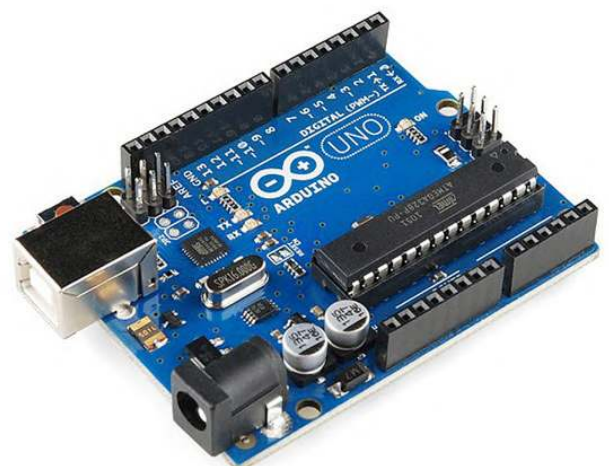


10.4 Placa Arduino UNO

La Arduino UNO es una placa de desarrollo electrónica basada en el microcontrolador ATmega328P y es una de las más utilizadas para aprender programación y electrónica. Sirve para controlar sensores, motores, pantallas, luces y muchos otros componentes electrónicos mediante código.

La placa recibe instrucciones programadas desde una computadora mediante USB y luego ejecuta esas acciones automáticamente. Tiene pines digitales y analógicos que permiten conectar diferentes dispositivos externos.

Cuenta con 14 pines digitales y 6 entradas analógicas. También incluye alimentación de 5V, regulador de voltaje, cristal oscilador y conexión USB para programarla y alimentarla.



Se utiliza muchísimo en robótica, automatización, domótica, proyectos escolares, sistemas de control, alarmas, sensores y prototipos electrónicos porque es económica, fácil de usar y compatible con muchas librerías y módulos electrónicos.

10.5 Buzzer (zumbador)

El buzzer de Arduino es un zumbador utilizado para generar sonidos, pitidos o alarmas en proyectos electrónicos controlados por la placa Arduino.

Funciona convirtiendo señales eléctricas en sonido y se conecta normalmente a un pin digital de la placa.

Puede usarse para emitir alertas, notificaciones, melodías o señales acústicas en proyectos como alarmas, robots, temporizadores, sensores y sistemas de seguridad. Existen buzzers activos, que suenan solo con alimentarlos, y pasivos, que necesitan señales PWM para generar distintos tonos y frecuencias.

Es un componente muy popular porque es barato, pequeño y fácil de programar con Arduino.



10.6 LEDs

Los LEDs verde y rojo son diodos emisores de luz utilizados en Arduino y otros circuitos electrónicos para indicar estados o señales visuales. Funcionan permitiendo el paso de corriente en un solo sentido y transformando energía eléctrica en luz.

Normalmente el LED rojo se usa para indicar error, apagado, alarma o advertencia, mientras que el LED verde suele indicar funcionamiento correcto, encendido o estado activo. Cada LED tiene dos terminales: ánodo (positivo) y cátodo (negativo).



Para evitar que se quemen, se conectan con una resistencia que limita la corriente eléctrica.

Se utilizan mucho en proyectos electrónicos para mostrar información visual de manera simple, rápida y económica.

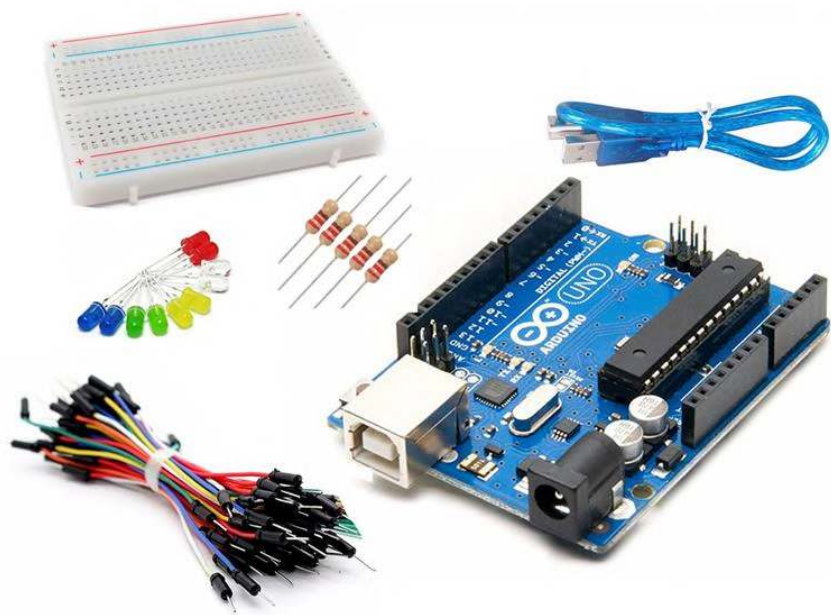
10.7 Cables, resistencias y protoboard

Los cables, resistencias y la protoboard son componentes básicos utilizados en proyectos con Arduino y electrónica en general para construir y probar circuitos sin necesidad de soldar.

Los cables sirven para conectar eléctricamente los componentes y transportar corriente o señales entre ellos. Existen diferentes tipos, como macho-macho, macho-hembra y hembra-hembra, dependiendo de la conexión necesaria.

Las resistencias son componentes que limitan el paso de corriente eléctrica para proteger dispositivos como LEDs, sensores o microcontroladores. Su valor se mide en ohmios (Ω) y se calcula usando la Ley de Ohm.

