UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE TELECOMUNICACIONES

INFOREM FINAL DEL TRABAJO DE INTEGRACIÓN CURRICULAR,

PROYECTO DE INVESTIGACIÓN

TEMA:

Implementación de métodos avanzados de autenticación en cerraduras electromecánicas de bajo costo para entornos residenciales.

Trabajo de titulación previo a la obtención del título de Ingeniería en Telecomunicaciones

Línea de investigación: Desarrollo, aplicación de software y cyber security (seguridad cibernética)

AUTOR:

Jessica Aracely Ruano Benavides

DIRECTOR:

MSc. Luis Edilberto Suárez Zambrano

Ibarra, Ecuador 2025

UNIVERSIDAD TÉCNICA DEL NORTE
BIBLIOTECA UNIVERSITARIA

IDENTIFICACIÓN DE LA OBRA

La Universidad Técnica del Norte dentro del proyecto Repositorio Digital Institucional, determinó la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD	1004636179		
APELLIDOS Y NOMBRES	Ruano Benavides Jessica Aracely		
DIRECCIÓN	San Antonio- San Agustín Ramon Teanga		
EMAIL	jaruanob@utn.edu.ec - jaruanob23@gmail.com		
TELÉFONO FIJO		**TELÉFONO MÓVIL:**	0969217162

DATOS DE LA OBRA	
TÍTULO	Implementación de métodos avanzados de autenticación en cerraduras electromecánicas de bajo costo para entornos residenciales.
AUTOR	Ruano Benavides Jessica Aracely
FECHA:	15/07/2015
PROGRAMA	[x] **PREGRADO** [ ] **POSGRADO**
TITULO	Ingeniera en Telecomunicaciones
DIRECTOR	MSc. Suárez Zambrano Luis Edilberto

AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, Ruano Benavides Jessica Aracely, con cédula de identidad Nro. 1004636179, en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de integración curricular descrito anteriormente, hago entrega del ejemplar respectivo en formato digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión; en concordancia con la Ley de Educación Superior Artículo 144.

Ibarra, a los 15 días del mes de julio de 2025

EL AUTOR:

Ruano Benavides Jessica Aracely

CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 15 días del mes de julio de 2025.

EL AUTOR:

Ruano Benavides Jessica Aracely

CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Ibarra, 15 de julio de 2025

Msc. Luis Edilberto Suárez Zambrano

DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:

Haber revisado el presente informe final del trabajo de Integración Curricular, el mismo que se ajusta a las normas vigentes de la Universidad Técnica del Norte; en consecuencia, autorizo su presentación para los fines legales pertinentes.

MSc. Luis Edilberto Suárez Zambrano

C.C.: 1002304291

APROBACIÓN DEL COMITÉ CALIFICADOR

El comité calificador del trabajo de Integración Curricular Diseño de un "Implementación de métodos avanzados de autenticación en cerraduras electromecánicas de bajo costo para entornos residenciales", elaborado por Ruano Benavides Jessica Aracely, previo a la obtención del título de Ingeniería en Telecomunicaciones, aprueba el presente informe de investigación en nombre de la Universidad Técnica del Norte.

MSc. Luis Edilberto Suárez Zambrano

C.C.: 1002304291

MSc. Carlos Alberto Vásquez Ayala

C.C.:1002424982

DEDICATORIA

Dedicada a todas las personas que tienen cerraduras eléctricas.

Ruano Benavides Jessica Aracely

AGRADECIMIENTO

Agradezco a mis padres Carmen y Zoilo por apoyarme en el transcurso de mis estudios. Agradezco a mi Ray y Monsita, a mi abuelita Charito, mi tía Rosita y mi tío Rami, por siempre ser un apoyo, son la mejor familia.

Christopher la vida me dio la oportunidad de conocerte precisamente en esta epata de estudiante, quiero decirte que te admiro mucho, eres amable, paciente y muy inteligente. Gracias por apoyarme a cumplir mis metas.

Expreso mi más sincero agradecimiento a mi director Ing. Luis Suárez, gracias por confiar en mí en todo momento, gracias por su apoyo y orientación durante el desarrollo de este trabajo de titulación. También agradezco a mi asesor Ing. Carlos Vásquez por su compromiso de enseñarnos a ser buenos profesionales, de principios y valores.

Ruano Benavides Jessica Aracely

RESUMEN

Este proyecto tiene como objetivo la implementación de métodos avanzados de autenticación en cerraduras electromecánicas con el fin de fortalecer la seguridad en entornos residenciales, garantizando una mejora a este tipo de cerraduras tradicionales. Inicialmente la investigación se enfocará en el análisis de las tecnologías de autenticación y los mecanismos de acceso más avanzados, evaluando sus vulnerabilidades y limitaciones con el propósito de establecer una base teórica sólida. Con base en estos hallazgos, se desarrollará un sistema de seguridad basado en una llave electrónica programable que incorporará múltiples métodos de autenticación, incluyendo contraseñas numéricas, tecnología RFID y códigos de acceso temporales.

El proyecto seguirá la metodología en cascada, lo que permitirá un desarrollo estructurado y secuencial a través de distintas fases: requerimientos, diseño, implementación y pruebas. En la primera fase se determinan las exigencias del sistema a nivel de requerimientos operacionales, de usuarios, software y hardware. Durante la fase de implementación, se integrará el sistema en un prototipo funcional que combinará tanto componentes de hardware como de software. Además, se desarrollará una aplicación móvil que facilitará la gestión y supervisión del acceso en tiempo real de manera remota. En la etapa final, se llevarán a cabo las pruebas de cada método de autenticación para validar la funcionalidad seguridad y usabilidad del sistema, garantizando el cumplimiento de los requisitos establecidos, con el fin de ofrecer una mejora significativa en comparación con los sistemas de acceso residenciales convencionales.

Palabras clave: llave de circuito programable, cerradura inteligente, hash, salt, aplicación, mecanismos de acceso, tecnologías de autenticación.

ABSTRACT

This project aims to implement advanced authentication methods in electromechanical locks to strengthen security in residential environments, ensuring an improvement over traditional lock types. Initially, the research will focus on analyzing authentication technologies and the most advanced access mechanisms, evaluating their vulnerabilities and limitations to establish a solid theoretical foundation. Based on these findings, a security system will be developed based on a programmable electronic key that will incorporate multiple authentication methods, including numeric passwords, RFID technology, and temporary access codes.

The project will follow the waterfall methodology, allowing for a structured and sequential development through different phases: requirements, design, implementation, and testing. In the first phase, the system's specifications are determined at the level of operational, user, software, and hardware requirements. During the implementation phase, the system will be integrated into a functional prototype that combines both hardware and software components. Additionally, a mobile application will be developed to facilitate the remote management and supervision of access in real-time. In the final stage, tests of each authentication method will be carried out to validate the system's functionality, security, and usability, ensuring compliance with the established requirements, in order to offer a significant improvement compared to conventional residential access systems.

Keywords: programmable circuit key, smart lock, hash, salt, application, access mechanisms, authentication technologies.

Tabla de contenidos

Tabla de figuras

Contenido de tablas

1 CAPÍTULO I: ANTECEDENTES

1.1 Tema

IMPLEMENTACIÓN DE MÉTODOS AVANZADOS DE AUTENTICACIÓN EN CERRADURAS ELECTROMECÁNICAS DE BAJO COSTO PARA ENTORNOS RESIDENCIALES.

1.2 Problema

Según la información digital proporcionada por el diario El Comercio (2023) afirma que los ciudadanos han comenzado a priorizar la búsqueda de inmuebles que ofrezcan mayores niveles de protección. Además, la inseguridad se ha convertido en un factor determinante a la hora de elegir una vivienda en Ecuador. Según una encuesta inmobiliaria realizada en mayo de 2023, el 97% de los encuestados anteponen la seguridad de las viviendas a cualquier otra ventaja, lo que evidencia la importancia crucial de este aspecto en las decisiones de compra y alquiler de propiedades (El Comercio, 2023).

Si bien es cierto que la inseguridad es una preocupación creciente, aún se sigue utilizando en muchos casos cerraduras tradicionales, las cuales presentan vulnerabilidades como la pérdida de llaves, la falsificación y la apertura forzada. No obstante, existen cerraduras más avanzadas que ofrecen mejoras significativas en seguridad. Entre estas opciones más seguras se encuentran las cerraduras electrónicas, que utilizan mecanismos para dificultar el acceso no autorizado. Por otro lado, las cerraduras inteligentes permiten el control remoto a través de aplicaciones móviles y pueden integrarse con otros dispositivos de seguridad del hogar, como cámaras y alarmas, proporcionando una protección integral.

La evolución de la tecnología ha permitido el desarrollo de diversos dispositivos y sistemas destinados a mejorar la seguridad en los entornos residenciales. Entre estos, los sistemas de accesos han sido una herramienta clave en la comunicación entre los visitantes

y los residentes, permitiendo la verificación y el control de acceso a las propiedades. Sin embargo, los accesos convencionales se basan principalmente ya sea en la verificación visual, auditiva o incorporado los dos, lo que deja margen para suplantaciones de identidad y accesos no autorizados. A pesar de las ventajas que ofrecen estas tecnologías de seguridad avanzadas, aún no son ampliamente adoptadas debido a su costo, el desconocimiento de los usuarios y la resistencia al cambio. Sin embargo, invertir en estas opciones tecnológicas podría mejorar significativamente la seguridad de las residencias, brindando mayor comodidad y tranquilidad a los habitantes.

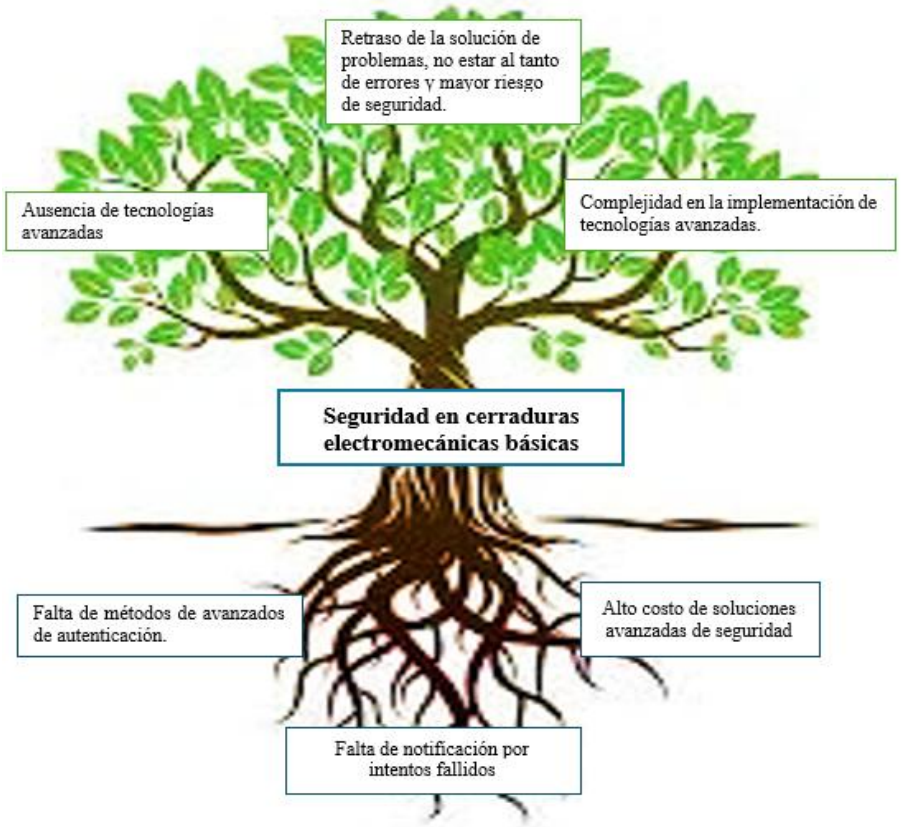
El incremento en la sofisticación de los métodos utilizados por los delincuentes para acceder a propiedades privadas subraya la necesidad de mejorar la seguridad de los sistemas de acceso. Las tecnologías avanzadas de autenticación, como las contraseñas numéricas y RFID, han demostrado ser efectivas en diversos contextos (Martínez, 2023). Sin embargo, su implementación en sistemas de acceso no ha sido ampliamente explorada, dejando una brecha significativa en la seguridad de estos dispositivos.

Por lo tanto, este proyecto se propone adaptar métodos de autenticación robustos a sistemas de accesos, con el objetivo de ofrecer una mayor protección contra accesos no autorizados, garantizar la integridad y confidencialidad de las comunicaciones y ser escalable es decir puede ser aplicado para una residencia o para un conjunto residencial. Esta innovación no solo abordará las vulnerabilidades actuales de los sistemas de acceso, sino que también proporcionará una solución económica y accesible, respondiendo a las demandas de seguridad de los ciudadanos en entornos residenciales. En la **Figura 1** se

presenta un árbol de problemas que ilustra las principales causas y efectos de la inseguridad en los sistemas de acceso residenciales.

Figura 1

Árbol de problemas.



1.3 Objetivos

1.3.1 Objetivo general

Implementar métodos avanzados de autenticación en cerradura electromecánica para entornos residenciales, con el fin de mejorar significativamente la seguridad a bajo costo.

1.3.2 Objetivos específicos

- Elaborar el estado del arte de las tecnologías de autenticación y mecanismos de acceso residencial, así como diseño de circuitos programables.

- Diseñar un método de autenticación basado en llave de circuito programable que pueda ser incorporado de manera eficiente y económica.
- Implementar las tecnologías de autenticación un sistema de acceso para residencias.
- Realizar las pruebas necesarias para validar la efectividad del sistema desarrollado.

1.4 Alcance

El presente trabajo de titulación tiene como objetivo incorporar nuevos sistemas de seguridad mediante la autenticación en mecanismos básicos residenciales. El propósito es mejorar significativamente la seguridad y reducir costos en comparación con alternativas más costosas haciendo estas tecnologías más accesibles para un mayor número de usuarios, tanto de residencias individuales como conjuntos (condominios, edificios, hoteles, etc.). Además, se pretende aprovechar las nuevas tecnologías que permiten a los propietarios gestionar y monitorear el acceso desde cualquier lugar.

Por otra parte, el proyecto se fundamentará en la metodología waterfall, lo que significa que seguirá un enfoque secuencial-lineal, en el desarrollo y la ejecución del proyecto (DIGITAL TALENT AGENCY, 2018). Se dividirá en diferentes fases siguiendo el orden de los objetivos específicos los cuales se llevarán a cabo de manera ordenada y progresiva.

En la primera fase, se llevará a cabo una investigación de bases bibliográficas y del estado del arte de las tecnologías de autenticación y mecanismos de acceso, así como diseño de circuitos programables. Esta investigación incluirá la revisión de literatura académica, artículos técnicos y patentes, con el fin de identificar las tecnologías más avanzadas y relevantes. Asimismo, se analizarán las vulnerabilidades y limitaciones de los sistemas de acceso residenciales actuales, lo que permitirá establecer una base teórica sólida y actualizar el conocimiento técnico necesario para el desarrollo del proyecto.

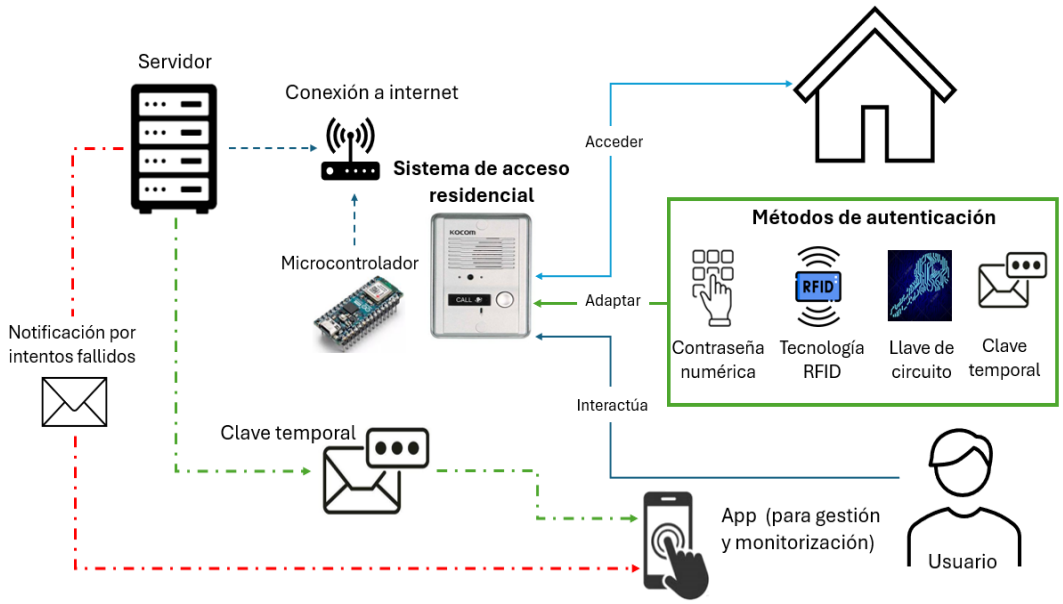
En la fase de diseño, se desarrollará un mecanismo de acceso mediante circuito programable (llave electrónica programable). Este proceso incluirá la elaboración del diagrama de circuito, el diseño detallado de la placa base y la especificación de los componentes de hardware y software necesarios para la implementación.

En la fase de implementación, se integrarán las tecnologías de autenticación como contraseña numérica, RFID, clave temporal, así como la llave electrónica programable en un sistema de acceso para residencias. Se desarrollará el código necesario para la implementación de estos métodos y se ensamblarán los componentes de hardware en un prototipo funcional. Además, se incluirá la integración de una aplicación móvil que permita al usuario gestionar y monitorear el acceso, la aplicación tendrá una interfaz intuitiva y de fácil manejo. Se llevarán a cabo pruebas unitarias para asegurar que cada componente funcione correctamente antes de la integración completa del sistema.

En la siguiente **Figura 2** se muestra un diagrama de un sistema de acceso mejorado con tecnologías de autenticación para incrementar la seguridad en el acceso a un hogar.

Figura 2

Integración de seguridad en accesos básicos.



Nota. En la **Figura 2** muestra cómo se interconectan los diferentes componentes para proporcionar una solución de seguridad integral para un acceso residencial.

Finalmente, se realizarán pruebas exhaustivas de funcionalidad, seguridad y usabilidad del prototipo desarrollado. Se validará el correcto funcionamiento del sistema, asegurando que cumple con los requerimientos establecidos y ofrece una mejora tangible en la seguridad de los accesos residenciales. Se documentarán todos los procedimientos y resultados de las pruebas, incluyendo cualquier observación o sugerencia para futuras mejoras.

1.5 Justificación

En un mundo cada vez más conectado y globalizado, la seguridad en los espacios habitables, laborales y públicos se ha convertido en una prioridad absoluta. Las cerraduras

tradicionales, o sistemas básicos de seguridad, si bien han cumplido su función durante décadas o años, presentan limitaciones en cuanto a su capacidad para brindar protección efectiva, comodidad y eficiencia (Miño, 2017). Por ello, el desarrollo de mejoras en el sistema de accesos surge como una alternativa innovadora y necesaria para responder y mejorar la seguridad, y además se pretende que sean sistemas no tan costosos.

En este contexto, la creciente tasa de robos y accesos no autorizados en áreas residenciales resalta la urgente necesidad de mejorar los sistemas de seguridad. Implementar métodos avanzados de autenticación para proteger mejor a los residentes y sus propiedades (Vaca, 2021). Sin embargo, los sistemas de seguridad más avanzados y robustos suelen tener un costo elevado, lo que dificulta su adquisición por la mayoría de los propietarios de viviendas. Esto genera una brecha en la protección, ya que solo las personas con mayores recursos pueden acceder a soluciones de seguridad más efectivas.

Por lo tanto, con el avance de la tecnología, los sistemas de seguridad tradicionales deben actualizarse para estar a la par de las nuevas amenazas. La integración de microcontroladores y servidores en sistemas de accesos permite una gestión más eficiente y segura. Por consiguiente, este proyecto busca desarrollar un sistema de autenticación de bajo costo que pueda ser fácilmente integrado en los accesos residenciales convencionales. De esta manera, se podrá mejorar significativamente la seguridad de estos dispositivos sin que los usuarios deban incurrir en gastos excesivos (Duarte, 2013).

Por otra parte, la seguridad en los sistemas de accesos residenciales es un aspecto importante que ha sido pasado por alto. Muchos de estos carecen de mejoras y de incorporación de nuevas tecnologías, lo que los vuelve vulnerables a ataques y accesos no autorizados. Estos sistemas, que en su mayoría se basan en la verificación visual y auditiva sin ningún tipo de autenticación avanzada, pueden ser fácilmente manipulados por personas con intenciones maliciosas (Nossa Jiménez, 2021). Como resultado, la privacidad

y la seguridad de los habitantes de las residencias se ven comprometidas, exponiéndolos a riesgos significativos. Los intrusos pueden interceptar comunicaciones, obtener información sensible o incluso lograr acceso físico a las propiedades, lo que aumenta la preocupación entre los residentes y la necesidad de una solución más robusta y confiable. Esta vulnerabilidad en los accesos convencionales subraya la importancia de implementar tecnologías avanzadas de autenticación que puedan ofrecer una barrera efectiva contra accesos no autorizados y garantizar la integridad y confidencialidad de las comunicaciones.

Asimismo, la incorporación de tecnologías de autenticación permitirá a los propietarios tener un mayor control y monitoreo del acceso a sus viviendas, incluso desde dispositivos móviles. La capacidad de gestionar y monitorear el acceso al hogar de forma remota a través de dispositivos móviles aumenta la conveniencia para los usuarios, permitiéndoles responder rápidamente a intentos de acceso no autorizados desde cualquier lugar (Revelo & Andrade, 2020). Esto les brindará una mayor tranquilidad y les permitirá gestionar de manera más eficiente la seguridad de sus hogares.

Además, implementar tecnologías de autenticación robustas, como PIN, tarjetas RFID o biometría, reduce significativamente el riesgo de accesos no autorizados, incrementando la seguridad de las áreas residenciales (Garcia & Mesa, 2022). Igualmente, la función de notificación de intentos fallidos permite a los usuarios estar siempre informados sobre posibles amenazas a su seguridad, facilitando una respuesta rápida y efectiva ante cualquier intento de intrusión.

Este trabajo de tesis se alinea con el Objetivo de Desarrollo Sostenible (ODS) 11: "Ciudades y comunidades sostenibles", específicamente con la meta 11.1 que busca "Asegurar el acceso de todas las personas a viviendas y servicios básicos adecuados, seguros y asequibles" (Cepal, 2019). Al mejorar la seguridad de los sistemas de acceso

residenciales se contribuye a la creación de comunidades más seguras y resilientes. Además, ayuda a que los sistemas de acceso cumplan con las normativas y estándares de seguridad modernos, asegurando que las soluciones implementadas sean sostenibles y efectivas a largo plazo.

Finalmente, mejorar la seguridad del hogar puede resultar en una reducción de los costos asociados a robos y daños a la propiedad, así como una posible reducción en las primas de seguros, ofreciendo beneficios económicos tangibles a los residentes. La evolución hacia hogares inteligentes requiere que todos los dispositivos, se integren de manera segura en la red del hogar. Este proyecto proporciona una base para futuras integraciones con otros dispositivos IoT, mejorando tanto la funcionalidad como la seguridad del hogar.

2 CAPÍTULO II. MARCO TEÓRICO

El presente capítulo se centra en la revisión del estado del arte sobre la seguridad en oficinas, residencias, departamentos y condominios, así como de tecnologías de autenticación, riesgos y amenazas respecto a la autenticación en sistemas electrónicos. Además, se detalla como las tecnológicas han transformado la protección de datos y el control de accesos mediante las tecnologías de autenticación.

2.1 Seguridad

Según el Consejo Nacional para la Igualdad de Discapacidades (2016) plantea que la seguridad ciudadana:

Es una política de Estado, destinada a fortalecer y modernizar los mecanismos necesarios para garantizar los derechos humanos, en especial el derecho a una vida libre de violencia y criminalidad, la disminución de los niveles de delincuencia, la protección de víctimas y el mejoramiento de la calidad de vida de todos los habitantes del Ecuador.

Por otra parte, la seguridad se puede decir que es la ausencia de peligros o de ataques contra la integridad de un ser humano o pertenencias de este, no es nada más que la naturaleza de vivir sin temor y sin amenazas.

2.1.1 Residencias y hogares

La seguridad es un concepto fundamental que abarca la protección de personas y bienes frente a diversas amenazas. Según Lozano (2016) en la actualidad, la seguridad se ha convertido en prioridad tanto a nivel personal como en el ámbito de la seguridad residencial y de los hogares, mientras que la demanda de servicios de seguridad privada en el sector

residencial es considerable, debido a que la delincuencia común está en constante evolución y se especializa en diversos métodos de robo, como fraudes telefónicos, extorsiones y asaltos a mano armada. Estos actos delictivos no solo ponen en riesgo los bienes materiales de los residentes, sino que también afectan su tranquilidad y sensación de seguridad. Este concepto incluye diferentes aspectos, entre los cuales se destacan la prevención de robos y actos de vandalismo.

Según la Policía Nacional, el 87% de las denuncias por robos a bienes inmuebles corresponden a casas y el 16% a departamentos. Como resultado, en 2022, se registraron un total de 141.986 delitos contra la propiedad a nivel nacional. De esta cifra, 8.112 denuncias (5.7%) fueron clasificadas como robos en domicilios, incluyendo tanto casas como departamentos.

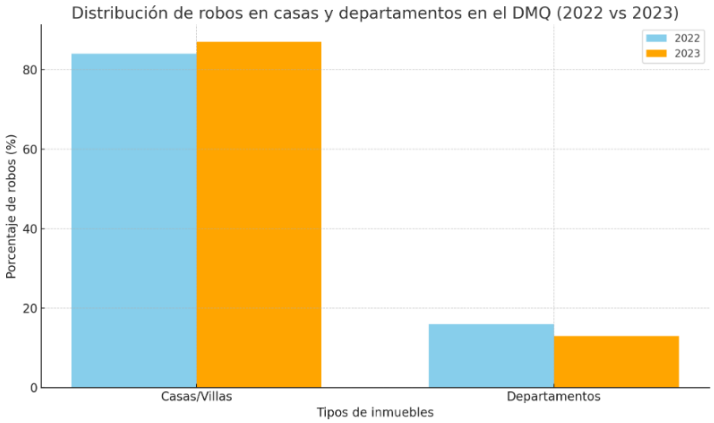
Específicamente en el Distrito Metropolitano de Quito (DMQ), casi nueve de cada diez denuncias indican que el robo ocurrió en una casa o villa. Esto significa que el 84% de los robos a bienes inmuebles en Quito estuvo relacionado con casas o villas, mientras que el 16% correspondió a robos en departamentos.

De igual forma, los datos se mantienen en el año 2023, se registró que el 87% de las denuncias de robos corresponde a casas, mientras que el 13% se refiere a departamentos, condominios.

Además, las modalidades de robo en hogares son variadas. Una de las más recientes en darse a conocer involucra el uso de tecnología avanzada. Según un reportaje de La Hora (2023), se detalló que los delincuentes utilizan drones para mapear zonas, especialmente conjuntos residenciales, con el fin de llevar a cabo sus robos. En la siguiente **Figura 3** se detallan los porcentajes de robos tanto en casas como en departamentos, en los años 2022 y 2023 el porcentaje de robo es más frecuente en casa y villas.

Figura 3

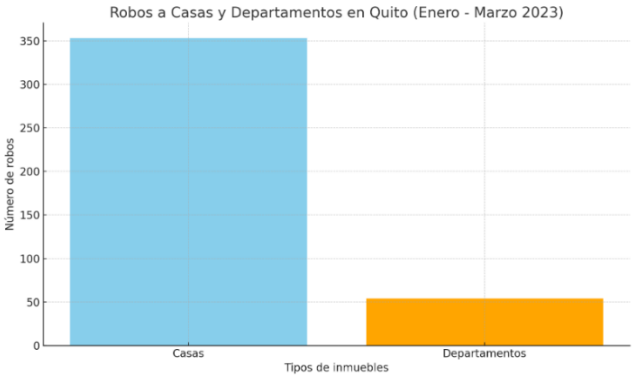
Porcentaje de robos en casas y departamentos.



Según un reportaje de La Hora, entre enero y marzo de 2023, se registraron 400 denuncias por robos a domicilios en Quito. De este total, 353 corresponden a robos en casas, lo que representa una parte significativa, mientras que 54 denuncias se refieren a robos en departamentos. En la **Figura 4** muestra gráficamente estos datos, evidenciando que la mayoría de los robos ocurren en viviendas unifamiliares, lo que subraya la necesidad de implementar medidas de seguridad efectivas.

Figura 4

Robos en casas y departamentos en la ciudad de Quito.



Importancia de la seguridad física

En términos generales, la seguridad física es esencial para las personas, independientemente de si viven en condominios, departamentos u hogares. Sin embargo, en los condominios, la necesidad de empresas de seguridad privada es más pronunciada debido al alto flujo de personas y la mayor complejidad en la gestión de accesos. Estas empresas brindan la tranquilidad de contar con profesionales capacitados que vigilan y están preparados para actuar ante cualquier amenaza.

Para hogares individuales, la contratación de servicios de seguridad privada puede ser costosa y poco práctica. En estos casos, la implementación de sistemas de seguridad completos, como cámaras de vigilancia, alarmas, sensores de movimiento o métodos de autenticación, es una opción más accesible y eficiente.

Es importante destacar que estos sistemas no solo protegen los bienes materiales, sino que también contribuyen significativamente al bienestar emocional y psicológico de las personas, al proporcionar un ambiente de confianza y tranquilidad. Una seguridad bien implementada garantiza que los residentes se sientan seguros en su entorno y que cualquier situación de riesgo se maneje de manera efectiva.

2.1.2 Oficinas

La seguridad en oficinas es un aspecto crucial para garantizar un entorno de trabajo seguro y productivo. Este concepto abarca diversas medidas y estrategias destinadas a proteger tanto a los trabajadores como a los activos de la empresa.

Entre las principales áreas de enfoque se encuentran el control de acceso, que implica la implementación de sistemas de identificación y credenciales para limitar el ingreso a áreas restringidas; según (Pastrana et al., 2010) “la instalación de sistemas automáticos de vigilancia y sistemas de alarma para monitorear y responder a incidentes en tiempo real”; y la capacitación

del personal en protocolos de seguridad, que les permita identificar y reaccionar adecuadamente ante situaciones de riesgo. Por otra parte, la seguridad en oficinas no solo protege a las personas y bienes, sino que también contribuye a fomentar un ambiente de confianza y bienestar.

2.1.3 Medidas de seguridad

Históricamente, las medidas de seguridad han evolucionado significativamente desde tiempos antiguos. En épocas pasadas, la seguridad de los hogares y comunidades se basaba principalmente en métodos físicos y rudimentarios. Las personas utilizaban barreras físicas, como muros, cercas y puertas de madera reforzadas, para protegerse de intrusiones. Además, las cerraduras eran simples, a menudo de metal, y su efectividad dependía de la habilidad de los ladrones para forzarlas.

Actualmente, entre las estrategias de seguridad más efectivas para entornos residenciales, hogares y oficinas se destaca el uso de métodos de autenticación de ingreso, cerraduras electrónicas y sistemas de control de acceso, como tarjetas magnéticas o cerraduras inteligentes. Otra estrategia importante es la instalación de sistemas de vigilancia, que abarcan cámaras de seguridad y alarmas. Estos métodos son fundamentales para garantizar que solo las personas autorizadas tengan acceso a espacios privados. Por otra parte, los dispositivos de vigilancia permiten no solo el monitoreo en tiempo real, sino que también registran eventos que pueden ser esenciales para futuras investigaciones.

Adicionalmente, las alarmas o notificaciones informan a los propietarios y a las autoridades en caso de intrusiones, facilitando así una respuesta rápida ante cualquier eventualidad.

2.1.4 Mecanismos de acceso

Como señala (Mojica Francisco, 2014) los métodos o sistemas de seguridad han sido utilizados por el hombre desde tiempos inmemoriales, prácticamente desde que él hizo su aparición sobre la tierra, el hombre sintió la imperiosa necesidad de defenderse de su entorno que para entonces era muy hostil y no le resultaba muy fácil sobrevivir, empíricamente fue ideando una serie de métodos de seguridad, los cuales fueron evolucionando poco a poco marcando con algunas características muy interesantes las diferentes épocas de la historia.

Los mecanismos de acceso son dispositivos fundamentales para garantizar la seguridad de espacios físicos.

2.1.4.1 Tecnologías tradicionales

En el ámbito de la seguridad, se han desarrollado diversas tecnologías, siendo las cerraduras tradicionales las más comunes. A continuación, se describen algunos de los tipos más representativos:

Tabla 1

Tipos de cerraduras y especificaciones.

Cerraduras tradicionales		
Tipos	**Características**	**Funcionamiento**
Cerraduras multipunto Permiten asegurar una puerta en múltiples puntos a lo largo de su marco, típicamente entre 3 y 5 puntos.	• Aumento de la seguridad gracias a múltiples puntos de fijación. • Ideal para puertas de entrada y zonas con alto riesgo de intrusión. • Puede incluir sistemas de cierre automático.	Al insertar la llave y girarla, varios pestillos se desplazan simultáneamente, bloqueando la puerta en múltiples puntos. Esto dificulta la apertura forzada, ya que requiere que todos los puntos de anclaje sean manipulados.
Cerraduras antibumping Estas cerraduras cuentan con características especiales que impiden que los pines se alineen	• Alta resistencia a técnicas de manipulación. • A menudo cuentan con certificaciones de seguridad.	Estas cerraduras están diseñadas con elementos que impiden que la técnica

de manera adecuada, evitando así su apertura no autorizada.	• Disponibles en diferentes niveles de complejidad y precio.	de bumping[1] tenga éxito. Suelen incorporar pines adicionales o mecanismos de seguridad que solo se alinean con llaves específicas.
Cerraduras magnéticas Este tipo de cerradura es común en sistemas de control de acceso. Se basan en un sistema electromagnético que mantiene la puerta cerrada.	• Alta seguridad sin partes móviles, lo que reduce el desgaste. • Ideal para puertas de vidrio y sistemas de control de acceso. • Requiere una fuente de energía constante para funcionar.	Las cerraduras magnéticas funcionan mediante imanes para mantener la puerta cerrada. Cuando se activa el electroimán, se genera una fuerza magnética que asegura la puerta. Al desactivar el sistema eléctrico, la puerta se puede abrir fácilmente.
Cerraduras de sobreponer Las cerraduras que se sobreponen es decir se instalan sobre la superficie de la puerta, en lugar de estar empotradas en ella.	• Fácil instalación y mantenimiento. • Proporcionan un nivel adicional de seguridad. • Disponibles en diversos diseños estéticos.	Se instalan sobre la superficie de la puerta y operan mediante un mecanismo de palanca o cilindro. Al accionar la llave, se bloquea o desbloquea el pestillo.

2.1.4.2 Tecnologías actuales

Los avances y desarrollos tecnológicos se pueden definir como versiones mejoradas de un sistema que buscan resolver problemas, mejorar condiciones y explorar nuevas modalidades. Además, mediante la innovación constante, las tecnologías modernas procuran optimizar procesos, incrementar la eficiencia y ofrecer soluciones innovadoras que se adapten a las necesidades cambiantes de la sociedad.

En el ámbito de los mecanismos de acceso residenciales, los avances tecnológicos se traducen en la implementación de cerraduras inteligentes, electrónicas, biométricas de doble o triple autenticación, lo cual añade un nivel adicional de seguridad, asegurando que solo los usuarios autorizados puedan acceder, también pueden permitir el control remoto a través de aplicaciones móviles. En consecuencia, estas innovaciones no solo optimizan la seguridad, sino que también ofrecen una experiencia más conveniente y personalizada, reflejando así el

[1] La técnica de bumping implica insertar una llave modificada, llamada bump key, que coloca los pistones de la cerradura en la posición más baja. Al golpear la llave, se separan los pitones de los contrapitones, permitiendo así desbloquear la cerradura.

compromiso de la tecnología por adaptarse a las necesidades de un mundo en constante evolución.

Tabla 2

Tipos de cerraduras modernas.

Cerraduras modernas		
Tipos	**Características**	**Funcionamiento**
Electrónicas Son cerraduras que utilizan componentes electrónicos para controlar el acceso, en lugar de llaves tradicionales.	• Pueden ser operadas a través de un control remoto o aplicación móvil. • Uso de un código numérico para desbloquear. • Pueden almacenar un registro de quién accedió y cuándo.	Generalmente su funcionamiento es mediante un teclado numérico o un sistema de tarjetas RFID. Al ingresar el código correcto o presentar la tarjeta, un motor interno desbloquea la cerradura.
Inteligentes Son cerraduras que se conectan a Internet y pueden ser controladas desde dispositivos móviles, ofreciendo características avanzadas.	• Conectividad Wi-Fi o Bluetooth, para permitir el control a través de una aplicación desde smartphones. • Integración con sistemas de domótica, es decir pueden conectarse con otros dispositivos inteligentes en el hogar. • Permiten crear códigos de acceso temporales para visitantes.	Utilizan tecnología de conectividad para comunicarse con aplicaciones móviles. El usuario puede desbloquear la puerta a distancia, recibir notificaciones de actividad y gestionar el acceso de otros usuarios.
Cerraduras biométricas Utilizan características biométricas (huellas dactilares, reconocimiento facial) para identificar al usuario.	• Dificultad para ser falsificadas. • No requieren llaves ni códigos que recordar. • Desbloqueo inmediato al reconocer la biometría.	Escanean la huella dactilar o el rostro del usuario y comparan con una base de datos interna. Si coincide, la cerradura se desbloquea.
Cerradura de llave electrónica Utilizan una llave electrónica que puede ser una tarjeta o un dispositivo similar, en lugar de una llave física.	• Solo se necesita presentar la llave electrónica. • Dificultad para copiar. • Permiten restringir el acceso a ciertas áreas.	Al insertar o presentar la llave electrónica, un lector verifica la autenticidad y desbloquea la cerradura si es válida.

