

Sistema de control de acceso a un almacén

Manuel Erices, Braulio Jeldres y Diego Uribe

Instituto de Electricidad y Electrónica

Universidad Austral de Chile

Resumen—Este documento detalla el desarrollo e implementación de un sistema de control de acceso automatizado que utiliza tarjetas RFID, sensores ultrasónicos, y comunicación entre microcontroladores (Arduino Uno y Arduino Mega) y un sistema Python con interfaz gráfica. El proyecto incluye registro automático de eventos en una base de datos local y generación de alertas para accesos no autorizados.

I. INTRODUCCIÓN

El sistema de control de acceso a un almacén tiene como objetivo principal automatizar la gestión de entradas y salidas, garantizando un registro preciso y en tiempo real de accesos autorizados y no autorizados. Estos datos se almacenan localmente en archivos en formato .csv, evitando el uso de soluciones basadas en la nube debido a ciertas limitaciones asociadas, como la dependencia de una conexión a internet, lo cual se discutirá más adelante en este informe.

Este documento detalla el funcionamiento del sistema, incluyendo las iteraciones y ajustes realizados durante su desarrollo hasta alcanzar la versión final. Además, se presentan diagramas de flujo que ilustran las operaciones del sistema, junto con los resultados obtenidos y propuestas de trabajos futuros para optimizar y mejorar aún más la funcionalidad del sistema.

II. DESCRIPCIÓN DEL PROYECTO

El sistema se compone de:

- **Arduino Uno:** Registra nuevas tarjetas RFID y las asocia a usuarios en la base de datos.
- **Arduino Mega:** Gestiona los lectores RFID de entrada y salida, el sensor ultrasónico, y el servomotor para el acceso.
- **Python:** Proporciona una interfaz gráfica de usuario (GUI) para el registro y consulta de usuarios, y gestiona la base de datos en formato CSV.

II-A. Objetivos

- Registrar usuarios mediante una GUI en Python.
- Automatizar el acceso utilizando tarjetas RFID.
- Registrar eventos en una base de datos local.

III. METODOLOGÍA Y DESARROLLO

En esta sección se darán a conocer los pasos que se siguieron para completar el proyecto **sistema de control de acceso a un almacén**. No dando detalles tan específicos, pero sí contando los pasos que se siguieron o, mejor dicho, las iteraciones que se hicieron para llevar a cabo el proyecto.

III-A. Primera Propuesta

En una primera instancia, se propuso un proyecto titulado “Sistema de alarma y control de acceso con ESP32”. Esta propuesta inicial planteaba la implementación de tres entradas y una salida, utilizando un sensor ultrasónico, un lector RFID y un módulo sensor de efecto Hall, los cuales serían gestionados por una ESP32 para la toma de decisiones. Según los datos obtenidos de estos sensores, la salida consistiría en la activación de un pestillo electrónico solenoide para el control de acceso.

Tras un análisis detallado de esta propuesta y considerando los componentes disponibles, se decidió realizar modificaciones a la idea original, lo que dio lugar al desarrollo de una segunda propuesta.

III-B. Segunda Propuesta

En la segunda propuesta, se conservaron las tres entradas originales del sistema, pero se añadieron más salidas para crear un sistema más robusto y adecuado para su implementación en un almacén. Las salidas incluían un pestillo electrónico solenoide, un servomotor SG90, una pantalla LCD de 16x2 líneas con comunicación I2C y una alarma diseñada para activarse en caso de que se intentara acceder con una tarjeta no registrada en la base de datos alojada en la nube.

El funcionamiento propuesto consistía en que, al pasar una tarjeta por el lector RFID, la ESP32 enviaría el UID a la base de datos en la nube para su validación. Si la tarjeta estaba registrada, la base de datos respondería con un mensaje de “acceso concedido”, lo que activaría el pestillo electrónico y el servomotor para abrir automáticamente la puerta, además de mostrar en la pantalla LCD el mensaje “Acceso concedido”. En caso de que se utilizara una tarjeta no registrada, el sistema activaría la alarma, alertando al encargado del almacén o al guardia responsable, para permitir una respuesta inmediata y resguardar las instalaciones.