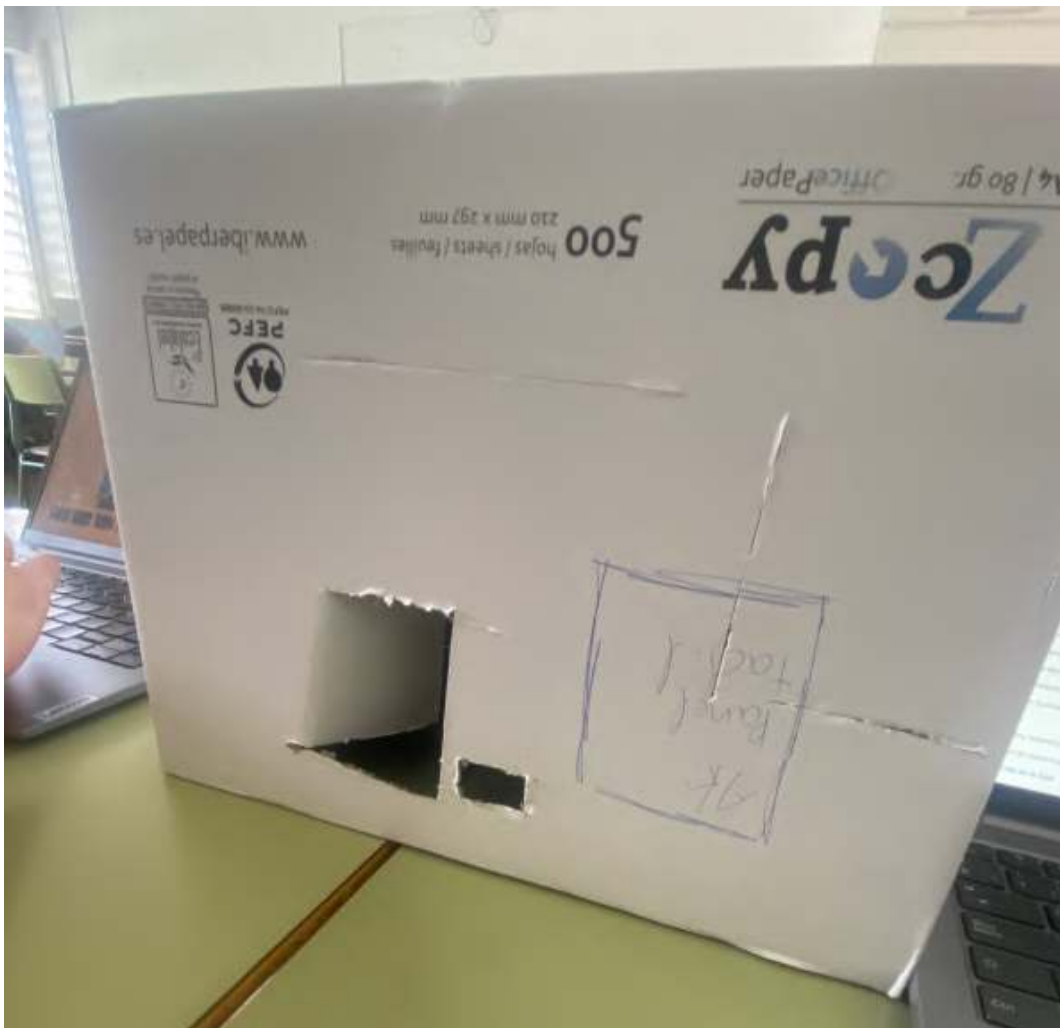
La protoboard es una placa de pruebas que permite montar circuitos temporalmente sin soldadura. Internamente tiene conexiones metálicas que unen ciertos agujeros, facilitando conectar componentes y hacer pruebas rápidas de manera segura y ordenada. Se usa mucho para prototipos, aprendizaje y desarrollo de circuitos electrónicos.



11. Creación de nuestro proyecto

11.1 Montaje de la Caja

Utilizamos una caja de cartón como estructura principal para nuestro proyecto. La caja tiene unas dimensiones aproximadas de 30 cm de ancho y 10 cm de largo, lo que la hace compacta y fácil de manejar. Este tipo de material resulta ideal para prototipos, ya que es ligero, económico y sencillo de perforar o modificar según las necesidades del proyecto.



Esta es nuestra idea mas o menos de como va ser la caja, obviamente esta foto es de nuestra caja de pruebas lo primero que hemos hecho es hacer un hueco en la caja para la puerta principal que es de 3cm de largo y de ancho.

Después otro hueco con el cutter para la pantalla IIC Oled Module y un ejemplo para el teclado numérico.

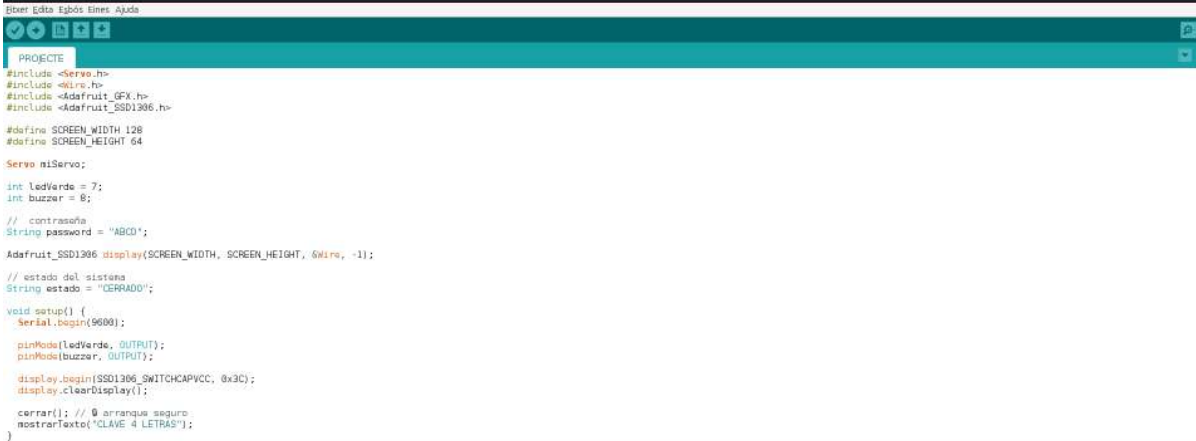
En un principio, nuestra idea era construir la caja con hierro para que el resultado fuera mucho más profesional y resistente. Sin embargo, debido a la falta de tiempo y de materiales adecuados, no pudimos llevar a cabo esa opción, por lo que optamos por una solución más sencilla y accesible.



Como se ve en la imagen, Hemos instalado el servomotor ES08A en la parte trasera de la puerta para que pueda realizar su función de abrir y cerrar la puerta de la caja fuerte de manera eficiente. De esta forma, el mecanismo queda oculto y protegido, asegurando un funcionamiento más seguro y discreto.

Además, hemos colocado la pantalla IIC Oled Module en un hueco previamente realizado con un cúter, ajustándolo con precisión para que encaje perfectamente. Esto permite una integración limpia del componente en la estructura, facilitando la visualización de la información sin afectar al diseño del conjunto.

11.2 Programacion



```
PROJECTE
#include <Servo.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64

Servo miServo;

int LedVerde = 7;
int buzzer = 8;

// contraseña
String password = "ABCD";
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

// estado del sistema
String estado = "CERRADO";

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

  pinMode(LedVerde, OUTPUT);
  pinMode(buzzer, OUTPUT);

  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  display.clearDisplay();

  cerrar(); // 0 arranque seguro
  mostrarTexto("CLAVE 4 LETRAS");
}
```

Primero, las librerías.

Al inicio del código se incluyen varias librerías. Cada una sirve para una cosa concreta. La librería Servo permite controlar el servomotor. Wire permite la comunicación I2C, que es la que usa la pantalla OLED. Adafruit_GFX es una librería gráfica para poder escribir texto y dibujar, y Adafruit_SSD1306 es la que controla específicamente la pantalla OLED.

Después viene la configuración de la pantalla.

Se define el tamaño de la pantalla con dos valores: ancho 128 y alto 64. Esto es necesario para que el Arduino sepa cómo trabajar con la pantalla.

Luego se crean los objetos y variables.

Se crea un objeto llamado miServo, que representa el servomotor. También se definen los pines donde están conectados el LED verde y el buzzer, en este caso los pines 7 y 8. Se define la contraseña como un texto, por ejemplo "ABCD". También se crea el objeto display, que representa la pantalla OLED. Finalmente hay una variable llamada estado que guarda si el sistema está en "CERRADO" o "ABIERTO". Esto evita que el servo reciba órdenes repetidas.