Nota. Obtenido de *(Enrique et al., 2017).*

2.2 Tecnologías de autenticación

En esta sección se detallan las características, su funcionalidad y detalles relevantes de las tecnologías de autenticación con el propósito de explorar todo lo necesario acerca de estas tecnologías que van a hacer utilizadas en el desarrollo de este proyecto.

2.2.1 RFID

La tecnología de Identificación por Radiofrecuencia (RFID) es un sistema de identificación que emplea ondas de radio para transferir datos entre un lector y un dispositivo equipado con un chip RFID. Esta tecnología permite la identificación y el seguimiento de objetos, animales o personas sin requerir contacto directo ni línea de visión.

De acuerdo con (Alvarez-Marin & Castillo-Vergara, 2015) el concepto de RFID se remonta a la Segunda Guerra Mundial, cuando se utilizó una tecnología similar para identificar aviones amigos o enemigos. Sin embargo, el desarrollo comercial de RFID comenzó en la década de 1970, y en 1973 se realizó la primera patente para un dispositivo **RFID Activo**[2] por parte de Mario Cardullo, en ese mismo año, Charles Walton recibió una patente para un dispositivo RFID pasivo.

2.2.1.1 Aplicaciones de RFID

En la actualidad esta tecnología es muy aplicada según Portillo García et al. (2008) detalla que la tecnología RFID tiene varias aplicaciones en diferentes sectores como:

[2] **RFID Activo** permite la identificación por radiofrecuencia usando etiquetas o que tienen una fuente de energía interna, generalmente una batería, que permite una mayor distancia de lectura y la capacidad de transmitir señales de manera autónoma.

- **Logística y Gestión de Inventarios**: Se utiliza en centros comerciales para el seguimiento de productos desde el fabricante hasta el consumidor.
- **Comercio Minorista:** Gestión de inventarios y prevención de pérdidas.
- **Control de Acceso:** Tarjetas de identificación para entrada y salida de personal.
- **Transporte y Peajes:** Sistemas de peaje automatizados y rastreo de vehículos.
- **Salud:** Seguimiento de pacientes y equipos médicos.
- **Agricultura:** Identificación y seguimiento de ganado.
- **Bibliotecas:** Gestión y seguimiento de libros.
- **Ganadería:** Registro y rastreo de animales, identificar aquellos expuestos a enfermedades y gestionar la prevención y erradicación de amenazas veterinarias.

2.2.1.2 Funcionamiento

Consiste en una pequeña etiqueta constituida por un chip de circuito integrado y una antena, que tiene la habilidad de responder a ondas de radio transmitidas desde el lector RFID con el propósito de enviar, procesar, y almacenar información. Dentro del sistema, se distinguen tres componentes básicos: la etiqueta (tag), el lector (reader), y el equipo que gestiona el procesamiento de datos. Por otra parte, la etiqueta contiene información única del objeto al cual es adherida; el lector emite y recibe ondas de radio para leer la información almacenada en la etiqueta, y el equipo de procesamiento de datos procesa toda la información recolectada.

2.2.1.3 Tipos de etiquetas RFID

Una etiqueta RFID según Portillo García et al. (2008) está compuesta por una antena, un transductor de radio y un chip que incluyen datos que se quieren comunicar. Además, el

chip tiene una memoria interna cuya capacidad puede oscilar entre unas pocas decenas y varios miles de bytes.

Etiquetas pasivas

No tienen una fuente de alimentación propia, por lo que utilizan como fuente de alimentación la corriente eléctrica inducida en la antena por la señal de escaneo del lector RFID. Son los dispositivos de este tipo de menor costo.

Etiquetas activas

Tienen una fuente de energía propia, lo que permite mayores rangos de lectura y mayor espacio de memoria. También tienen la posibilidad de recibir y almacenar información.

Cuadro comparativo entre etiquetas pasivas y activas

En la tabla 2 se resume las etiquetas activas y pasivas. Cabe mencionar que hay otro tipo de etiqueta, semi-pasivas estás incorporan una pequeña batería que permite que el circuito integrado esté constantemente alimentado. Por lo cual le da la posibilidad de tener mejores tiempos de respuesta.

Tabla 3

Comparación de etiquetas activas y etiquetas pasivas.

	Etiquetas activas	**Etiquetas pasivas**
Integración de batería	Si	No
Precio	Mayor	Menor
Vida útil	Limitado	Casi ilimitado
Alcance-cobertura	Mayor	Menor
Capacidad de datos	Mayor	Menor

Nota. Obtenido de (Portillo García et al., 2008).

2.2.1.4 Rangos de frecuencias

Hay 4 tipos diferentes de sistemas RFID de acuerdo con el rango de frecuencia en el que operan:

Tabla 4

Rango de frecuencias para sistemas RFID.

Especificación	**Rangos de frecuencias**	**Alcance (Metros)**
Frecuencia baja (LF)	125 a 134.2 KHz	0.5
Frecuencia alta (HF)	13.56 MHz	1 a 3
Ultra alta frecuencia (UHF)	433 y 868 a 956 MHz	3 a 10
Microondas	2.4 GHz	Más de 10

Nota. Adaptado de (Huidobro, 2010).

2.2.2 Llave electrónica

Consiste en un circuito electrónico programable el cual contiene un código, este circuito interactúa con un receptor de manera física y envía la información contenida dentro del circuito. Además, esta llave generalmente puede tener una batería incluida o energizarse por medio del receptor.

2.2.2.1 Funcionamiento y tecnología subyacente

El funcionamiento de la llave electrónica se basa en la comunicación entre el circuito electrónico y el receptor. Al acercar la llave al receptor, se establece una conexión que permite la transmisión del código. El receptor, a su vez, verifica la validez del código y, si es correcto, permite el acceso o la activación del dispositivo que controle. Esta tecnología emplea sistemas de encriptación para asegurar la transmisión de datos, lo que proporciona un nivel adicional de seguridad.

2.2.2.2 Aplicaciones y casos de uso

Las llaves electrónicas tienen diversas aplicaciones y casos de uso en distintos ámbitos:

Vehículos

En la industria automotriz, las llaves electrónicas se emplean para el acceso y la puesta en marcha de vehículos. Al acercar la llave al automóvil, se activa el sistema de apertura de puertas y el motor puede encenderse sin necesidad de una llave física tradicional.

Casas – Hogares

En el sector residencial, las llaves electrónicas están implementadas en cerraduras inteligentes, lo que permite a los propietarios controlar el acceso a sus hogares de forma remota. Mediante aplicaciones móviles, los usuarios pueden otorgar acceso temporal a visitantes o monitorear la entrada y salida de la vivienda.

2.2.3 Usuario-contraseña y códigos PIN

Una contraseña es una forma de autenticación que utiliza información secreta para controlar el acceso a recursos específicos. En este contexto, en inglés, se distinguen dos términos: "password" (palabra de acceso) y "pass code" (código de acceso). Mientras que la primera puede no ser necesariamente una palabra existente, la segunda se asocia más

frecuentemente con códigos numéricos, como los PIN. Por otro lado, en español, los términos "clave" y "contraseña" a menudo se utilizan de manera intercambiable.

Su función principal es mantener la información protegida, requiriendo que los usuarios ingresen una clave para acceder a ella. Históricamente, el uso de contraseñas se remonta a la antigüedad, donde los centinelas solicitaban un "santo y seña" para permitir el paso(Infante, 2021).

Actualmente, la autenticación mediante usuario y contraseña o PIN es uno de los métodos más sencillos y extendidos para verificar la identidad en diversos sistemas. Este enfoque se basa en la premisa de que solo el usuario conoce la información de acceso, lo que le permite acceder a sus datos. De esta forma, la seguridad de la información recae en el usuario, quien es responsable de elegir su propia contraseña o PIN. No obstante, un desafío significativo es la posibilidad de que atacantes intercepten esta información, lo que compromete la integridad del sistema de seguridad establecido.

Figura 5

Seguridad en base a contraseña o PIN.

Nota. La **Figura 5** muestra un diseño gráfico que representa una forma de inicio de sesión mediante PIN o usuario y contraseña. En general sería un proceso de autenticación en un sistema digital.

2.2.3.1 Almacenamiento de contraseñas

El almacenamiento seguro de contraseñas es crucial para la protección de datos en sistemas informáticos. No obstante, algunos sistemas aún guardan las contraseñas en archivos de texto simples. En consecuencia, si un atacante consigue acceder a estos archivos, todas las contraseñas se verán comprometidas. Además, si los usuarios emplean la misma contraseña para múltiples cuentas, la vulnerabilidad se magnifica, ya que todas esas cuentas estarán en riesgo.

Por otro lado, los sistemas más seguros optan por almacenar contraseñas utilizando métodos criptográficos. De esta manera, se dificulta el acceso a las contraseñas incluso si un atacante obtiene acceso interno al sistema. Como indica Mendoza Gómez (2017) “un método común es el uso de hash”, el cual consiste en almacenar solo el valor codificado de la contraseña. Cuando un usuario introduce su contraseña, se genera un código hash a partir de ella mediante un algoritmo. Si este hash coincide con el valor almacenado, el acceso es concedido.

Para crear el hash de la contraseña, se aplica una función criptográfica que utiliza la contraseña junto con un valor adicional conocido como "salt". Este valor adicional es fundamental, ya que previene que los atacantes puedan generar listas de posibles contraseñas comunes.

2.2.3.2 Contraseñas vulnerables

Uno de los problemas más comunes es el uso de contraseñas débiles. La gran mayoría de las personas optan por contraseñas fáciles de recordar o comunes, lo que facilita el acceso no autorizado a sus cuentas, ya que estas contraseñas son predecibles. Según lo mencionado por Gonzalo & Salmerón (2021) lo más común en contraseñas sin complejidad es:

- Utilizar contraseñas predeterminadas del sistema
- Cadena de caracteres consecutivos
- Contraseñas comunes o usar un mínimo de caracteres
- Uso de información personal (fechas de nacimiento, nombres de familiares, eventos relevantes, número celular, número de identificación, etc.)
- Usar combinaciones solo de números o solo de letras

Esta falta de complejidad en las contraseñas permite que los atacantes utilicen técnicas de fuerza bruta o diccionario para adivinar las credenciales rápidamente. Diferentes estudios en seguridad determinan cuanto tiempo tomaría a un atacante descubrir una contraseña. Entre ellos, HIVE SYSTEMS se destaca por su amplio análisis de la evolución de la seguridad de las contraseñas en función de los avances tecnológicos. Por otra parte, HIVE SYSTEMS ha creado tablas comparativas anuales que muestran cómo los diferentes algoritmos de hash y configuraciones de hardware impactan en el tiempo necesario para descifrar una contraseña.

Además, estas tablas no solo ilustran la fortaleza relativa de las contraseñas según su longitud y complejidad, sino que también reflejan cómo los avances en la capacidad de procesamiento han influido en la seguridad de las contraseñas a lo largo de los años.

Desde otra perspectiva, la Red de Bibliotecas Universitarias (REBIUN) brinda materiales de formación para estudiantes de los cuales destaca “Contraseñas seguras” y se puede obtener la siguiente información.

Tabla 5

Tipos de contraseñas y tiempo de descifrado.

Número de caracteres	Contraseñas	Tiempo estimado para descifrar una contraseña según su complejidad
4	1234	Al instante
6	123456	Menos de un segundo
7	Alicia	Menos de un segundo

8	alcal123	1 minuto
9	Japan2024	4 días
10	¡Era$e1vez!	6 años

Nota. Basado en (Red de Bibliotecas Universitarias (REBIUN), 2019).

Otro punto que recalcar son los diversos tipos de ataques cibernéticos, que buscan engañar a los usuarios para que revelen información confidencial o comprometan sus dispositivos. Por otro lado, conectarse a redes inseguras también aumenta el riesgo de interceptación de datos, esto debido a la falta de capacitación o información, la cual puede llevar a la exposición accidental de información.

Finalmente, la ausencia de métodos de autenticación multifactor y la presencia de vulnerabilidades en las aplicaciones agravan la situación de seguridad. Sin una capa adicional de autenticación, las cuentas son más susceptibles a accesos no autorizados. En conjunto, todas estas amenazas resaltan la importancia de mantener buenas prácticas de seguridad digital.

2.2.3.3 Contraseñas robustas

La robustez de una contraseña es fundamental para la seguridad de la información. Una contraseña fuerte es aquella que resulta difícil de adivinar o descifrar por los ciberdelincuentes. Para crear contraseñas robustas, la UNED recomienda seguir las siguientes pautas:

- **Longitud:** La contraseña debe tener al menos 8 caracteres. Cuanto más larga sea, más tiempo tardará un atacante en descubrirla.
- **Combinación de caracteres:** Utiliza una mezcla de letras mayúsculas, minúsculas, números y símbolos. Esto aumenta la complejidad y dificulta su adivinanza.

- **Uso de frases nemotécnicas:** Una técnica efectiva es utilizar frases memorables. Por ejemplo:
    - ✓ Selecciona una frase: «el sol brilla en la mañana».
    - ✓ Aplica mayúsculas: «El Sol Brilla en la Mañana».
    - ✓ Incluye el servicio: «El Sol Brilla en la Mañana Instagram».
    - ✓ Añade números: «El Sol Brilla en la Mañana Instagram 2024».
    - ✓ Incorpora caracteres especiales: «El Sol Brilla en la Mañana Instagram 2024! *».
    - ✓ Comprime la frase utilizando la primera letra de cada palabra: «ESbElM2024! *».

2.2.4 Biometría

La autenticación biométrica representa una evolución significativa en los métodos de identificación y verificación de identidad. Específicamente, esta tecnología se basa en la medición de características biológicas únicas de los individuos, como huellas dactilares, reconocimiento facial, iris y otros rasgos físicos. Mediante sistemas avanzados que integran matemáticas, inteligencia artificial y aprendizaje automático, la biometría permite una identificación rápida y precisa, facilitando el acceso sin la necesidad de recordar contraseñas o portar identificaciones físicas.

En este sentido, la implementación de sistemas biométricos se ha vuelto más común en diversos sectores, aunque su adopción generalizada ha enfrentado desafíos como el costo y el mantenimiento. No obstante, la seguridad que ofrecen, al ser más difíciles de falsificar que métodos tradicionales, hace que su uso sea cada vez más atractivo.

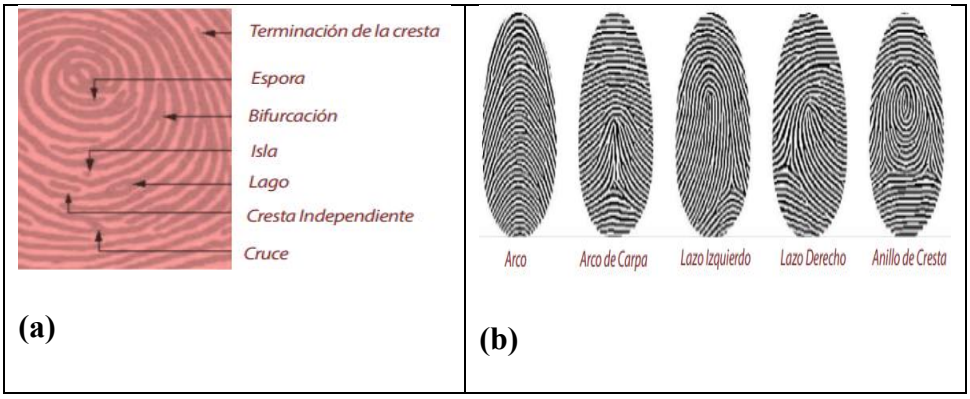
2.2.4.1 Verificación mediante huella dactilar

La verificación de huellas dactilares es uno de los métodos más antiguos y efectivos para la identificación biométrica. En este sentido, la singularidad de las huellas garantiza que no existen dos dedos con patrones idénticos, debido a esto se ha convertido en una herramienta confiable en entornos enfocados a seguridad.

Además, según INCIBE (2016) explica que hay dos métodos utilizados para la búsqueda o comparación de muestras de huellas dactilares, las basadas en minucias[3] y las basadas en correlación[4].

Figura 6

Comparación de huellas dactilares.



Nota. El literal a es huella dactilar de minucias y el literal b es de patrones o correlación

Obtenido de (INCIBE, 2016).

En cuanto al proceso de autenticación, este es sencillo: el usuario coloca su dedo en un área de lectura, donde se captura una imagen que se normaliza mediante espejos para corregir ángulos. A partir de esta imagen, el sistema extrae las minucias - los arcos, bucles y remolinos

[3] Minucias se refieren a los puntos donde los bordes de las líneas de la huella terminan, se bifurcan o presentan variaciones.

[4] Esta técnica analiza la imagen completa de la huella dactilar, enfocándose en la estructura general y los patrones de las crestas (como arcos, bucles y espirales).

característicos de la huella que se comparan con los datos almacenados en la base de datos. Es crucial destacar que el análisis se centra en la posición relativa de estas minucias, no en la huella en sí. En la **Figura 7** muestra la representación de la huella dactilar sin embargo se debe tener en cuenta que las huellas presentan diversas configuraciones, estas características hacen que las huellas dactilares sean altamente efectivas y confiables para sistemas de seguridad.

Figura 7

Representación de huella dactilar.

Nota. En la figura las características que se indican son basadas en el método de minucias, es decir se registra cada tipo de minucia y la posición.

Sin embargo, a pesar de su costo relativamente bajo en comparación con otras tecnologías biométricas, la verificación de huellas enfrenta algunos desafíos. Por un lado, lesiones temporales en el dedo, como cortes o quemaduras, pueden afectar la precisión del sistema, al igual que factores como la suciedad, la presión aplicada sobre el lector o el estado de la piel.

2.2.4.2 Retina

La autenticación basada en patrones retina[5] se fundamenta en la singularidad de la vasculatura[6] retinal de cada individuo, lo que la convierte en un método altamente efectivo para la identificación de usuarios. Como primera instancia, el sistema requiere que el usuario realice una serie de pasos, incluyendo el ajuste de la distancia interocular[7] y la alineación de la cabeza, antes de mirar a un punto específico y activar el escaneo. Durante el proceso, se utiliza radiación infrarroja de baja intensidad para escanear la retina en un patrón espiral, capturando los nodos y ramas de los vasos sanguíneos. Posteriormente, esta información se compara con los datos almacenados en una base de datos. Si los patrones coinciden, se concede el acceso al usuario.

Figura 8

Biometría por retina

Este método no solo ofrece un alto nivel de seguridad, al ser difícil de falsificar, sino que también destaca por su precisión y rapidez en la identificación, convirtiéndolo en una opción prometedora en el ámbito de la autenticación biométrica.

[5] Retina membrana interior del ojo constituida por varias capas de células, está compuesta por células que detectan la luz y la convierten en señales eléctricas que son enviadas al nervio óptico.

[6] Vasculatura conjunto de los vasos sanguíneos (arterias, venas y capilares) del organismo, que transportan la sangre a través del cuerpo o de un órgano en específico, y su principal función es transportar oxígeno y nutrientes a los tejidos.

[7] Interocular perteneciente o relativo al interior del ojo o situado entre los ojos.

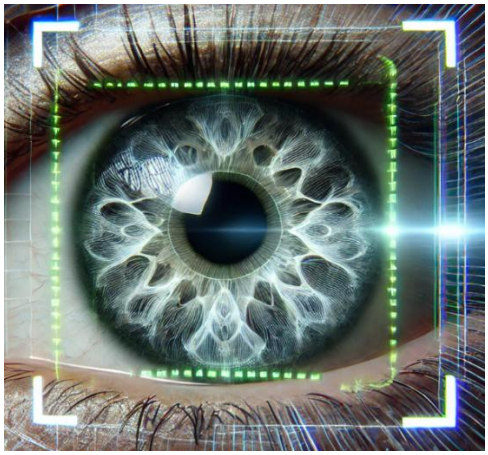
2.2.4.3 Verificación mediante patrones oculares

La verificación de patrones oculares es una de las tecnologías más avanzadas en autenticación biométrica, la cual se divide principalmente en dos enfoques: el análisis de patrones retinales y el reconocimiento del iris. Estas metodologías ofrecen una probabilidad de coincidencia casi nula en poblaciones extensas, lo que las convierte en opciones altamente seguras.

Sin embargo, a pesar de su eficacia, estos sistemas enfrentan retos significativos en términos de aceptación por parte de los usuarios. La incomodidad de utilizar dispositivos como binoculares y la desconfianza hacia la tecnología que analiza el ojo generan resistencia. Muchos usuarios temen que el escaneo ocular pueda revelar información médica sensible, como el consumo de sustancias, lo que añade una capa de inquietud sobre la privacidad.

Figura 9

Biometría por iris.



Adicionalmente, el alto costo de estos sistemas y la lentitud del proceso de autenticación en grandes poblaciones limitan su implementación, restringiendo su uso principalmente a entornos de alta seguridad, como instalaciones militares. A pesar de las

garantías de los fabricantes sobre la protección de la privacidad, la percepción pública tiende a ser escéptica, lo que representa un obstáculo para la adopción masiva de esta tecnología.

2.2.4.4 Reconocimiento por rostro

Reconocimiento facial o por rostro es una técnica que permite identificar a una persona a partir de una imagen o fotografía. Para ello, se utilizan programas informáticos que analizan las características de los rostros humanos. Entre los factores clave considerados para la comparación se encuentran medidas como la separación entre los ojos, la longitud de la nariz y el contorno de la mandíbula. A diferencia de otros métodos biométricos, el reconocimiento facial se puede emplear para fines de vigilancia, generalmente mediante cámaras de video. En la **Figura 10** se muestra una representación del reconocimiento facial para un sistema de seguridad.

Figura 10

Reconocimiento facial.

Por otra parte, el análisis facial, ya sea en 2D o 3D, es un método eficaz para identificar a un individuo. Sin embargo, el estudio en 3D es preferible, ya que ofrece mejores resultados en comparación con el análisis de imágenes 2D que dificulta los ataques fraudulentos es decir

por el uso de fotografías, ya que los sistemas 2D no pueden diferenciar entre un rostro real y una fotografía de un rostro, lo que puede comprometer la seguridad del sistema.

2.2.4.5 Reconocimiento por voz

Los sistemas de reconocimiento de voz no se centran en entender el contenido de lo que el usuario dice, sino en identificar patrones sonoros y características acústicas como la frecuencia o tonalidad de voz, timbre, el ritmo y velocidad. para verificar la identidad del hablante. Para que un sistema de reconocimiento de voz funcione correctamente, es esencial contar con condiciones óptimas de grabación, como la ausencia de ruidos, reverberaciones o ecos. Idealmente, estas condiciones deben ser consistentes cada vez que se requiera la autenticación. En la **Figura 11** se representa como seria la autenticación por voz la cual el usuario pronuncia una serie de palabras o frases la cual posteriormente se valida o deniega.

Figura 11

Autenticación por voz.

Durante el proceso de autenticación, el usuario pronuncia una serie de frases que son clave para la seguridad del sistema. Existen dos tipos de modelos en el reconocimiento de voz:

- Texto Dependiente: En este modelo, el sistema reconoce un conjunto limitado de frases predefinidas. Por ejemplo, el usuario podría ser requerido a pronunciar solo su nombre. Sin embargo, este enfoque ofrece una seguridad limitada debido a su naturaleza restringida.
- Texto Independiente: El modelo de texto independiente es más robusto, ya que permite al sistema proponer palabras o frases de un conjunto amplio. Las frases elegidas suelen ser características, lo que maximiza la cantidad de datos útiles para el análisis, como la entonación, la pronunciación de diptongos o palabras con múltiples vocales.

A medida que el usuario habla, el sistema registra información relevante. Al finalizar la frase, el sistema evalúa los datos recopilados y los compara con los almacenados en la base de datos para decidir si se concede o se deniega el acceso.

Tabla 6

Comparación de métodos biométricos

	Ojo-Iris	**Ojo-Retina**	**Huellas dactilares**	**Facial**	**Voz**
Fiabilidad	Muy alta	Muy alta	Alta	Alta	Alta
Facilidad de uso	Media	Baja	Alta	Alta	Alta
Prevención contra ataques	Muy alta	Muy alta	Alta	Alta	Media
Aceptación	Media	Media	Media	Alta	Alta
Estabilidad	Alta	Alta	Alta	Media	Media
Identificación y autenticación	Ambas	Ambas	Ambas	Ambas	Autenticación

Interferencias	Gafas	Irritaciones	Suciedad, heridas, asperezas	Rasgos similares otra persona.	Ruido, cambio en a la voz
Utilización	Instalaciones nucleares, servicios médicos, centros penitenciarios de máxima seguridad	Instalaciones nucleares, servicios médicos, centros penitenciarios de máxima seguridad	Policial, instituciones universitarias	Instituciones universitarias (en sistema de acceso), celulares	Accesos remotos en bancos o base de datos
Estándares			ANSI	ISO/IEC 19794-5	SVAPI

Nota. Obtenido de (Carro Frau, 2015).

Ventajas

- Alta Fiabilidad: Métodos como el reconocimiento del iris y la retina ofrecen una alta precisión en la identificación.
- Dificultad de Falsificación: La mayoría de los métodos biométricos son difíciles de replicar, lo que aumenta la seguridad.
- Facilidad de Uso: Métodos como las huellas dactilares y la firma son intuitivos y ampliamente aceptados por los usuarios.
- Conveniencia: Los métodos, como el reconocimiento de voz, permiten una autenticación rápida y sin contacto físico.
- Largo Plazo: Características biométricas como las huellas dactilares y la geometría de la mano son generalmente estables a lo largo del tiempo.

Desafíos

- Costo: La implementación de sistemas biométricos puede ser costosa, especialmente para tecnologías avanzadas como el reconocimiento del iris.
- Costo: La implementación de sistemas biométricos puede implicar una inversión significativa, especialmente para tecnologías avanzadas como el reconocimiento de iris.
- Preocupaciones de Privacidad: La gestión de datos biométricos puede generar inquietudes relacionadas con la privacidad y la protección de la información personal de los usuarios.
- Condiciones Ambientales: La eficacia de algunos métodos biométricos puede verse comprometida por factores externos como la iluminación o el ruido ambiental.
- Aceptación del Usuario: La disposición de los usuarios para adoptar ciertos métodos biométricos puede variar, lo que podría dificultar su implementación generalizada.
- Vulnerabilidad a Ataques: A pesar de su dificultad para ser falsificados, algunos métodos, como el reconocimiento de voz, pueden ser susceptibles a ataques de suplantación.
- Requerimientos Técnicos: Ciertos métodos biométricos exigen hardware y software específicos, lo que puede complicar su integración con sistemas ya existentes.

2.3 Riesgos y amenazas en sistemas de acceso electrónico

Los riesgos y amenazas en sistemas de acceso electrónico son peligros que pueden comprometer la seguridad de la información. Además, las vulnerabilidades del sistema pueden surgir de errores de programación o configuraciones inadecuadas, las cuales facilitan el acceso

de los atacantes a los recursos protegidos. Asimismo, el acceso no autorizado es un riesgo constante, especialmente cuando se utilizan contraseñas débiles o se omiten medidas de autenticación adecuadas.

2.3.1 Ciberataques y hacking

Los ciberataques, que incluyen técnicas como el phishing y el malware (programa maligno), representan una de las principales preocupaciones en la actualidad, ya que buscan acceder de manera no autorizada a datos sensibles.

El phishing es una técnica de fraude en línea empleada por ciberdelincuentes para engañar a las personas y obtener información personal sensible, como contraseñas, números de tarjetas de crédito o datos bancarios. Generalmente, los atacantes envían correos electrónicos o mensajes de texto que aparentan ser de entidades legítimas, tales como bancos, empresas de servicios o plataformas de redes sociales.

En estos mensajes, los delincuentes suelen incluir enlaces a sitios web diseñados para parecer auténticos. Cuando las víctimas hacen clic en estos enlaces, son redirigidas a páginas fraudulentas que les solicitan ingresar su información personal. Adicionalmente, estos correos pueden contener advertencias urgentes o promociones atractivas para incrementar la probabilidad de que la víctima caiga en la trampa.

2.3.2 Phishing

El phishing es una técnica de fraude por ciberdelincuentes para engañar a las personas y obtener información personal sensible, como contraseñas, números de tarjetas de crédito o datos bancarios. Generalmente, los atacantes envían correos electrónicos o mensajes de texto que aparentan ser de entidades legítimas, como bancos, empresas de servicios o plataformas de

redes sociales. Cuando las víctimas hacen clic en estos enlaces, son redirigidas a páginas fraudulentas que solicitan ingresar su información personal. Además, estos correos pueden contener advertencias urgentes o promociones atractivas para aumentar la probabilidad de que la víctima caiga en la trampa.

El phishing puede manifestarse de diversas maneras. Por ejemplo, el spear phishing se enfoca en individuos específicos mediante mensajes personalizados, mientras que el whaling tiene como objetivo a altos ejecutivos de empresas. Asimismo, existen variantes como el smishing, que se realiza a través de mensajes de texto, y el vishing, que se lleva a cabo por teléfono (Tarazona, 2023).

En este caso las cerraduras que utilizan tecnología conectada a Internet para gestionar el acceso a hogares y edificios, las hace vulnerables a ataques de phishing, que pueden comprometer la seguridad de estos sistemas.

2.3.3 Fuerza bruta

Consiste en descifrar la contraseña o claves realizando distintas combinaciones hasta dar con la contraseña correcta. Sin embargo, este ataque puede ser efectivo si la contraseña es de pocos dígitos, pero también es muy lento si la contraseña es muy larga y compleja, según Alfonso Pagán (2019) “con una capacidad de cómputo adecuada y un tiempo relativamente largo, es teóricamente viable este ataque”.

2.3.4 Robo de identidad y suplantación

Los atacantes pueden utilizar técnicas de phishing para engañar a los usuarios y obtener sus credenciales de acceso. En específico, de acuerdo con Grupo de Regulación de AUTELSI (2021) “la suplantación o robo de identidad es un delito en el que un atacante obtiene y utiliza

la información personal de otra persona sin su consentimiento, con el fin de hacerse pasar por esa persona". Por consiguiente. esto puede incluir el uso de datos como nombres, números de seguro social, direcciones y detalles financieros para cometer fraudes o acceder a servicios de manera ilegal.

En consecuencia, los atacantes pueden suplantar la identidad de la persona cuyas credenciales han robado, utilizándolas para acceder a propiedades, realizar transacciones o incluso manipular sistemas de seguridad. Por ejemplo, podrían modificar los códigos de acceso o añadir nuevas credenciales para mantener dicho acceso.

2.3.5 Fallos y vulnerabilidades del sistema

Según Grupo de Regulación de AUTELSI (2021) "una vulnerabilidad de seguridad es un error o deficiencia en el diseño, la implementación, el funcionamiento o la gestión de un sistema, que puede ser aprovechado para infringir la política de seguridad de este."

2.4 Implementación de medidas de seguridad

En esta sección se aborda temas de seguridad como autenticación de múltiples factores, cifrado de datos y la importancia que tiene reforzar la seguridad en algún sistema que lo requiera.

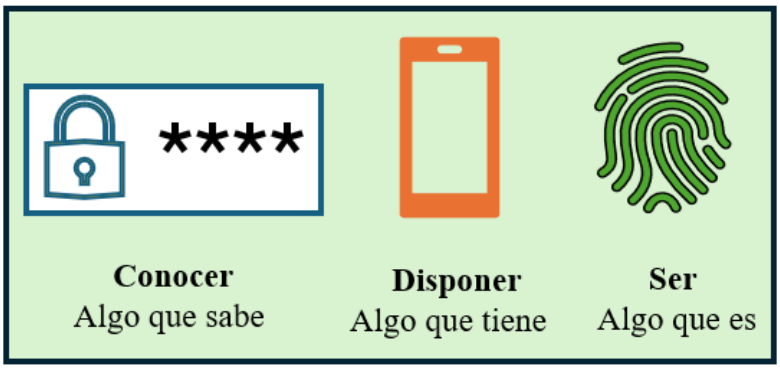
2.4.1 Autenticación múltiples factores

La autenticación unifactorial, que se basa tradicionalmente en el uso de nombre de usuario y contraseña, ya no proporciona el nivel de seguridad necesario. Según lo planteado por Veriddica (2022) en los últimos años, la autenticación de 2 o más factores ha ganado una gran relevancia, ya que, ofrecen servicios críticos a través de plataformas en línea. De hecho, tal como su nombre sugiere, la autenticación de múltiples factores requiere que los usuarios se

autentiquen utilizando al menos dos métodos distintos. En la **Figura 12** se muestran tres niveles de seguridad, cada uno representando un tipo diferente de verificación, el primer factor es algo que el usuario conoce, como una contraseña, el segundo factor se refiere a algo que el usuario posee, como una tarjeta inteligente, para finalizar el tercer factor implica algo que el usuario es, que incluye características biométricas.

Figura 12

Factores o niveles de autenticación.



Adicionalmente, la integración de estos factores en un sistema de autenticación fortalece la seguridad, ofreciendo una protección más eficaz contra accesos no autorizados.

Tabla 7

Ejemplos de autenticación de varios niveles.

Niveles	Ejemplo
Simple	• Contraseña • Tarjeta RFID • Huella dactilar
Doble	• RFID + código PIN • RFID + Reconocimiento facial • Huella dactilar + contraseña
Triple	• Escaneo de retina + PIN + Biometría

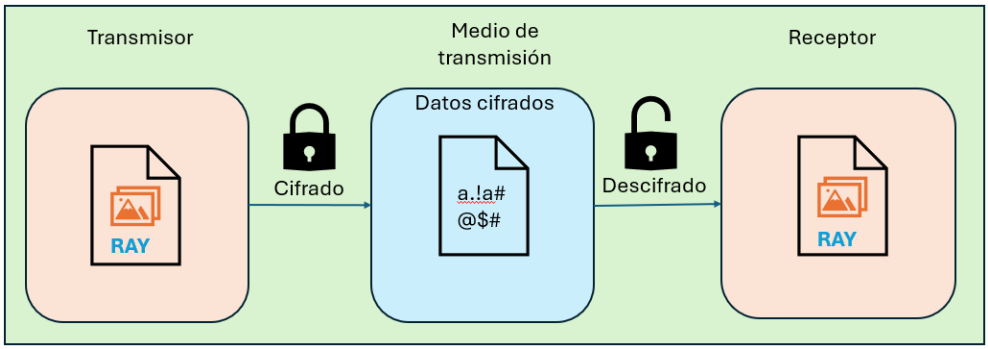
2.4.2 Cifrado de datos

Es una técnica de seguridad que transforma información legible a un formato encriptado, lo que garantiza que solo las personas o sistemas autorizados puedan acceder a ella. Además, este proceso es esencial para proteger tanto la confidencialidad como la integridad de la información, especialmente en entornos digitales, donde los datos pueden ser vulnerables a accesos no autorizados.

Por otra parte, el cifrado implica la conversión de información legible (texto plano) en un formato ilegible (texto cifrado) con el fin de salvaguardar la confidencialidad de los datos. Por lo tanto, para llevar a cabo este cifrado, se utilizan algoritmos y claves que permiten únicamente descifrar y acceder a la información original las personas autorizadas.

Figura 13

Representación de encriptación de mensaje.



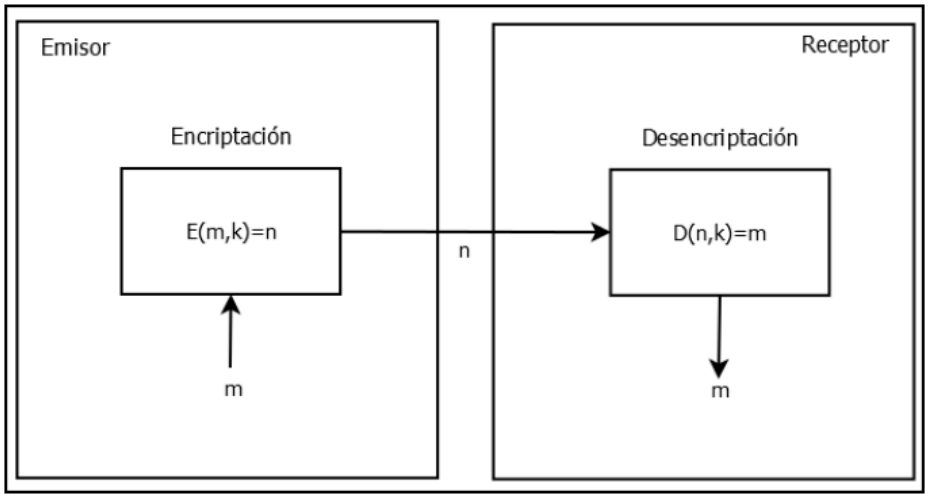
2.4.3 Cifrado simétrico

El cifrado simétrico es un método de encriptación que utiliza la misma clave para cifrar y descifrar la información. De tal modo que, en este tipo de cifrado, tanto el emisor como el receptor deben conocer y mantener en secreto la clave, lo que garantiza que solo las partes autorizadas puedan acceder a los datos originales (Ortega Chulde, 2023).

En particular este método es rápido y eficiente, especialmente al manejar grandes volúmenes de datos, ya que requiere menos recursos computacionales. Sin embargo, una de sus principales desventajas radica en la necesidad de una distribución segura de la clave, ya que, si se logra capturar datos y obtener la clave por un tercero, la seguridad de la información se ve comprometida. Según el autor Ortega Chulde (2023) explica que en la **Figura 14** se describe el proceso de encriptación simétrica, el cual consiste en aplicar una función matemática a un mensaje para hacerlo ilegible para quien no tenga la clave privada. Por otra parte, el descifrado manifiesta que es el proceso inverso, en el que se aplica la función inversa para recuperar el mensaje original.

Figura 14

Encriptación simétrica.



Nota. Obtenido de (Ortega Chulde, 2023).

Además, en la **Figura 14** la representación de "la encriptación simétrica utiliza la misma contraseña "k" para la encriptación y desencriptación de los datos" (Ortega Chulde, 2023).