A continuación, se presenta el diagrama de flujo correspondiente al funcionamiento propuesto para este sistema.

Tras presentar la segunda propuesta, se desarrolló un prototipo básico para ilustrar y explicar el sistema de forma más visual, lo que permitió recibir retroalimentación. Una de las observaciones más relevantes fue que el uso de una base de datos alojada en la nube en sistemas embebidos puede ser una solución ambivalente. Si bien ofrece accesibilidad y escalabilidad, depende completamente de una conexión estable a internet (WiFi) para su correcto funcionamiento. Esto introduce una limitación crítica, ya que la implementación del sistema quedaría condicionada a la calidad de la señal WiFi.

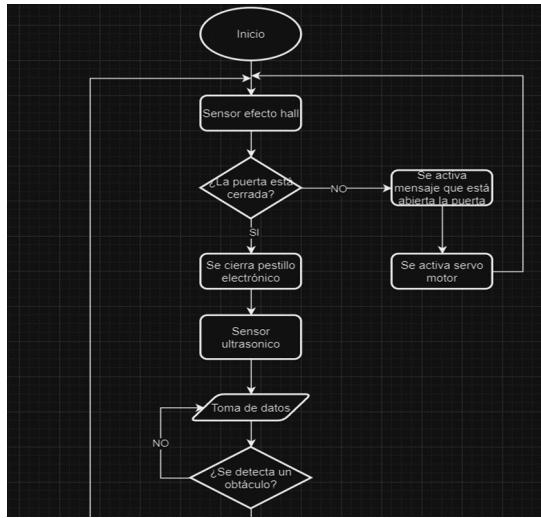


Figura 1: Primera parte del diagrama de flujo.

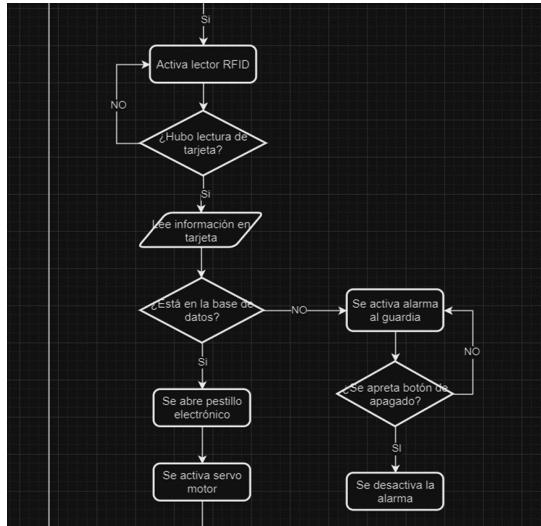


Figura 2: Segunda parte del diagrama de flujo.

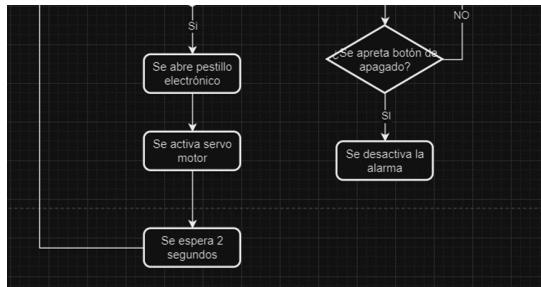


Figura 3: Tercera parte y final del diagrama de flujo.

en el lugar, así como a la ausencia de interrupciones en la conectividad.

En consecuencia, se decidió optar por una base de datos local, almacenando los datos directamente en un computador conectado al sistema. Si bien esta solución tiene desventajas, como un límite de almacenamiento y el riesgo potencial de pérdida de datos, presenta la ventaja de depender únicamente del suministro eléctrico, lo que la hace más adecuada para las

necesidades del proyecto y el entorno específico en el que se planea implementar.