La función setup se ejecuta una sola vez al encender el Arduino.

Primero se inicia la comunicación serie con el ordenador, lo que permite escribir la contraseña desde el monitor serie. Después se configuran los pines del LED y del buzzer como salidas. Luego se inicializa la pantalla OLED y se limpia. A continuación se llama a la función cerrar para asegurarse de que el sistema empieza en estado cerrado. Por último, se muestra en la pantalla el mensaje "CLAVE 4 LETRAS".

La función loop se repite continuamente.

Dentro de esta función, el Arduino comprueba si hay datos disponibles en el puerto serie, es

decir, si has escrito algo en el monitor serie. Si hay datos, los lee hasta que detecta un salto de línea (cuando pulsas enter). Luego elimina espacios sobrantes. Después compara lo que has escrito con la contraseña guardada. Si coinciden, llama a la función abrir. Si no coinciden, llama a la función error.

La función abrir se ejecuta cuando la contraseña es correcta.

Primero comprueba si el estado no es ya "ABIERTO". Si no lo es, entonces conecta el servo al pin 9, lo mueve a 90 grados (posición de abierto), espera medio segundo para que llegue a la posición y luego lo desconecta para evitar vibraciones. Después enciende el LED verde, activa el buzzer durante medio segundo para hacer un pitido y muestra "ABIERTO" en la pantalla. Luego cambia el estado a "ABIERTO".

Después de eso, el sistema espera unos segundos (por ejemplo 3 segundos) con la puerta abierta. Pasado ese tiempo, llama a la función cerrar para volver a la posición inicial y vuelve a mostrar "CLAVE 4 LETRAS".

La función cerrar devuelve el sistema al estado cerrado.

Primero comprueba si el estado no es ya "CERRADO". Si no lo es, conecta el servo, lo mueve a 0 grados (posición cerrada), espera medio segundo y lo desconecta para evitar temblores. Luego apaga el LED y el buzzer y cambia el estado a "CERRADO".

La función error se ejecuta cuando la contraseña es incorrecta.

Primero muestra en la pantalla el mensaje "INCORRECTO". Luego hace dos pitidos con el buzzer utilizando un bucle que se repite dos veces. En cada repetición el buzzer suena un corto tiempo, se apaga, y hay una pequeña pausa. Después de los pitidos, el sistema espera un momento y vuelve a mostrar "CLAVE 4 LETRAS" en la pantalla.

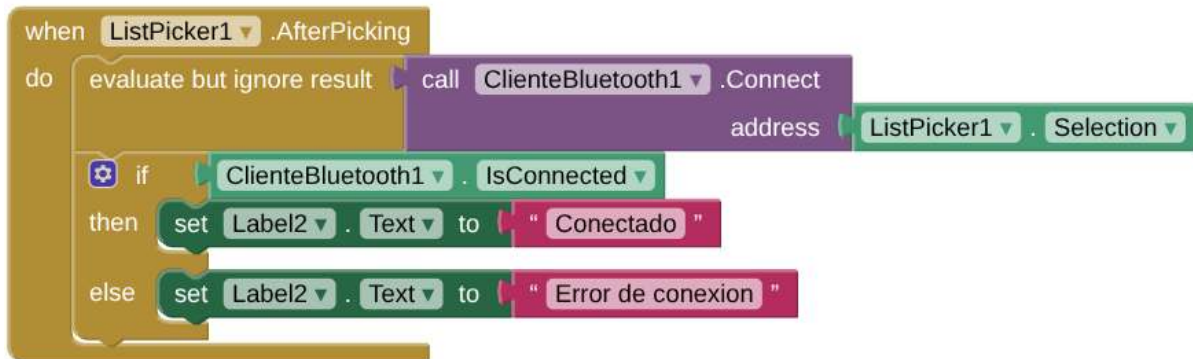
La función mostrarTexto se encarga de la pantalla OLED.

Cada vez que se llama, borra la pantalla, configura el tamaño del texto y el color. Luego coloca el cursor en una posición y escribe el texto fijo "CLAVE 4 LETRAS". Después coloca el cursor un poco más abajo y escribe el mensaje que se le pasa como parámetro, como "ABIERTO" o "INCORRECTO". Finalmente actualiza la pantalla para que se vea el contenido.

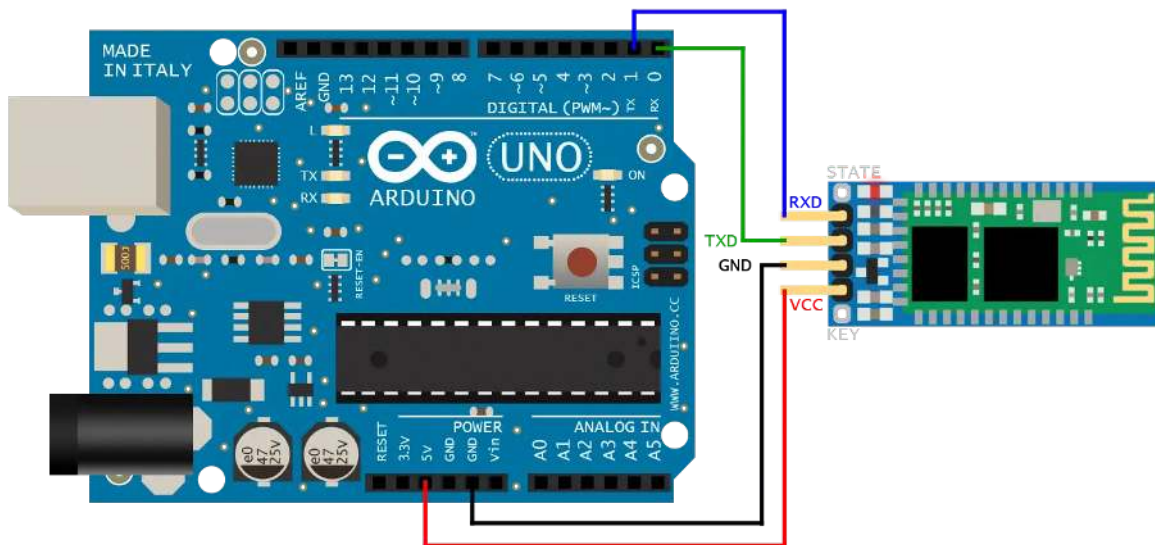
En conjunto, el programa funciona como una cerradura electrónica básica. Al encender, está cerrado y pide una contraseña. Si introduces la correcta, abre, avisa con luz y sonido, y luego se vuelve a cerrar. Si la contraseña es incorrecta, muestra un error y emite dos pitidos.

Al parecer todo funcionaba bien en nuestra placa arduino, pero nos estuvimos informando y para usar correctamente la aplicación móvil que ya teníamos preparada y prácticamente hecha, necesitábamos un aparato para poder conectar correctamente por Bluetooth la placa arduino a nuestra aplicación móvil.

Dentro de nuestra aplicación tenemos una especie de instrucciones que nosotros las llamamos bloques



en esta instrucción lo que hacemos es conectar la placa arduino a la aplicación mediante bluetooth.