Donde:

- **m** = mensaje a transmitir
- **n** = mensaje cifrado o encriptado
- **k** = clave privada
- **E(m, k) = n** → Función para la encriptación.
- **D(n, k) = m** →Función para la desencriptación.

Ejemplos comunes de algoritmos de cifrado simétrico incluyen el AES (Advanced Encryption Standard) y el DES (Data Encryption Standard), los cuales son ampliamente utilizados en diversas aplicaciones de seguridad informática.

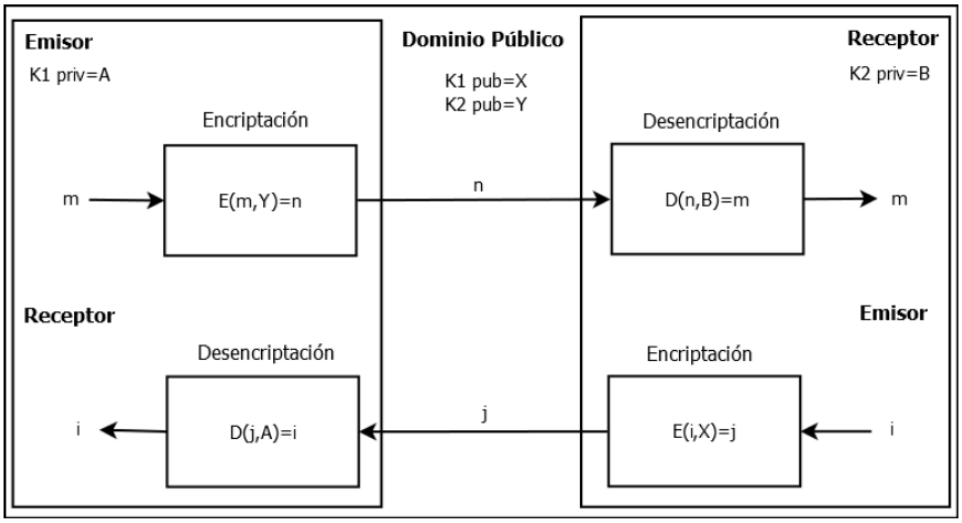
2.4.4 Cifrado asimétrico

Es un método de encriptación que utiliza un par de claves: una clave pública para cifrar y una clave privada para descifrar. La clave pública se puede compartir abiertamente y se utiliza para cifrar la información, mientras que la clave privada se mantiene en secreto y se usa para descifrar los datos. En este sentido el cifrado asimétrico permite que cualquier persona pueda enviar información segura a un destinatario específico, quien es el único capaz de descifrarla con su clave privada (Ortega Chulde, 2023).

En la **Figura 15** según Ortega Chulde indica cómo el proceso de cifrado permite a un usuario encriptar un mensaje utilizando la clave pública de otra persona. De esta manera, solo la persona poseedora de la clave privada correspondiente podrá descifrarlo. Esta característica confiere a la criptografía asimétrica una alta seguridad, ya que, aunque un tercero intercepte el mensaje cifrado, no podrá descifrarlo sin la clave privada (2023).

Figura 15

Encriptación asimétrica.



Nota. Obtenido de (Ortega Chulde, 2023)

Además, en la **Figura 15** se representa a la criptografía de clave privada, donde tanto el emisor como el receptor utilizan su propio par de claves, para generar un nivel alta seguridad.

Donde:

- $\boldsymbol{m}$, $\boldsymbol{i}$ → Menajes para transmitir
- $\boldsymbol{n}$, $\boldsymbol{j}$ → Mensajes Encriptados
- $\boldsymbol{A}$, $\boldsymbol{B}$ → Claves privadas (Utilizadas para desencriptar mensajes)
- $\boldsymbol{X}$, $\boldsymbol{Y}$ → Claves públicas (Utilizadas para encriptar mensajes)
- $\boldsymbol{E}\ (\boldsymbol{m}, \boldsymbol{k}) = \boldsymbol{n}$ → Función para la Encriptación
- $\boldsymbol{D}\ (\boldsymbol{n}, \boldsymbol{k}) = \boldsymbol{m}$ → Función para la Desencriptación

Algoritmos como RSA (Rivest-Shamir-Adleman) y ECC (Elliptic Curve Cryptography) son ejemplos destacados de cifrado asimétrico, y son ampliamente utilizados en aplicaciones de seguridad, como en la transmisión de datos a través de internet y en la creación de certificados digitales.

2.4.5 Porque es importante cifrar los datos

Debido a que hoy en día los datos o la información son un recurso valioso tal como en el ámbito personal, la información como datos financieros, historial médico o preferencias de consumo se considera altamente sensible, ya que puede ser utilizada para manipular comportamientos, realizar fraudes o incluso comprometer la privacidad de los individuos. Por otro lado, en el contexto corporativo, los datos sobre clientes, operaciones o tendencias del mercado son datos sensibles que necesitan ser protegidos. Además, cifrar los datos significa que, cada vez que se desee acceder a ellos, es necesario realizar un proceso de descifrado (UNAM, 2020). Este procedimiento, aunque añade un nivel de complejidad al acceso simple, es esencial para proteger la confidencialidad de la información. En primer lugar, el cifrado actúa como una barrera contra accesos no autorizados, garantizando que solo aquellos con las claves adecuadas puedan visualizar o manipular los datos.

Sin embargo, es importante considerar que este aumento en la seguridad conlleva un costo en términos de rendimiento. Es decir, el proceso de cifrado y descifrado puede reducir la velocidad de acceso a la información, lo que podría ser un inconveniente en situaciones que requieren respuestas rápidas. A pesar de esta desventaja, las organizaciones y los individuos reconocen que los beneficios de proteger la información sensible superan ampliamente los inconvenientes asociados con la reducción de la velocidad. Por lo tanto, cifrar los datos se convierte en una práctica indispensable para salvaguardar la privacidad y la integridad de la información en un entorno digital cada vez más amenazante.

2.4.6 Qué tipo de información debe ser cifrada

Hoy por hoy, es fundamental identificar qué tipo de información debe ser cifrada para garantizar la seguridad y la privacidad de los datos. Como punto de partida, los datos personales, como nombres, direcciones, números de teléfono y detalles financieros, deben ser

objeto de cifrado, debido a que la exposición de esta información puede resultar en robos de identidad y fraudes financieros, lo que afecta gravemente a la privacidad del individuo. Otro tipo de información es la relacionada con la salud, como historiales médicos y datos de tratamientos, su cifrado es esencial para prevenir el acceso no autorizado y garantizar la confidencialidad de pacientes (UNAM, 2020).

Por otro lado, en el ámbito corporativo, los datos empresariales, que abarcan desde estrategias comerciales hasta información sobre clientes y proveedores, deben ser cifrados para proteger la ventaja competitiva de la empresa. La filtración de datos estratégicos puede llevar a pérdidas financieras significativas y dañar la reputación de la organización. Así, las empresas deben priorizar el cifrado de cualquier información que pueda comprometer su seguridad o la de sus clientes.

Finalmente, los datos de acceso, como contraseñas y credenciales de usuario, son otro tipo de información que debe ser cifrada. La protección de estas credenciales es crucial, ya que su exposición puede permitir a los atacantes acceder a sistemas críticos y robar información sensible. Por lo tanto, implementar cifrado en estos datos es una práctica esencial para mantener la integridad y la seguridad de los sistemas informáticos.

3 CAPÍTULO III: DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

En este apartado se describe el desarrollo del sistema de control de accesos, basado en la metodología en cascada, inicialmente se establecen los requerimientos necesarios para el buen funcionamiento para la ejecución del proyecto, otra fase del proyecto es el diseño del sistema de cerradura inteligente, la llave de circuito programable, el diseño de la base de datos y diseño de la aplicación, finalmente el desarrollo completo del prototipo e integración de todo el sistema implementación.

3.1 Metodología de gestión del proyecto

La metodología en cascada recibe este nombre debido a su estructura, en la que las fases del proyecto se organizan de manera secuencial, es decir una sobre la otra. De esta forma, el trabajo avanza de manera lineal y progresiva, el cual sigue un orden descendente. Además, este enfoque agrupa el desarrollo del proyecto en una serie de fases que deben completarse de forma estricta y en un orden determinado, el cual se puede observar en la **Figura 16**. Por otra parte, esta metodología se caracteriza porque cada etapa debe concluirse completamente antes de pasar a la siguiente, lo que asegura que los avances en el proyecto sigan un flujo continuo y descendente, similar al de una cascada.

Figura 16

Fases de la metodología en cascada.

Nota. Adaptado de (DIGITAL TALENT AGENCY, 2018).

3.2 Primera fase: Análisis de requerimientos

Esta primera fase del proyecto permite establecer los requerimientos necesarios de: **Stakeholders**[8], Sistema y Arquitectura, así como se detalla en la **Tabla 8,** cada requerimiento tiene una nomenclatura que lo identifica, y recopila las necesidades y expectativas funcionales y no funcionales, expresando claramente lo que se espera del sistema.

Además, para el análisis de requisitos, se tomó como base las recomendaciones de normas internacionales, como la norma ISO/IEC/IEEE 29148:2018 que establece lineamientos para la ingeniería de requisitos, incluyendo la identificación de stakeholders, la formulación de necesidades del sistema y criterios para la calidad de los requisitos(ISO/IEC/IEEE 42010:2018, 2024).

[8] **Stakeholders** se puede denominar así a un individuo o grupo que tiene cierto interés en las actividades o resultados de una empresa, como ejemplo en una ISP los stakeholders serian clientes, proveedores, empleados, accionistas y directivos.

Finalmente, para el diseño de la arquitectura del sistema se tomó en cuenta la norma ISO/IEC/IEEE 42010:2022, la cual proporciona un marco para describir la arquitectura desde distintas perspectivas (ISO/IEC/IEEE 42010:2022, 2022).

Tabla 8

Etiquetado de los requerimientos.

Requerimiento	Siglas
Stakeholders	StSR
Sistema	SiSR
Arquitectura	ArSR

Del mismo modo es necesario establecer la importancia que se designa a cada uno de los requisitos, con el objetivo de identificar la importancia de cada uno y el cumplimiento obligatorio, dentro del desarrollo del sistema.

Tabla 9

Nivel de importancia de los requisitos.

Importancia	Especificación
Alta	La valoración o prioridad alta es la parte fundamental para el cumplimiento del desarrollo del sistema.
Media	La valoración media en los requisitos cumple una función relevante en el desarrollo del sistema y su funcionalidad.
Baja	Este tipo de valoración es tomado en cuenta, pero no afecta el desarrollo del proyecto.

3.2.1 Requerimientos de stakeholders

La intención de definir este requerimiento es identificar las partes interesadas y el rol que cumplen para lograr el desarrollo del proyecto. En la **Tabla 10** se definen a los stakeholders quienes pueden afectar o verse afectados por los resultados obtenidos.

Tabla 10

Identificación de partes interesadas.

Partes interesadas	**Descripción del rol**
Jessica Ruano	Tesista
Msc. Luis Suárez	Director del proyecto
Msc. Carlos Vásquez	Asesor del proyecto
Ing. Christopher Ortega	Experto en electrónica

3.2.1.1 Requisitos stakeholders – usuarios y operacionales

En la **Tabla 11** se detallan los requisitos de usuarios y operacionales, los cuales tienen una ponderación de prioridades, con el fin de establecer que tan críticos son para el éxito del sistema y del proyecto en general. El establecer requisitos de usuarios y operacionales ayudan a definir las condiciones necesarias para el funcionamiento del sistema dentro de su entorno técnico y el cumplimiento de estos asegura que el sistema funcione de manera eficiente y confiable.

Tabla 11

Requisitos stakeholders.

StSR				
N °	**Requerimiento**	**Prioridad**		
		Baja	**Media**	**Alta**
	usuario			
StSR1	Requieren ingreso de datos mediante la aplicación.			x
StSR2	Los mecanismos de acceso deben ser de uso fácil para garantizar el acceso rápido.			x
StSR3	La interfaz de la aplicación debe ser amigable e intuitiva.		x	
StSR4	El servidor debe tener una conexión a internet constante para emitir las alertas y claves temporales.		x	
StSR5	Se debe recibir notificaciones por la APP y correo electrónico.		x	
StSR6	Desde la aplicación se debe visualizar el historial de accesos.		x	
StSR7	La clave temporal se recibe por mensajes, el tiempo de duración de la clave se elige en APP.		x	
StSR8	Mediante la App se puede registrar de la clave de acceso.			x
StSR9	Tecla de activación para el ingreso por contraseña en el teclado.			x
StSR10	Funcionamiento de la cerradura (llave tradicional).		x	

StSR11	Desde la aplicación se debe habilitar/deshabilitar los métodos de acceso.	x	
	Operacionales		
StSR12	La cerradura debe ser compacta, con un tamaño adecuado para instalarse en una puerta estándar.		x
StSR13	Debe incluir al menos tres métodos de autenticación, como acceso mediante RFID, llave programable, y clave numérica (PIN)	x	
StSR14	El sistema debe contar con conexión a Internet para el almacenamiento y envió de datos en el servidor, así como una conexión directa con la cerradura eléctrica.		x
StSR15	El sistema debe tener una pantalla que permita al usuario visualizar el ingreso de la contraseña y navegar por el menú de configuración, también luces indicadoras que reflejen el estado del sistema (por ejemplo, desbloqueo o fallos).	x	
StSR16	La cerradura tiene conexión a la red eléctrica estándar de 110-120 voltios, asegurando una fuente de alimentación continua.	x	
StSR17	La llave debe contener un identificador único en su código.	x	
StSR18	La llave programable debe ser pequeña y portátil.	x	

StSR19	La llave programable debe estar alimentada por una batería recargable.	x	
StSR20	Se debe permitir el registro de múltiples tarjetas RFID	x	
StSR21	El sistema debe aceptar contraseñas numéricas de 6 dígitos.		x
StSR22	Envío de notificación alerta después de tres intentos fallidos (contraseña numérica).		x
StSR23	Los mecánicos de accesos (RFID, llave programable, contraseña numérica, y clave temporal) están almacenados en una base de datos.		x
StSR24	El servidor en la nube y el mecanismo de acceso deben estar operativos permanentemente.		x
StSR25	Se debe considerar un plan de mantenimiento, que incluya el pago del plan del servidor en la nube.		x

3.2.2 Requerimientos del sistema

En la **Tabla 12** se detallan los requisitos del sistema con el fin de establecer una base clara sobre las funcionalidades, características, tareas y procesos que se debe cumplir. A su vez, estos requisitos son esenciales para el diseño, ya que proporcionan la información necesaria para construir soluciones efectivas. De igual manera, permiten establecer criterios de aceptación, lo que facilita la evaluación del progreso del proyecto y asegura que se cumplan las expectativas de los stakeholders.

Tabla 12

Requerimientos del sistema.

SySR				
N °	**Requerimiento**	**Prioridad**		
		Baja	**Media**	**Alta**
SySR1	El sistema debe utilizar al menos tres métodos de autenticación y el ingreso por clave temporal.			x
SySR2	El sistema debe contar con de acceso tradicional mecánico.			x
SySR3	El sistema debe notificar ante intentos fallidos.			x
SySR4	Se debe permitir los accesos temporales o limitados para usuarios.		x	
SySR5	Es necesario un servidor para gestionar las conexiones, claves y notificación de intentos fallidos.			x
SySR6	El servidor debe estar alojado en la nube para garantizar el acceso remoto a la aplicación.		x	
SySR7	El servidor debe almacenar y gestionar los datos del sistema.		x	
SySR8	La base de datos almacenada en el servidor debe ser capaz de guardar un registro de los accesos.		x	

3.2.3 Requerimientos de arquitectura

Finalmente, otra parte esencial son los requisitos de arquitectura, lo que permiten establecer las bases sobre las cuales se construye la estructura del sistema. En la **Tabla 13** se

detallan los requisitos de arquitectura con el fin de proporcionar una visión clara del sistema, para facilitar la comprensión de como interactúan las diferentes partes. Al igual que los requisitos de sistema son necesarios para la planificación y el diseño.

Tabla 13

Requerimiento de arquitectura.

ArSR				
N °	**Requerimiento**	**Prioridad**		
		Baja	**Media**	**Alta**
	Software			
ArSR1	El IDE de Arduino debe ser compatible con el hardware (microcontroladores) a utilizar para el sistema.			x
ArSR2	La aplicación debe realizarse en una plataforma de fácil manejo y experiencia previa.			x
ArSR3	La APP debe ser de fácil manejo para el usuario.			x
ArSR4	La APP debe contar con una interfaz gráfica fácil de usar, con menús simples.			x
	Hardware			
ArSR5	El sistema embebido debe permitir la conexión con la cerradura eléctrica, el manejo de un teclado matricial, lector de tarjetas RFID, transmisor y receptor de RF.			x
ArSR6	El sistema debe tener una pantalla o interfaz para el usuario.		x	

ArSR7	El sistema de acceso residencial debe contar con un microcontrolador capaz de soportar la programación de distintos métodos de autenticación.		x
ArSR8	El microcontrolador para la llave electrónica debe ser de pequeñas dimensiones.		x
ArSR9	El microcontrolador debe ser de bajo consumo energético tanto para la llave electrónica y el sistema.		x
ArSR10	El microcontrolador para el sistema debe ser de bajo costo.	x	
ArSR11	El microcontrolador para el sistema debe tener tecnología Wi-Fi.		x
ArSR12	El microcontrolador para la llave debe soportar las funciones de hash.	x	

3.3 Segunda fase: Diseño del sistema

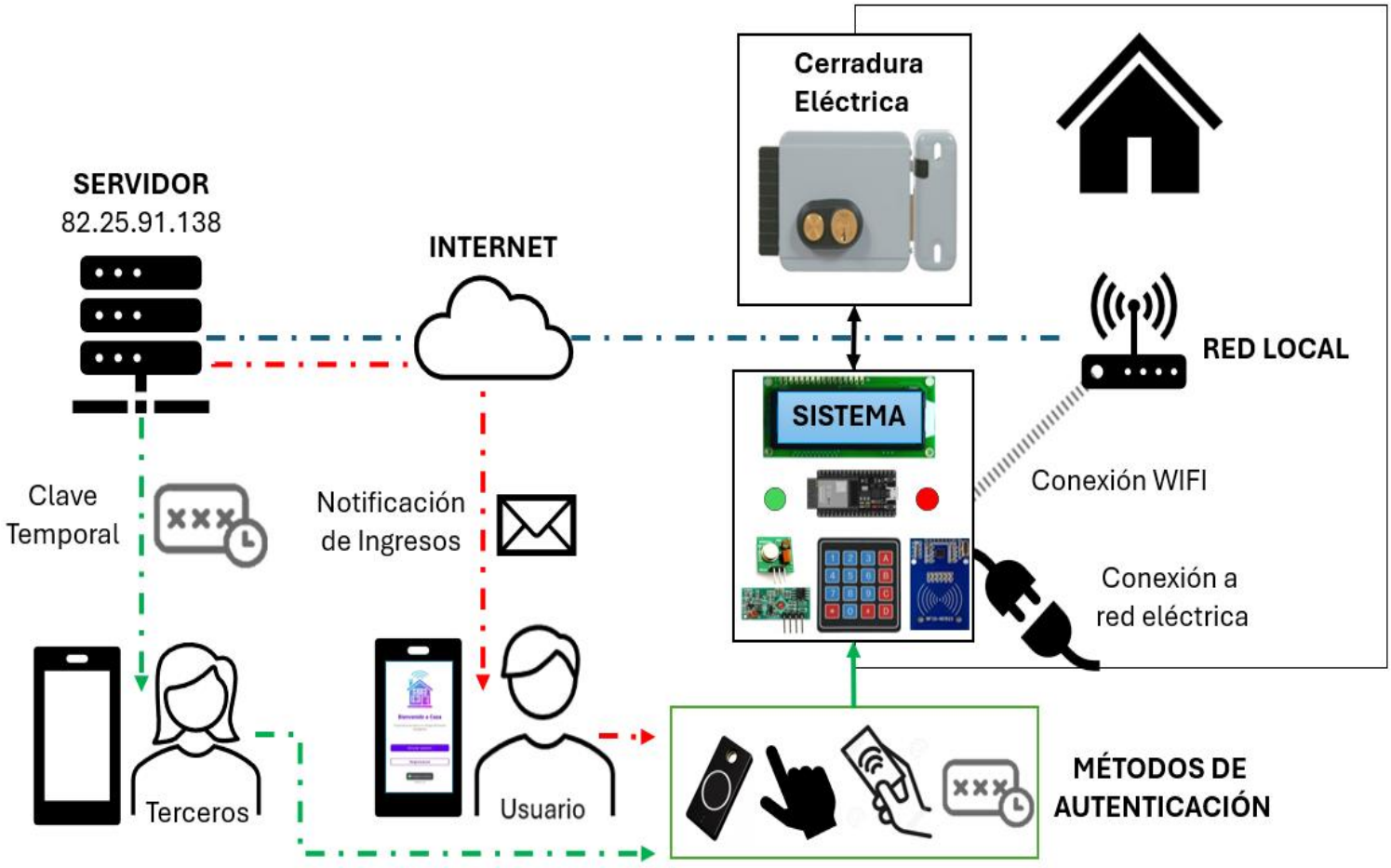
La idea general del presente proyecto es implementar diferentes métodos de autenticación en cerraduras eléctricas tradicionales, entre los métodos propuestos se encuentra el mecanismo de **autenticación por contraseña (PIN)**, el cual permite ingresar un PIN de 6 dígitos. Este PIN puede ser modificado por al usuario cuando lo desee a través de la aplicación. Otro método de autenticación es el **acceso por Tarjetas RFID**, cada integrante del domicilio puede registrar una o más tarjetas para su acceso personal. Asimismo, se implementará el

método de **autenticación por llave de circuito programable**, que utiliza módulos de radiofrecuencia (RF) para la comunicación.

Adicional, se incluye el método de **acceso mediante clave temporal aleatoria**, generada desde la aplicación y enviada por SMS. Esta clave tiene una validez limitada y está diseñada para otorgar acceso a terceros de forma segura y controlada. Finalmente, para garantizar la seguridad, el sistema envía **notificaciones de los accesos** de cada integrante. En la **Figura 17** se muestra un ejemplo del sistema con todos los mecanismos de acceso.

Figura 17

Sistema de acceso para entornos residenciales.



Nota. En la Figura se observa la interacción de múltiples métodos de autenticación con el usuario principal, así como la generación de claves temporales para el ingreso de terceros.

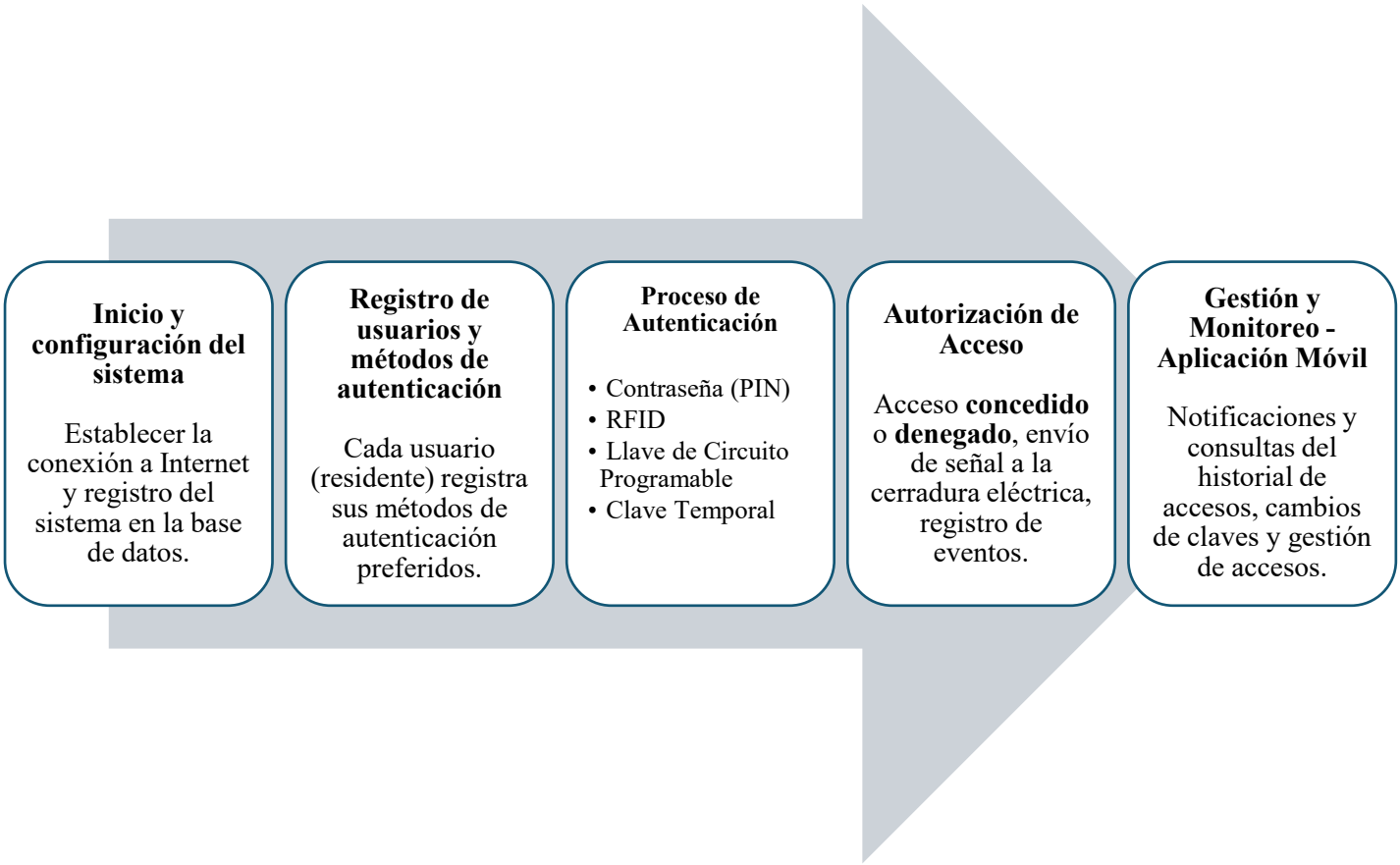
3.3.1 Diagrama de bloques del sistema

El sistema basado en múltiples métodos de autenticación para el acceso residencial se divide en varios procesos o etapas involucrando el almacenamiento de información en la base de datos, registro de los usuarios en la aplicación, generación de claves aleatorias, interconexión con sistemas eléctricos, entre otros.

La **Figura 18** representa el diagrama general del sistema, en donde se describen la secuencia de los procesos principales del sistema, posteriormente se profundizará a detalle cada uno de estos.

Figura 18

Diagrama de bloques del sistema.



Nota. En la Figura se observa los procesos generales a utilizar para el acceso a entornos residenciales, los procesos de registro de usuarios y métodos de autenticación pueden variar y ser aplicados para cualquier tipo de entorno.

3.3.2 Selección de hardware y software

En este apartado se procede a seleccionar tanto el hardware como el software necesario para el diseño del sistema, con el objetivo de cumplir los requisitos establecidos y de esta manera confirmar si es la opción más adecuada para garantizar un sistema eficiente, seguro y adaptable a los requerimientos del proyecto.

Para la evaluación de cada componente, se aplica un sistema de valoración binario, 1 que determina que el componente cumple con el requisito y 0 indica que no cumple.

3.3.2.1 Sistema general (hardware)

Cerraduras

Para el sistema general se debe seleccionar el tipo de cerradura que se acople al diseño o requerimientos necesarios. Por ello, se ha elegido cuatro tipos de cerraduras: eléctrica, mecánica, biométrica y magnética. Cada una ha sido analizada en función de su capacidad para cumplir con requisitos primordiales relacionados con facilidad de uso, métodos de autenticación, conectividad, autonomía energética y compatibilidad con el hardware del sistema. Además, la elección final depende de la cantidad de requisitos cumplidos, priorizando aquellos que garanticen la funcionalidad, seguridad y adaptabilidad del diseño.

Tabla 14

Selección de cerradura

	Requisitos									
Hardware	**StSR2**	**StSR11**	**StSR13**	**StSR14**	**StSR15**	**StSR17**	**SySR1**	**SySR2**	**ArSR5**	**Valoración**
Cerradura eléctrica	1	1	1	1	1	1	1	1	1	9
Cerradura mecánica	1	1	1	0	0	0	0	1	0	4
Cerradura biométrica	1	0	1	1	0	1	0	0	1	5
Cerradura magnética	1	0	1	0	0	0	0	0	0	2

Nota. En la **Tabla 14,** se puede apreciar que la cerradura eléctrica es la que cumple con la mayor cantidad de requisitos establecidos, obteniendo la valoración más alta en comparación con las demás opciones evaluadas. Este tipo de cerradura destaca por satisfacer aspectos clave, como facilidad de uso (StSR2), tamaño compacto (StSR13), soporte para múltiples métodos de autenticación (StSR14) y conectividad para acceso remoto (StSR15). Por el contrario, las cerraduras mecánica, biométrica y magnética no cumplen con un número suficiente de requisitos, lo que no las hace óptimas para el diseño del sistema.

Placas de desarrollo

La elección de la unidad de control programable es fundamental para la integración de múltiples componentes. Con el objetivo de asegurar el cumplimiento de los requisitos del sistema, se han evaluado diversas opciones de hardware, incluyendo el ESP32, Arduino Uno, Raspberry Pi y Arduino Mega. Esta evaluación se realizó en función de criterios específicos, representados como requisitos (StSR21, StSR25, SySR1, etc.), asignando a cada placa una puntuación basada en su capacidad para satisfacerlos.

Tabla 15

Placas de desarrollo

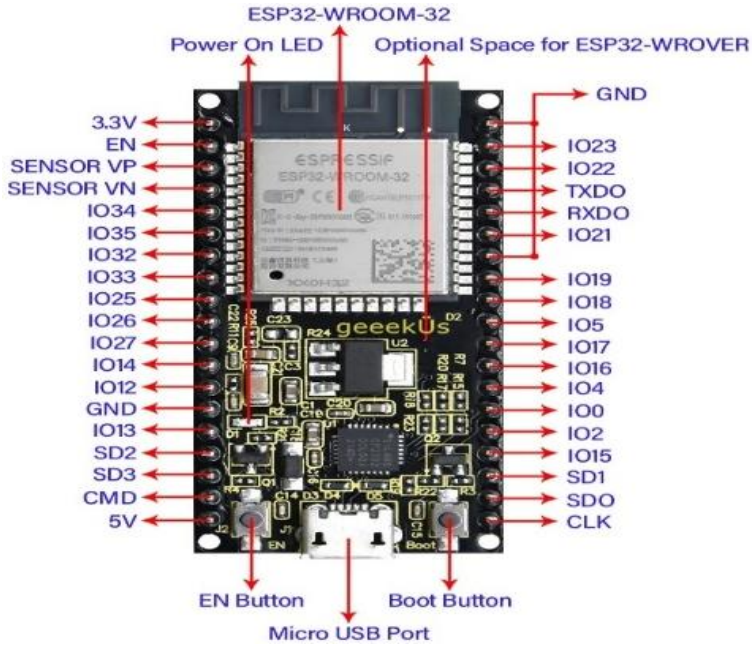
	Requisitos									
Hardware	**StSR21**	**StSR25**	**SySR1**	**SySR3**	**SySR4**	**ArSR1**	**ArSR7**	**ArSR9**	**ArSR11**	**Valoración**
ESP32	1	1	1	1	1	1	1	1	1	9
Arduino uno	1	1	1	1	1	1	1	0	0	7
Raspberry Pi	1	1	1	1	1	0	1	0	1	7
Arduino Mega	1	1	1	1	1	1	1	0	0	7

Nota. En este contexto, en la **Tabla 15** el ESP32 se perfila como la opción más adecuada, se destacó al obtener una valoración total de 9 puntos, lo que indica su alta compatibilidad con los requisitos establecidos. Además, es capaz de integrar fácilmente componentes adicionales, conectividad y manejo de periféricos. Por otro lado, tanto Arduino Uno, Raspberry Pi, Arduino Mega también su puntuación es de 7 puntos. Aunque presentan un buen nivel de cumplimiento de los requisitos, su capacidad en términos de conectividad o rendimiento puede limitar su efectividad en comparación con el ESP32 en este proyecto.

Siendo el ESP32 la principal opción para el desarrollo del sistema, es necesario detallar las características principales de esta placa de desarrollo.

Figura 19

ESP-WROOM32 pines.



Nota. Obtenido de (ABRA, 2025).

Es importante que la información técnica en este caso de este tipo de placas electrónicas sea obtenida de su ficha técnica oficial (datasheet). En la siguiente **Tabla 16** se destacó las características generales de operación y funcionamiento de la placa ESP-WROOM32.

Tabla 16

ESP-WROOM32 especificaciones.

Categorías		Ítems	Detalles
Wi-Fi		Protocolos	802.11b/g/n (802.11n arriba de 150 Mbps)
		Rango de Frecuencia en la que opera en la banda de 2.4GHz.	2412 ~ 2484 MHz
Operación de	Voltaje y corriente	Voltaje de operación	3.0V ~ 3.6V
		Corriente	Un promedio de 80mA
		Corriente mínima desde fuente de alimentación	500mA

Procesador	Arquitectura	Dual-Core Xtensa 32-bit LX6, bajo consumo.
	Núcleos	2 núcleos independientes, multitarea eficiente.
	Rendimiento	Ideales para aplicaciones IOT
Memoria	ROM	448 KB, utilizada para funciones de arranque y operaciones básicas.
	SRAM principal	520 KB para datos y ejecución de instrucciones.

Nota. Elaboración propia. Modificado de (Espressif Systems, 2023).

3.3.2.2 Llave electrónica

Hardware (Microcontrolador)

Para el diseño de la llave de circuito programable, se van a evaluar distintas opciones de microcontroladores Arduino Nano, Arduino Mini, Raspberry Pi Pico y ATtiny85. Entre los aspectos a analizar es la capacidad de modificar códigos de acceso (StSR18), la portabilidad del dispositivo (StSR19), el soporte de alimentación externa (StSR20), la independencia de conectividad (StSR24), el tamaño compacto (ArSR1), el bajo consumo energético (ArSR9), el costo accesible (ArSR10) y la capacidad de memoria (ArSR12).

Tabla 17

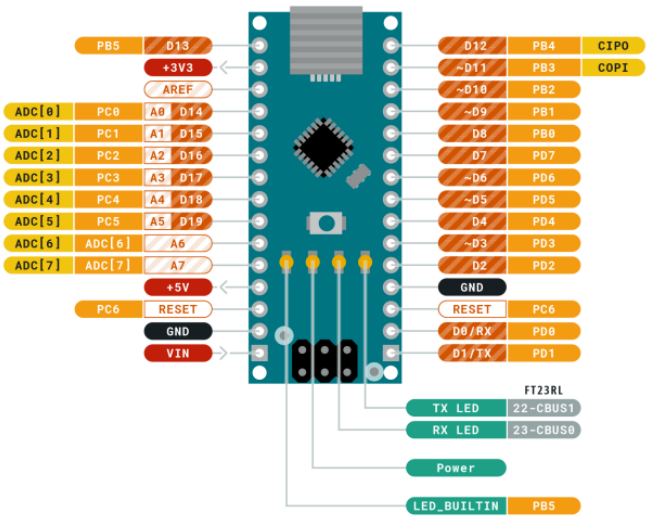
Microcontroladores – llave electrónica.

	Requisitos									
Hardware	**StSR18**	**StSR19**	**StSR20**	**StSR24**	**ArSR1**	**ArSR8**	**ArSR9**	**ArSR10**	**ArSR12**	**Valoración**
ATtiny85	1	1	1	1	1	1	1	1	0	8
Arduino Mini	1	0	1	1	1	0	0	0	0	4
Arduino Nano	1	1	1	1	1	1	1	1	1	9
Raspberry pi pico	1	0	1	1	1	0	0	0	0	4

Nota. Después de evaluar cada hardware en la **Tabla 17**, se destaca que el Arduino Nano cumple con todos los requisitos establecidos, obteniendo la máxima valoración de 9 puntos, lo que lo posiciona como la opción más adecuada para el diseño de la llave electrónica programable.

Figura 20

Esquema Arduino nano.



Nota. Obtenido de (Arduino, 2025).

Tabla 18

Características de importancia del microcontrolador Arduino Nano

Categoría	Ítem	Detalle
Voltaje de operación	Regular	5V (PIN 27)
	Irregular	Rango de 7-15V (PIN 30)
Entradas/Salidas	Pines	20 digitales, 8 analógicos, 6 salidas PWM
Memoria	Flash	32 KB Flash
Consumo energético	Modo	Bajo consumo
Categoría	Placa	Microcontrolador de 8 bits ATmega328 a 16 MHz.

Nota. Elaboración propia. Especificaciones obtenidas de (Arduino, 2025).

Software de diseño electrónico

Para el diseño del circuito de la llave de circuito programable, se evaluaron diferentes opciones de software en función de su capacidad para cumplir con los requerimientos establecidos, como la posibilidad de modificar códigos de acceso (StSR18), portabilidad del diseño (StSR19) y compatibilidad con fuentes de alimentación externas (StSR20). Se analizaron cuatro alternativas: Proteus, Autodesk Eagle, OrCAD y Altium Designer. La selección del software se basa en la cantidad de requisitos cumplidos, priorizando aquellos que aseguren precisión, funcionalidad y facilidad de uso en el desarrollo del sistema.

Tabla 19

Software de diseño de placas.

	Requisitos			
Software	**StSR18**	**StSR19**	**StSR20**	**Valoración**
Proteus	1	1	1	3
Autodesk Eagle	0	1	0	1
OrCAD	0	1	0	1
Altium Designer	0	1	0	1

Nota. En la **Tabla 19** se concluye que Proteus es el software que cumple con todos los requerimientos establecidos, obteniendo la máxima valoración. Su versatilidad y capacidad para gestionar modificaciones en los códigos de acceso (StSR18), así como su portabilidad (StSR19), lo convierten en la opción ideal para el diseño del circuito. Por otro lado, las alternativas Autodesk Eagle, OrCAD y Altium Designer cumplen únicamente con un requisito y presentan limitaciones significativas, por lo que no se consideran óptimas para este proyecto.