IV. PROYECTO FINALIZADO

Se analizaron las retroalimentaciones obtenidas en las etapas previas del proyecto, y se identificaron algunas limitaciones importantes en el diseño. Entre estas, se destacó que el sistema, tal como estaba planteado, solo permitía la gestión de entradas al almacén, sin incluir un mecanismo para registrar salidas. En respuesta a esta observación, se decidió cambiar el enfoque del proyecto para abordar estas deficiencias, lo que dio lugar al desarrollo de una versión final estructurada en tres fases principales:

IV-A. Diseño Inicial

En esta etapa, se identificaron los componentes clave del sistema y se diseñó una arquitectura modular que permitiera una integración eficiente de hardware y software. Para el almacenamiento de los datos, se optó por el uso de archivos en formato CSV, organizados en hojas separadas para *Usuarios*, *Eventos* y *Ultrasonido*, con el objetivo de mantener la información estructurada y accesible de forma local.

IV-B. Implementación

- Hardware:** Dos lectores RFID se configuraron para operar de manera simultánea en el Arduino Mega, gestionando los conflictos del bus SPI mediante el uso de pines SS dedicados para cada lector, como se muestra en la figura 4. Además, se integraron un servomotor y un sensor ultrasónico para gestionar la apertura de la puerta y detectar proximidad. Por otro lado, un tercer lector RFID fue instalado en el Arduino Uno con la finalidad de registrar tarjetas nuevas. Este lector permite, mediante una interfaz gráfica de usuario (GUI) desarrollada en Python, registrar automáticamente las nuevas tarjetas en la base de datos, habilitando su posterior acceso a través de los lectores de entrada y salida.

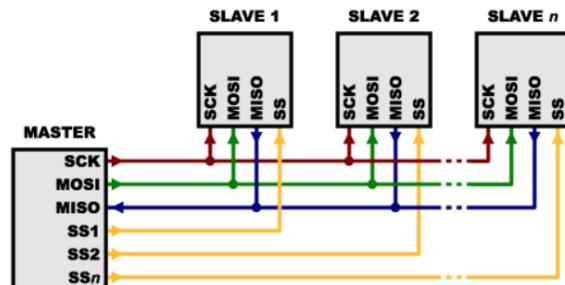


Figura 4: Protocolo SPI [1].

- Software:** Se desarrolló un programa en Python en conjunto con Arduino IDE para gestionar la comunicación serial entre los Arduinos y el computador. Este programa valida los accesos mediante la base de datos local y registra eventos en tiempo real, permitiendo una interacción fluida y precisa entre los componentes del sistema.

IV-C. Solución a Problemas

Dentro de la nueva solución se tuvieron algunos problemas, donde en esto se destacan los siguientes errores y soluciones respectivamente:

- Conflictos en la comunicación serial: Se implementó limpieza de buffers para evitar datos residuales.
- Gestión de estados: Se mejoró el control de permisos para tarjetas autorizadas en entradas y salidas consecutivas.

V. FUNCIONAMIENTO DEL SISTEMA

El flujo de trabajo se divide en tres procesos principales:

V-A. Registro de Nuevas Tarjetas

El Arduino Uno detecta el UID y lo envía a Python. La GUI solicita los datos del usuario y los almacena en la base de datos local.

V-B. Validación de Accesos

El Arduino Mega verifica los UID en los lectores de entrada y salida, y consulta a Python si tienen permiso. Si está autorizado, el servomotor abre la puerta.

VI. RESULTADOS Y VISUALIZACIÓN

Antes de presentar resultados, vamos a visualizar los diagramas de flujo con los que se construyó el sistema.

VI-A. Diagramas de flujo

El primer diagrama de flujo es con respecto al sistema de Arduino Uno (figura 5).

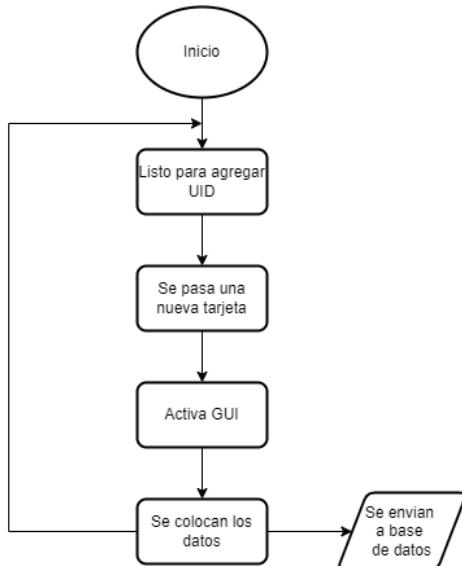


Figura 5: Diagrama de flujo para Arduino Uno.