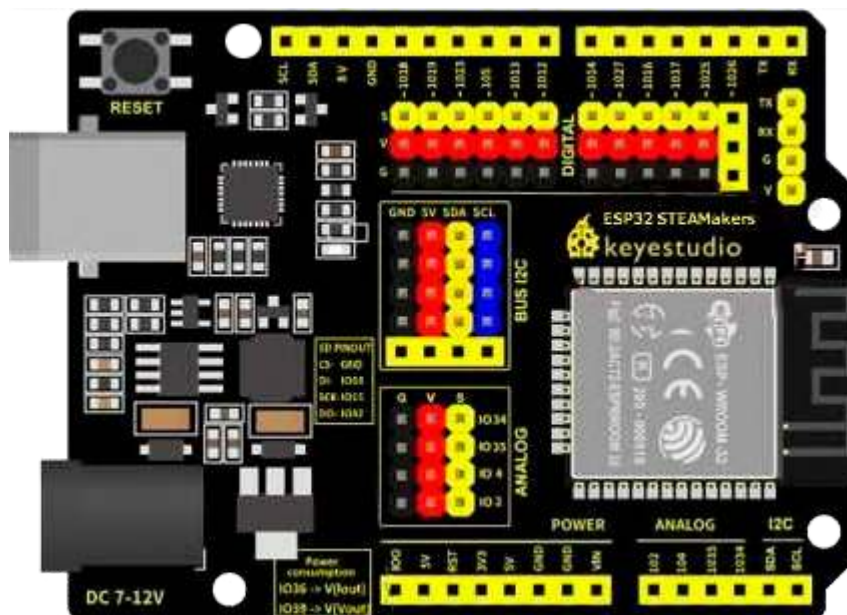


Como podemos ver en la imagen tendríamos que usar otra placa, esta placa externa se llama HC-05 Bluetooth Module, que sirve básicamente para esto.

El módulo HC-05 Bluetooth Module sirve para conectar un Arduino con otros dispositivos mediante Bluetooth, como un móvil o un ordenador. Se usa para enviar y recibir datos sin cables, por ejemplo para controlar luces, motores o robots desde una app. Funciona como una comunicación serie inalámbrica y es uno de los módulos más usados en proyectos de Arduino porque es barato y fácil de configurar.

Pero estuvimos buscando varias opciones más, y incluso hablamos con el señor Jaime para que nos diese este módulo, pero estuvimos y vimos que teníamos otras opciones aun más interesante aunque más complejas de usar y nuevas para nosotros.

ESP32 STREAMAKERS



Y llegamos a esta nueva placa muy muy parecida a la placa arduino convencional,

El nombre de esta placa es ESP32, las principales diferencias entre la placa arduino convencional son estas:

El ESP32 STREAMAKERS es más moderno y potente que un Arduino con módulo Bluetooth aparte. La principal diferencia es que el ESP32 ya lleva Bluetooth y Wi-Fi integrados, así que no hace falta conectar un HC-05. También es más rápido, tiene más memoria y permite hacer proyectos más avanzados como domótica, control por móvil o conexión a internet.

En cambio, el HC-05 solo sirve para añadir Bluetooth a un Arduino y tiene menos funciones. El ESP32 es más completo y práctico para proyectos actuales. El ESP32 DevKit es más moderno y potente que un Arduino con módulo Bluetooth aparte. La principal diferencia es que el ESP32 ya lleva Bluetooth y Wi-Fi integrados, así que no hace falta conectar un

HC-05. También es más rápido, tiene más memoria y permite hacer proyectos más avanzados como domótica, control por móvil o conexión a internet.

En cambio, el HC-05 solo sirve para añadir Bluetooth a un Arduino y tiene menos funciones. El ESP32 es más completo y práctico para proyectos actuales

11.3 Creación de aplicación

Para el desarrollo de la aplicación móvil de nuestro proyecto utilizamos MIT App Inventor, una herramienta de programación visual que permite crear aplicaciones para dispositivos Android de forma sencilla e intuitiva.

Esta plataforma funciona mediante bloques gráficos, lo que facilita el desarrollo sin necesidad de escribir código complejo. Gracias a esto, pudimos diseñar la interfaz de la aplicación y programar su funcionamiento de manera rápida, adaptándola a las necesidades de nuestro sistema.

La aplicación desarrollada permite la interacción con el proyecto, enviando y recibiendo información para controlar su funcionamiento. De este modo, el usuario puede gestionar el sistema de forma cómoda desde su dispositivo móvil.

El uso de App Inventor ha sido clave en el proyecto, ya que ha simplificado el proceso de desarrollo y ha permitido centrarse en la lógica y funcionalidad de la aplicación.

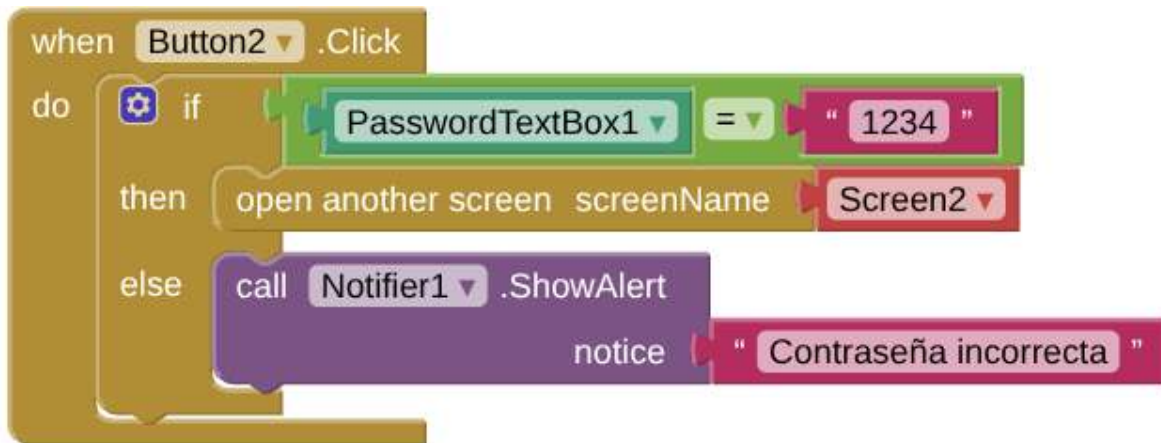
Básicamente la parte de la aplicación se basa en dos partes bastante diferenciadas. La primera parte lo que hemos hecho es tener en cuenta la parte gráfica de nuestra aplicación, hemos intentado tener bastante en cuenta la parte visual de lo que viene siendo la aplicación ya que es prácticamente lo único que vemos dentro de la aplicación, hemos puesto colores bastante vivos, algunos logos del instituto y otras cosas que tienen que ver para que a la mayoría de gente pueda usar la aplicación.

Esta es la primera pantalla de nuestra aplicación, como podemos ver es bastante sencilla, ya que uno de nuestros propósitos dentro de la aplicación es que cualquier tipo de persona pueda usarla sin ningún tipo de problema.