3.3.2.3 Servidor

Para garantizar la operatividad, conectividad y seguridad del sistema, se evaluaron distintas plataformas de servidores en la nube en función de su capacidad para cumplir con los requerimientos establecidos. Entre los aspectos considerados se encuentran la conexión constante a internet (StSR4), la operación ininterrumpida del servidor (StSR25), la gestión de pagos y mantenimiento (StSR26), así como la posibilidad de administrar conexiones, claves y notificaciones (SySR5, SySR6, SySR7 y SySR8). Las plataformas para evaluar son Hostinger, Amazon Web Services (AWS), Microsoft Azure y Google Cloud Platform (GCP).

Tabla 20

Plataformas de computación en la nube.

	Requisitos							
Software	**StSR4**	**StSR25**	**StSR26**	**SySR5**	**SySR6**	**SySR7**	**SySR8**	**Valoración**
Hostinger	1	1	1	1	1	1	1	7
Amazon Web Services (AWS)	1	1	0	1	1	1	1	6
Microsoft Azure	1	1	0	1	1	1	1	6
Google Cloud Platform (GCP)	1	1	0	1	1	1	1	6

Nota. En base a la **Tabla 20,** se destaca que Hostinger es la plataforma que cumple con la mayor cantidad de requisitos evaluados, obteniendo la valoración más alta con 7 puntos. Esto la posiciona como la opción más adecuada para alojar el servidor en la nube del sistema.

Hostinger ofrece algunos servicios, de los cuales el tipo de servicio adecuado dependerá de las necesidades que el proyecto requiera. En la siguiente **Tabla 21** se presenta una tabla con los servicios que ofrece Hostinger y sus principales características de cada plan.

Tabla 21

Tipo de servicios de Hostinger.

Tipo de servicio Hostinger	Plan	Espacio SSD	Recursos	AB (Ancho de banda)
Web hosting	Single	50 GB	Hasta dos (limitado)	100 GB
	Premium	100 GB	Ilimitadas	Ilimitado
	Business	200 GB	Ilimitadas	Ilimitado
Cloud hosting	Startup	200 GB	Ilimitadas	Ilimitado
	Profesional	250 GB	Ilimitadas	Ilimitado
VPS Hosting	VPS 1	20 GB	Ilimitadas	1 TB
	VPS 2	40 GB	Ilimitadas	2 TB

Nota. Elaboración propia. Basado de (HOSTINGER, 2025).

3.3.2.4 Aplicación

Para el desarrollo de la aplicación del sistema, se evaluaron diferentes entornos de programación en función de su capacidad para cumplir con los requisitos establecidos. Entre los aspectos considerados se encuentran la facilidad de ingreso de datos mediante la aplicación (StSR1), una interfaz amigable e intuitiva (StSR3), la recepción de notificaciones (StSR5), la visualización del historial de accesos (StSR6) y la gestión de claves temporales (StSR8). También se analizaron requerimientos adicionales como la capacidad de realizar cambios de claves desde la aplicación (StSR9), la gestión de registros de acceso (StSR12) y el diseño de una interfaz gráfica fácil de usar (ArSR2, ArSR3, ArSR4). Las herramientas evaluadas fueron Android Studio, Flutter, React Native y Xamarin, considerando su valoración total según los requisitos cumplidos.

Tabla 22

Software para aplicación.

Software	StSR1	StSR3	StSR5	StSR6	StSR8	StSR9	StSR12	ArSR2	ArSR3	ArSR4	Valoración
Android Studio	1	1	1	1	1	1	1	1	1	1	10
Flutter	1	1	1	1	1	1	1	0	1	1	9
React Native	1	1	1	1	1	1	1	0	1	1	9
Xamarin	1	1	1	1	1	1	1	0	1	1	9

Nota. En la **Tabla 22** se observa que Android Studio cumple con todos los requisitos evaluados, alcanzando la valoración máxima de 10 puntos. Por otro lado, Flutter, React Native y Xamarin también muestran un desempeño destacado, cumpliendo con 9 de los 10 requisitos evaluados, pero presentan limitaciones menores, la aplicación debe realizarse en una plataforma de fácil manejo y experiencia previa (ArSR2). Aunque todas son opciones viables, Android Studio se postula como la alternativa más completa y adecuada.

La siguiente **Tabla 23** detalla información general, esta plataforma no se limita a ciertos sistemas operativos.

Tabla 23

Características Android Studio.

	Ítem	Detalle
Plataforma	Windows(64bits)	1.2GB
	Mac(64bits)	1.3GB
	Linux(64bits)	1.3GB
	Chrome OS	1.0GB
Compatibilidad	Android	Android

Nota. Elaboración propia, Información obtenida de (ANDROID STUDIO, 2025).

3.3.3 Diseño de llave de circuito programable

La Llave de Circuito Programable representa un mecanismo de acceso adicional dentro del sistema propuesto, diseñado para complementar los métodos ya existentes: tarjeta RFID (*identificación por radiofrecuencia*), teclado físico (*entrada por pulsaciones*), llave mecánica y clave temporal (*enviada por SMS*). Este tipo de llave electrónica se basa en la transmisión de señales mediante ondas electromagnéticas de mayor alcance, utilizando módulos de radiofrecuencia RF de 433 MHz (transmisor y receptor).

Además de servir como un método de acceso, esta llave electrónica cumple el rol de **dispositivo cliente** en un **protocolo de autenticación seguro** basado en el mecanismo de **desafío-respuesta**. Esto significa que, antes de permitir el acceso, el sistema pone a prueba a la llave, la cual debe responder correctamente para demostrar que posee una contraseña válida, sin necesidad de enviarla directamente.

Para lograrlo, se utiliza un método llamado ***salt-hash***[9], en el cual el sistema genera un número aleatorio (salt) que se combina con una contraseña predefinida para luego ser procesado mediante una función hash. Este valor hasheado se transmite para su verificación, garantizando así la integridad y confidencialidad de la credencial sin exponer directamente la clave original. Su propósito es garantizar la autenticación confiable del usuario antes de habilitar el acceso físico a la residencia.

En este apartado se presenta el diseño y desarrollo de la llave de circuito programable, abarcando cada una de sus etapas clave. En primer lugar, se describe el funcionamiento lógico, así como las funciones utilizadas, diseño del circuito electrónico y la elaboración del PCB, fundamentales para la integración de los componentes. Posteriormente, se detalla el proceso de fabricación, que incluye la impresión del circuito y la preparación de la placa.

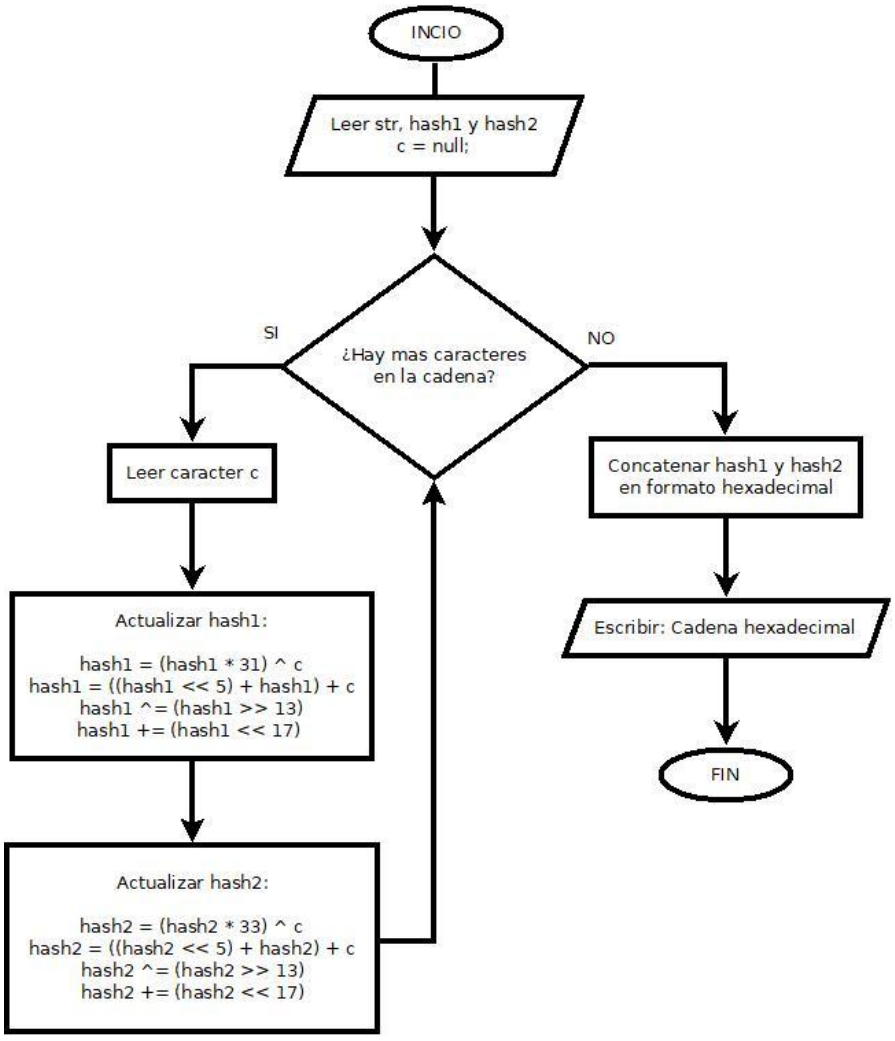
[9] Técnica criptográfica en la que se añade un valor aleatorio (salt) a una contraseña antes de aplicar una función hash.

3.3.3.1 Generación de Hashes

Para que la llave electrónica sea capaz de utilizar funciones de resumen (*hash*) se diseñó la función **MD5_16(…)**, esta es una implementación ligera de un algoritmo **MD5** y diseñada para generar una cadena hexadecimal de 16 caracteres a partir de una entrada de texto. Su propósito principal es ser utilizada en entornos con recursos limitados, como microcontroladores Arduino Nano o ATtiny85, donde algoritmos de hashing más complejos (como MD5 o SHA-256 completos) podrían exceder la capacidad de procesamiento o memoria disponible. la **Figura 21** muestra el diagrama de flujo de esta función.

Figura 21

Diagrama de flujo MD5_16.



A continuación, se detalla el funcionamiento del **Algoritmo 1** llamado MD5_16(..) Esta función emplea una serie de operaciones bit a bit y multiplicaciones para generar dos valores hash intermedios (hash1 y hash2), que luego se combinan y se formatean como una cadena hexadecimal. Esta función es adecuada para aplicaciones que requieren una identificación única de datos o una verificación de integridad simple en entornos embebidos.

```
Input: (char *) str
Output: (char *) output

 (1)   hash1 ← 475381
 (2)   hash2 ← 0x12345678
 (3)   While ((c = *str++) != '\0')
 (4)      hash1 ← (hash1 * 31) XOR c
 (5)      hash2 ← (hash2 * 33) XOR c
 (6)      hash1 ← (hash1 << 5 + hash1) + c
 (7)      hash2 ← (hash2 << 5 + hash2) + c
 (8)      hash1 ← hash1 XOR (hash1 >> 13)
 (9)      hash2 ← hash2 XOR (hash2 >> 13)
(10)      hash1 ← hash1 + (hash1 << 17)
(11)      hash2 ← hash2 + (hash2 << 17)
(12)   output ← sprintf(hash1 & hash2)
(13)   return output
```

Algoritmo 1. Generación de hash hexadecimal de 16 caracteres

3.3.3.2 Diagrama de funcionamiento llave electrónica

En la **Figura 22** se muestra el flujo de funcionamiento del sistema, que consta de los siguientes pasos:

Identificación y validación inicial:

- La llave electrónica cuenta con un identificador único (ID) $\boldsymbol{n}$ y una contraseña asociada $\boldsymbol{PASS_n}$.
- Por su parte, la cerradura almacena en su base de datos una lista de identificadores válidos ($\boldsymbol{IDs = n, n-1, n-2, \ldots}$) junto con las contraseñas correspondientes ($\boldsymbol{PASS_{BD}}$: $\boldsymbol{PASS_n, PASS_{n-1}, PASS_{n-2}, \ldots}$).

- Tanto la llave electrónica como la cerradura, utiliza el mismo proceso para la generación del hash $\boldsymbol{h_{md5}(x)}$, utilizando una versión de $\boldsymbol{md5}$ diseñada para equipos con bajo rendimiento, **Algoritmo 1** muestra el proceso para la generación del hash.

Inicio del proceso de autenticación:

- Cuando se presiona el botón en la llave electrónica, esta envía su identificador $\boldsymbol{n}$ a la cerradura a través del medio de comunicación inalámbrico.
- La cerradura, al recibir este ID $\boldsymbol{n}$, verifica si está registrado en su base de datos. Si el ID no es válido, la cerradura permanece en modo de escucha sin realizar ninguna acción adicional. Si el ID es válido, genera un código aleatorio de 10 caracteres $\boldsymbol{RANDOM_n}$, que es enviado a la llave electrónica como parte del proceso de autenticación.
- La llave electrónica recibe el código aleatorio $\boldsymbol{RANDOM_n}$ enviado por la cerradura y responde con un acuse de recibido $\boldsymbol{ACK_n}$. Si la cerradura no recibe este acuse en un tiempo determinado $\boldsymbol{t}$, reenvía el mismo código aleatorio hasta un máximo de tres intentos.
- Una vez que la cerradura recibe el acuse de recibido, ambas partes combinan la contraseña $\boldsymbol{PASS_n}$ con el código aleatorio $\boldsymbol{RANDOM_n}$ y utilizan la función $\boldsymbol{h_{md5}(x)}$ para generar un hash único $\boldsymbol{HASH_n}$.

Validación del hash:

- Ambas partes (llave y cerradura) generan el hash $\boldsymbol{HASH_n}$ de manera independiente utilizando la misma combinación $\boldsymbol{PASS_n + RANDOM_n}$. La cerradura almacena este valor temporalmente durante el proceso de autenticación.

- La llave una vez ha generado el hash $\boldsymbol{HASH_n}$ lo envía a la cerradura. Al recibir el hash, la cerradura responde con un acuse de recibido $\boldsymbol{ACK_{CI}}$ para confirmar la recepción correcta del hash por parte de la llave electrónica.

Finalización del proceso:

- Si la llave electrónica recibe el acuse de recibido $\boldsymbol{ACK_{CI}}$, elimina de su memoria los valores temporales $\boldsymbol{RANDOM_n}$ y $\boldsymbol{HASH_n}$, dando por finalizado su parte del proceso de autenticación. Si la llave no recibe el acuse en un tiempo determinado, reenvía el hash hasta un máximo de tres intentos.
- La cerradura valida el hash recibido $\boldsymbol{HASH_n}$, si este es válido, procede a abrir la puerta y elimina de su memoria los valores temporales generados $\boldsymbol{RANDOM_n}$ y $\boldsymbol{HASH_n}$, concluyendo de esta forma el proceso de autenticación.

Ejemplo:

- El identificador único de la llave electrónica $\boldsymbol{n}$ es $\mathbf{001}$ y su contraseña asociada $\boldsymbol{PASS_n}$ corresponde a un dato único, como un número de cédula o un valor similar; por ejemplo, $\mathbf{1001234567}$.
- En primer lugar, la llave electrónica envía su ID a la cerradura $\mathbf{001}$, y esta verifica si el identificador está registrado en su base de datos. Si el ID es válido, la cerradura genera un código aleatorio de 10 caracteres, como $\boldsymbol{abcdf123456}$ denotado como $\boldsymbol{RANDOM_n}$, y lo envía de vuelta a la llave como parte del proceso de autenticación.

$$\boldsymbol{Llave} \Rightarrow \quad \mathbf{001} \quad \Rightarrow \boldsymbol{Cerradura} \qquad \text{(Ej. 1)}$$

$$\boldsymbol{Llave} \Leftarrow \quad \boldsymbol{abcdf123456} \quad \Leftarrow \boldsymbol{Cerradura} \qquad \text{(Ej. 2)}$$

- A continuación, la llave electrónica combina el valor de su contraseña $\boldsymbol{PASS_n}$ con el código aleatorio recibido $\boldsymbol{RANDOM_n}$, formando la cadena $\mathbf{1001234567}\boldsymbol{abcdf}\mathbf{123456}$. Usando esta combinación, tanto la llave como la cerradura calculan de manera independiente un hash único $\boldsymbol{HASH_n}$ mediante un algoritmo previamente definido.

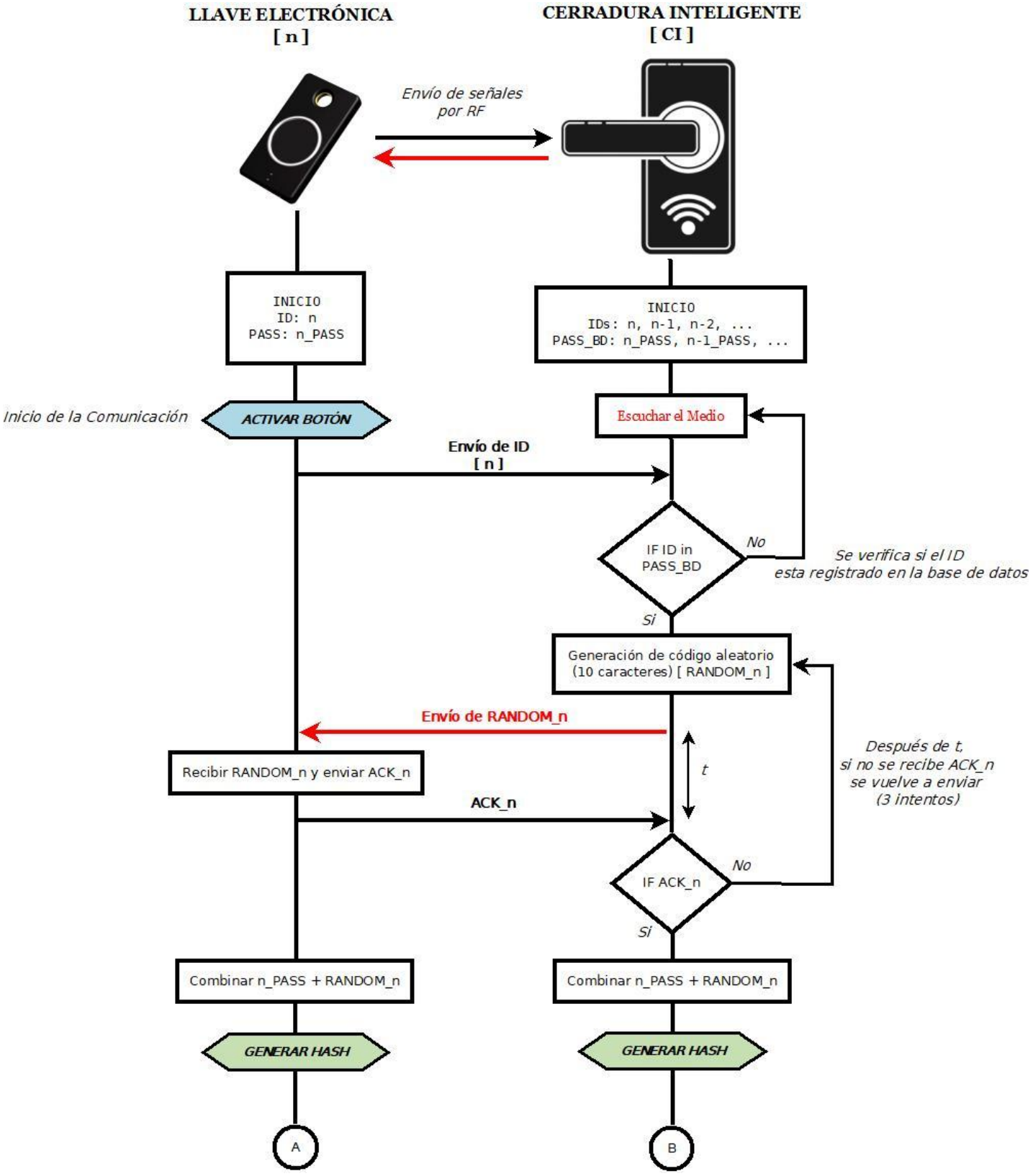
$$\boldsymbol{Llave} \Longrightarrow \boldsymbol{h_{md5}}(\mathbf{1001234567}\boldsymbol{abcdf}\mathbf{123456}) = \boldsymbol{ee}\mathbf{033}\boldsymbol{ea}\mathbf{411529}\boldsymbol{c}\mathbf{62} \quad \text{(Ej. 3)}$$

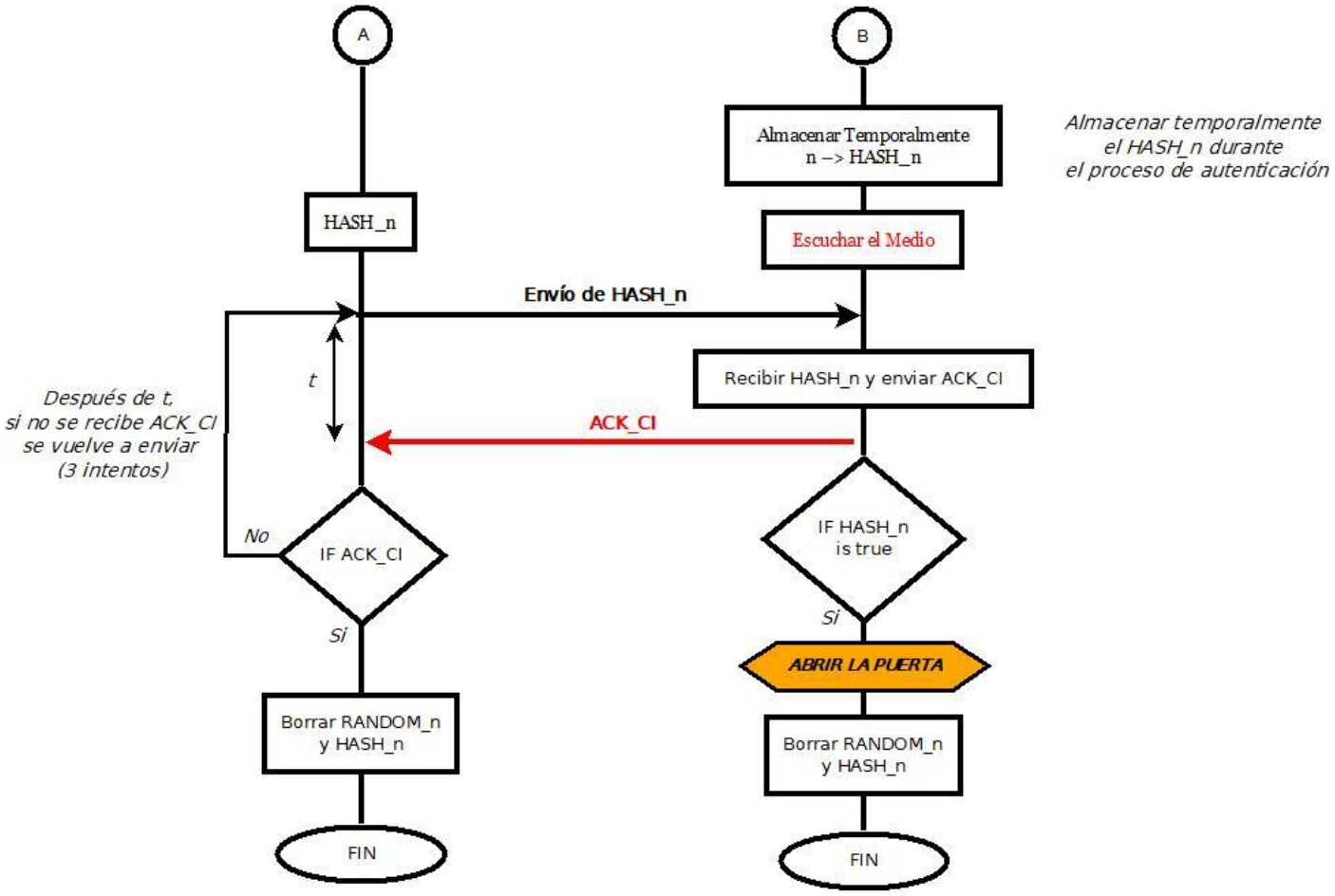
$$\boldsymbol{Cerradura} \Longrightarrow \boldsymbol{h_{md5}}(\mathbf{1001234567}\boldsymbol{abcdf}\mathbf{123456}) = \boldsymbol{ee}\mathbf{033}\boldsymbol{ea}\mathbf{411529}\boldsymbol{c}\mathbf{62} \quad \text{(Ej. 4)}$$

- Posteriormente, la llave envía el hash generado $\boldsymbol{HASH_n} = \boldsymbol{ee}\mathbf{033}\boldsymbol{ea}\mathbf{411529}\boldsymbol{c}\mathbf{62}$ a la cerradura. Una vez recibido, la cerradura lo compara con el valor que ella misma ha calculado. Si ambos hashes coinciden, la cerradura considera que la autenticación fue exitosa y procede a desbloquear el mecanismo de apertura de la puerta.
- Finalmente, para garantizar la seguridad y evitar la reutilización de los datos, tanto la llave como la cerradura eliminan de su memoria todos los valores temporales utilizados en el proceso, incluyendo el código aleatorio $\boldsymbol{RANDOM_n}$ y el hash $\boldsymbol{HASH_n}$. De esta forma, se completa el ciclo de autenticación de manera segura y eficiente.

Figura 22

Diagrama de flujo del funcionamiento de la llave electrónica.





Nota. En este sistema, el **hash**[10] y el **salt**[11] contribuyen significativamente a garantizar la seguridad del proceso de autenticación entre la llave electrónica y la cerradura. Esto protege la confidencialidad del sistema, ya que cualquier información interceptada por un atacante carece de utilidad sin el **salt** correspondiente y la contraseña original. Gracias a esta combinación, el sistema refuerza la autenticidad y la integridad del proceso, previniendo accesos no autorizados a la cerradura.

3.3.3.3 Diseño del Circuito

El desarrollo del circuito para la llave electrónica inicia con la integración de los principales componentes electrónicos. En este sentido, el microcontrolador *Arduino nano* actúa

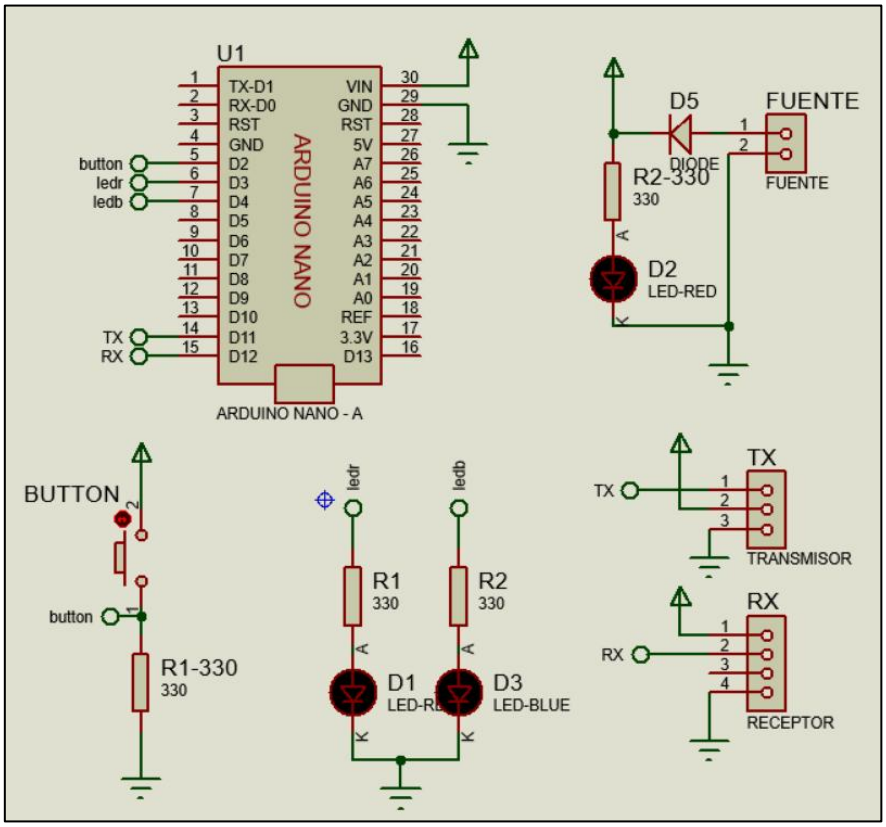
[10] **Hash:** es un valor único que se genera al procesar una entrada (archivo, contraseñas o texto).
[11] **Salt:** es un valor aleatorio que se añade a una entrada.

como la unidad central de control, permitiendo la interacción entre los diferentes elementos del sistema.

En la **Figura 23** se muestra el diseño realizado en Proteus donde se muestra la interconexión lógica de todos los componentes, para la comunicación inalámbrica se incorporaron dos conexiones de 3 y 4 pines para las antenas de transmisión y recepción de radiofrecuencia, además de un pulsador, el cual se utiliza como un activador para generar las señales de radiofrecuencia. Como elementos secundarios, se añadieron 2 LED que funcionan como indicadores visuales del estado del sistema. Finalmente, la fuente garantiza el suministro estable de energía necesario para el funcionamiento del sistema.

Figura 23

Componentes utilizados para el diseño de la llave electrónica.



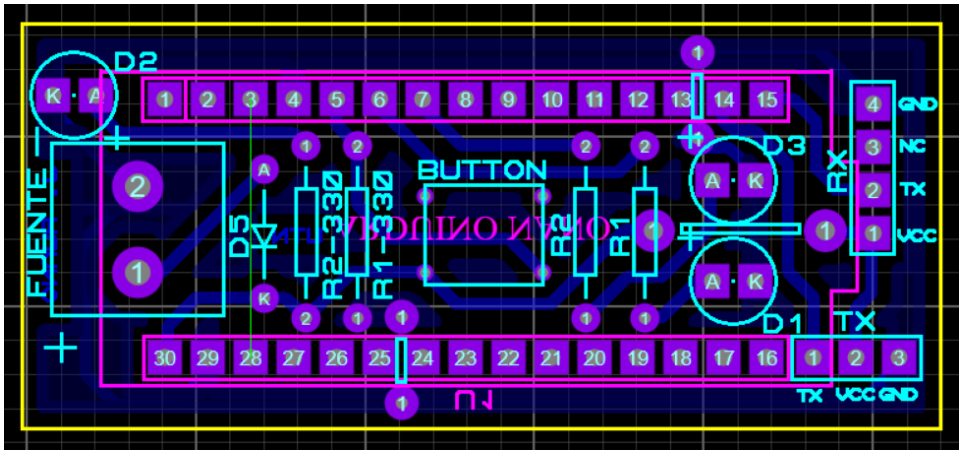
Nota. En conjunto, este diseño permite validar cada una de las funciones del sistema mediante simulaciones que garantizan la correcta interacción entre los componentes antes de proceder a la fabricación física de la placa.

3.3.3.4 Diseño del PCB

La **Figura 24** muestra el diseño de la placa de circuito impreso (PCB) para la implementación física de la llave de circuito programable. El trazado de las pistas se realizó considerando la correcta interconexión entre los componentes principales, y se optimizó la disposición de los elementos en la PCB para minimizar el espacio ocupado y garantizar una distribución eficiente de las señales y la energía. Las pistas fueron dimensionadas adecuadamente para soportar los requerimientos de corriente, y se aplicaron reglas de diseño que aseguran la separación entre pistas, reduciendo el riesgo de interferencias y cortocircuitos. Además, se integraron puntos de soldadura y etiquetado claro para facilitar el montaje y la identificación de los componentes durante el ensamble de la llave.

Figura 24

Diseño de la placa de circuito impreso (PCB) – llave de circuito.



Nota. En el diseño se muestra al Arduino nano en la cara posterior para optimizar el espacio de los componentes, permitiendo una disposición compacta de los elementos en la placa. Este

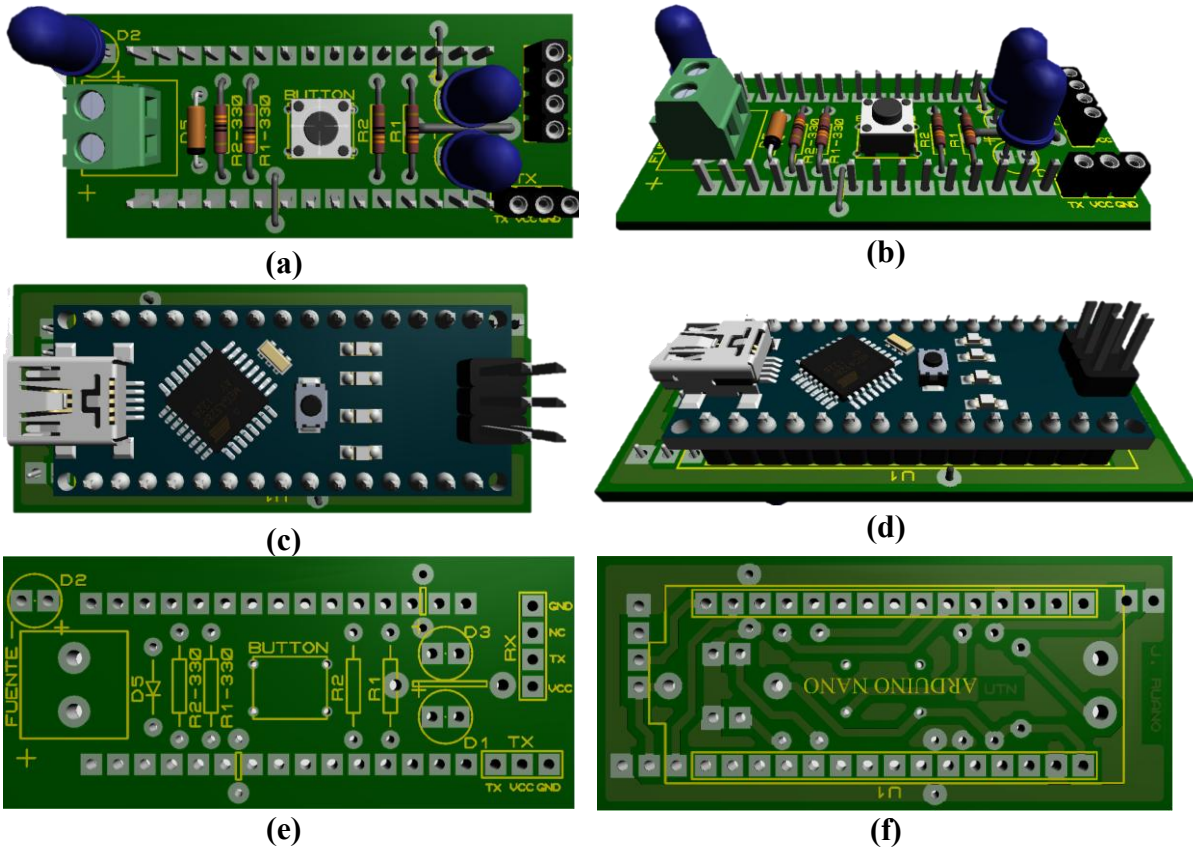
enfoque los pines del Arduino se conectan a través de pistas a los componentes en la cara frontal.

3.3.3.5 Diseño 3D

En la **Figura 25** esta etapa se presenta la vista tridimensional del montaje del PCB, donde se visualiza la distribución de los componentes electrónicos sobre la placa. Además, el diseño físico refleja la ubicación optimizada de cada elemento para asegurar la funcionalidad y la eficiencia del circuito. Adicional se observan las vistas reversas, en las cuales se aprecia el trazado de las pistas que interconectan los componentes.

Figura 25

Vista tridimensional del montaje del PCB.



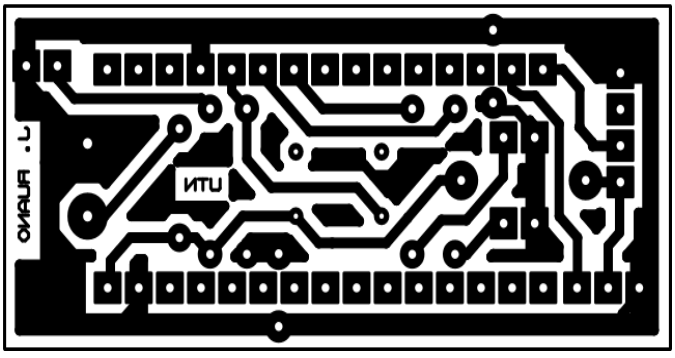
Nota. Las vistas tridimensionales **(a)** y **(b)** muestran los componentes principales del circuito montados en la cara superior del PCB, incluyendo el botón, resistencias y bornes de conexión. Las vistas **(c)** y **(d)** revelan la integración del Arduino Nano en la cara inferior, lo que facilita una solución compacta y funcional. Las vistas **(e)** y **(f)** muestra la serigrafia para el montaje de los componentes y el diseño de las pistas.

3.3.3.6 Impreso

La **Figura 26** presenta el diseño final del trazado de pistas para la placa de circuito impreso (PCB) correspondiente al sistema de la llave electrónica, el siguiente paso es la transferencia del patrón de las pistas a la placa de cobre virgen, para lo cual se empleará la **técnica de planchado**. Este método permite la transferencia precisa del tóner que define el circuito desde un papel especial al sustrato de cobre, preparando la placa para el proceso de grabado químico.

Figura 26

Diseño final del trazado de pistas – Llave de circuito programable.