A continuación, se presentan los diagramas de flujo correspondientes al funcionamiento del Arduino Mega. Para facilitar su comprensión, el flujo se ha dividido en dos partes principales. La primera corresponde al manejo del sensor ultrasónico, que se encarga de registrar datos sobre

la proximidad al lector RFID de entrada. Este sensor registra eventos detallados con fecha, hora, minuto y segundo cada vez que detecta que alguien se aproxima al área, lo que contribuye al seguimiento de actividad en el almacén.

Por otro lado, la segunda parte abarca los dos lectores RFID restantes, que operan en conjunto para gestionar los accesos de entrada y salida. Estos lectores mantienen una comunicación constante con el programa en Python y la base de datos local, validando tarjetas y registrando eventos en tiempo real.

En la figura 6, se muestra el diagrama de flujo correspondiente al funcionamiento del sensor ultrasónico.

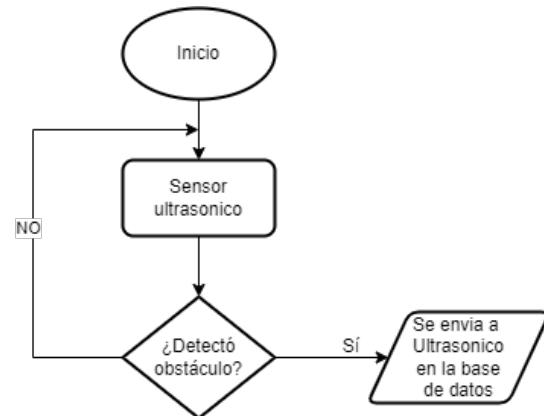


Figura 6: Diagrama de flujo para sensor ultrasónico en Arduino Mega.

Para los lectores RFID, tanto de entrada como de salida, se presentan los diagramas de flujo en la figura 7 y la figura 8. Cabe destacar que ambos sistemas funcionan de manera similar, compartiendo el mismo flujo general de operación. Sin embargo, se elaboraron diagramas de flujo independientes para cada lector con el fin de resaltar las diferencias específicas en su implementación, que son mínimas pero importantes para la claridad del sistema. Estos detalles permiten identificar cómo el sistema gestiona las operaciones de validación de acceso y registro en cada caso.

VI-B. Base de Datos

La base de datos está estructurada en tres hojas:

- **Usuarios:** Contiene UID, nombre y permisos.
- **Eventos:** Registra UID, fecha, tipo (entrada/salida) y resultados.
- **Ultrasonico:** Registra todos los movimientos cercano al lector RFID en entrada.

VI-C. Resultados

Para ilustrar la estructura y funcionalidad del sistema, se incluyen imágenes que muestran cómo está conectado cada componente y cómo interactúan entre sí, junto con ejemplos del sistema en funcionamiento.

En la figura 9, se presenta una vista general del hardware utilizado en el proyecto, que incluye todos los componentes integrados para la finalización del sistema. Entre estos se destacan los lectores RFID, el sensor ultrasónico y una caja equipada con un servomotor que simula la puerta de acceso

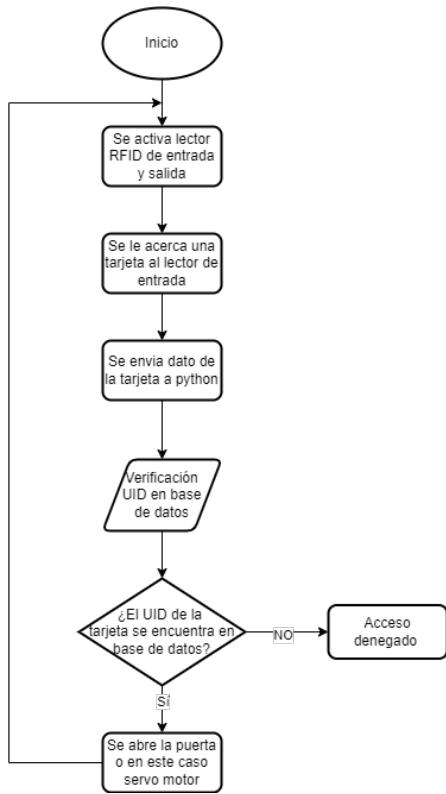


Figura 7: Diagrama de flujo para lector RFID de entrada en Arduino Mega.