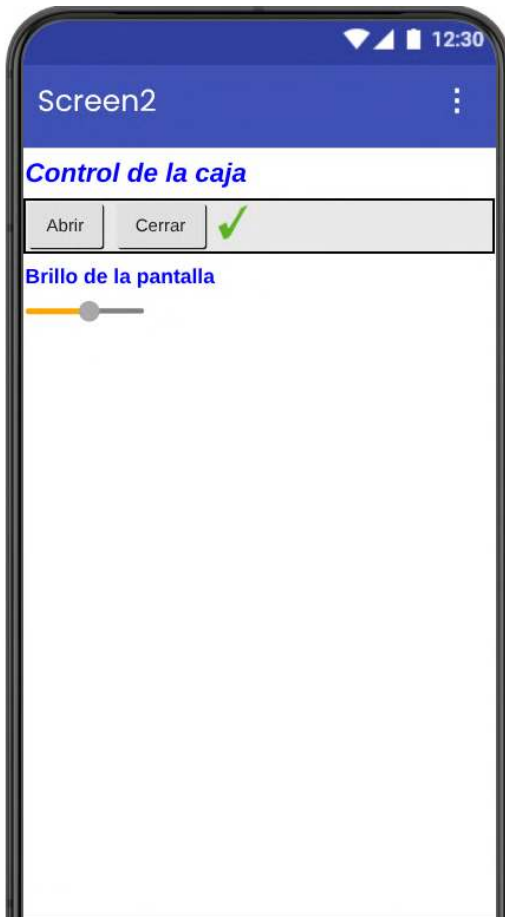
Tenemos dos grandes controles, el primero y más importante es la parte de control de Bluetooth, si no conectamos la aplicación a Arduino, no podremos disfrutar de los beneficios que nos da la aplicación.

Y posteriormente tenemos la parte de la contraseña, como en casi todas las aplicaciones para desbloquear otra pantalla tenemos que saber la contraseña para posteriormente poder acceder a abrir lo que viene siendo la caja fuerte.



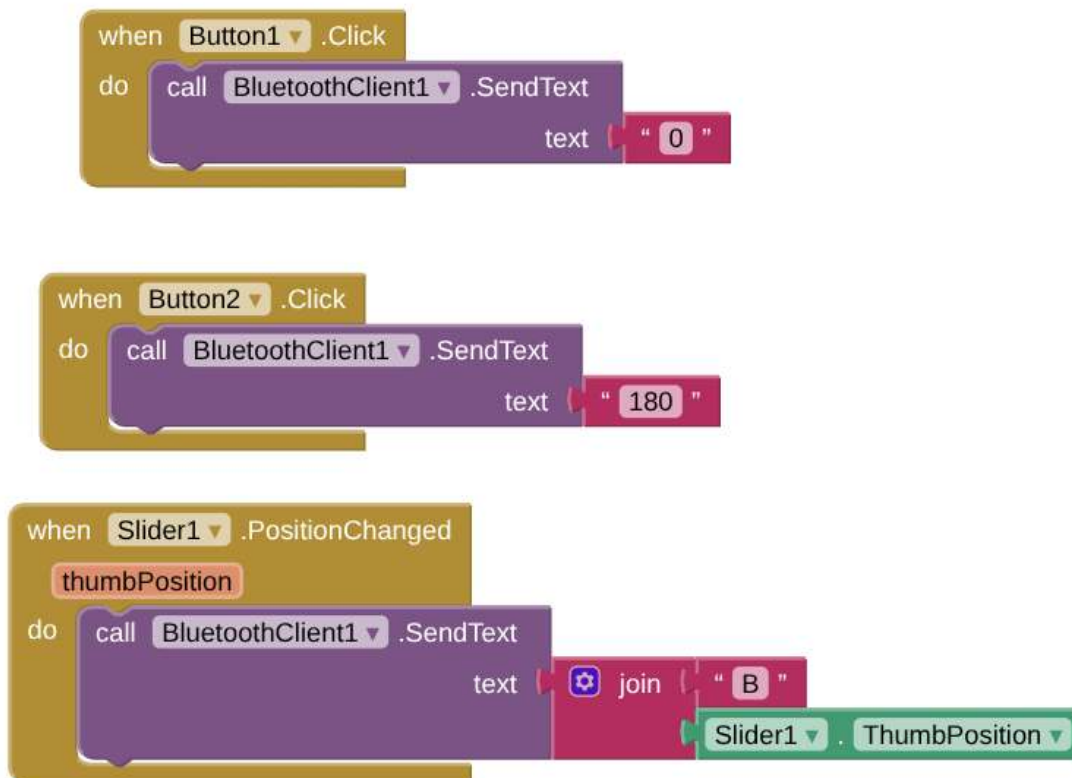


y por otro lado tenemos la parte de los bloques, que básicamente lo que hace es la parte grafica tenga sentido y podamos hacer que todo funcione correctamente.



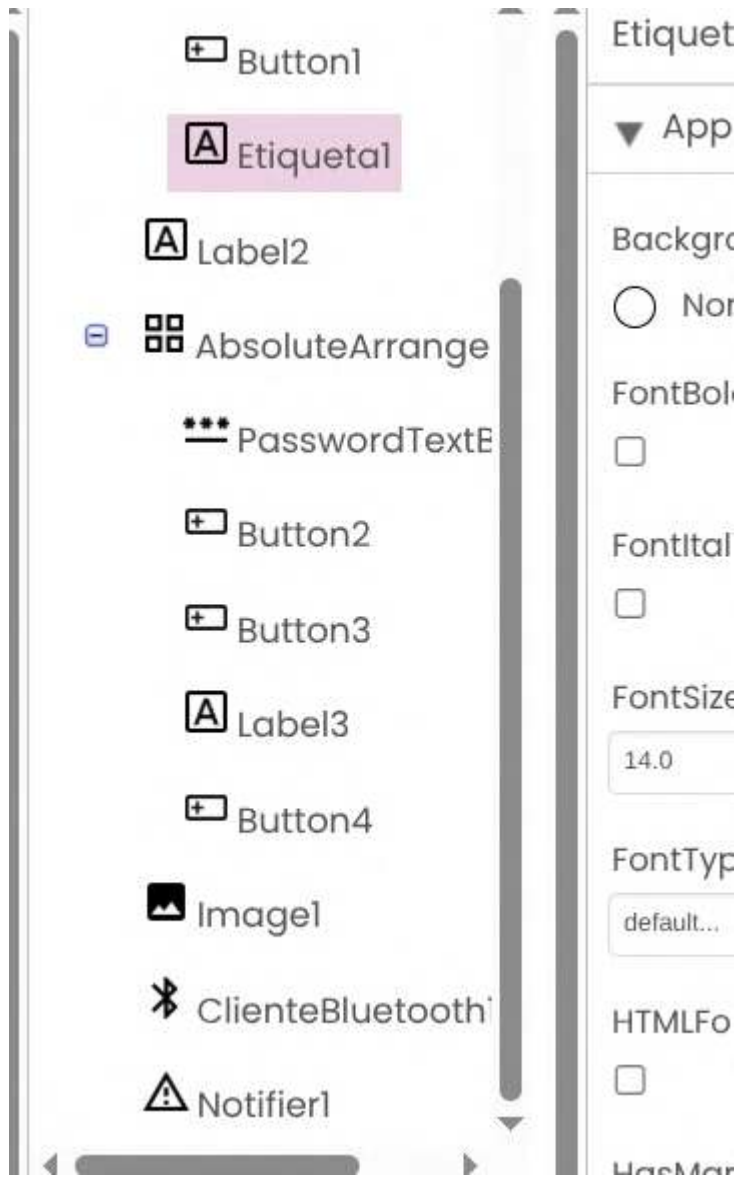
Al acertar la contraseña lo que nos sale esta pantalla que nos deja abrir o cerrar la caja y poder cambiar la intensidad de las luces que tiene la pantalla lcs y de las leds.