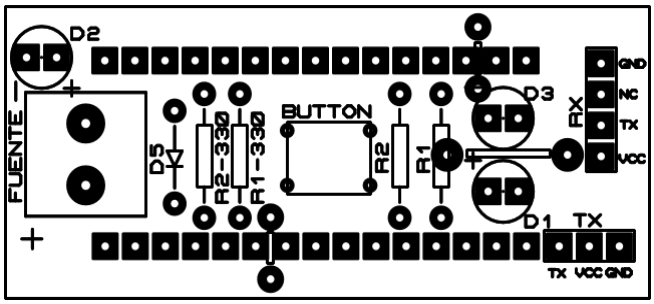
Nota. Es importante destacar que, para la fabricación de una placa de circuito impreso, existen diversas técnicas disponibles. Por ejemplo, se pueden enviar los archivos de diseño en formato

Gerber[12] a proveedores especializados para su producción profesional, lo que permite obtener PCBs de alta complejidad, doble capa o multicapa con acabados de máscara de soldadura y serigrafía. Sin embargo, en este caso específico y con fines de prototipado, se ha optado por implementar la técnica de planchado. Para asegurar una transferencia exitosa es crucial que la impresión de los patrones sea de alta calidad laser. La correcta aplicación de calor y presión durante el planchado garantizará que las pistas queden firmemente adheridas al cobre.

Para el diseño superior, se representa la serigrafía de la placa de la llave electrónica. En la **Figura 27** se detallan las posiciones exactas de los componentes. Además, el marcado incluye etiquetas que especifican las conexiones eléctricas, como GND, DATA y VCC, lo que facilita la interpretación del esquema durante el montaje..

Figura 27

Diseño superior (Serigrafía) – Llave de circuito programable.



Nota. La serigrafía no solo facilita el proceso de ensamblaje manual al indicar dónde va cada componente y en qué orientación, sino que también es crucial para la identificación de puntos de prueba y para futuras tareas de diagnóstico o reparación.

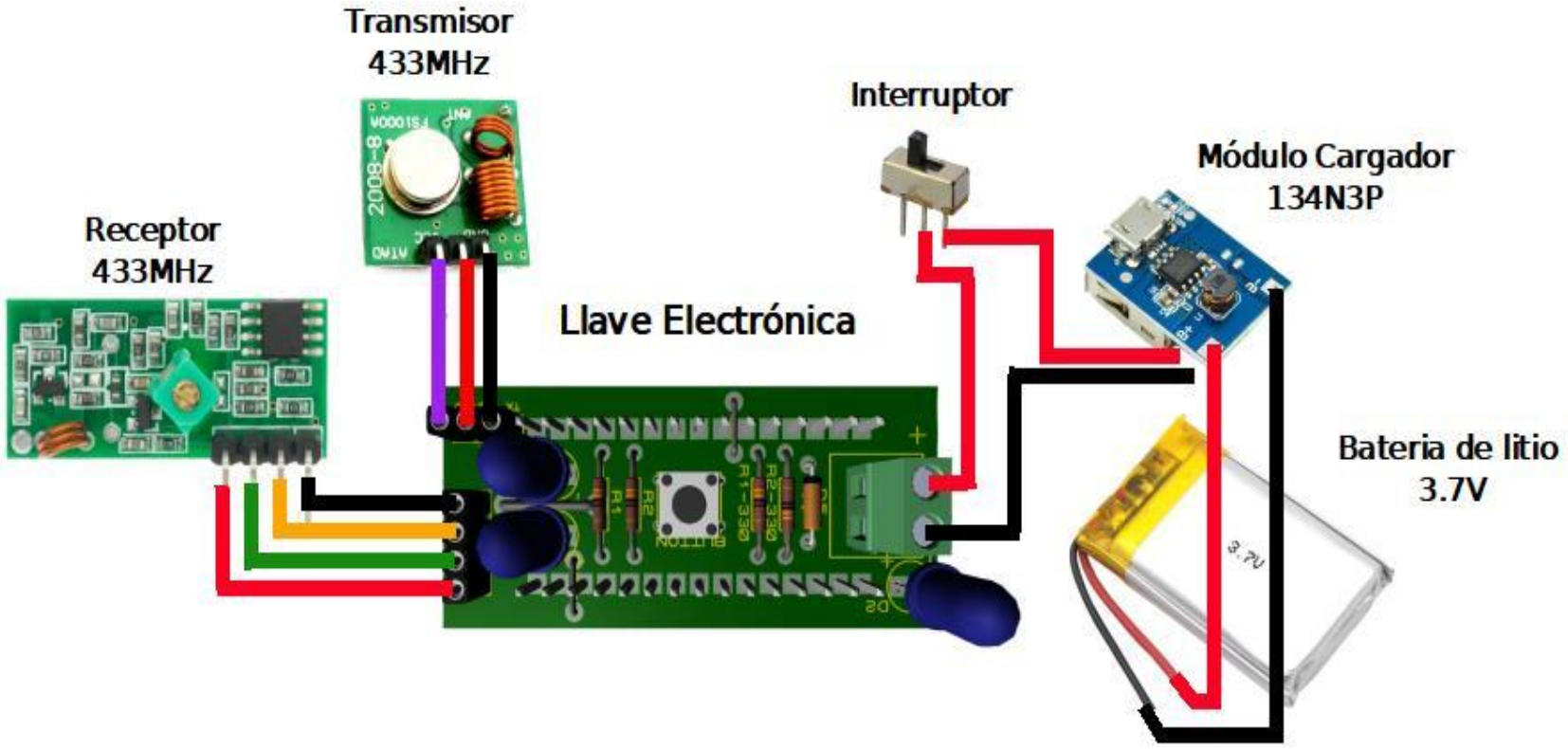
[12] El formato Gerber es el estándar de facto en la industria de la electrónica para describir las imágenes y la información de fabricación de las placas de circuito impreso (PCB).

3.3.3.7 Diagrama de conexiones

La **Figura 28** muestra el diagrama de conexiones eléctricas de los componentes externos que interactúan con la placa de control principal. Este esquema detalla cómo se interconectan el módulo transmisor, el módulo receptor, el módulo cargador de batería, la batería de litio de 3.7V y un interruptor, estableciendo el sistema de alimentación y comunicación inalámbrica del conjunto. La conexión entre los distintos módulos y el circuito principal está organizada mediante cables de diferentes colores. Esto facilita la identificación de las rutas de señal y alimentación.

Figura 28

Diagrama de conexiones de la llave de circuito programable.



Nota. El módulo cargador (134N3P) gestiona la carga de la batería de litio de 3.7V, la cual a su vez alimenta la placa de control a través de un interruptor para activar o desactivar el circuito. La placa es la encargada de suministrar el voltaje adecuado a todos los componentes incluido el microcontrolador.

3.3.4 Diseño de la cerradura inteligente

La cerradura electrónica es el componente central del sistema, ya que es la encargada de recibir las solicitudes de acceso de los diferentes métodos de autenticación **(PIN, tarjetas RFID, llave de circuito programable y clave temporal)**, validarlas y, en caso de ser correctas, activar el mecanismo de apertura. Además de gestionar las notificaciones de acceso, enviando estos datos a la aplicación y al correo electrónico de los usuarios.

El diseño de la cerradura electrónica se enfoca en la robustez, la seguridad y modularidad, permitiendo la integración de múltiples mecanismos de acceso de manera eficiente.

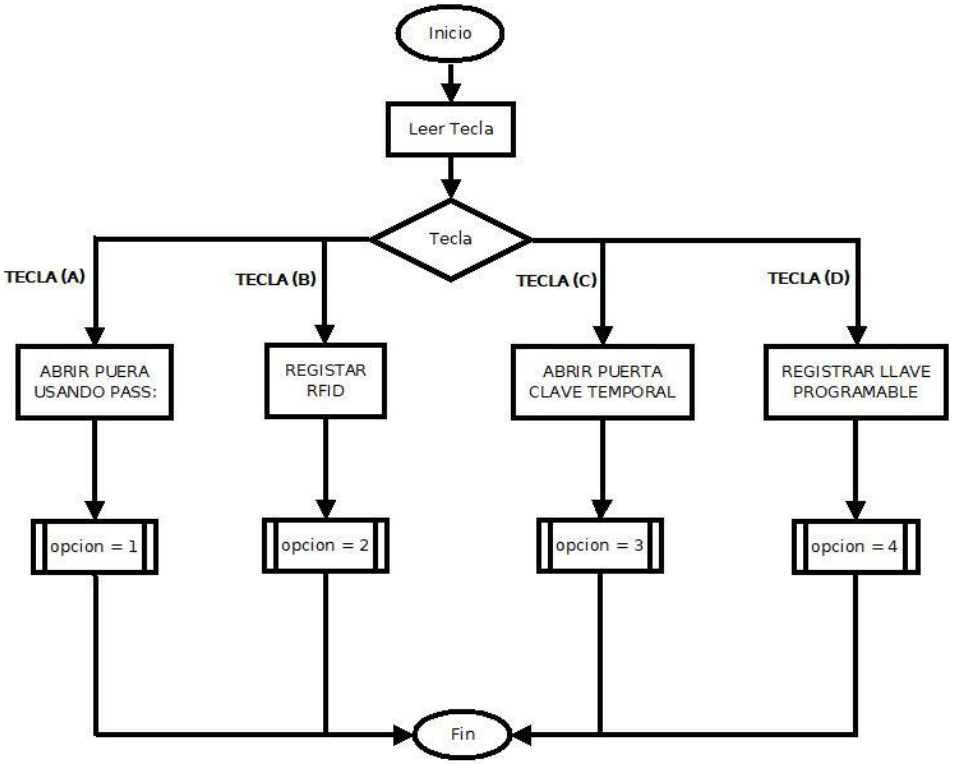
A continuación, se detallan las funciones, componentes y procesos involucrados en su diseño y funcionamiento:

3.3.4.1 Menú del sistema

Para que la cerradura sea capaz de gestionar los diferentes métodos de acceso cuenta con varios algoritmos, uno de ellos es la función de menú, la cual controla un sistema de menú interactivo mediante un teclado matricial y una pantalla LCD. Su propósito principal es presentar al usuario diferentes opciones de interacción con el sistema dependiendo de la tecla presionada **(A, B, C, D)**, se muestra las opciones en la pantalla LCD: **"Abrir puerta"**, **"Registrar nuevo RFID"**, **"Clave temporal"** o **"Registrar nueva llave"**, y establece un valor en la variable global $\boldsymbol{opcion}$, la cual se utiliza posteriormente en el programa principal para ejecutar la acción seleccionada. En la **Figura 29** se muestra el diagrama de flujo de la función $\boldsymbol{menu}(..)$.

Figura 29

Diagrama de flujo del menú del sistema.



A continuación, se detalla el funcionamiento del **Algoritmo 2** llamado $\boldsymbol{menu}(..)$ Esta función evalúa la tecla presionada y actualiza el valor de la $\boldsymbol{variable\ opcion}$ para que el sistema sepa qué función ejecutar.

```
Input: (char) customKey
Output: (int) opcion

 (1)   switch (customKey)
 (2)      case 'A':
 (3)         print(ABRIR PUERTA USANDO PASS)
 (4)         opcion ← 1
 (5)         break
 (6)      case 'B':
 (7)         print(REGISTRAR RFID)
 (8)         opcion ← 2
 (9)         break
(10)      case 'C':
(11)         print(ABRIR PUERTA CLAVE TEMPORAL)
(12)         opcion ← 3
(13)         break
(14)      case 'D':
(15)         print(REGISTRAR LLAVE PROGRAMABLE)
(16)         opcion ← 4
```

```
(17)         break
(18)      default:
(19)         opcion ← 0
(20)         Break
```

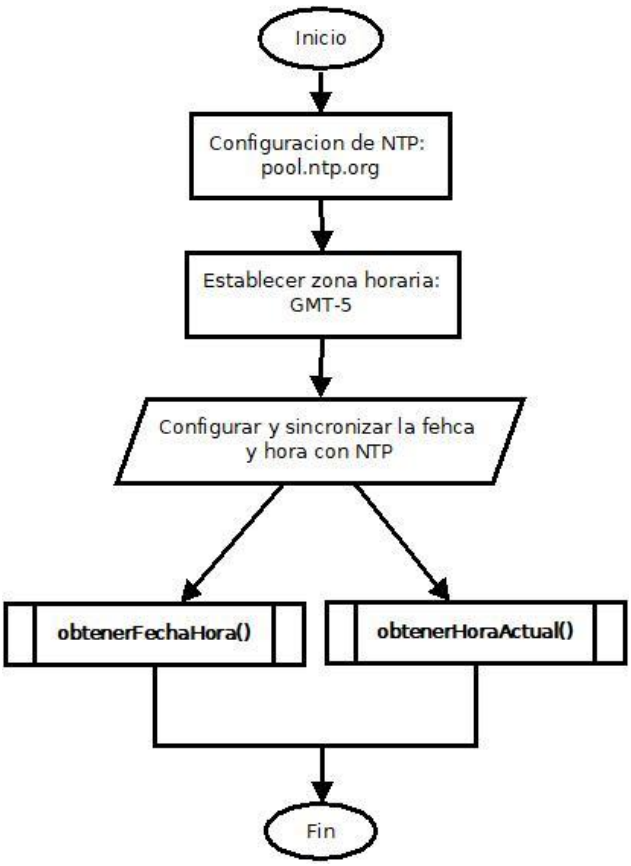
Algoritmo 2. Menú de control basado en entrada de teclado

3.3.4.2 Sincronización de tiempo con NTP

Una parte crucial para el registro de notificaciones es mantener la sincronización NTP, en este caso el ESP32 gracias a la capacidad de conexión Wi-Fi permite realizar consultas con un servidor en Internet **(pool.ntp.org)**, luego obtener en un determinado formato la fecha y hora actual. La **Figura 30** muestra el diagrama de flujo que utiliza el ESP32 para la conexión y sincronización NTP.

Figura 30

Diagrama de flujo de la sincronización NTP.

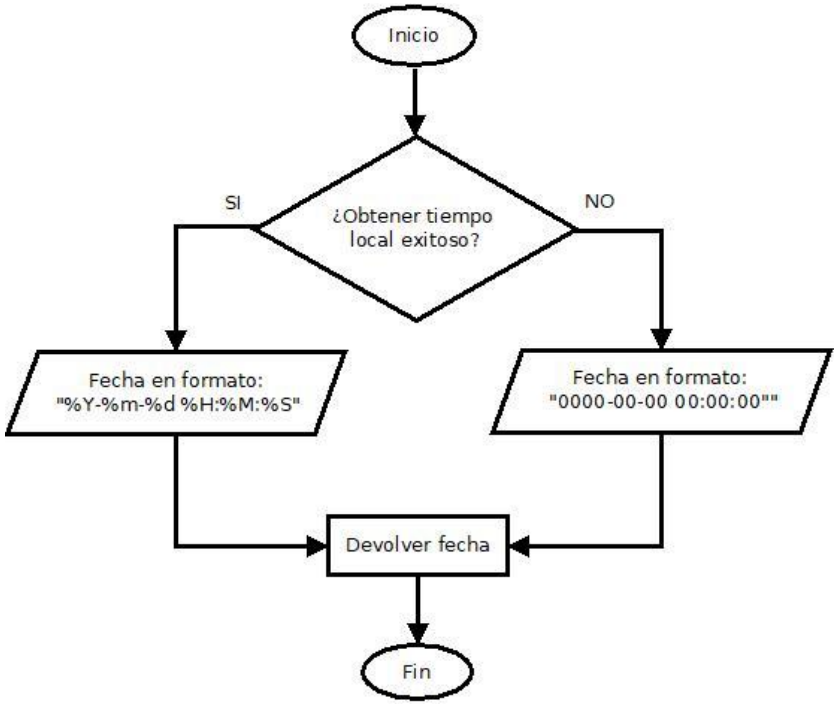


A partir de la sincronización inicial se procede a utilizar las funciones de $\boldsymbol{obtenerFechaHora}(..)$ y $\boldsymbol{obtenerHoraActual}(..)$, estas permiten obtener la fecha y hora mostrando en un formato de "**YYYY-MM-DD HH:MM:SS**" ideal para el registro de notificaciones, y formato solo de tiempo "**00:00:00**" para la visualización del tiempo en el sistema.

En la **Figura 31** se muestra el diagrama de flujo de la función $\boldsymbol{obtenerFechaHora}(..)$, esta función es llamada continuamente y se utiliza para el registro de notificaciones. Se utiliza principalmente para: Registrar accesos permitidos/denegados, Marcar el momento exacto de las notificaciones y Registrar el tiempo de eventos RFID y de llave electrónica.

Figura 31

Diagrama de flujo de la función $\boldsymbol{obtenerFechaHora}(..)$**.**



El **Algoritmo 3** muestra la estructura de la función $\boldsymbol{obtenerFechaHora}(...)$.

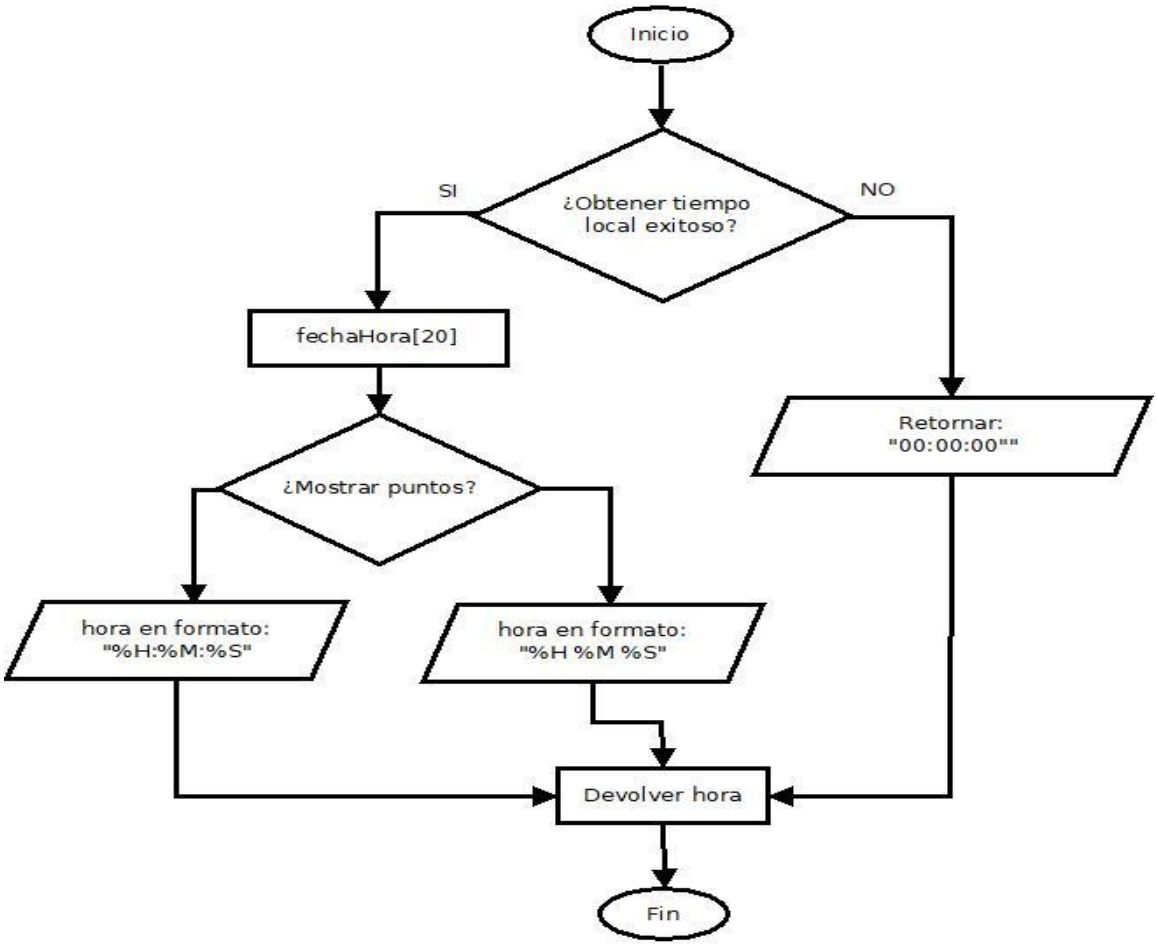
```
Input: Ninguno
Output: (String) fechaHora

(1)   struct tm timeinfo
(2)   if !getLocalTime(&timeinfo):
(3)      print("Error al obtener la fecha y hora")
(4)      return "0000-00-00 00:00:00"
(5)   else
(6)      char fechaHora[20]
(7)      formato (fechaHora,"%Y-%m-%d %H:%M:%S")
(8)      return String(fechaHora)
```

Algoritmo 3. Obtención de Fecha y Hora en formato específico.

En la **Figura 32** se muestra el diagrama de flujo de la función "**obtener hora actual**" esta función es utilizada para mostrar la hora actual al usuario mediante la pantalla LCD, El parámetro **mostrarPuntos** permite crear un efecto de parpadeo en la pantalla (alternando entre mostrar y ocultar los puntos).

Figura 32

Diagrama de flujo de la función ***obtenerHoraActual*****(..).**



El **Algoritmo 4** muestra la estructura de la función ***obtenerHoraActual***(...).

```
Input: (bool) mostrarPuntos
Output: (String) hora

(1)   struct tm timeinfo
(2)   if !getLocalTime(&timeinfo):
(3)      print("Error al obtener la hora")
(4)      return "00:00:00"
(5)   Else
(6)      char hora[9]
(7)      if mostrarPuntos == true:
(8)         formato (hora, “%H:%M:%S")
(9)      else:
(10)        formato (hora, “%H %M %S")
(11)     return String(hora)
```

Algoritmo 4. Obtención Hora en formato específico.

3.3.4.3 Control de apertura de la cerradura

Para que el sistema pueda realizar la conexión con la cerradura electromecánica, es necesario que se implemente un módulo relé permitiendo así que el sistema envíe 3v de activación lógica y permita el paso de 12V AC, La función ***triggerDoorUnlock***(..) permite ejecutar este proceso y controlar el proceso de apertura y cierre de una puerta, mostrando mensajes en el display LCD y activando/desactivando un relé (**doorPin**) para controlar el mecanismo de bloqueo. La función también reinicia el contador de errores (errores) y restablece la variable ***opcion*** para permitir nuevas interacciones con el sistema.

El **Algoritmo 5** indica el funcionamiento y mensajes de la función para la apertura y cierre de la puerta.

```
Input: Ninguno
Output: doorPin state

(1)   print(Puerta Abierta)
(2)   delay(1000)
(3)   errores ← 0
(4)   doorPin ← HIGH
(5)   greenLED ← HIGH
(6)   delay(lockTurnTime)
(7)   doorPin ← LOW
(8)   greenLED ← LOW
(9)   print(Puerta Abierta)
(10)  delay(1000)
(11)  print(SMART DOOR LOCK)
(12)  opcion ← 0
```

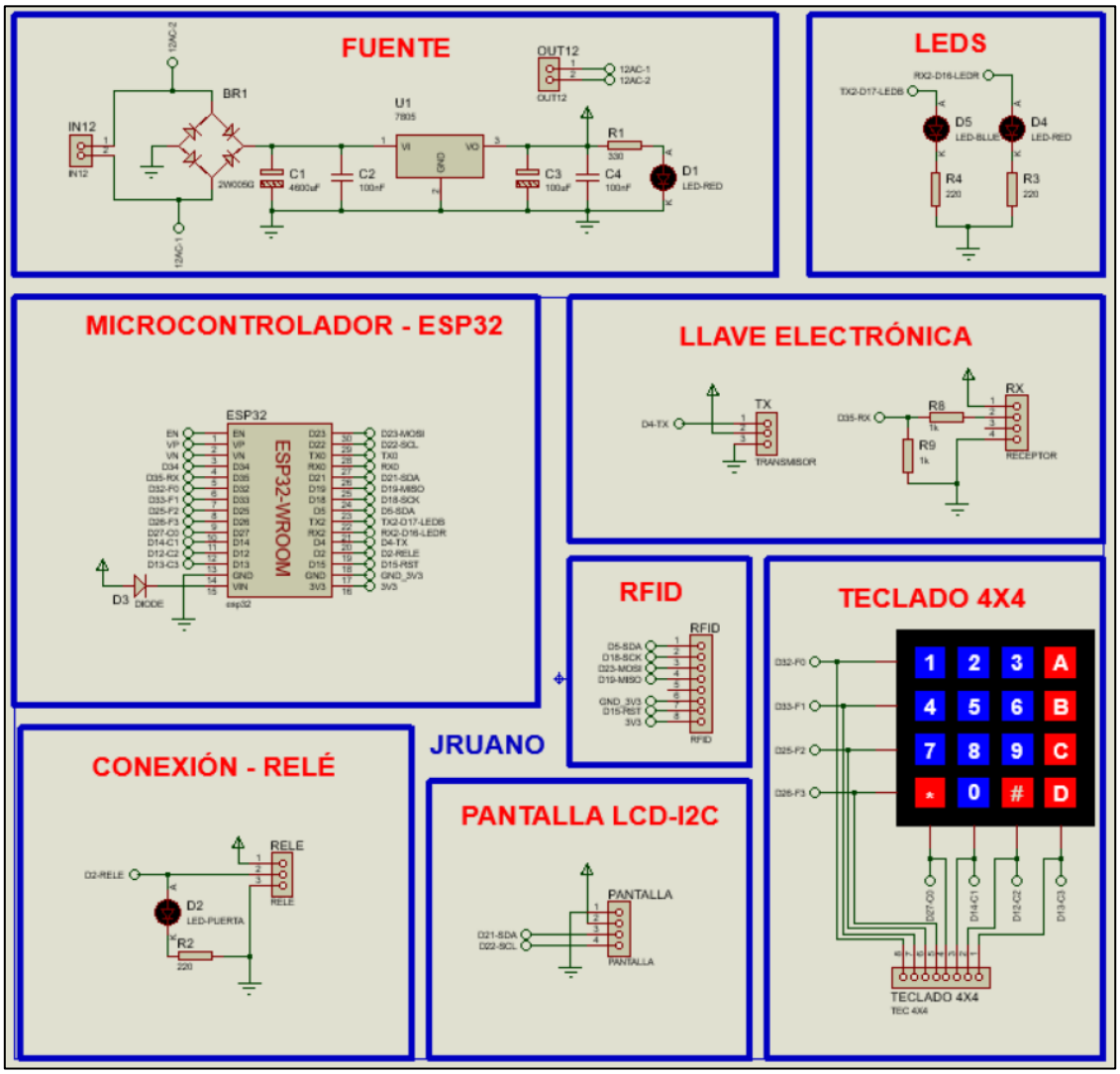
Algoritmo 5. Control de apertura/cierre de puerta.

3.3.4.4 Diseño del circuito en Proteus

Para el diseño de la placa base para el circuito de la cerradura se inicia diseñando las conexiones lógicas de todos los componentes en el simulador Proteus. Este esquema permite validar el correcto funcionamiento del sistema antes de implementarlo físicamente.

Figura 33

Diseño de la cerradura electrónica.



Nota. La figura muestra el diagrama esquemático del sistema en Proteus, donde se visualizan los módulos principales: fuente de alimentación, microcontrolador ESP32, teclado 4x4, lector RFID, pantalla LCD-I2C, LEDs, llave electrónica por RF y control del relé de apertura.

3.3.4.5 Diseño de la fuente

La primera etapa es el diseño de la fuente de alimentación, diseñada para generar el voltaje requerido por los componentes del sistema, convertir una entrada de corriente alterna (AC) en una salida de corriente directa (DC) regulada, indispensable para garantizar el correcto

funcionamiento de todo el circuito. De este modo, la fuente está diseñada para proporcionar tanto 12V AC y 5V DC, cumpliendo con las necesidades energéticas de todos los elementos del sistema.

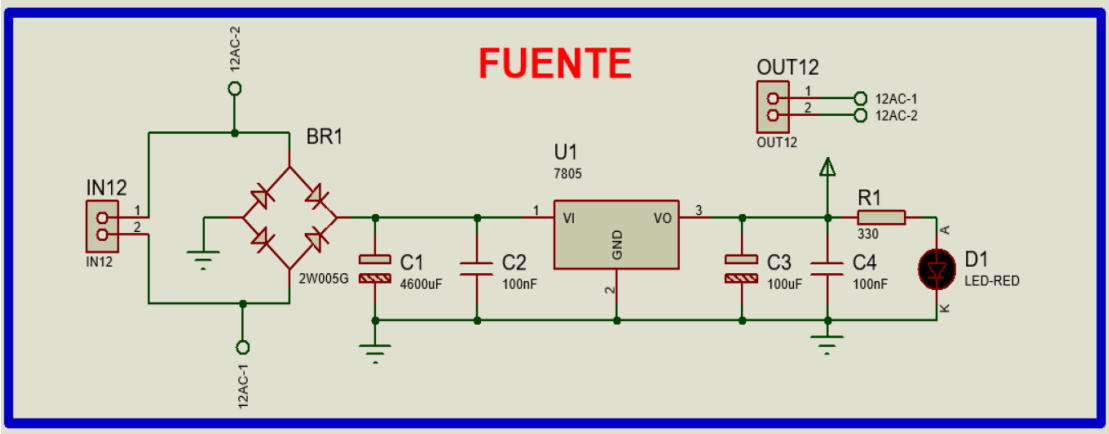
El regulador de voltaje 7805 garantiza una salida estable de 5V la mayor parte de componentes eléctricos. Además, se proporciona 12V AC directamente desde el transformador, necesarios para la operación de las cerraduras eléctricas.

El diseño incluye los siguientes componentes principales:

- Puente rectificador (BR1): Conversión de AC a DC.
- Condensadores (C1, C2, C3, C4): Filtrado y estabilización de la señal.
- Regulador de voltaje (U1): Regulación de la salida a 5V DC.
- Resistor (R1): Limitación de corriente para el LED indicador.
- LED indicador (D1): Señalización visual del funcionamiento de la fuente.

Figura 34

Diseño de fuente de alimentación para la cerradura eléctrica.



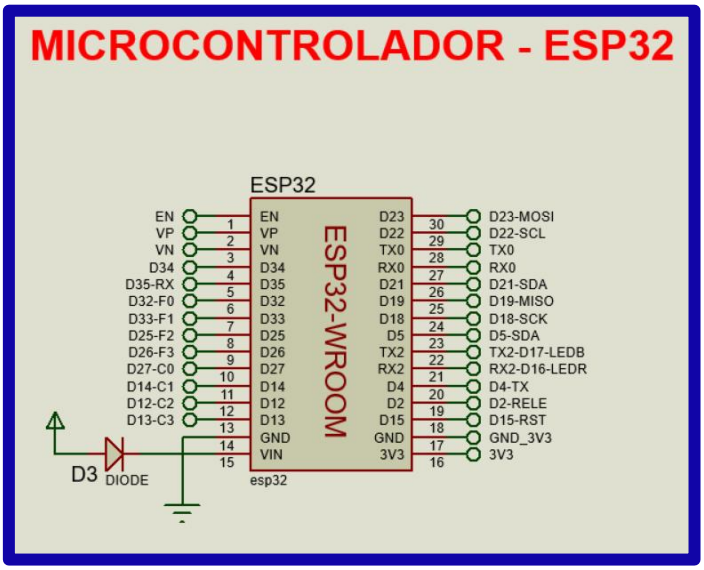
Nota. El circuito mostrado en la figura inicia con la conexión de la entrada de corriente alterna de 12V AC, para después pasar por las etapas de rectificación y así poder proporcionar tanto 5 V DC como 12 V AC.

3.3.4.6 Diagrama central del microcontrolador ESP-32

En la **Figura 35** se observa el diseño de conexión del microcontrolador ESP32-WROOM, destacando sus pines de entrada y salida, así como una conexión adicional con un diodo (D3) para protección del circuito. Este microcontrolador es el núcleo principal del sistema, encargado de controlar y gestionar los dispositivos conectados.

Figura 35

Diagrama de Pines del Microcontrolador ESP32.

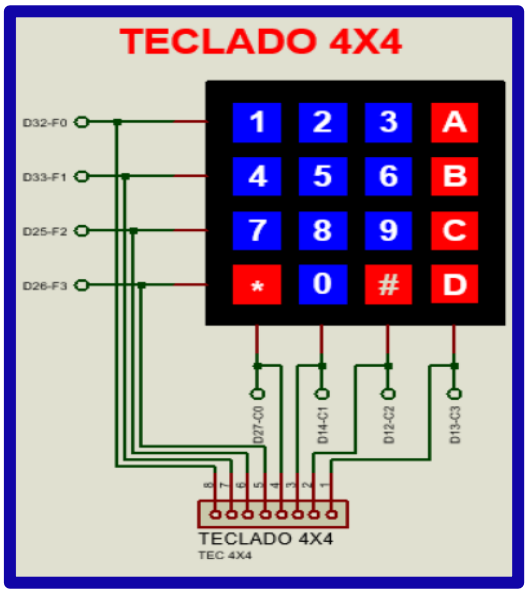


3.3.4.7 Acceso por teclado numérico

En la segunda etapa, se implementa un módulo de acceso mediante teclado numérico. Este sistema permite configurar combinaciones de 6 caracteres para activar la cerradura, ofreciendo una opción de autenticación sencilla y práctica.

Figura 36

Conexiones del esquema del teclado matricial 4x4 con ESP32.



El presente diagrama de flujo representa el proceso lógico de validación de acceso mediante un teclado matricial, implementado en un sistema de control de accesos utilizando un ESP32. Este flujo contempla tanto los escenarios de ingreso de datos, validación exitosa como los de intentos fallidos, las notificaciones en la aplicación y envió al correo electrónico.

Estructura y Funcionamiento

- *Inicio del proceso*

El sistema inicia con la activación de una letra o tecla del menú principal, que selecciona el modo de ingreso por teclado.

- *Verificación de acceso habilitado*

Antes de permitir la introducción del PIN, el sistema verifica si el método de acceso por teclado se encuentra habilitado, esta funcionalidad el usuario modificar en la aplicación.

- *Ingreso del PIN*

Si el acceso está habilitado, el usuario procede a ingresar una contraseña numérica (PIN) de seis dígitos mediante el teclado.

- *Validación del PIN*

El sistema valida los datos ingresados a través de una solicitud HTTP al servidor remoto. Si los datos son correctos, se activa el mecanismo de apertura de la puerta (mediante un relé), se enciende el LED verde como indicador visual de éxito y se registra una notificación de acceso permitido en la base de datos.

- *Control de intentos*

El usuario dispone de un máximo de tres intentos para ingresar el PIN correctamente. Si aún no se ha alcanzado el límite:

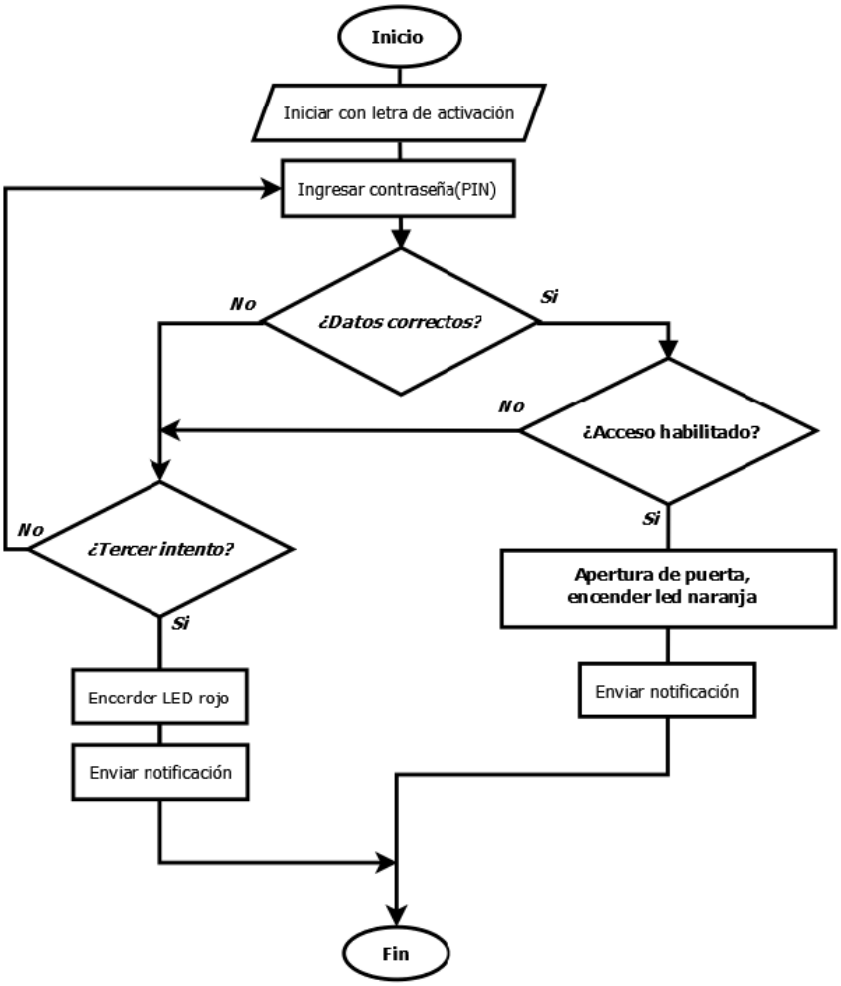
  - El sistema permite reintentar la introducción del PIN.
  - Si se alcanza el tercer intento sin éxito, se activa una alerta visual mediante el encendido del LED rojo, y se registra un evento de acceso denegado en la base de datos, enviando también una notificación a la aplicación y al correo registrado de cada integrante de la residencia.

- *Fin del proceso*

El flujo finaliza tras un intento exitoso o después de tres intentos fallidos. En ambos casos, el sistema retorna al menú principal, permitiendo nuevos intentos o cambios de modo de autenticación.

Figura 37

Diagrama de flujo de ingreso por PIN.



El **Algoritmo 6** muestra el sistema de validación de contraseña mediante teclado matricial para control de acceso, así también utiliza funciones anteriormente explicadas y el envío de notificaciones a la base de datos.

```
Input: (char) tecla, (int) contpass, (int) presion
Output: Ninguno (efectos: LCD y base de datos)

 (1)   if tecla es dígito (0-9):
 (2)      Almacenar en verpass[contpass]
 (3)      Mostrar * en LCD
 (4)      contpass++
 (5)      presion++
 (6)   if tecla == 'E' y contpass > 0:
 (7)      Borrar último * en LCD
 (8)      contpass--
 (9)      presión--
(10)   if contpass == 6:
(11)      if validarClaveIngresada(claveIngresada):
```

```
(12)        Mostrar "CORRECTO"
(13)        triggerDoorUnlock()
(14)        obtenerFechaHora();
(15)        Registrar acceso permitido en BD
(16)     else:
(17)        errores++
(18)        if errores == 3:
(19)           Mostrar "INCORRECTO"
(20)           obtenerFechaHora();
(19)           Registrar acceso denegado en BD
(20)           Resetear contpass y presion
(21)         else:
(21)           Permitir reintento
```

Algoritmo 6. Validación de contraseña por teclado matricial.