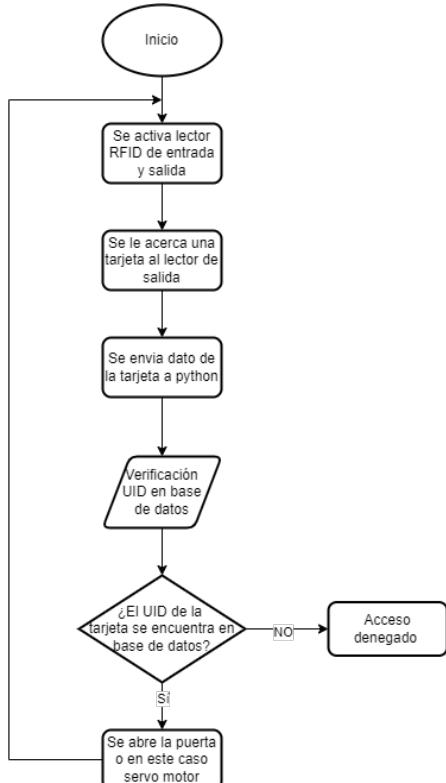


Figura 8: Diagrama de flujo para lector RFID de salida en Arduino Mega.

al almacén. Esta configuración permite visualizar claramente cómo los elementos trabajan en conjunto para garantizar el control de acceso de manera eficiente.

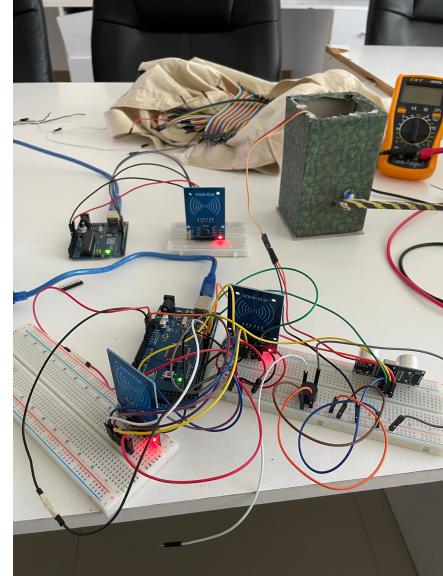


Figura 9: Hardware completo.

El sistema fue sometido a pruebas funcionales, registrando los resultados para evaluar su desempeño. Como se observa en la figura 10, el sistema opera correctamente, activando la barrera que simula la puerta del almacén al detectar una tarjeta autorizada. Además, en la figura 14 se evidencia cómo los eventos se registran en tiempo real, almacenándose de manera ordenada en la base de datos *eventos*. Esto confirma que el sistema cumple con su objetivo de registrar y gestionar de forma eficiente las interacciones dentro del sistema de control de acceso.

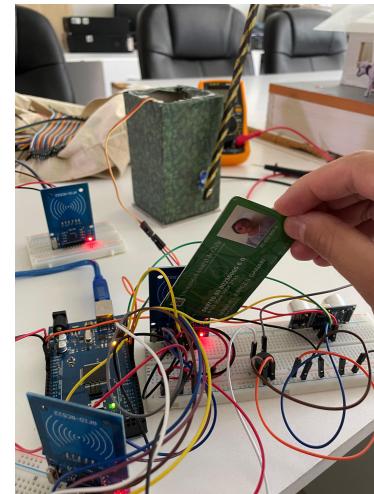


Figura 10: Prueba para lector de entrada.

De manera similar, el funcionamiento del sistema se replicó para el lector de salida, como se aprecia en la figura 11. En esta prueba, se observa cómo una tarjeta es pasada por el lector de salida, activando la barrera simulada y permitiendo el paso de las personas que desean abandonar el almacén. Al igual

que en el caso del lector de entrada, los eventos generados por el lector de salida se registran en tiempo real, tal como se muestra en la figura 14, dejando constancia de que la tarjeta fue utilizada para realizar una salida.