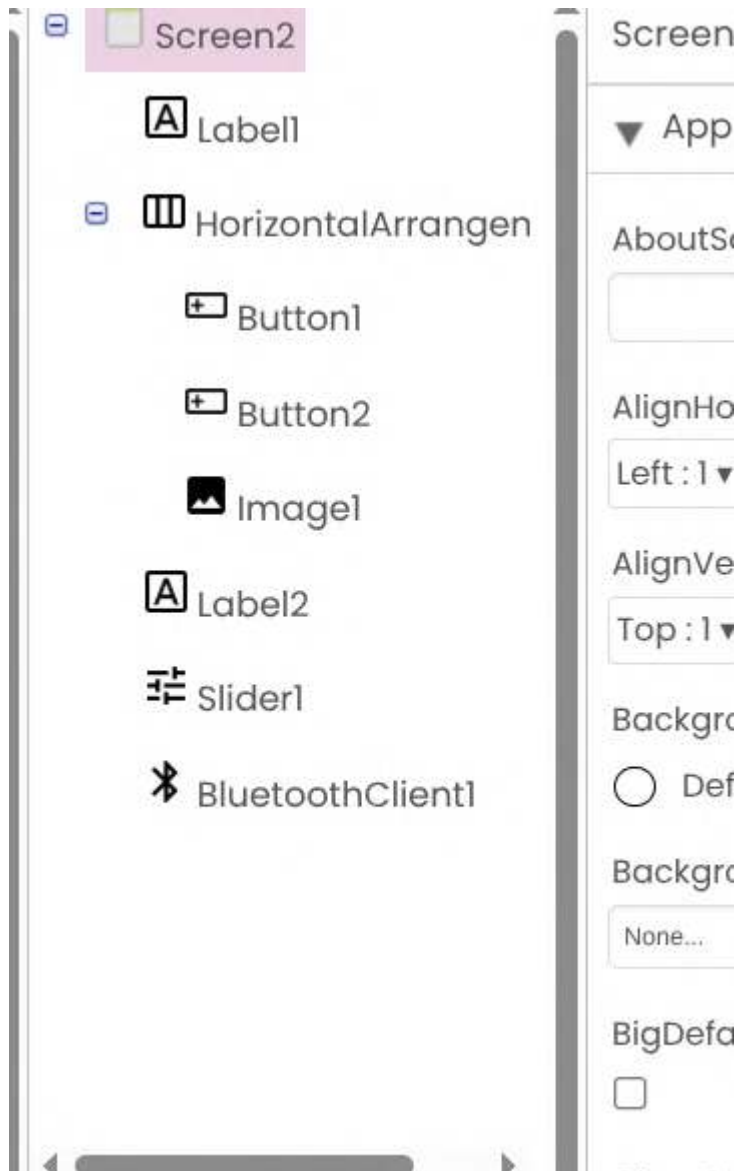
Los bloques que hemos usado para que todo funcione son estos.



Al hacer la aplicación de la caja fuerte con App Inventor me encontré con varios problemas típicos, por ejemplo a veces el móvil no detectaba el Bluetooth o el HC-05 no se conectaba bien porque no estaba emparejado correctamente. También tuve errores con la comunicación entre la app y Arduino porque algunos datos no se enviaban como esperaba o la velocidad de transmisión no coincidía. En algunas pruebas el servo no respondía bien por falta de alimentación y el Arduino incluso se reiniciaba. Otro problema fue controlar bien los bloques de App Inventor para enviar los datos correctos y comprobar si el Bluetooth estaba conectado antes de usar la app. Además, el alcance del Bluetooth a veces fallaba y había desconexiones si el móvil se alejaba demasiado

Al hacer la app con App Inventor, una de las cosas que más nos costó fue que a veces era difícil encontrar exactamente los bloques que necesitábamos, porque hay muchos y no siempre está claro cuál usar. También nos pasó que queríamos hacer una función concreta, pero no sabíamos cómo construirla con los bloques que ofrece la herramienta, así que teníamos que probar varias combinaciones hasta que funcionaba. Otra dificultad fue entender bien cómo organizar la lógica de la app, ya que si pones los bloques en mal orden no funciona como esperas. En general, aunque es una herramienta fácil de usar, al principio cuesta un poco acostumbrarse y encontrar todo lo que necesitas.





Estas cosas que pongo son todos los botones y utilidades que hemos tenido que programar para poder hacer que nuestra aplicación funcione de una forma correcta, són bastantes pero la parte más difícil a sido programarlas y hacer algunas veces prueba y error para que todo funcione, ya que la mayoría de veces había fallos que prácticamente no se podían ni ver.

La librería:

```
#include "BluetoothSerial.h"
```

se utiliza para activar la comunicación Bluetooth clásica del ESP32 y permitir la conexión inalámbrica con una aplicación móvil. Esta librería crea un puerto serie virtual por Bluetooth, funcionando de forma similar al monitor serie USB (Serial), pero sin necesidad de cables. Para utilizarla se crea primero un objeto Bluetooth con la instrucción `BluetoothSerial`

SerialBT;, que será el encargado de gestionar toda la comunicación entre el ESP32 y el dispositivo móvil.

En la función `setup()` se inicia el Bluetooth mediante la instrucción `SerialBT.begin("ESP32_Cerradura");`. Esto hace que el ESP32 active su módulo Bluetooth y aparezca visible en el teléfono móvil con el nombre "ESP32_Cerradura". Desde la aplicación móvil el usuario puede buscar dispositivos Bluetooth disponibles y conectarse al ESP32 seleccionando ese nombre.

Una vez establecida la conexión, el programa comprueba continuamente si llegan datos desde la aplicación mediante la instrucción `if (SerialBT.available())`. Cuando la aplicación envía información, el ESP32 la recibe usando `char datoAPP = SerialBT.read();`, almacenando el carácter recibido en la variable `datoAPP`.

El sistema funciona enviando caracteres simples desde la aplicación móvil. En este proyecto se programó que cuando la app envía el carácter 'A', el ESP32 ejecuta la función `abrir()`, encargada de abrir la cerradura mediante el servomotor. Cuando la aplicación envía el carácter 'B', el ESP32 ejecuta la función `cerrar()`, cerrando nuevamente la cerradura. De esta forma, la aplicación móvil actúa como un mando inalámbrico para controlar el sistema de apertura y cierre.