3.3.4.8 Acceso por RFID

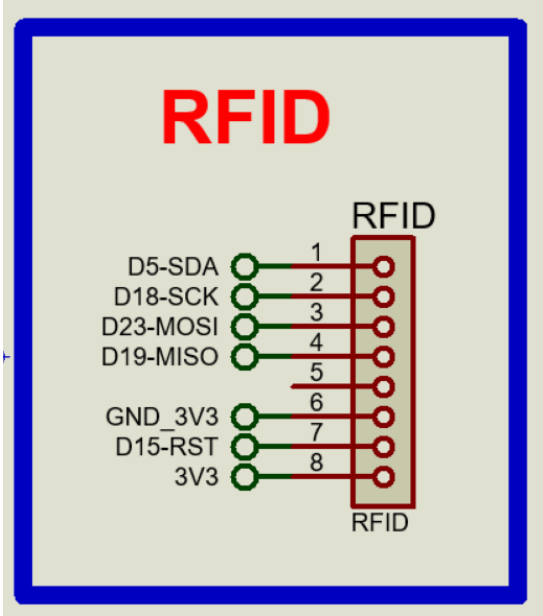
En la tercera etapa el módulo permite registrar tarjetas o etiquetas RFID autorizadas, que los usuarios pueden utilizar para activar la cerradura de forma rápida y sin contacto. La comunicación entre el lector RFID y el microcontrolador asegura que solo las credenciales válidas activen el sistema.

El módulo RFID mostrado en la **Figura 38** se conecta al sistema utilizando una interfaz SPI (Serial Peripheral Interface), que facilita la comunicación entre el microcontrolador y el módulo. De esta manera, los pines principales para la transmisión de datos son D21-SDA, que actúa como la línea de datos serial para el intercambio de información; D18-SCK, funciona como la línea de reloj para sincronizar la comunicación; D23-MOSI, utilizada para enviar datos desde el microcontrolador al módulo; y D19-MISO, permite recibir datos desde el módulo RFID al microcontrolador.

Adicionalmente, el módulo incluye el pin D22-RST, que se utiliza para reiniciar o inicializar el módulo en caso necesario, garantizando su correcto funcionamiento. El módulo recibe su alimentación a través del pin 3V3, que suministra un voltaje de 3.3 V, y se conecta a tierra mediante el pin GND.

Figura 38

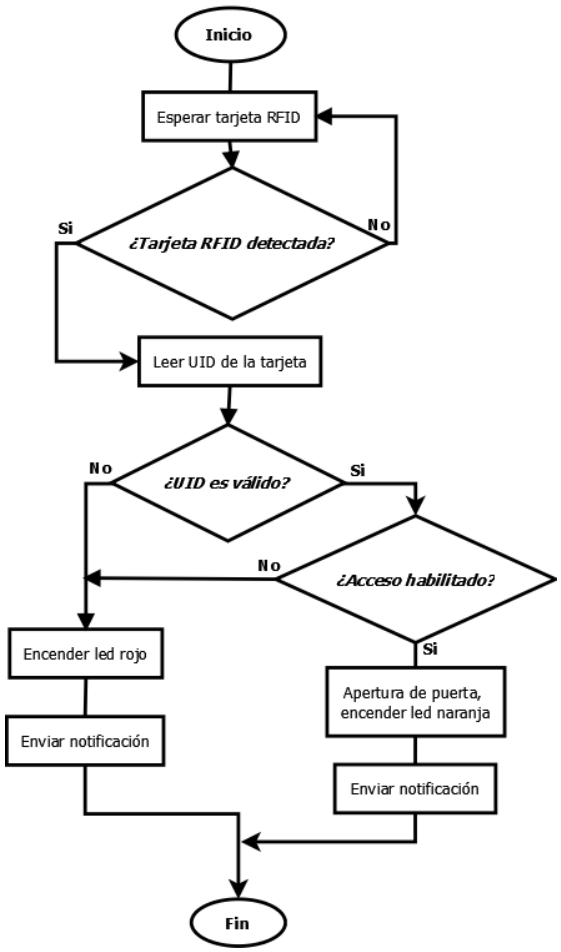
Conexiones de RFID.



El diagrama de la **Figura 39** muestra el flujo del proceso de autenticación mediante lectura RFID. El sistema inicia con la detección de entrada, seguida por la lectura de la etiqueta RFID. Los datos obtenidos son procesados por un ESP32 y, posteriormente, enviados a un servidor para su validación. Si los datos son correctos, se activa un indicador LED verde y se habilita la apertura de la puerta. En caso contrario, si los datos son incorrectos, se enciende un indicador LED rojo, denegando el acceso. Este flujo garantiza un control de acceso eficiente y seguro, fundamentado en la tecnología RFID y la validación en tiempo real.

Figura 39

Diagrama de flujo de ingreso por RFID.



El **Algoritmo 7** muestra el funcionamiento del acceso por tarjeta RFID, Cuando se detecta una tarjeta, convierte su UID a una cadena hexadecimal en minúsculas y lo valida contra una base de datos. Si es válido, desbloquea la puerta utilizando $\boldsymbol{triggerDoorUnlock}(..)$. Y cada evento se registra y grada en la base de datos.

```
Input: (int) opcion, (MFRC522) mfrc522
Output: Ninguno (efectos: LCD y base de datos)

(1)   if opcion ≠ 2 y mfrc522.PICC_IsNewCardPresent()
(2)      uidString ← ""
(3)      for i from 0 to mfrc522.uid.size-1:
```

```
(4)         uidString += String(mfrc522.uid.uidByte[i])
(5)      if validarRfid(uidString):
(6)         triggerDoorUnlock()
(7)         Registrar acceso Permitido en BD
(8)      else:
(9)         Registrar acceso Denegado en BD
(10)     delay(1000)
```

Algoritmo 7. Validación de tarjetas RFID para control de acceso.

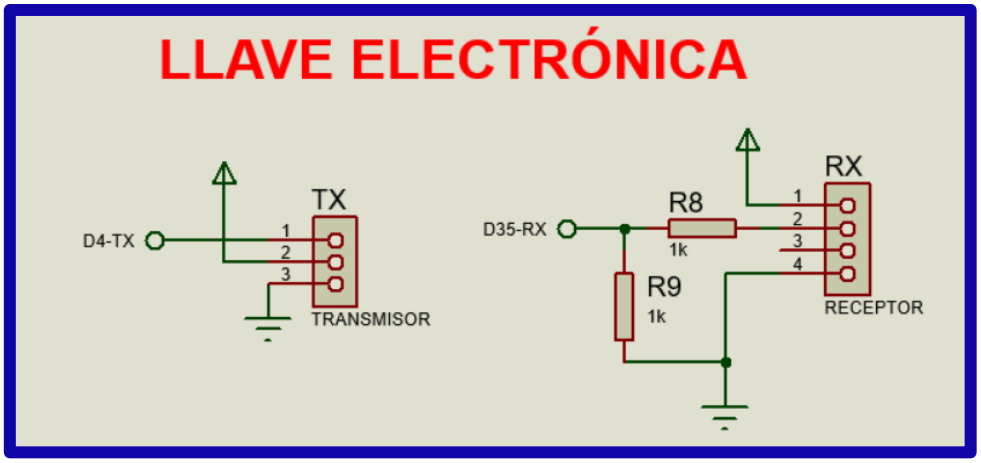
3.3.4.9 Acceso por llave electrónica

La cuarta etapa es acceso por llave electrónica permite la comunicación bidireccional entre el microcontrolador y la llave electrónica mediante los módulos de transmisión y recepción. Los datos se transfieren de manera segura a través de los pines TX y RX.

Por otra parte, como se observa en la **Figura 40** el transmisor puede ser enviar los datos con valores lógicos de entre 0 y 3V, sin embargo, el receptor genera valores de 0 y 5V los cuales el ESP32 no puede entender, para ello se realiza un divisor de voltaje en el pin de datos del Receptor permitiendo que los datos puedan ser leídos de 0 a 2.5V.

Figura 40

Acceso físico para llave electrónica.



Nota. La llave electrónica utiliza algoritmos de seguridad avanzados, como hash y cifrado en acceso inalámbrico (RF), para evitar su clonación o manipulación.

El diagrama de funcionamiento se observa en el apartado Diagrama de funcionamiento llave electrónica donde se muestra el cada uno de los eventos e interacción entre la llave electrónica y la cerradura.

El **Algoritmo 8** implementa un protocolo de autenticación seguro en dos fases para llaves RF. En la **Fase 1**, valida la llave RF y envía un código de desafío aleatorio. En la **Fase 2**, verifica un hash $\boldsymbol{MD5\_16}(..)$ generado a partir del código de desafío y una contraseña almacenada. Si la autenticación es exitosa, desbloquea la puerta $\boldsymbol{triggerDoorUnlock}(..)$ y registra el acceso.

```
Input: (bool) esperando_hash, (RH_ASK) driver
Output: Ninguno (efectos: LCD y base de datos)

(1)   if esperando_hash y tiempo > tiempo_espera_maximo:
(2)      esperando_hash ← false
(3)   if opcion ≠ 4 y driver.recv(buf, buflen):
(4)      recibido ← String(buf)
(5)      if not esperando_hash:
(6)         if validarLlaveRF(recibido):
(7)            Obtener contraseña_real
(8)            if contraseña_real ≠ "":
(9)               code = generarCodigoAleatorio(10)
(10)              send(code)
(11)              esperando_hash ← true
(12)              Iniciar espera
(13)           else:
(14)              accesoDenegadoRF()
(15)        else:
(16)           accesoDenegadoRF()
(17)     else:
(18)        Hash ← simpleMD5_16(code + contraseña_real)
(19)        if recibido.equals(Hash):
(20)           triggerDoorUnlock()
(21)           Registrar acceso Permitido en BD
(22)        else:
(23)           accesoDenegadoRF()
(24)        esperando_hash ← false
```

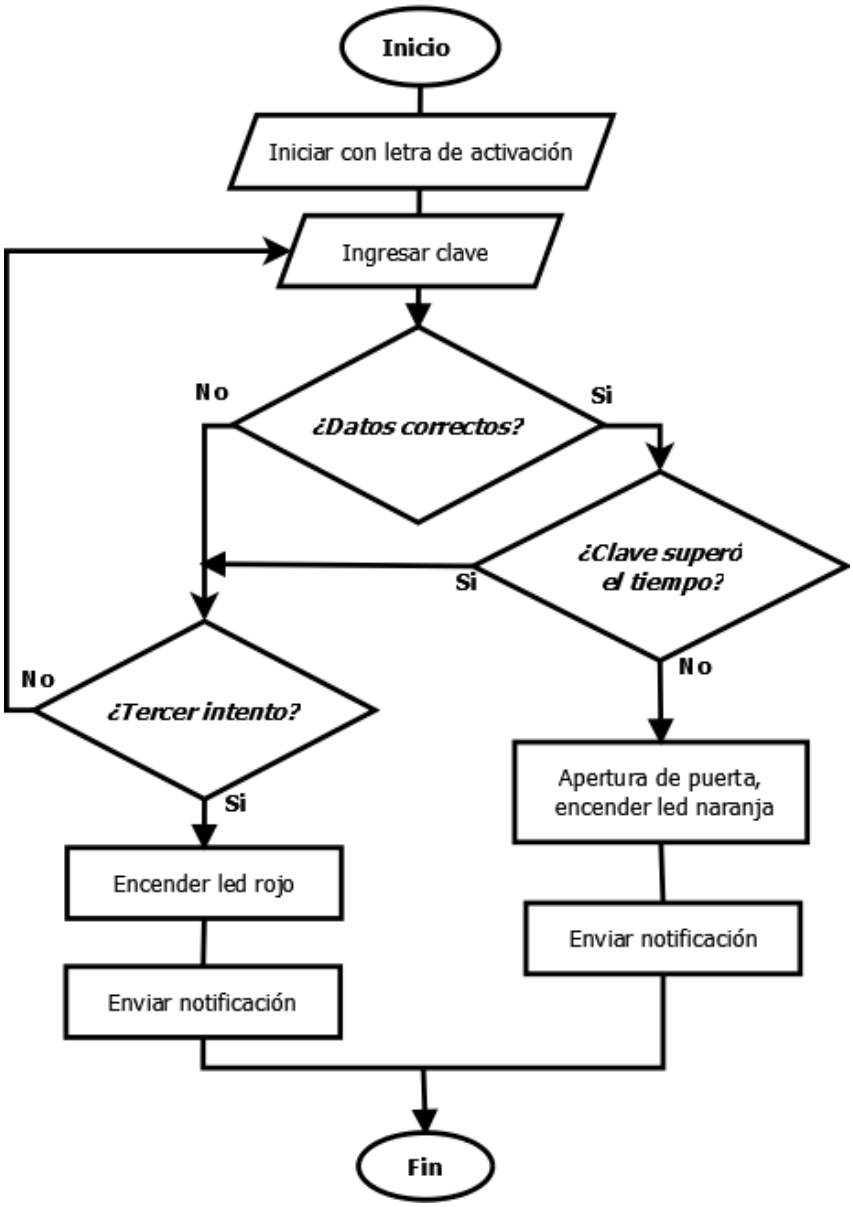
Algoritmo 8. Autenticación para llaves RF.

3.3.4.10 Acceso por clave temporal (SMS)

La quinta etapa, para cumplir con esta funcionalidad se diseña una aplicación móvil que permite al usuario, ingresar el número celular para enviar la clave temporal además de elegir el tiempo de validez de la clave.

Figura 41

Diagrama de flujo – ingreso por clave temporal.



El **Algoritmo 9** implementa un sistema de validación de claves temporales mediante teclado matricial para control de acceso, similar al acceso por teclado **Algoritmo 6**.

```
Input: (char) tecla, (int) contpass, (int) presion
Output: Ninguno (efectos: LCD y base de datos)

 (1)   if tecla es dígito (0-9):
 (2)      Almacenar en verpass[contpass]
 (3)      Mostrar * en LCD
 (4)      contpass++
 (5)      presion++
 (6)   if tecla == 'E' y contpass > 0:
 (7)      Borrar último * en LCD
 (8)      contpass--
 (9)      presión--
(10)   if contpass == 6:
(11)      if validarClaveTemporal(claveIngresada):
(12)         Mostrar "CORRECTO"
(13)         triggerDoorUnlock()
(14)         obtenerFechaHora();
(15)         Registrar acceso permitido en BD
(16)      else:
(17)         errores++
(18)         if errores == 3:
(19)            Mostrar "INCORRECTO"
(20)            obtenerFechaHora();
(19)            Registrar acceso denegado en BD
(20)            Resetear contpass y presion
(21)          else:
(21)            Permitir reintento
```

Algoritmo 9. Validación de claves temporales por teclado.

3.3.4.11 Acceso mecánico llave tradicional

Como medida de respaldo, el sistema conserva la opción de acceso mecánico mediante una llave física tradicional. Este método garantiza la operatividad de la cerradura en casos de fallos eléctricos, pérdida de conexión o preferencia del usuario por un método más convencional. Este diseño híbrido asegura que la cerradura mantenga su funcionalidad en cualquier circunstancia, aumentando la confiabilidad del sistema

Figura 42

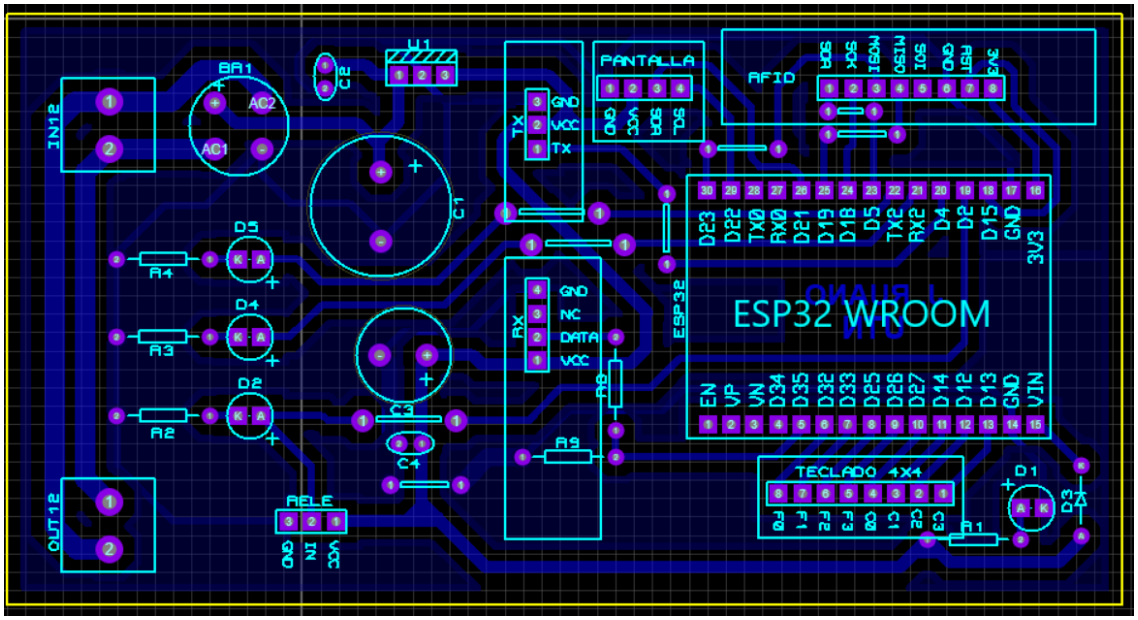
Ingreso por llave tradicional.

3.3.4.12 Diseño del PCB

La **Figura 43** muestra el diseño del PCB (Printed Circuit Board) del sistema basado en el microcontrolador ESP32-WROOM, en el cual se integran las diferentes etapas y módulos necesarios para el funcionamiento del sistema.

Figura 43

Diseño de la placa de circuito impreso (PCB) – cerradura.



3.3.4.13 Diseño 3D

En la **Figura 44** se muestran el diseño tridimensional del PCB del sistema, donde se aprecian la distribución y ubicación física de los componentes electrónicos, asegurando una disposición lógica y eficiente para facilitar el ensamblaje y el funcionamiento correcto del sistema.

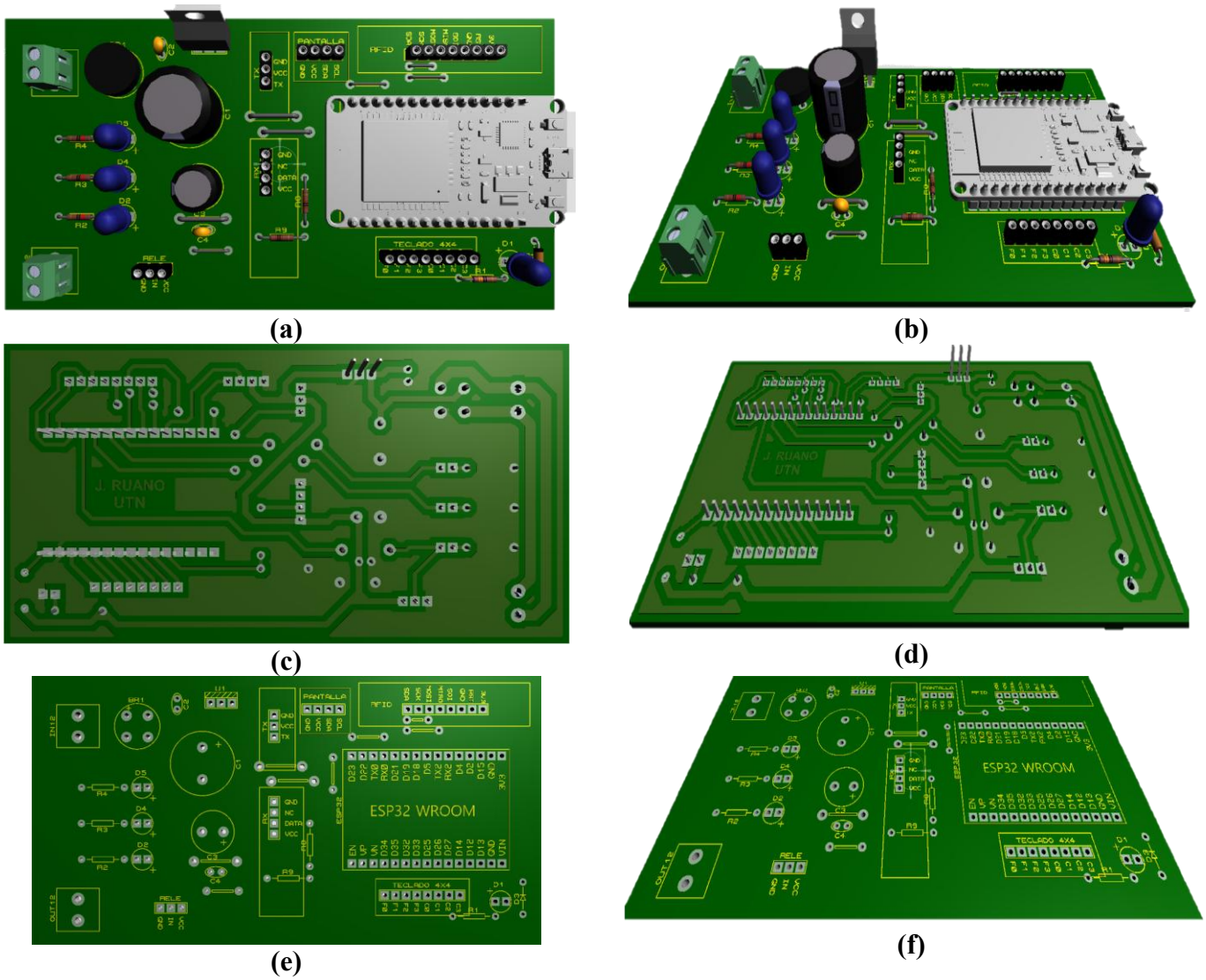
En la parte izquierda del PCB se encuentra la sección destinada a la fuente de alimentación, que integra la entrada de 12V para el transformador, un puente rectificador de diodos, los condensadores y el regulador de voltaje. Estos componentes están dispuestos estratégicamente, separando la sección de potencia de la de datos en el PCB, lo que garantiza un flujo de energía adecuado hacia todos los módulos del sistema.

El microcontrolador ESP32-WROOM está montado en la parte derecha del PCB, asegurando una conexión directa con los periféricos a través de los pines GPIO. Su posición centralizada facilita el acceso y las conexiones con otros componentes, además de permitir un acceso sencillo al puerto USB-C para implementar actualizaciones de código y obtener registros (logs).

En la parte superior se encuentran los conectores para periféricos como el teclado 4x4 y el módulo RFID. Finalmente, los LEDs y las resistencias están ubicados en una línea central en el PCB, proporcionando retroalimentación visual sobre los eventos que ocurren en el sistema.

Figura 44

Vista tridimensional del montaje del PCB.



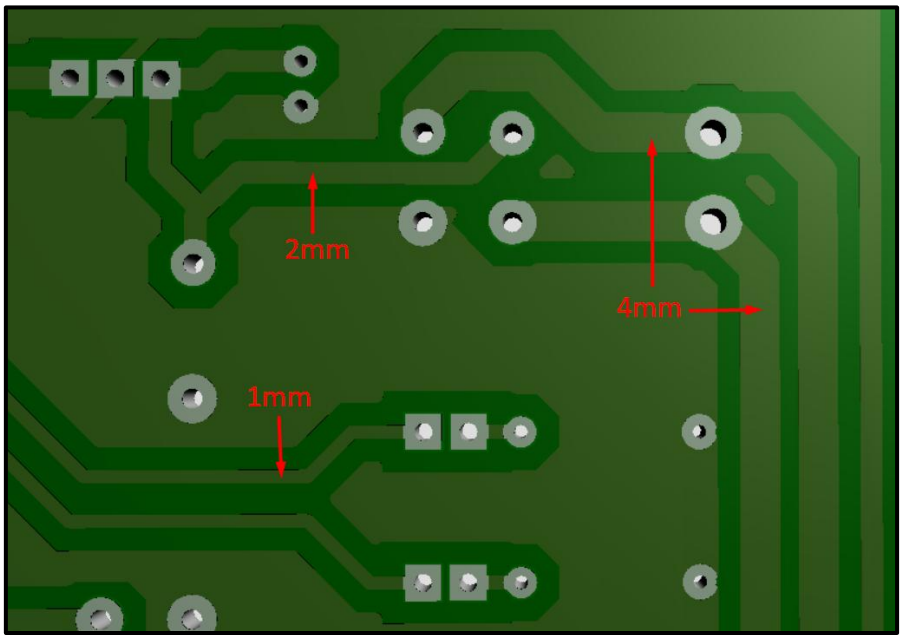
(a) **(b)** **(c)** **(d)** **(e)** **(f)**

Nota. Las vistas tridimensionales **(a)** y **(b)** muestran los componentes principales del circuito montados en la cara superior del PCB, incluyendo el microcontrolador, puertos para la conexión de los demás periféricos, resistencias, condensadores, etc. Las vistas **(c)** y **(d)** muestran el diseño de las pistas en la cara inferior del PCB, donde se destacan las pistas de cobre encargadas de conectar eléctricamente los diferentes componentes del sistema. Estas pistas están diseñadas de manera estratégica para garantizar un flujo eficiente de corriente y datos. Las vistas **(e)** y **(f)** muestra la serigrafia para el montaje de los componentes.

En el diseño presentado se observa una diferenciación clara en el ancho de las pistas, lo cual es fundamental para asegurar la integridad eléctrica y térmica del circuito. Las pistas han sido dimensionadas de acuerdo con los niveles de corriente y voltaje que manejarán. En la **Figura 45** se muestra el tamaño de pistas.

Figura 45

*Tamaño de pistas para circuito PCB***.**



Las pistas de 4 mm están diseñadas para transportar corriente alterna de 12V a 2A proveniente del transformador. Este ancho es adecuado para dicho nivel de corriente, considerando tanto la capacidad de conducción como la disipación térmica en una PCB. Además, contribuye a evitar el sobrecalentamiento y garantiza una baja caída de voltaje.

Por otro lado, las pistas de 2 mm están destinadas a transportar voltaje continuo de 12V con una corriente menor a 1A, lo que es ideal para la alimentación general del sistema. Este ancho proporciona un margen de seguridad y es suficiente para alimentar todos los módulos.

Finalmente, las ***pistas de 1 mm*** están destinadas para señales de control y comunicación a **5V** DC. Este ancho es más que suficiente para señales digitales de bajo consumo en el orden

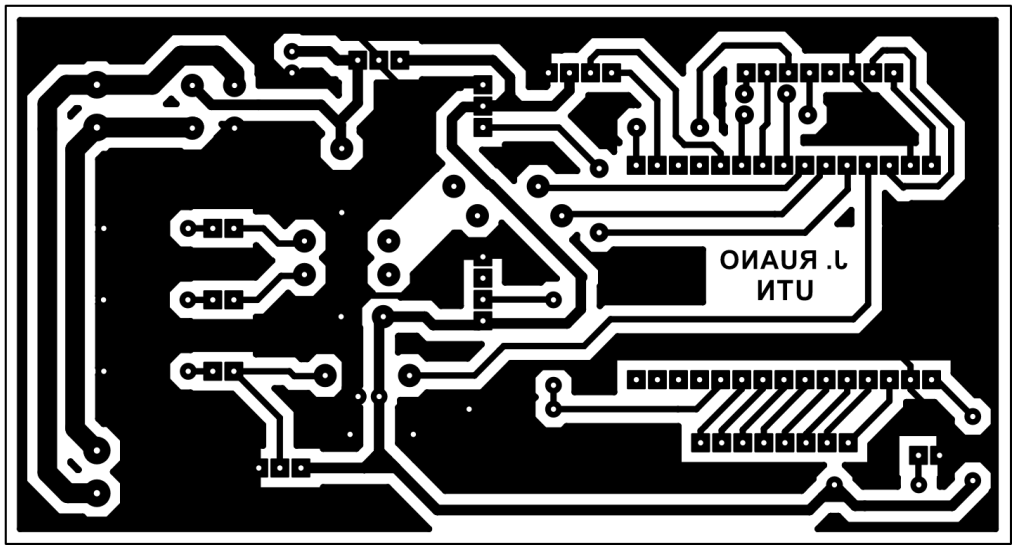
de **mA**, y permite un trazado más fino en zonas de alta densidad sin comprometer la integridad de la señal.

3.3.4.14 Impreso

La **Figura 46** exhibe el diseño final del trazado de pistas para la placa de circuito impreso de la cerradura eléctrica. Se puede observar el enrutamiento de las conexiones eléctricas que componen el circuito, el cual fue optimizado para asegurar una correcta distribución de señales y alimentación. Este diseño se generó meticulosamente para garantizar la integridad en las conexiones, respetando los anchos de pista adecuados según la corriente que transportan. Una vez que este trazado está finalizado, el siguiente paso en el proceso de manufactura es transferir el patrón a una placa de cobre utilizando la técnica de planchado.

Figura 46

Diseño del circuito impreso.

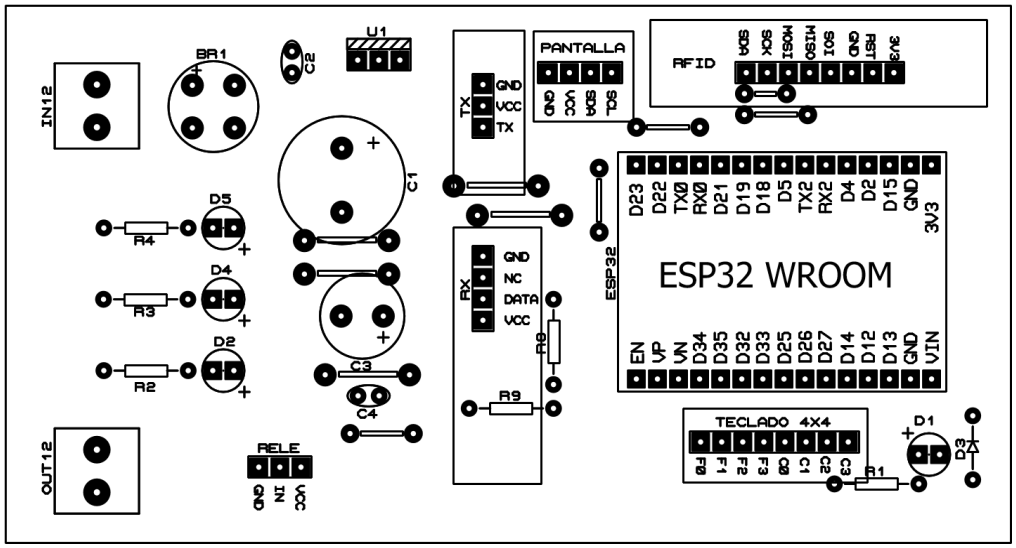


Finalmente, en la **Figura 47** se muestra la vista superior de la placa de circuito impreso, correspondiente al diseño de la serigrafía. Donde se muestra los identificadores de cada uno de

los componentes, polaridades, etiquetas y contornos, que facilitan el montaje manual de los elementos electrónicos. La serigrafía no solo contribuye a una mejor organización visual del circuito, sino que también ayuda a prevenir errores durante el proceso de ensamblaje. Esta etapa complementa el diseño técnico, sirviendo como guía para la correcta colocación de los dispositivos sobre la placa.

Figura 47

Placa de circuito impreso – cerradura.

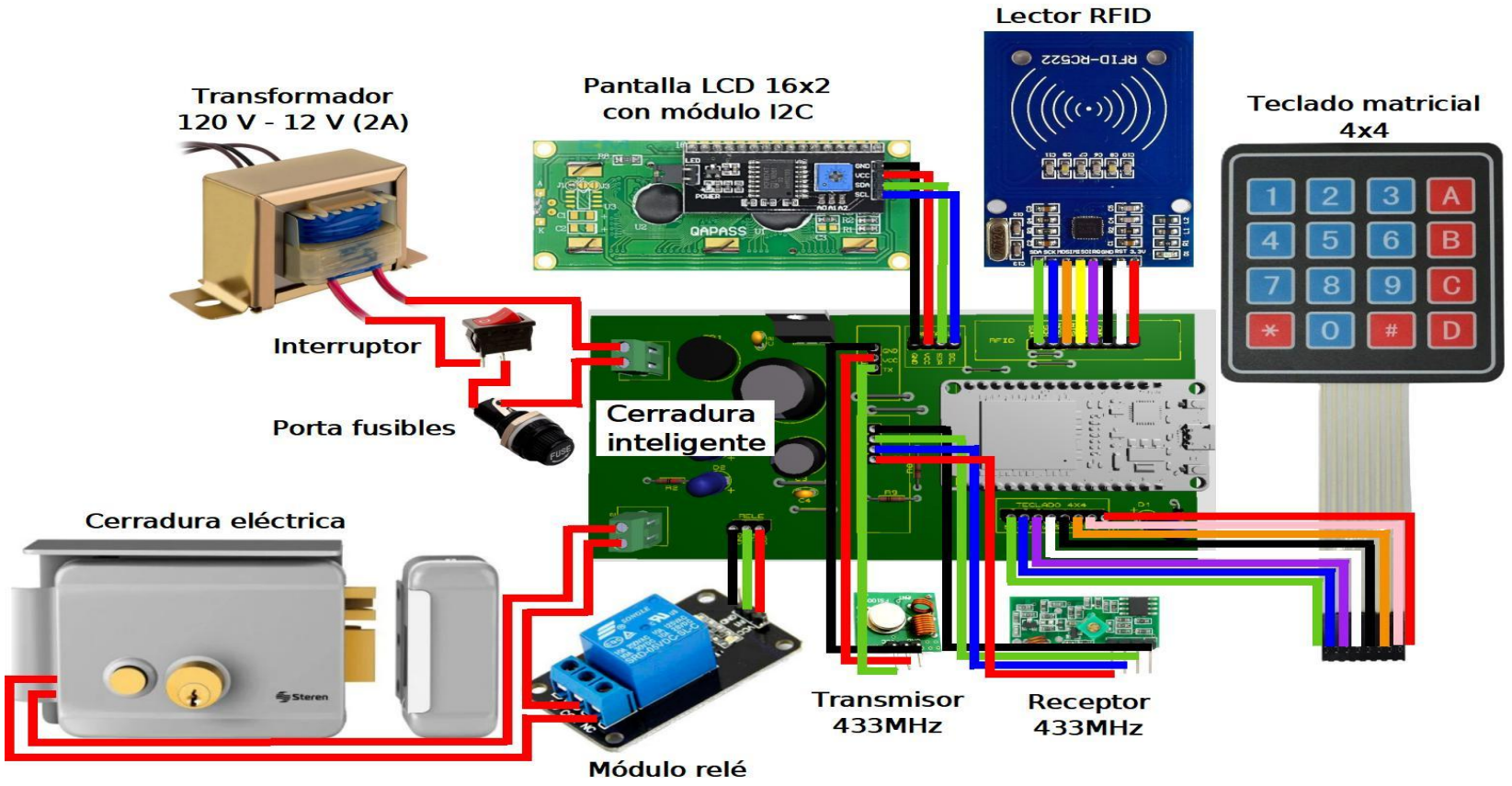


3.3.4.15 Diagrama de conexiones

La **Figura 48** muestra el diagrama de conexión de los componentes que conforman el sistema de cerradura inteligente. En él se visualizan las interconexiones eléctricas y lógicas entre el transformador, la cerradura eléctrica, el módulo relé, la pantalla LCD, el lector RFID, el teclado matricial, y los módulos transmisor y receptor 433MHz, demostrando el cableado necesario para su correcto funcionamiento.

Figura 48

Diagrama de conexiones de la cerradura inteligente.



Nota. La alimentación de la cerradura inteligente es protegida por un fusible el cual evita que corrientes elevadas puedan dañar los componentes internos, así también la alimentación de 12V de entrada es habilitada por medio de un relé para abrir la cerradura eléctrica.

3.3.5 Servidor en la nube

La integración de un servidor en la nube dentro del diseño del sistema representa un componente fundamental para garantizar la disponibilidad, escalabilidad y accesibilidad remota del sistema. El servidor actúa como el núcleo de almacenamiento en el cual contiene la base de datos y el servicio web, permitiendo los usuarios puedan consultar información en tiempo real desde cualquier parte del mundo a través de la aplicación móvil, así también la cerradura inteligente necesita enviar sus eventos y notificaciones al servidor por medio de una conexión a Internet.

La utilización de servicios en la nube optimiza la gestión de recursos, sin depender de infraestructura física local. A continuación, se detallan los aspectos clave del diseño del sistema en el servidor:

3.3.5.1 Diseño de Base de datos

La base de datos para el sistema debe contener todas las tablas necesarias y el diseño debe permitir estructurar y organizar los datos de manera eficiente. En este contexto, se realiza el modelo de entidad-relación (ER), el cual define la arquitectura de una base de datos destinada a la gestión de un sistema de control de accesos.

El modelo ER se emplea ampliamente en el diseño de bases de datos relacionales, ya que facilita la visualización y documentación de la estructura de los datos de forma clara y organizada. En este caso, dicho modelo identifica las principales entidades que intervienen en el sistema de control de accesos, tales como "***casas", "claves temporales", "usuarios", "accesos", "permisos acceso", "tipos acceso", etc***. junto con sus respectivos atributos y relaciones.

Esta representación gráfica resulta fundamental para comprender la manera en que la información se encuentra estructurada y cómo interactúan los distintos elementos del sistema como se observa en la **Figura 49**. Las tablas disponen de un conjunto de atributos, y se relacionan con otras de la siguiente forma:

Usuarios – Casas

- Muchos usuarios pueden pertenecer a una casa.
- El administrador puede o no pertenecer a una casa.