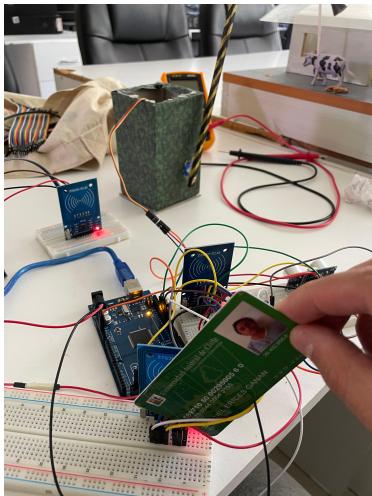


Figura 11: Prueba para el lector de salida.

Finalmente, se presenta el funcionamiento del sistema basado en el Arduino Uno, el cual desempeña un rol fundamental en el registro de nuevas tarjetas RFID. Como se muestra en la figura 12, al intentar registrar una tarjeta que no está previamente almacenada en la base de datos, el sistema activa una interfaz gráfica de usuario. En la figura 13, se observa cómo esta ventana muestra el UID de la tarjeta y solicita el nombre del usuario asociado. Una vez ingresada la información, los datos se guardan en la base de datos *usuarios*, permitiendo que la nueva tarjeta sea reconocida por los lectores de entrada y salida en futuras operaciones.

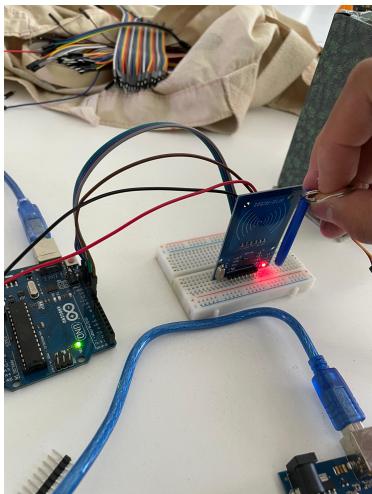


Figura 12: Registrando una nueva tarjeta.

VII. DIAGRAMA DE CONEXIONES

- Circuito de Registro:** en la figura 15 se observa la conexión entre el RFID para registrar a los usuarios y el arduino UNO en donde SDA, SCK, MOSI, MISO y



Figura 13: Registro de usuario.



Figura 14: Eventos que ocurren a tiempo real.

RST se conectan a los pines D10, D13, D11, D12 y D9 respectivamente utilizando el protocolo SPI.

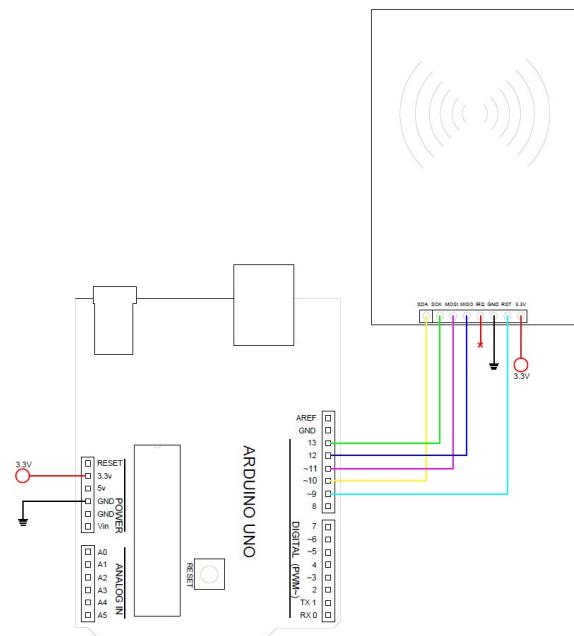


Figura 15: Circuito de registro.

- Circuito de identificación y actuación:** en la figura 16 existen 2 RFID en donde se tienen pines comunes para SPI en donde SCK, MOSI y MISO se conectan a D52,

D51 y D50 respectivamente, para el RFID de entrada se utiliza SDA y RST conectados a D7 y D8 respectivamente mientras que para el de salida se conectan a los pines D10 y D9. Para la apertura de la puerta se controla el servomotor mediante el pin D4 mientras que para el sensor ultrasónico se utiliza D5 y D6 conectados a TRIG y ECHO respectivamente. Se debe considerar que los RFID utilizan 3.3V para ser alimentados mientras que para el servomotor y el sensor ultrasónico se utiliza 5V.