La comunicación se realiza mediante Bluetooth clásico tipo SPP (Serial Port Profile), lo que permite utilizar aplicaciones sencillas de control Bluetooth o aplicaciones desarrolladas específicamente para el proyecto. Gracias a este sistema, el usuario puede controlar la cerradura electrónica desde el teléfono móvil de forma inalámbrica, cómoda y rápida.

```
void loop() {  
  // Bluetooth  
  if (SerialBT.available()) {  
    char datoAPP = SerialBT.read();  
    if (datoAPP == 'A') {  
      abrir();  
    }  
    else if (datoAPP == 'B') {  
      cerrar();  
      input = "";  
      mostrarInput();  
    }  
  }  
}
```

CAJA FUERTE

Para empezar con el proyecto de la caja fuerte con Arduino, utilizamos una caja de cartón como prototipo inicial para hacer todas las pruebas antes de construir la versión final. La idea era simular una caja fuerte real de una forma sencilla y económica, para poder comprobar que todo el sistema funcionaba correctamente sin arriesgarnos a dañar materiales más caros o definitivos.

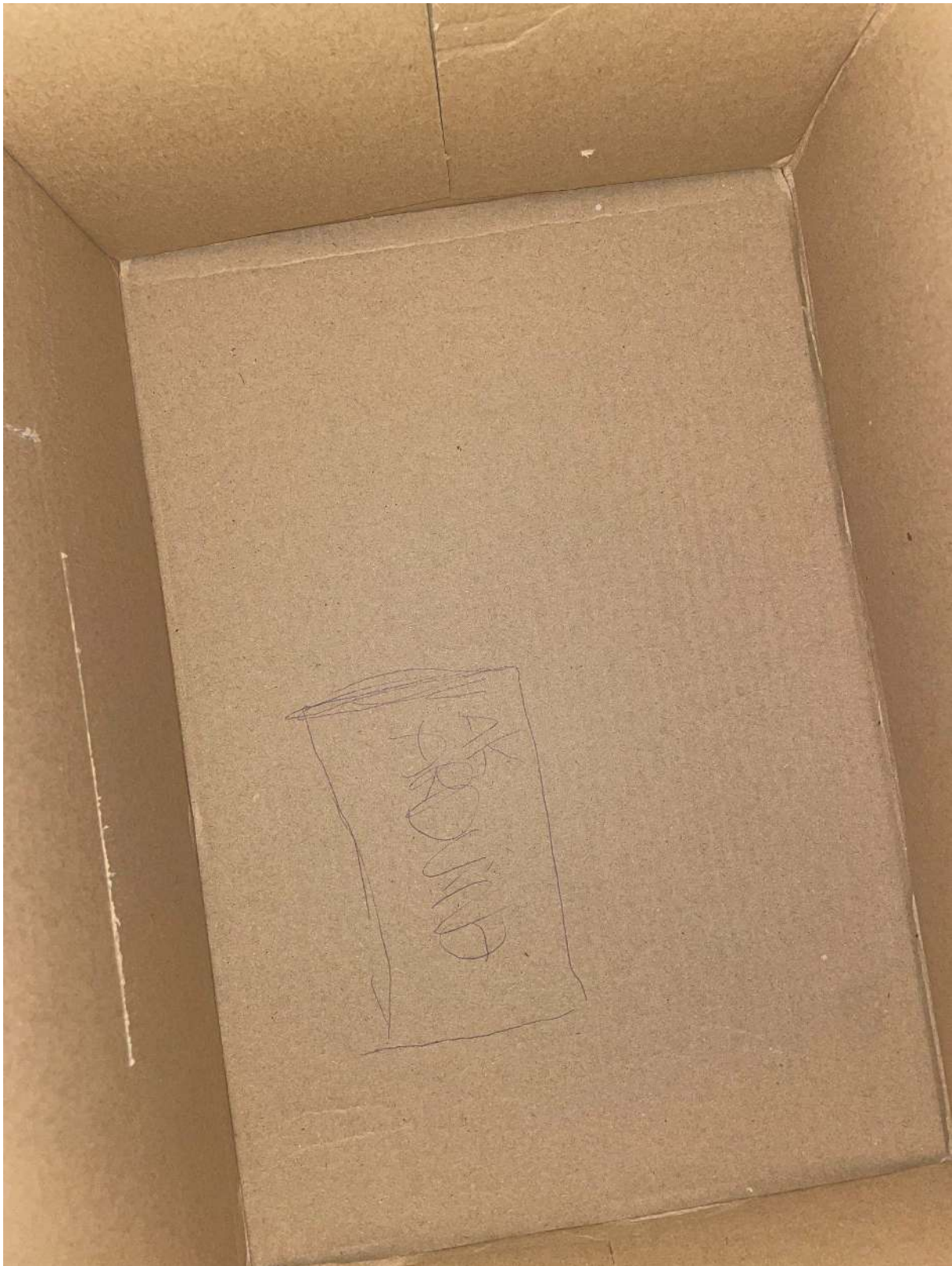
Primero adaptamos la caja de cartón haciendo una pequeña puerta en uno de los lados, como si fuera la puerta de una caja fuerte real. Después colocamos el mecanismo de cierre, que en nuestro caso era un servo motor conectado al Arduino, de forma que pudiera abrir y cerrar la “puerta” de la caja. El servo lo fijamos con cinta y cartón reforzado para que no se moviera al hacer pruebas.

También organizamos dentro de la caja todos los componentes electrónicos, como el Arduino, los cables y el módulo Bluetooth, intentando que quedaran bien colocados y no se soltaran mientras hacíamos las pruebas. La caja de cartón nos ayudó mucho porque podíamos ver fácilmente cómo se comportaba el sistema y hacer cambios rápidos si algo no funcionaba bien.

Gracias a este prototipo pudimos comprobar si la comunicación entre la app y el Arduino funcionaba correctamente, si el servo respondía a las órdenes y si el sistema de apertura y cierre era estable. Además, nos permitió detectar fallos como cables mal conectados, problemas de alimentación o errores en el código sin complicarnos demasiado.

En general, usar la caja de cartón fue muy útil porque nos permitió hacer pruebas reales del proyecto de la caja fuerte de forma sencilla, rápida y económica, y así mejorar el diseño antes de pasar a una versión más definitiva.





Para el proyecto final de la caja fuerte con Arduino, vamos a seguir utilizando una caja de cartón como base, pero esta vez será una versión más cuidada y pensada como el prototipo del sistema. La caja será completamente lisa, sin dibujos ni marcas, ya que la idea es que tenga un aspecto más limpio y profesional, como si fuera una caja fuerte de verdad pero en modo de prueba.

Además, la caja estará pintada de color azul para darle un mejor acabado visual y diferenciarla claramente del prototipo inicial que usamos para las primeras pruebas. Este cambio de diseño nos ayudará a presentar el proyecto de una forma más ordenada y atractiva, sin perder la función principal que es comprobar el funcionamiento del sistema.