Usuarios — Accesos

- Un usuario puede tener múltiples accesos.

Usuarios — Claves_temporales

- Un usuario puede generar muchas claves temporales.

Usuarios — Permisos_acceso

- Un usuario puede tener muchos permisos de acceso.

Accesos – Tipos_acceso

- Cada acceso está ligado a un tipo de acceso.

Tipos_acceso — Permisos_acceso

- Un tipo de acceso puede estar presente en muchos permisos.

Contraseñas_llave – Accesos

- Cada contraseña_llave está ligada a un acceso por llave.

Notificaciones — Usuarios, Tipos_acceso, Casas

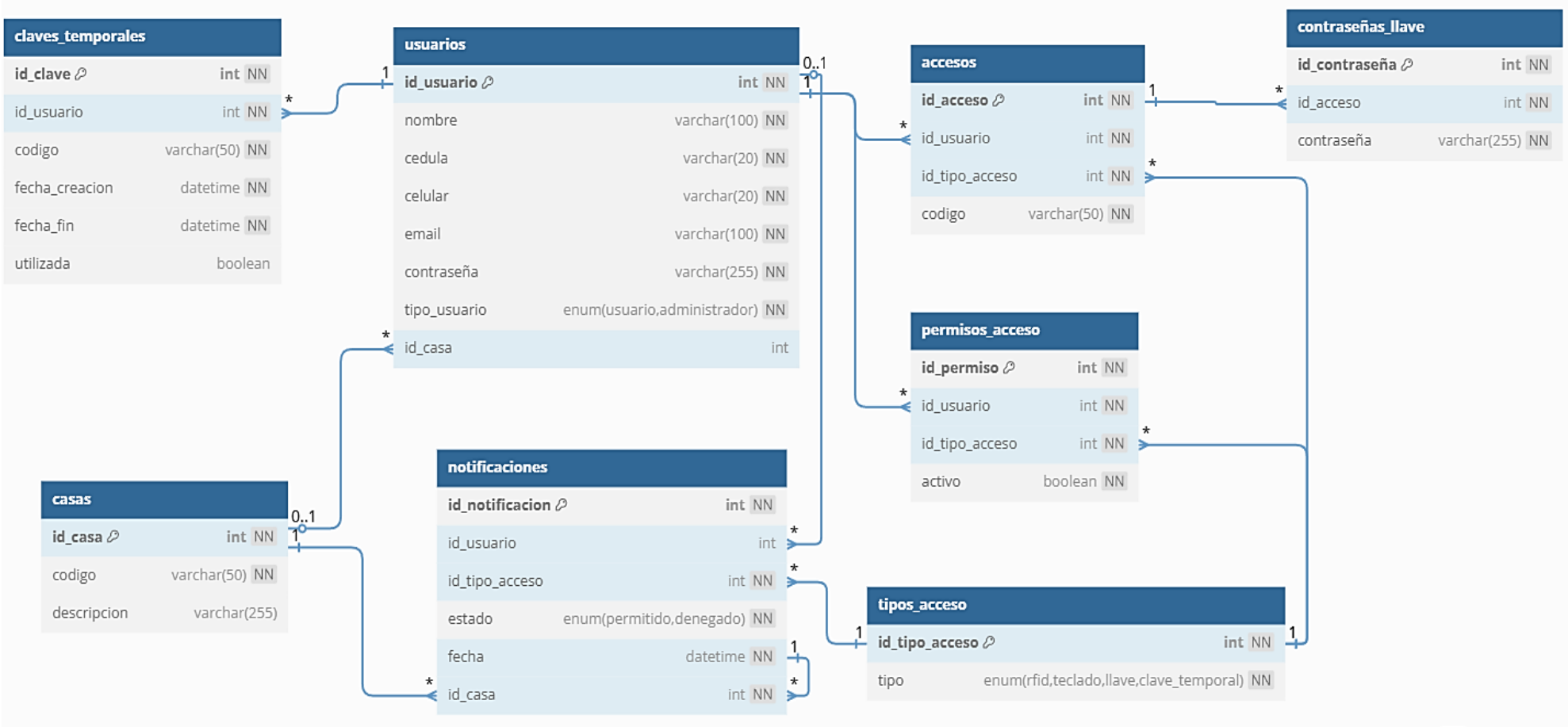
Cada notificación se relaciona con:

- Un usuario
- Un tipo de acceso
- Una casa

Cada notificación indica que un usuario intentó acceder con un tipo de acceso a una casa. El **Anexo I – Diseño entidad-relación para la base de datos** muestra la estructura para la generación del diagrama entidad relación.

Figura 49

Diagrama entidad relación.



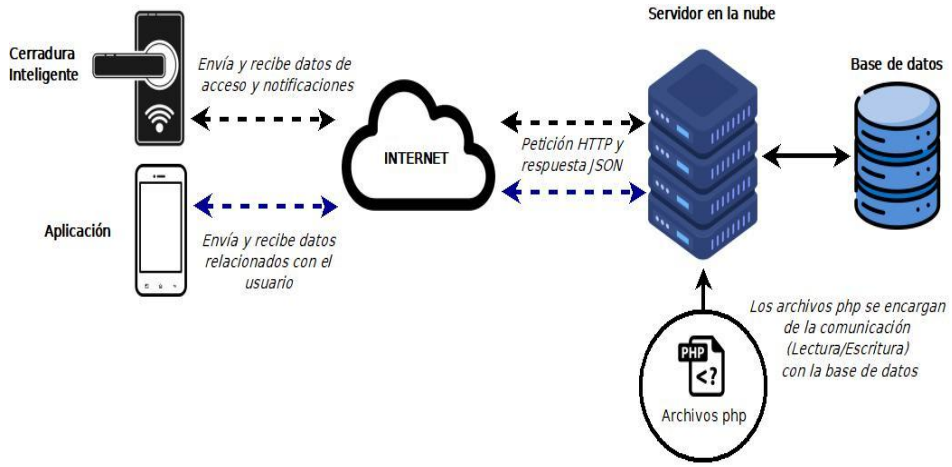
3.3.5.2 Conexiones PHP

Las conexiones PHP juegan un papel vital como intermediarios entre la aplicación cliente y la base de datos alojada en el servidor en la nube. A través de scripts escritos en este lenguaje, se implementan funciones que permiten insertar registros de acceso, consultar información relevante y gestionar usuarios de forma segura. PHP, por su naturaleza ligera y su compatibilidad con múltiples sistemas de gestión de bases de datos, representa una solución flexible y eficiente para desarrollar APIs que conecten la lógica del sistema físico con los servicios web en la nube.

En este contexto la cerradura inteligente y aplicación para los usuarios realizará conexiones HTML publicando datos y obteniendo resultados en formato JSON con la información requerida. En la **Figura 50** se muestra la comunicación del sistema con la base de datos y como el servidor web actúa como el puente vital entre el sistema y la base de datos en la nube, permitiendo la lectura y escritura de información de manera segura y eficiente.

Figura 50

Comunicación entre la cerradura inteligente, la aplicación móvil y la base de datos a través de archivos PHP.



Nota. La figura muestra el rol fundamental de los archivos PHP como capa intermedia entre los dispositivos y la base de datos. Estos archivos procesan peticiones HTTP ya sea para registrar accesos, enviar notificaciones o consultar datos de usuario. La comunicación se realiza utilizando el formato JSON.

3.3.6 Diseño de la aplicación

La aplicación es un componente fundamental del sistema, sirviendo como la interfaz principal entre el usuario y la cerradura electrónica. Su diseño contempla dos niveles de interacción: el modo usuario y el modo administrador. Cada uno de estos modos ofrece funcionalidades específicas que garantizan la seguridad, personalización y una gestión eficiente del sistema.

En el modo usuario, la aplicación permite:

- Crear y eliminar contraseñas numéricas para el acceso habitual.
- Generar claves temporales con validez limitada, ideal para visitantes o servicios externos.
- Habilitar y deshabilitar los tipos de acceso.
- Visualizar el historial de accesos registrados por la cerradura.
- Gestionar un perfil personalizado, donde se puede editar información básica o actualizar credenciales de ingreso.

Por su parte, el modo administrador integra herramientas avanzadas de control:

- Creación y asignación de "códigos de casa" (identificadores únicos para cada unidad residencial), junto con información asociada.
- Filtrado de usuarios registrados mediante criterios como código de casa o número de cédula, facilitando la administración masiva.
- Eliminación selectiva de usuarios.

- Consulta detallada de notificaciones, donde el administrador puede visualizar el historial de accesos filtrando por código de casa y rangos de fechas específicos (desde/hasta), lo que permite búsquedas precisas y seguimiento de actividades.

La arquitectura de la aplicación prioriza una navegación intuitiva, con menús jerárquicos y retroalimentación visual inmediata (como confirmaciones de operaciones exitosas o alertas de errores). Adicionalmente, todas las transacciones sensibles (ej. cambio de contraseñas o eliminación de usuarios) requieren autenticación reforzada para prevenir accesos no autorizados.

3.3.6.1 Diagrama de funcionamiento de la aplicación

El siguiente diagrama de flujo **Figura 51** se detalla el procedimiento de funcionamiento de una aplicación que permite a los usuarios, tanto normales como administradores, acceder y gestionar diversas funcionalidades. El proceso comienza con el inicio de la aplicación y sigue un flujo lógico para permitir el registro, inicio de sesión y acceso a las diferentes opciones según el tipo de usuario.

Inicio del Proceso

El flujo comienza con el Inicio de la aplicación. En este punto, el usuario tiene dos opciones principales:

- Registro: Si el usuario no está registrado.
- Iniciar sesión: Si el usuario ya está registrado.

Registro de Usuario

Si el usuario decide registrarse, debe seguir el siguiente procedimiento:

- Ingresar datos : El usuario debe completar todos los campos obligatorios, incluyendo información personal y otros datos requeridos (como cédula, código de casa, etc.).
- Validación de campos: Se verifica si todos los campos están completos. Si algún campo falta, se muestra un mensaje indicando que (Todos los campos son obligatorios) y se vuelve al paso de ingreso de datos.
- Validación de cédula: Se comprueba si la cédula ingresada es válida. Si la cédula no es válida, se muestra un mensaje de error (Cédula inválida) y se vuelve al paso de ingreso de datos.
- Validación del código de casa: Se verifica si el código de casa ingresado existe en el sistema. Si el código no existe, se muestra un mensaje de error (El código de casa no existe) y se vuelve al paso de ingreso de datos.
- Confirmación de contraseñas: El usuario debe ingresar dos veces la contraseña para confirmarla. Si las contraseñas no coinciden, se muestra un mensaje de error (Contraseñas no coinciden) y se vuelve al paso de ingreso de datos.
- Registro exitoso: Si todos los pasos anteriores se cumplen correctamente, el registro se considera exitoso y el usuario puede continuar con el proceso.

Inicio de Sesión

Si el usuario ya está registrado, puede optar por iniciar sesión. El proceso es el siguiente:

- **Ingresar usuario (C.I.) y contraseña**: El usuario debe ingresar su identificación (C.I.) y contraseña para iniciar sesión.
- **Verificación de credenciales**: El sistema verifica si los datos ingresados son correctos. Si los datos son incorrectos, se muestra un mensaje de error (Datos incorrectos) y se le da la opción de reintentar o volver al inicio.

- **Opción de recuperación de contraseña**: Si el usuario olvida su contraseña, puede seleccionar la opción "¿Olvidó su contraseña?". En este caso, se le solicitará ingresar sus datos de recuperación (por ejemplo, correo electrónico) y se enviará una clave temporal al correo proporcionado. El usuario podrá restablecer su contraseña utilizando esta clave temporal.
- **Acceso al sistema**: Si los datos de inicio de sesión son correctos, el usuario accede al sistema.

Identificación del Tipo de Usuario

Una vez que el usuario inicia sesión, el sistema verifica si es un usuario normal o un administrador:

Usuario Normal

Accede al Menú principal, donde tiene disponibles varias funcionalidades:

- **Generar clave:** Permite generar una nueva clave de acceso mediante el ingreso de un número celular y la elección de un tiempo de validez.
- **Registro de contraseña:** Permite registrar una nueva contraseña de 6 dígitos y guardarla.
- **Tipos de accesos:** Muestra el estado de acceso (habilitado o deshabilitado), así como la lista de accesos realizados.
- **Notificaciones:** Permite ver la lista de notificaciones relacionadas con el acceso.
- **Perfil:** Permite editar los datos personales y cambiar la contraseña.

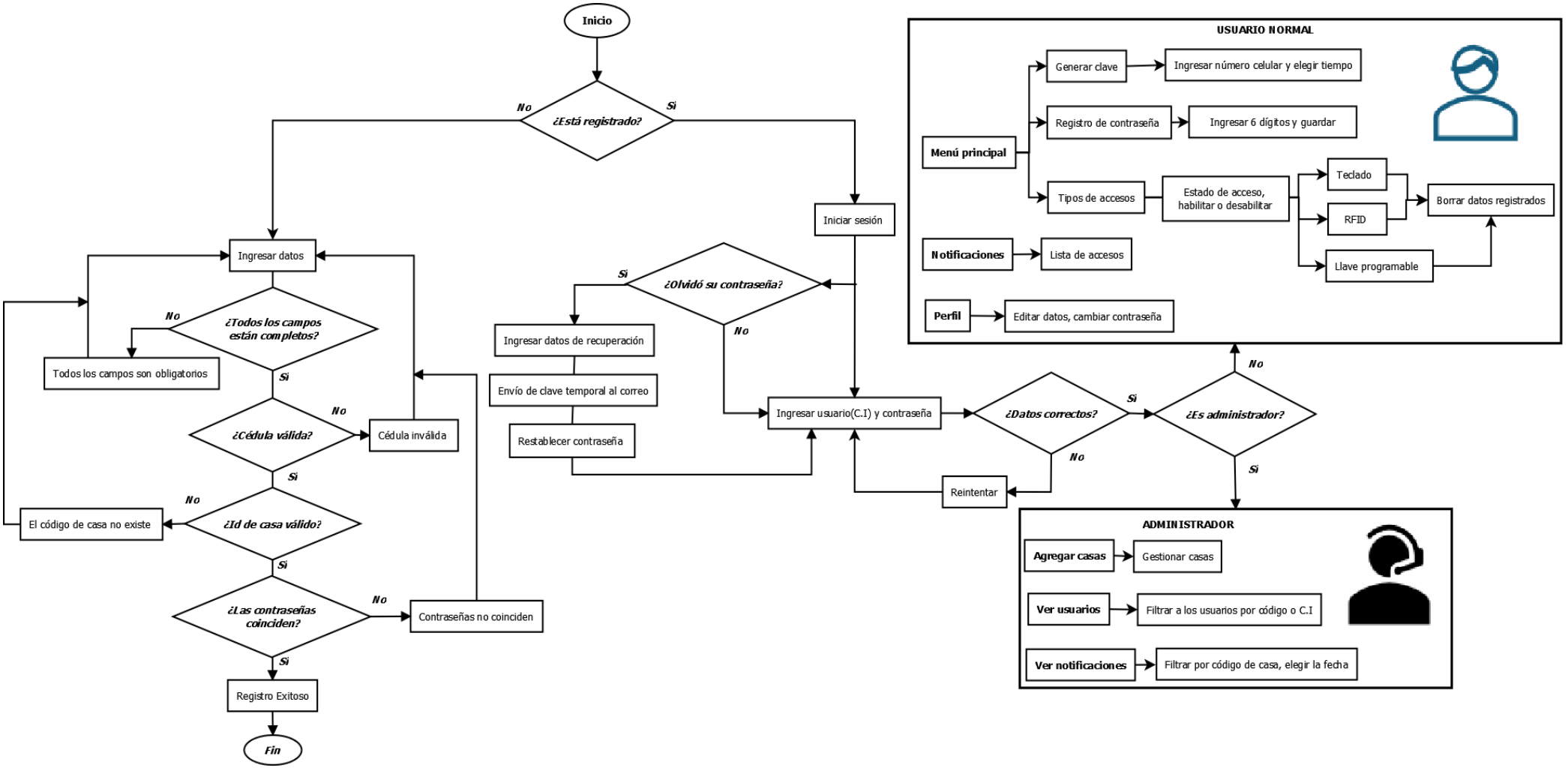
Administrador

Tiene acceso a funciones adicionales específicas para la gestión del sistema:

- **Agregar casas**: Permite agregar nuevas casas al sistema y gestionarlas.
- **Ver usuarios**: Filtra a los usuarios por código o C.I., facilitando la administración de cuentas.
- **Ver notificaciones**: Filtra las notificaciones por código de casa y fecha, permitiendo un seguimiento más detallado de eventos.

Figura 51

Diagrama de flujo del funcionamiento de la aplicación.



3.4 Tercera fase: Implementación

En esta fase se lleva a cabo la implementación y desarrollo del sistema de cerradura inteligente, poniendo en práctica todos los diseños elaborados previamente. Se inicia con la materialización de la llave y la cerradura inteligente a nivel de prototipo, utilizando una placa de pruebas (protoboard) para verificar el correcto funcionamiento de las conexiones eléctricas y la lógica de control. Para posteriormente, proceder con la fabricación de la placa de circuito impreso (PCB) para montar los componentes de forma permanente y compacta.

En paralelo, se desarrolla la aplicación móvil en Android Studio, la cual permite al usuario gestionar accesos, recibir notificaciones y controlar el sistema. Asimismo, se implementa la base de datos en MySQL, alojada en un servidor en la nube, y se crean los scripts en PHP que posibilitarán la comunicación entre los dispositivos físicos y el sistema web. Esta fase abarca tanto la parte electrónica como la de software, garantizando una integración funcional y segura del sistema completo.

3.4.1 Montaje del prototipo en protoboard

En el protoboard se realizan las conexiones de todos los componentes electrónicos sobre una placa de pruebas (protoboard), permitiendo realizar conexiones temporales para validar la funcionalidad del sistema antes de fabricar el PCB. Durante este proceso, se pueden realizar cambios en el diseño, detectar errores, revisar la alimentación de voltaje, y comportamiento del sistema ante solicitudes válidas e inválidas, garantizando una lógica robusta y coherente. Tanto para la llave como para la cerradura.

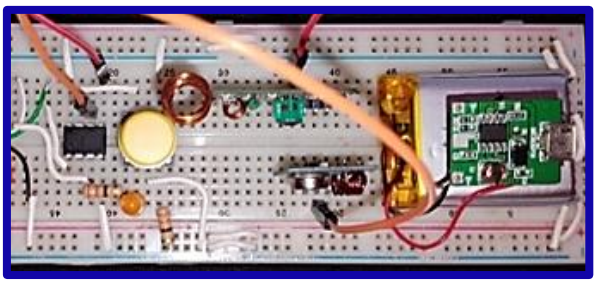
3.4.1.1 Llave electrónica

Para el desarrollo de la llave electrónica se utilizó inicialmente un protoboard de un solo canal, en el cual se implementó el armado preliminar del circuito. En esta etapa se integró un microcontrolador Arduino nano , seleccionado debido a su tamaño compacto, ideal para la miniaturización del dispositivo. Este microcontrolador se conectó a los módulos de radiofrecuencia (RF) encargados de la comunicación inalámbrica con la cerradura.

El objetivo principal de esta primera iteración fue explorar la viabilidad de construir una llave compacta sin comprometer su funcionalidad básica. Al momento de realizar las pruebas de transmisión de la señal y recepción no hubo complicaciones.

Figura 52

Diseño en protoboard – llave electrónica



3.4.1.2 Cerradura Inteligente

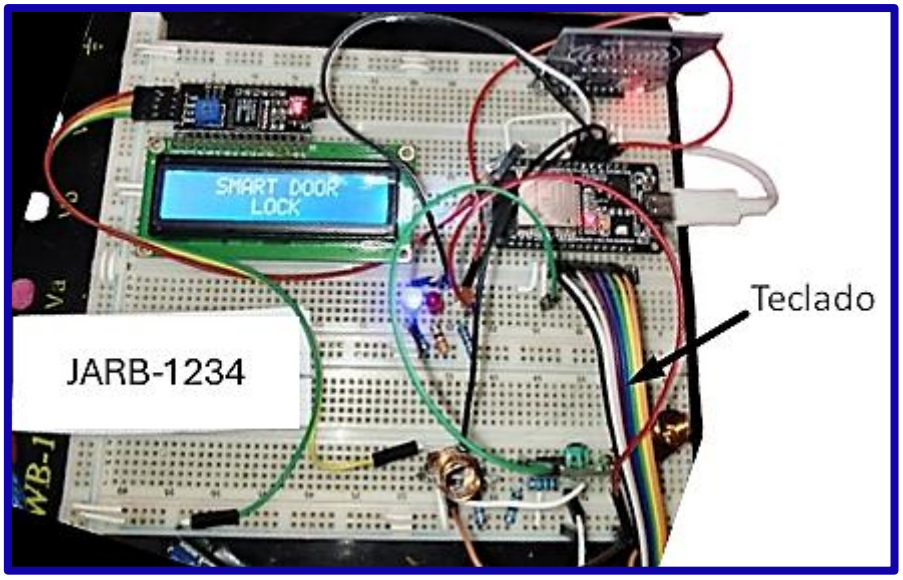
Este paso resulta fundamental para llevar a cabo las pruebas físicas del diseño teórico previamente desarrollado y simulado en el software Proteus. Además, se verifica que el circuito cumpla con todas las especificaciones técnicas establecidas antes de proceder a la fabricación de la placa de circuito impreso (PCB), lo cual asegura mayor fiabilidad y eficiencia en la producción final.

Una vez integrados todos los componentes electrónicos en el protoboard, se realizó el armado físico del circuito, lográndose comprobar el correcto funcionamiento del sistema. En

esta fase se ejecutaron pruebas preliminares correspondientes a los tres métodos de acceso propuestos: autenticación por teclado numérico, por tarjeta RFID y por llave de circuito programable. Estas pruebas permitieron validar el desempeño funcional del circuito bajo condiciones reales, identificando posibles inconsistencias entre la simulación y el comportamiento físico del sistema.

Figura 53

Diseño de cerradura inteligente en protoboard.



3.4.2 Fabricación e implementación en PCB- llave de circuito programable.

Una vez verificada la funcionalidad del prototipo, se procede a realizar el ensamblaje de la placa. En la **Figura 54** se muestra el patrón del circuito el cual se transfiere a la placa de cobre mediante técnicas como el planchado térmico, y posteriormente se lleva a cabo el proceso de revelado y soldadura de los componentes. El resultado es una versión funcional, compacta y permanente del sistema electrónico de la cerradura inteligente.

Se procede a la fabricación de la placa de circuito impreso (PCB). Este proceso inicia con el recorte de la baquelita, ajustándola al tamaño exacto del diseño. Esta acción garantiza que la placa se adapte adecuadamente a las dimensiones del circuito planeado.

Posteriormente, se lija la superficie de cobre de la baquelita con papel abrasivo fino, para que la transferencia del diseño sea uniforme y precisa. Una vez lijada, la placa se limpia con alcohol isopropílico.

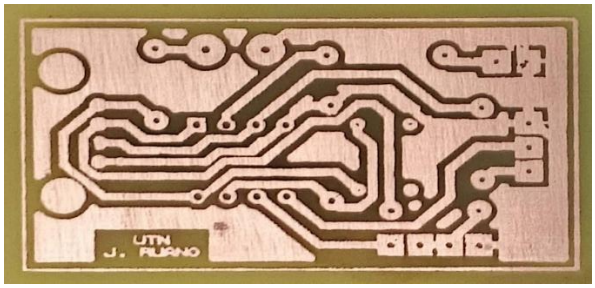
A continuación, se realiza la transferencia del diseño a la superficie de cobre , se coloca el diseño sobre la baquelita y se aplica calor mediante un planchado uniforme. Este paso permite que las trazas del circuito se adhieran al cobre, formando las pistas que posteriormente constituirán el diseño funcional de la PCB.

El siguiente paso es el grabado químico, donde la placa se sumerge en una solución de percloruro de hierro. Este compuesto químico reacciona con el cobre, eliminando las áreas que no forman parte del circuito.

Finalmente, una vez que el grabado está completo, la placa se retira del percloruro y se enjuaga con agua para detener la reacción química. La placa se seca y se inspecciona para asegurar que el diseño esté correctamente definido y que las pistas estén libres de interrupciones o defectos.

Figura 54

Diseño en baquelita de llave electrónica.



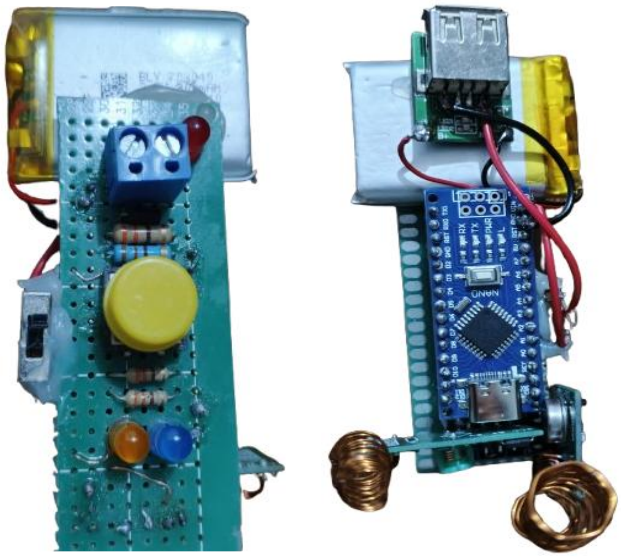
Nota. Se presenta el circuito ensamblado en su versión final, utilizando la placa de circuito impreso (PCB).

En la siguiente **Figura 55** se presenta el prototipo final de la llave electrónica. En el literal (a) , se muestra la vista frontal del dispositivo. Además del diseño básico, se incorporó un interruptor manual para permitir al usuario apagar y encender la llave cuando sea necesario. Esta modificación tiene como objetivo optimizar el consumo de energía de la batería, extendiendo su vida útil y garantizando un funcionamiento eficiente durante períodos prolongados.

En la vista posterior , ilustrada en el literal (b) , se observa la integración del Arduino Nano y el módulo de carga , lo cual permite mantener un tamaño compacto y portátil para la llave. La colocación estratégica de estos componentes contribuye a minimizar el volumen total del dispositivo, cumpliendo con los requisitos de miniaturización sin comprometer su funcionalidad.

Figura 55

Prototipo final de la llave.



(a) **(b)**

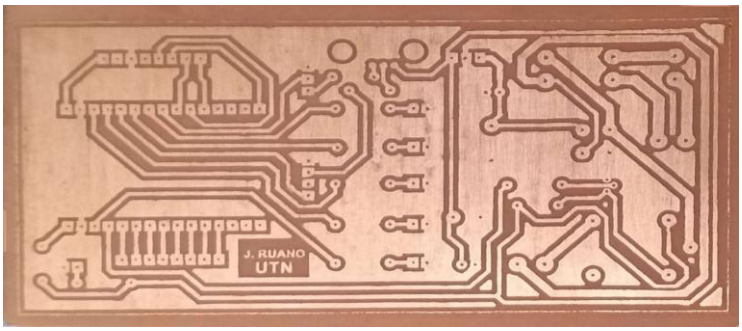
3.4.3 Cerradura

Para comprobar la integración de cada componente de la cerradura, se arma en protoboard basándonos en el diseño realizado en Proteus como se observa en la **Para el diseño de la placa base para el circuito de la cerradura se inicia diseñando las conexiones lógicas de todos los componentes en el simulador Proteus. Este esquema permite validar el correcto funcionamiento del sistema antes de implementarlo físicamente.**

La **Figura 56** ilustra la placa de circuito impreso (PCB) que fue diseñada tras las fases de validación en protoboard y simulación en software. Este diseño representa el paso final del proceso de desarrollo del circuito, el cual ya ha sido materializado en baquelita, evidenciándose en los trazos de cobre. La placa está lista para el ensamblaje de los componentes y su subsiguiente implementación en el sistema final.

Figura 56

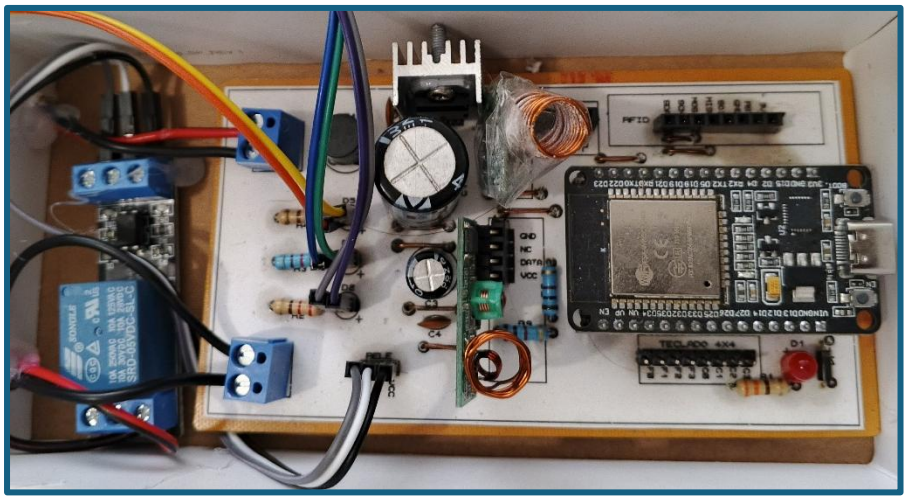
Diseño en baquelita – cerradura.



El dispositivo presentado corresponde al ensamblaje final de la placa electrónica diseñada para realizar pruebas de funcionamiento del sistema de acceso. Este desarrollo es el resultado de un proceso previo que incluyó la simulación del circuito en el software Proteus y su validación funcional en protoboard.

Figura 57

Placa de la cerradura eléctrica.



En la **Figura 58** muestra el prototipo final y en el literal (a) se observa en la pantalla el correcto funcionamiento al conectarse a internet, mientras que, en el literal (b) se muestra el menú inicial de sistema, en el cual muestra la hora (configurada con la zona horaria de Quito).

Figura 58

Inicio de sistema ya implementado.

(a) **(b)**

Este prototipo demuestra una solución efectiva para sistemas de acceso controlado, donde el usuario interactúa directamente con el sistema a través de un teclado numérico (PIN, clave temporal), RFID y llave programable. La combinación de contraseñas permanentes y claves temporales ofrece flexibilidad y seguridad, adaptándose a diversas necesidades de acceso. La interfaz intuitiva y la retroalimentación visual aseguran una experiencia de usuario fluida y confiable, consolidando el prototipo como una herramienta práctica y funcional para aplicaciones de cerraduras electrónicas. En la del literal (a) después de pulsar la tecla de activación A, indica que va acceder usando una contraseña. En el literal (b) al pulsar la tecla de activación C, en el sistema le muestra que es por clave temporal y debe ingresar la clave. En el literal (c), al presionar la tecla de activación B, el usuario puede registrar una tarjeta RFID. En el literal (d) al presionar la tecla de activación D, el usuario puede registrar una llave.

Figura 59

Prototipo final de cerradura inteligente.

(a)

(b)

(c)

(d)

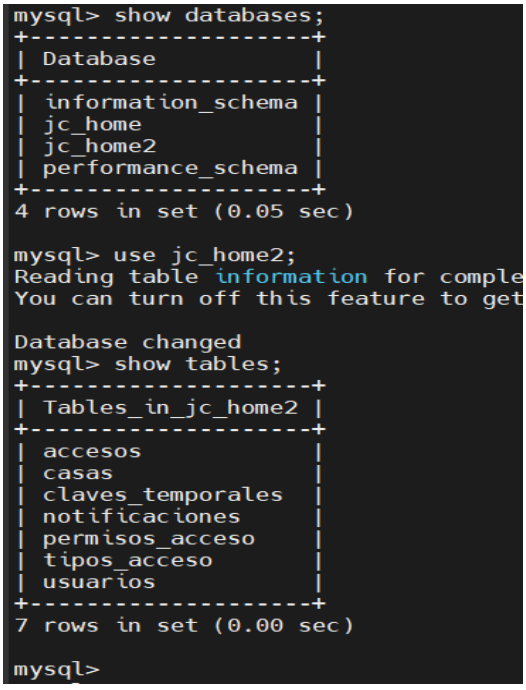
3.4.4 Base de Datos

Al consultar la estructura de la base de datos, se observan siete tablas principales: accesos, casas, claves_temporales, notificaciones, permisos_acceso, tipos_acceso y usuarios. Cada una de estas tablas cumple una función específica dentro del sistema, permitiendo la administración de usuarios, el registro de accesos y la gestión de permisos de manera integral.

En la **Figura** 60 ilustra la base de datos jc_home2, la cual ha sido diseñada con el propósito de gestionar el control de accesos residenciales. A través de esta base de datos, es posible administrar de manera eficiente los permisos de entrada, garantizando así un sistema seguro y organizado.

Figura 60

Base de datos jc_home2.

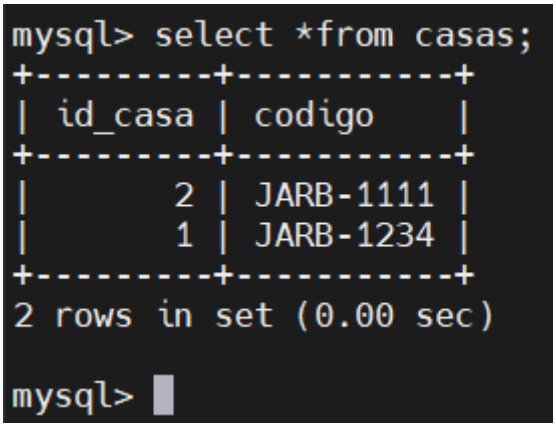


En la **Figura** 61, se muestra la tabla casas, perteneciente a la base de datos jc_home2, la cual almacena información sobre las viviendas registradas en el sistema de control de

accesos. Los datos han sido ingresados manualmente, permitiendo establecer identificadores únicos (id_casa) y códigos alfanuméricos (código) para cada propiedad.

Figura 61

Tabla casas



En la Figura 62 se muestra la tabla usuarios de la base de datos jc_home2 almacena la información de los residentes registrados en el sistema de control de accesos. Los datos han sido ingresados mediante la aplicación con fines de prueba.

Cada usuario dispone de un identificador único (id_usuario) y de información personal como nombre, número de cédula, celular y credenciales de acceso. Adicionalmente, el campo tipo_usuario permite diferenciar entre administradores y usuarios regulares.

Es importante señalar que los residentes están asociados a un código de casa mediante el campo id_casa, lo que facilita la identificación de la vivienda a la que pertenece cada usuario. Por ejemplo, los usuarios con id_usuario del 9 al 11 conforman una familia vinculada a la casa con id_casa = 1, mientras que los usuarios del 12 al 16 pertenecen a otra familia registrada bajo id_casa = 2.

Figura 62

Tabla de usuarios.

```
mysql> select * from usuarios;
+------------+-----------------+------------+------------+-------------------------------+--------------------------------------------------------------+--------------+------------+
| id_usuario | nombre          | cedula     | celular    | email                         | contraseña                                                   | tipo_usuario | id_casa    |
+------------+-----------------+------------+------------+-------------------------------+--------------------------------------------------------------+--------------+------------+
|          2 | Admin           | 10[redacted]503 | 09[redacted]321 |                               | $2y$10$qHh.tp2Wkd9BkYoJKnNO7OSnCJuOTNVmLS8oU3X1BuESJ7HgUocZ2 | administrador |       NULL |
|         27 | Zoilo Ruano     | 04[redacted]011 | 09[redacted]748 | j[redacted]es23@gmail.com     | $2y$10$dwQFvHf4s3r6gWTRS1Et9Or8iHmg41OFbj8N8Z2CWOLd99t5nxLNW | usuario      |         24 |
|         28 | Carmen Benavides | 04[redacted]498 | 09[redacted]162 | c[redacted]s16@gmail.com      | $2y$10$yPegdRkkDt61kz/psuFV9.Hz/gxaFhJwwIS4lRO3r0teko.gq2O8K | usuario      |         24 |
|         29 | Adamaris Ruano  | 10[redacted]260 | 09[redacted]404 | e[redacted]6@gmail.com        | $2y$10$.UzF5xbL1Qdb5zUf1a9jo.Q/TL4tLbQU3Sc87s5g.175RT9vbO6oi | usuario      |         24 |
|         30 | Maite Ruano     | 10[redacted]187 | 09[redacted]435 | r[redacted]@gmail.com         | $2y$10$59ie1mzYx8GUfmYO0pjIC.LaN9M8eOkKSdgwbNW3fSsXvw2Yq8BT2 | usuario      |         24 |
|         32 | Darwin Benavides | 21[redacted]682 | 09[redacted]876 | d[redacted]vides_123@hotmail.com | $2y$10$U9Km4eZF8uLR9GJgo7cEh.SrU.hxk3hdGDRygwxtg0IdnxzuhJF7i | usuario      |         25 |
|         34 | Marta Benavides | 04[redacted]686 | 09[redacted]847 | m[redacted]@gmail.com         | $2y$10$6vyFytzKLZYTTPJTPBfOFuORuG9rYkPZc0f6O3nvNQX59saEPz.Le | usuario      |         25 |
|         36 | Jessica Ruano   | 10[redacted]179 | 09[redacted]162 | j[redacted]6@utn.edu.ec       | $2y$10$RfnCyPehx9zyDTrLl1aTh.ANCtkNXqnhDgJ6qraQVbfOGbnUIRheC | usuario      |         24 |
+------------+-----------------+------------+------------+-------------------------------+--------------------------------------------------------------+--------------+------------+
8 rows in set (0.00 sec)
```

Finalmente, se presenta la tabla denominada tipos_acceso, la cual forma parte del modelo de datos del sistema de control de accesos. Esta tabla almacena los distintos métodos de autenticación disponibles, cada uno identificado por un id_tipo_acceso único y una descripción textual en el campo tipo.

De acuerdo con los datos registrados, se reconocen cuatro tipos de acceso: RFID, teclado, llave y clave temporal. Esta clasificación permite al sistema asociar con precisión el método de autenticación utilizado en cada intento de acceso, facilitando tanto el control como el registro de los eventos de entrada al sistema.