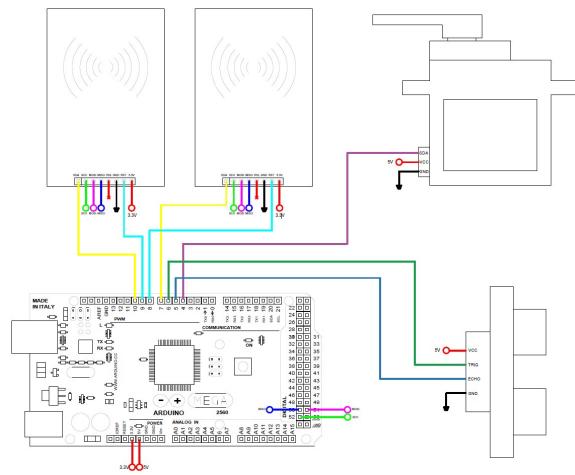


Figura 16: circuito de identificación y actuación.

VIII. TRABAJOS FUTUROS

Dado que los sistemas embebidos y de control de acceso siempre pueden ser optimizados y adaptados a nuevas necesidades, se proponen las siguientes mejoras e incorporaciones para el sistema desarrollado:

- **Gestión de eventos no autorizados:** Incorporar un sistema de alarma sonora, como un buzzer o sirena, que se active inmediatamente cuando una tarjeta no autorizada intente acceder. Además, registrar estos intentos en la base de datos para proporcionar un historial detallado de accesos no autorizados y permitir un análisis de seguridad más exhaustivo.
- **Optimización del manejo de datos:** Mejorar la eficiencia de la comunicación serial entre los sistemas embebidos (Arduino Uno y Arduino Mega) y Python, para minimizar los problemas de latencia y la acumulación de datos residuales en los buffers. Actualmente, en algunos casos, el sistema requiere de dos intentos con la tarjeta para registrar correctamente un cambio de estado, por lo que se sugiere implementar técnicas de control de flujo y validación de datos más robustas.
- **Integración con puertas reales:** Reemplazar el servomotor utilizado en las pruebas actuales por un pestillo electrónico o un actuador de puerta automático. Esto permitiría validar el sistema en un entorno más realista, como almacenes o edificios, mejorando la aplicabilidad y adaptabilidad del proyecto en implementaciones reales.

- **Autenticación biométrica:** Implementar sistemas de autenticación biométrica, como lectores de huellas dactilares o escáneres de retina, para complementar o sustituir el uso de tarjetas RFID. Esto no solo reduciría el riesgo de pérdida o uso indebido de tarjetas, sino que también aumentaría la seguridad al asociar directamente la identidad biométrica del usuario con el acceso.

IX. CONCLUSIONES

El sistema automatizado desarrollado cumplió con los objetivos planteados, destacando especialmente en el registro automatizado de usuarios. A diferencia de las propuestas iniciales, donde era necesario detener todo el sistema para registrar un nuevo usuario, la última iteración automatizada permite registrar y validar tarjetas de forma simultánea con el funcionamiento del control de acceso. Esto no solo mejora significativamente la eficiencia del sistema, sino que también incrementa su seguridad al evitar interrupciones operativas.

Se presentaron los diagramas de flujo que ilustran el funcionamiento interno del sistema, proporcionando una visión clara y detallada de sus componentes y procesos. Además, se explicaron las estructuras de datos utilizadas para registrar y almacenar la información generada, y se mostraron los resultados obtenidos a través de diversas pruebas, confirmando la efectividad del sistema propuesto.

La arquitectura modular del sistema facilita futuras ampliaciones y mejoras, como la implementación de autenticación biométrica, mencionada en la sección de trabajos futuros. Estas mejoras permitirán optimizar aún más el sistema y adaptarlo a una mayor variedad de entornos y aplicaciones, reforzando su robustez y flexibilidad.

REFERENCIAS

- [1] Prometec, “El bus spi,” 2023. <https://www.prometec.net/bus-spi/>.