Dentro de esta caja seguiremos integrando todos los componentes del proyecto, como el Arduino, el sistema de cierre con el servo motor y la conexión Bluetooth. La diferencia es que ahora intentaremos que todo quede mejor organizado y fijado para que el montaje sea más estable y se parezca más a un producto final.

En resumen, esta nueva caja de cartón azul será la versión definitiva de prueba del proyecto, donde podremos demostrar el funcionamiento completo de la caja fuerte de una manera más realista y visualmente más cuidada.

Para montar la caja fuerte del proyecto vamos a hacerlo paso a paso usando una caja de cartón, porque es lo más práctico para ir probando todo sin complicarnos ni gastar dinero en materiales más serios desde el principio.

Primero vamos a elegir una caja de cartón que sea resistente y que tenga un tamaño más o menos decente para poder meter dentro el Arduino, los cables y el sistema de cierre. Antes de empezar a cortar nada, vamos a reforzar un poco si hace falta, sobre todo las esquinas y la base, para que no se deforme cuando empecemos a manipularla.

Después vamos a marcar la puerta de la caja. Básicamente dibujamos un rectángulo en uno de los lados, que será la parte que se abrirá como si fuera una caja fuerte de verdad. No la vamos a cortar entera, solo tres lados, para que quede como una puerta que se puede abrir y cerrar sin separarse del todo de la caja.

Cuando tengamos eso hecho, pasamos a la parte estética. Vamos a pintar toda la caja de color azul, sin dibujos ni nada raro, simplemente todo uniforme. La idea es que quede más limpia y que parezca un prototipo serio, no una caja cualquiera. Dejamos que se seque bien la pintura porque si no luego es un desastre tocarla.

Luego ya viene la parte más importante, que es montar el sistema. Vamos a colocar un servo motor dentro de la caja, justo alineado con la puerta, para que sea el que haga de cierre. Este servo va a ser el que realmente bloquee o desbloquee la caja cuando reciba la señal del Arduino. Lo vamos a fijar bien con cinta fuerte o silicona caliente para que no se mueva cuando esté funcionando.

Después colocamos el Arduino dentro de la caja, intentando dejarlo en una posición fija y ordenada. Aquí también organizamos todos los cables del servo y del módulo Bluetooth para que no queden sueltos ni liados, porque eso luego puede dar fallos o desconexiones raras.

Una vez todo está montado dentro, conectamos el módulo Bluetooth para poder controlar la caja desde el móvil. En este punto comprobamos que la app se conecta bien con el Arduino y que los comandos llegan correctamente.

Y por último hacemos pruebas reales. Probamos abrir y cerrar la caja varias veces desde el móvil, vemos si el servo responde bien, si la puerta encaja como debe y si todo el sistema es estable. Si algo falla, pues tocamos cables, revisamos el código o ajustamos el servo hasta que funcione bien.

En general, con este proceso vamos a tener una caja fuerte funcional hecha con cartón, bastante realista y perfecta para enseñar cómo funciona todo el sistema sin necesidad de hacerlo complicado desde el principio.

12. Conclusiones

12.1 Conclusiones generales del proyecto

La realización del proyecto Keepy nos ha permitido aprender mucho más sobre el trabajo en equipo y mejorar nuestra capacidad de organización y comunicación durante el desarrollo de un proyecto tecnológico real. Además, hemos ampliado nuestros conocimientos en áreas que apenas habíamos trabajado anteriormente, como la programación con Arduino IDE, el uso de App Inventor para aplicaciones móviles y la integración de diferentes componentes electrónicos en un mismo sistema.

Consideramos que el resultado final ha sido muy positivo, ya que hemos conseguido cumplir los objetivos principales que nos habíamos planteado al inicio del proyecto. A pesar de que en algunos momentos tuvimos dificultades y nos encontramos atascados por falta de experiencia o información, conseguimos resolver los problemas mediante investigación, pruebas y trabajo constante.

Gracias a este proyecto hemos aprendido a desarrollar una caja fuerte electrónica automatizada capaz de abrirse mediante contraseña y controlarse también desde una aplicación móvil mediante Bluetooth. Además, el sistema puede ampliarse y mejorarse fácilmente en el futuro, incorporando nuevas tecnologías y funcionalidades más avanzadas. Por ello, creemos que Keepy no solo ha sido un proyecto educativo, sino también una idea con potencial para aplicarse en situaciones reales.

12.2 Consecución de los objetivos

Los objetivos planteados al comienzo de la memoria se han cumplido satisfactoriamente gracias al esfuerzo y dedicación invertidos durante todo el desarrollo del proyecto.

Se ha conseguido diseñar y montar correctamente el circuito electrónico completo, integrando componentes como el teclado numérico, la pantalla OLED, el servomotor, el buzzer, los LEDs y la placa ESP32/Arduino. También se ha logrado programar toda la lógica de funcionamiento del sistema, permitiendo validar contraseñas, abrir y cerrar la caja fuerte y mostrar mensajes informativos al usuario.

Uno de los mayores logros ha sido el desarrollo de la aplicación móvil mediante MIT App Inventor. Aunque era la primera vez que trabajábamos con este tipo de herramientas, y a pesar de todos los problemas que hemos tenido con la app, conseguimos crear una aplicación funcional capaz de conectarse por Bluetooth con la placa y controlar la caja fuerte de manera remota. Durante este proceso aprendimos conceptos básicos de programación visual, comunicación Bluetooth y organización lógica de bloques.

Además, también hemos aprendido a solucionar errores tanto de hardware como de software, realizando múltiples pruebas hasta conseguir que todos los componentes funcionaran de forma coordinada. Todo esto nos ha permitido adquirir experiencia práctica y conocimientos muy útiles relacionados con la electrónica, la programación y la automatización.

13.FINAL

Para finalizar, queremos destacar que la realización de este proyecto ha sido una experiencia muy importante y enriquecedora para nosotros, tanto a nivel académico como personal. Durante el desarrollo de Keepy no solo hemos aplicado muchos de los conocimientos aprendidos durante el ciclo formativo, sino que también hemos aprendido nuevas tecnologías y formas de trabajo que al principio parecían muy complicadas.

A lo largo del proyecto hemos tenido que enfrentarnos a diferentes problemas técnicos, errores de programación y dificultades de organización, pero gracias al esfuerzo, la constancia y el trabajo en equipo hemos conseguido superarlos y sacar adelante una caja fuerte electrónica completamente funcional.

Este proyecto nos ha permitido comprender mejor cómo se desarrolla un sistema tecnológico real, desde la idea inicial hasta el montaje físico y la programación final. Además, nos ha ayudado a mejorar habilidades como la investigación, la resolución de problemas y la capacidad de adaptación ante los imprevistos.

Finalmente, queremos agradecer la ayuda y apoyo recibido durante el proyecto, especialmente al profesor Jaime Morcillo, por orientarnos y proporcionarnos materiales y consejos que han sido de gran utilidad para continuar avanzando.

Nos sentimos satisfechos con el resultado obtenido y creemos que Keepy representa todo el esfuerzo y aprendizaje adquirido durante esta etapa del ciclo.

IMÁGENES FINALES