Figura 63

Tipos de acceso.

```
mysql> select *from tipos_acceso;
+----------------+----------------+
| id_tipo_acceso | tipo           |
+----------------+----------------+
|              1 | rfid           |
|              2 | teclado        |
|              3 | llave          |
|              4 | clave_temporal |
+----------------+----------------+
4 rows in set (0.01 sec)

mysql>
```

El sistema de control de accesos implementa un mecanismo flexible y dinámico para gestionar los permisos de acceso de los usuarios, utilizando la tabla permisos_acceso. Esta tabla contiene información clave sobre los tipos de acceso disponibles para cada usuario, así como su estado actual de activación.

Cada registro en la tabla está compuesto por los siguientes campos:

- id_permiso : Identificador único del permiso.
- id_usuario : Identifica al usuario al que se le asigna el permiso.
- id_tipo_acceso : Indica el tipo de acceso (por ejemplo: teclado numérico, RFID, llave programable, clave temporal).
- activo : Campo binario que define si el método de acceso está habilitado (1) o deshabilitado (0).

Desde la perspectiva funcional, esta tabla permite configurar qué métodos de autenticación puede utilizar cada usuario. Por ejemplo, un usuario puede tener acceso habilitado por teclado y RFID, pero no por llave electrónica, dependiendo de las necesidades específicas de seguridad o uso.

En la interfaz de la aplicación, se ha implementado una funcionalidad que permite a los usuarios finales gestionar los permisos de manera intuitiva. Cuando un usuario realiza un cambio en el estado de un método de acceso (ya sea habilitar o deshabilitar), la aplicación actualiza automáticamente el campo activo en la base de datos. Esto asegura que los cambios sean persistentes y se reflejen de inmediato en el sistema.

Este proceso garantiza que el usuario solo pueda acceder mediante los métodos que estén activos (estado 0), lo que cumple con los principios de personalización y seguridad del sistema. Este diseño brinda una alta flexibilidad, permitiendo adaptar los permisos de acceso a las necesidades individuales de cada usuario, sin afectar la integridad del sistema ni comprometer la experiencia del usuario final.

Figura 64

Tabla de permisos para los accesos.

```
mysql> select * from permisos_acceso;
+------------+------------+----------------+--------+
| id_permiso | id_usuario | id_tipo_acceso | activo |
+------------+------------+----------------+--------+
|        181 |         27 |              1 |      1 |
|        182 |         27 |              2 |      1 |
|        183 |         27 |              3 |      1 |
|        184 |         27 |              4 |      1 |
|        188 |         28 |              1 |      1 |
|        189 |         28 |              2 |      1 |
|        190 |         28 |              3 |      1 |
|        191 |         28 |              4 |      1 |
|        195 |         29 |              1 |      1 |
|        196 |         29 |              2 |      1 |
|        197 |         29 |              3 |      1 |
|        198 |         29 |              4 |      1 |
|        202 |         30 |              1 |      0 |
|        203 |         30 |              2 |      1 |
|        204 |         30 |              3 |      1 |
|        205 |         30 |              4 |      1 |
|        210 |         32 |              2 |      1 |
|        212 |         34 |              2 |      1 |
|        214 |         36 |              2 |      0 |
+------------+------------+----------------+--------+
19 rows in set (0.00 sec)
```

3.4.5 Aplicación móvil

La aplicación móvil desempeña un papel importante en la materialización de este proyecto, ya que constituye uno de los principales medios de interacción entre el usuario y la cerradura eléctrica. En los siguientes apartados, se presenta las pantallas principales de la aplicación móvil, diseñadas para ofrecer una experiencia de usuario intuitiva y eficiente.

Estas interfaces representan los puntos clave de interacción entre el usuario y el sistema, abordando los procesos de bienvenida, registro y acceso.

3.4.5.1 Interfaz de presentación

En la **Figura** 65 se muestra la pantalla de presentación tanto para inicio de sesión como para registro de usuarios.

Figura 65

Pantalla de bienvenida.

Nota. En cumplimiento con los requerimientos, la interfaz de bienvenida ha sido diseñada para ser intuitiva y de fácil uso, garantizando una experiencia accesible y agradable para todos los usuarios.

3.4.5.2 Interfaz de registro

La **Figura** 67 muestra la pantalla de registro de la aplicación, diseñada para recopilar la información necesaria de los usuarios. Al acceder a esta sección, el usuario encontrará un mensaje con las indicaciones a seguir. Es fundamental que el usuario complete cada uno de los campos del formulario, incluyendo el "ID de casa", "Nombre", "Cédula" y "Celular". Además, existen dos campos específicos para la creación de credenciales de acceso: "Ingrese una contraseña" y "Confirmar contraseña", lo que asegura que el usuario configure su contraseña de manera precisa y minimice la posibilidad de errores.

Figura 66

Pantalla de registro de usuarios.

Validación de datos en el formulario de registro

El proceso de registro del usuario está diseñado con múltiples capas de validación, cuya finalidad es garantizar la integridad, autenticidad y coherencia de los datos ingresados. Este mecanismo previene errores comunes del usuario y asegura que la información recopilada sea válida antes de enviarse al servidor.

En primer lugar, todos los campos del formulario son obligatorios, por lo que no se permite el envío de la solicitud si alguno de ellos está vacío.

En segundo lugar, se implementa una validación específica para la cédula, con base en el algoritmo oficial de verificación como se observa en la **Tabla 24.** Este algoritmo evalúa la estructura del número de cédula, que debe contener exactamente 10 dígitos, comenzar con un

código de provincia válido (entre 01 y 24) y cumplir con la lógica de control del dígito verificador. Este filtro garantiza que no se ingresen números incorrectos o inventados.

Asimismo, el campo de correo electrónico es de carácter obligatorio, ya que a través de este canal se enviarán notificaciones importantes al usuario, como claves temporales y notificaciones.

En cuanto a las credenciales, la contraseña debe ingresarse dos veces en los campos "Contraseña" y "Confirmar contraseña". El sistema verifica que ambas coincidan exactamente para evitar errores de escritura involuntarios. Solo si todos estos criterios se cumplen, la información se empaqueta y se envía mediante una solicitud HTTP POST al servidor para su almacenamiento en la base de datos.

Tabla 24

Resumen de reglas para formulario de registro

Paso	Reglas de validación	Retroalimentación del usuario
1	Ningún campo vacío	Todos los campos son obligatorios.
2	Cédula ecuatoriana coherente (10 dígitos, provincia 01-24, algoritmo de módulo 10)	Todos los campos son obligatorios.
3	Coincidencia exacta entre contraseña y confirmación	Todos los campos son obligatorios.
4	El correo es obligatorio	Todos los campos son obligatorios.

3.4.5.3 Interfaz de inicio de sesión

La pantalla de inicio de sesión está diseñada para autenticar a los usuarios registrados previamente en el sistema. En esta interfaz no se solicita un nombre de usuario convencional, ya que se ha definido que la cédula de identidad (C.I.) será utilizada como identificador único del usuario. Por lo tanto, el primer campo, denominado “Usuario”, permite ingresar el número de cédula registrado. El segundo campo, titulado “Contraseña”, está destinado al ingreso de la clave establecida durante el proceso de registro.

Ambos campos son obligatorios para acceder al sistema. Una vez validados los datos ingresados, el sistema permitirá el acceso a las funcionalidades disponibles dentro de la aplicación, de acuerdo con el tipo de usuario (administrador o usuario estándar).

Figura 67

Pantalla de registro e inicio.

Restablecimiento de contraseña

El sistema incorpora una funcionalidad para restablecer la contraseña en caso de que el usuario haya olvidado sus credenciales de acceso. Esta opción se encuentra accesible desde la pantalla de inicio de sesión, mediante el enlace "¿Olvidaste tu contraseña?". Al seleccionar esta opción, se despliega un cuadro de diálogo emergente titulado Recuperar contraseña como se observa en la **Figura** 68, que solicita tres datos imprescindibles para validar la identidad del usuario:

- Número de cédula
- Código de casa (por ejemplo: JARB-1234)

- Número de teléfono

Figura 68

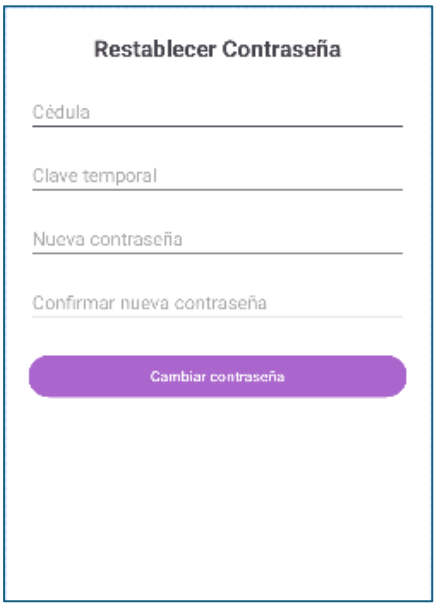
Recuperar contraseña.

Una vez ingresados y enviados estos datos, el sistema los verifica con la base de datos y, en caso de ser válidos, procede al envío de una clave temporal al correo electrónico asociado a ese usuario. Esta clave sirve como medio de verificación para completar el proceso de restablecimiento.

Posteriormente, el usuario es redirigido a una nueva pantalla titulada Restablecer Contraseña **Figura** 69, donde debe ingresar la clave temporal enviada por correo, la nueva contraseña y su confirmación. Esta validación doble garantiza que el proceso sea seguro y que sólo el propietario del correo pueda establecer una nueva contraseña.

Figura 69

Restablecer contraseña.



Esta característica mejora significativamente la usabilidad y seguridad del sistema, ya que permite una recuperación autónoma, rápida y confiable de las credenciales de acceso, sin necesidad de intervención administrativa.

3.4.5.4 Interfaz - Menú Inicio

La **Figura** 70 muestra, un menú de navegación la cual muestra la pantalla principal de la aplicación una vez que el usuario ha iniciado sesión. Está estructurada de la siguiente manera:

- **Saludo personalizado y descripción de la funcionalidad**

En la parte superior aparece un mensaje de bienvenida (“Hola”) seguido del nombre de la persona. Además, se describe de manera general las tres acciones principales que el usuario puede ejecutar desde esta vista.

- **Acciones principales**

Se presentan tres botones, cada uno acompañado de un icono representativo y un rótulo claro:

- ***Generar clave (icono de un SMS saliente)***

  Permite enviar de una clave temporal por mensaje de texto.

- ***Registro de contraseña (icono de un teclado numérico)***

  El usuario registra por primera vez, la contraseña numérica.

- ***Administrar Accesos (icono de una tarjeta RFID)***

  Da acceso a la gestión de métodos adicionales (RFID, llaves programables, etc.).

Figura 70

Interfaz de registro, cambio y envío de contraseña.

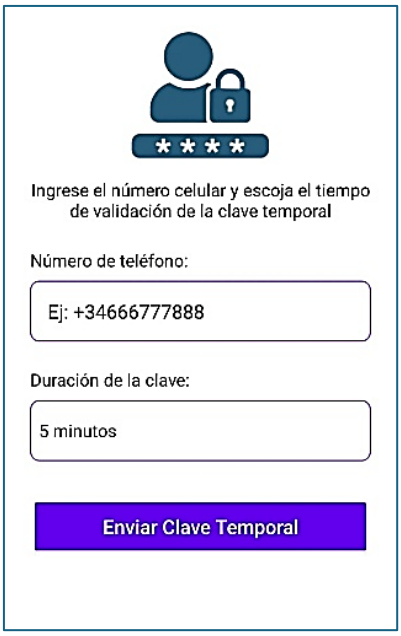
Nota. Cada elemento ha sido diseñado para ser fácilmente identificable, así como se establecen en los requerimientos.

Clave temporal

El diseño de esta interfaz permite al usuario ingresar el número de contacto al cual se enviará la clave temporal mediante un mensaje de texto (SMS). Además, ofrece la funcionalidad de seleccionar el tiempo de validez de la clave, que puede oscilar entre 5, 10, 30 minutos, o extenderse hasta un rango de 1 a 2 horas, según la preferencia del usuario.

Figura 71

Envió de clave temporal por SMS.



Registro de contraseña numérica

Se debe registrar una contraseña numérica única, la cual es almacenada de forma segura en la base de datos. Adicionalmente, se verificó la funcionalidad opcional para eliminar la

contraseña existente, así como el proceso de cambio de contraseña, garantizando que el usuario pueda actualizarla en caso de ser necesario.

La siguiente interfaz está diseñada para que **c**ada usuario, al ingresar por primera vez al sistema facilitar el registro inicial de este método de acceso por contraseña numérica. Si en la opción de accesos se elimina la contraseña se puede volver a registrar en esta opción.

Figura 72

Registro de contraseña para método de autenticación.

Tipos de accesos

La **Figura** 73 muestra la interfaz gráfica correspondiente a la pantalla de Tipos de Acceso, diseñada para permitir al usuario gestionar los métodos de autenticación que pueden estar en estado habilitado o deshabilitado. Se muestra los tres métodos que tienen datos de registro:

Teclado: permite visualizar las contraseñas almacenadas y asociadas al acceso por teclado. El interruptor superior habilita o deshabilita este método.

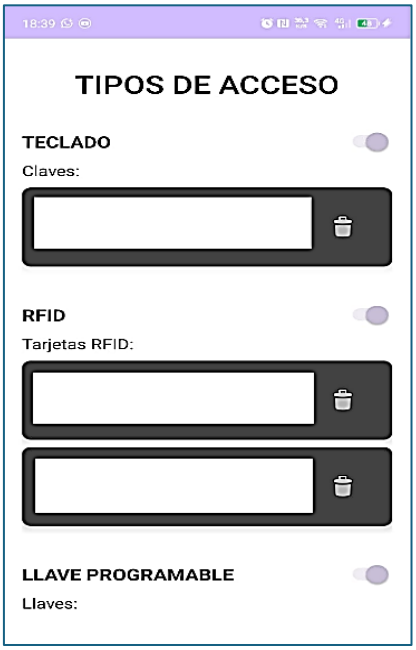
RFID: muestra las tarjetas RFID registradas. Se pueden visualizar y eliminar individualmente. El interruptor también permite activar o desactivar el uso del lector RFID.

Llave programable: en este apartado se listan las llaves generadas por radiofrecuencia. Al igual que los otros métodos, puede habilitarse o inhabilitarse según preferencia del usuario mediante el conmutador.

Esta funcionalidad ofrece al usuario final control total sobre qué medios están habilitados y qué credenciales están registradas.

Figura 73

Gestión de métodos de acceso.



3.4.5.5 Interfaz – menú de notificaciones

El menú de notificaciones está diseñado para proporcionar un registro detallado de los accesos realizados durante el día de cada integrante de la residencia. En este registro, se incluye información como el método de acceso utilizado (por ejemplo, RFID o llave electrónica), así como el estado del acceso, que puede ser permitido o denegado. Además, se detalla la fecha y

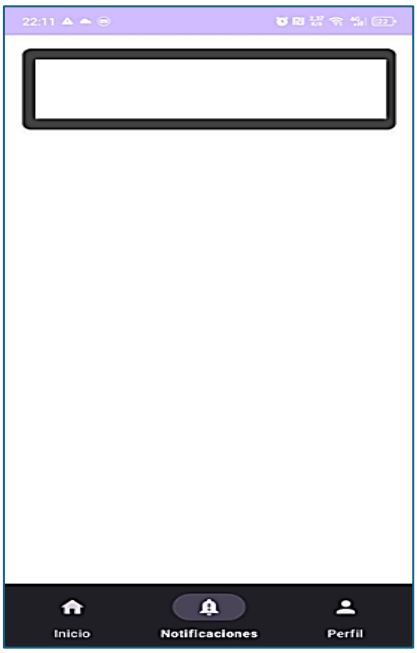
hora exacta en la que cada acceso tuvo lugar. De esta manera, el diseño asegura que el usuario pueda ver la lista de accesos de forma precisa y ordenada, el orden del listado se muestra

El menú de notificaciones ha sido diseñado con el objetivo de brindar al usuario un registro claro y detallado de los accesos efectuados en la residencia. Esta funcionalidad permite supervisar las actividades de ingreso de cada integrante del hogar, mostrando información clave como el método de autenticación empleado (por ejemplo, teclado, RFID o llave programable) y el estado del intento de acceso, que puede ser permitido o denegado.

Para asegurar una trazabilidad efectiva, cada registro incluye la fecha y hora exacta en la que se produjo el acceso. Un aspecto clave del diseño es cómo se presentan los datos: los registros más recientes aparecen al inicio de la lista, lo que facilita una revisión inmediata de los eventos más actuales. Este enfoque cronológico inverso permite al usuario identificar rápidamente cualquier intento sospechoso o acceso reciente sin tener que desplazarse por toda la lista. Gracias a esta organización estructurada y a la claridad visual, el usuario puede realizar un monitoreo efectivo y tomar decisiones informadas sobre la seguridad de su vivienda.

Figura 74

Notificaciones - listas de acceso.



3.4.5.6 Interfaz menú del perfil de usuario

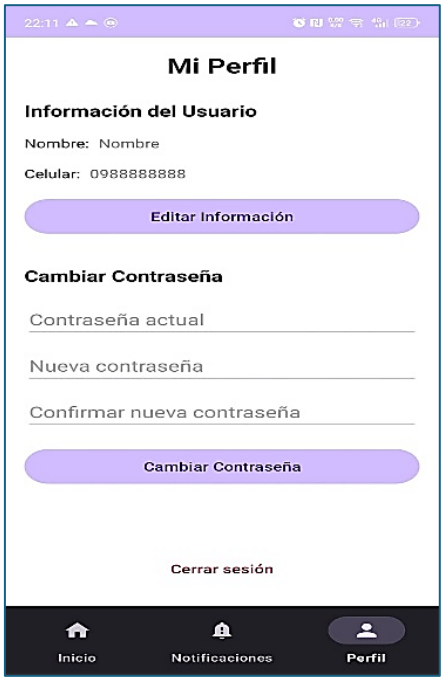
La interfaz correspondiente a la sección perfil ha sido diseñada para que el usuario pueda gestionar de manera intuitiva y segura tanto su información personal como sus credenciales de acceso. Tal como se observa en la **Figura** 75**,** se presenta un resumen de los datos personales registrados, específicamente el nombre del usuario y su número de celular. A través del botón Editar Información, el usuario puede actualizar estos datos en caso de ser necesario, garantizando así que la información registrada se mantenga actualizada.

En la parte inferior de la misma vista, se ha incluido una sección dedicada al cambio de contraseña. Aquí, el usuario debe ingresar su contraseña actual, una nueva contraseña y confirmarla para proceder con la actualización. Esta funcionalidad refuerza la seguridad del sistema al permitir al usuario modificar su clave en cualquier momento, especialmente en situaciones de olvido o ante la sospecha de uso no autorizado.

Finalmente, se incorpora una opción para cerrar sesión, ubicada al pie de la interfaz, que permite al usuario salir de forma segura de la aplicación.

Figura 75

Perfil de usuario.



3.4.5.7 Interfaz del Administrador

La interfaz presenta una pantalla optimizada para el rol de Administrador, caracterizada por un diseño limpio y funcional. En la sección superior, resalta el título "ADMINISTRADOR" en letras grandes y negritas, lo que enfatiza el propósito de esta área. Seguidamente, se encuentran tres botones, cada uno vinculado a una función esencial: "AGREGAR CASAS", "VER USUARIOS" y "VER NOTIFICACIONES". La disposición general de estos elementos es intuitiva y eficiente, facilitando al administrador un acceso rápido a las herramientas necesarias para la gestión de casas, usuarios y notificaciones.

Figura 76

Funcionalidades del administrador.

Agregar casas

En esta interfaz, el administrador puede agregar un código de casa junto con una descripción asociada. Asimismo, se dispone de la opción de filtrar los códigos de casa para facilitar una búsqueda rápida, especialmente en caso de contar con un gran número de registros.

Figura 77

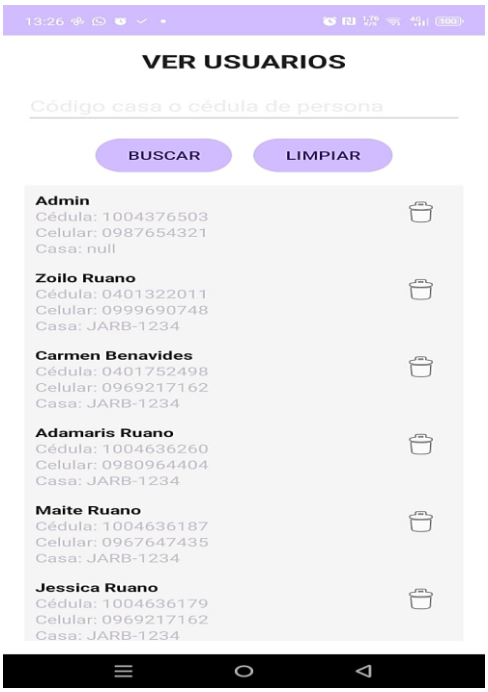
Agregar y eliminar casas.

Ver usuarios

En esta interfaz se muestra una lista de todos los usuarios registrados. El administrador tiene la posibilidad de filtrar o buscar usuarios ingresando el código de casa o la cédula del usuario, con el fin de realizar una búsqueda rápida. Asimismo, desde esta misma interfaz es posible eliminar a un usuario cuando sea necesario.

Figura 78

Interfaz ver usuarios.



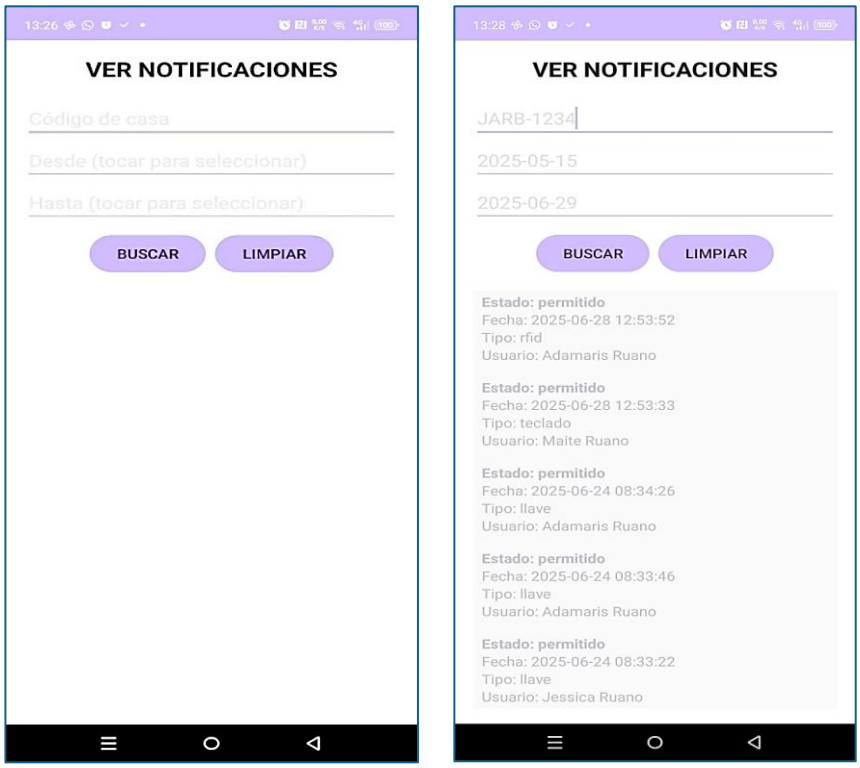
Ver notificaciones

En esta interfaz, el administrador puede visualizar los listados de acceso asociados a cada código de casa. Además, se permite seleccionar un rango de fechas para consultar las notificaciones desde una fecha específica, facilitando así la revisión de registros de acceso según sea necesario.

En la **Figura 79** en el literal (a), se observa inicial mente la interfaz sin realizar ninguna búsqueda, en el literal (b), se filtra por fechas y se muestran todas las notificaciones de accesos.

Figura 79

Ver notificaciones desde administrador.



(a) **(b)**

4 CAPÍTULO IV: PRUEBAS DE FUNCIONAMIENTO

En las pruebas de funcionamiento se describe el proceso de verificación y validación del sistema de cerradura electrónica desarrollado. Este capítulo se centra en evaluar el desempeño de los diferentes módulos integrados, como la fuente de alimentación, el microcontrolador ESP32-WROOM, los métodos de acceso (teclado numérico, RFID, llave electrónica, entre otros) y la aplicación móvil. Las pruebas tienen como objetivo garantizar que cada componente funcione correctamente, cumpla con los requisitos establecidos y se integre de manera eficiente dentro del sistema general. Además, se evalúan aspectos críticos como la estabilidad, la seguridad, la confiabilidad del acceso y la interacción entre los módulos físicos y digitales. A través de estas pruebas, se busca demostrar que el sistema es capaz de operar de manera óptima y cumplir con las expectativas de diseño planteadas en las fases anteriores.

4.1 Cuarta fase: Pruebas

La cuarta fase, enmarcada dentro de la metodología en cascada, se enfoca en validar el funcionamiento integral del sistema de acceso. Para lograr este objetivo, se implementó un plan de pruebas que evalúa los siguientes aspectos: pruebas de aplicación, ingreso por teclado, y el sistema de notificaciones. La **Tabla 25** detalla los casos de prueba diseñados, así como sus respectivos criterios de validación y los componentes involucrados en cada uno.

Tabla 25

Casos de prueba del sistema de acceso

Casos de prueba	Descripción	Resultados esperados	Herramientas
Caso 1: pruebas de aplicación	Verificar el registro de usuarios, inicio de	Cada usuario pueda acceder al sistema, al código de casa correspondiente, acceder sin	Aplicación, base de datos y correo.

	sesión, restablecimiento de contraseña.	problemas a las funcionalidades de la aplicación, en caso de olvidar la contraseña de la aplicación recuperar la contraseña.	
Caso 2: Ingreso por teclado	Permite ingresar con la contraseña registrada y la clave temporal.	Verificar el ingreso por teclado cuando este habilitado y deshabilitado. Verificar el ingreso por clave temporal con distintos tiempos y cuando esté caducada.	Base de datos y aplicación.
Caso 3: Acceso por RFID	Debe ingresar datos validos del usuario que quiere registrar una o más tarjetas RFID.	Verificar que el usuario pueda registrar una o más tarjetas desde el sistema físico.	Base de datos y aplicación.
Caso 4: llave de circuito programable	Debe enviar el código único.	Al presionar el botón pulsador ubicado en la placa de la llave, se efectúa una comunicación inalámbrica entre los módulos RF y se abre la puerta.	Base de datos y aplicación.
Caso 5: Notificaciones	El sistema debe notificar vía correo electrónico y en la aplicación.	Que se envíen las notificaciones a todos los correos registrados en ese código de casa. Que se muestre las notificaciones en la aplicación cuando este en primer plano	Correo

4.1.1 Pruebas de la aplicación

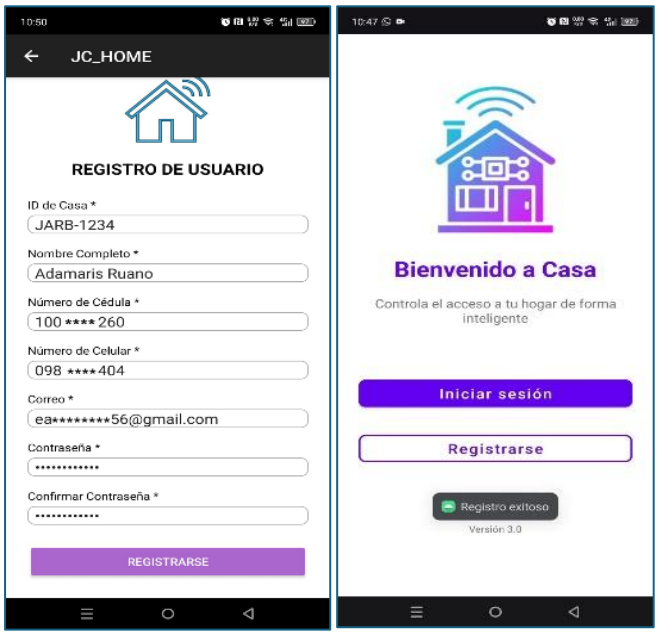
Se realizan tres pruebas, del registro de usuarios, inicio de sesión y pruebas de restablecer contraseña de la aplicación.

4.1.1.1 Registro de usuarios

Como parte de resultados se verifica el funcionamiento de la aplicación, como se observan en la **Figura** 80 para el registro de usuarios se llenan con datos válidos, si los datos son correctos, el registro es exitoso y puede iniciar sesión con sus credenciales.

Figura 80

Registro de usuario.

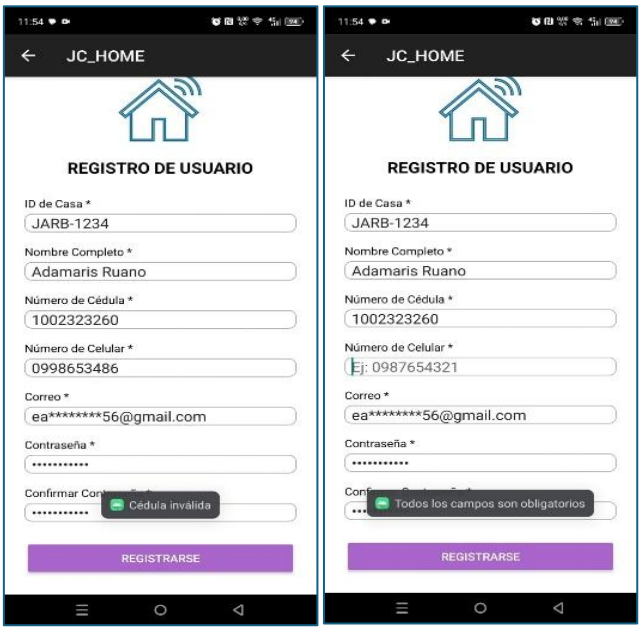


Como parte del registro, se lleva a cabo la validación específica sobre el número de cédula ingresado. En caso de no cumplir con los criterios, el sistema genera una alerta indicando **cédula inválida**, como se observa en la **Figura** 81.

Adicionalmente, todos los campos marcados con un asterisco (*) son obligatorios; por tanto, la omisión de cualquiera de ellos impide completar el proceso de registro. En estos casos, el sistema despliega una notificación emergente con el mensaje **todos los campos son obligatorios**, como se muestra en **Figura** 81 .

Figura 81

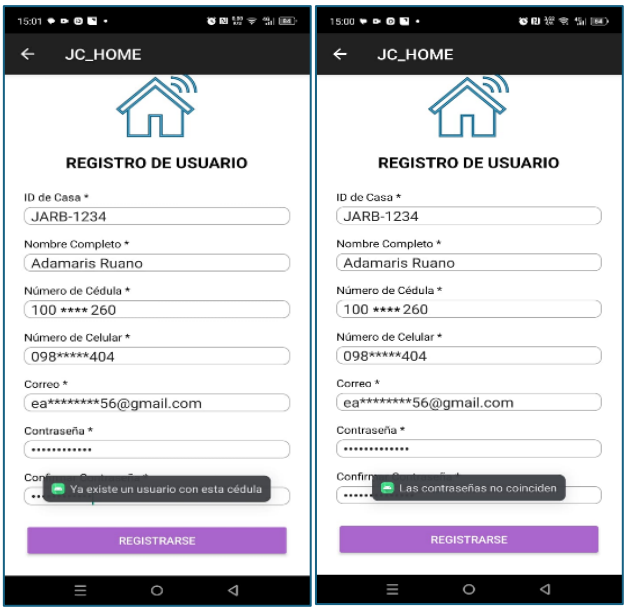
Validación de datos del Formulario de Registro.



Estas restricciones garantizan que únicamente usuarios con información completa y verificada puedan registrarse. Esta funcionalidad es esencial para preservar la fiabilidad del sistema de control de accesos, previniendo registros duplicados, datos erróneos o identidades inválidas, como se observa en la **Figura** 82.

Figura 82

Validación de datos del formulario de registro.



Como parte de evidencia se muestra en la siguiente **Figura** 83 la tabla casas que son las casas registradas, para pruebas se realizan los registros con el código de casa JARB-1234 con ID 24.

Figura 83

Contenido de la tabla casas.

```
mysql> select * from  casas;
+---------+-----------+--------------+
| id_casa | codigo    | descripcion  |
+---------+-----------+--------------+
|      24 | JARB-1234 | San Agustín  |
|      25 | JARB-0001 | La florida   |
+---------+-----------+--------------+
2 rows in set (0.00 sec)
```

En la **Figura** 84 se realizaron 5 registros de usuarios pertenecientes a un mismo id de casa 24, todos los registros fueron exitosos. Además, la base de datos registró adecuadamente otros usuarios pertenecientes al código de casa JARB-0001 con Id 25.

Figura 84

Contenido de tabla de registro de usuarios.

```
mysql> select * from usuarios;
+------------+----------------+----------+----------+-------------------------------+--------------------------------------------------------------+--------------+----------+
| id_usuario | nombre         | cedula   | celular  | email                         | contraseña                                                   | tipo_usuario | id_casa  |
+------------+----------------+----------+----------+-------------------------------+--------------------------------------------------------------+--------------+----------+
|          2 | Admin          | 100[redacted]503 | 098[redacted]321 | [redacted]                    | $2y$10$qHh.tp2Wkd9BkYoJKnNO7OSnCJuOTNVmLS8oU3X1BuESJ7HgUocZ2 | administrador |     NULL |
|         27 | Zoilo Ruano    | 040[redacted]011 | 099[redacted]748 | j[redacted]3@gmail.com        | $2y$10$dwQFvHf4s3r6gWTRS1Et9Or8iHmg41OFbj8N8Z2CWOLd99t5nxLNW | usuario      |       24 |
|         28 | Carmen Benavides | 040[redacted]498 | 096[redacted]162 | c[redacted]5@gmail.com        | $2y$10$yPegdRkkOt61kz/psuFV9.Hz/gxaFhJwwIS4lRO3r0teko.gq2O8K | usuario      |       24 |
|         29 | Adamaris Ruano | 100[redacted]260 | 098[redacted]404 | e[redacted]6@gmail.com        | $2y$10$.UzF5xbL1Qdb5zUf1a9jo.Q/TL4tLbQU3Sc87s5g.175RT9vbO6oi | usuario      |       24 |
|         30 | Maite Ruano    | 100[redacted]187 | 096[redacted]435 | r[redacted]a@gmail.com        | $2y$10$59ie1mzYx8GUfmYO0pjIC.LaN9M8eOkKSdgwbNW3fSsXvw2Yq8BT2 | usuario      |       24 |
|         31 | Jessica Ruano  | 100[redacted]179 | 096[redacted]162 | j[redacted]b@utn.edu.ec       | $2y$10$I4Uk9R.mg6dIpeOXo4A6j.tBYHGPJqA2zxX5H9bxAHa4bm1h01ZLK | usuario      |       24 |
|         32 | Darwin Benavides | 210[redacted]682 | 098[redacted]876 | d[redacted]23@hotmail.com     | $2y$10$U9Km4eZF8uLR9GJgo7cEh.SrU.hxk3hdGDRygwxtg0IdnxzuhJF7i | usuario      |       25 |
|         34 | Marta Benavides | 040[redacted]686 | 098[redacted]847 | m[redacted]s@gmail.com        | $2y$10$6vyFytzKLZYTTPJTPBfOFu0RuG9rYkPZc0f6O3nvNQX59saEPz.Le | usuario      |       25 |
+------------+----------------+----------+----------+-------------------------------+--------------------------------------------------------------+--------------+----------+
8 rows in set (0.00 sec)
```

4.1.1.2 Inicio de sesión

Se procedió a evaluar, el correcto funcionamiento de la interfaz de inicio de sesión de la aplicación, destinada a usuarios previamente registrados. La **Figura** 85 presentada corresponde a la pantalla de acceso, donde se validan credenciales como el nombre de usuario

(C.I) y la contraseña. Durante las pruebas, se verificó que los campos respondieran correctamente a la entrada de datos, mostrando mensajes de inicio exitoso, en caso de información incorrecta también muestre mensajes de usuario no encontrado o contraseña incorrecta como se muestra en la **Figura** 86**.**

Figura 85

Validar credenciales, para inicio de sesión.

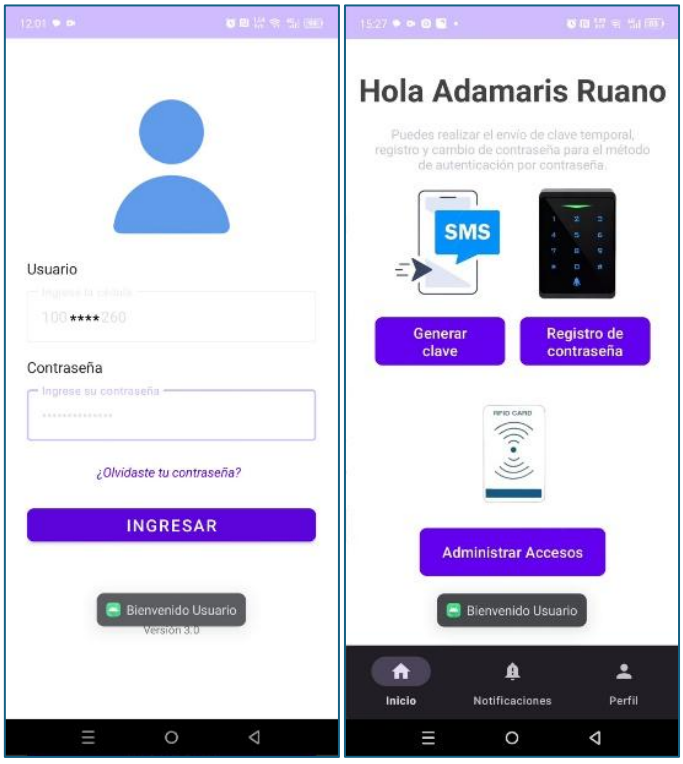
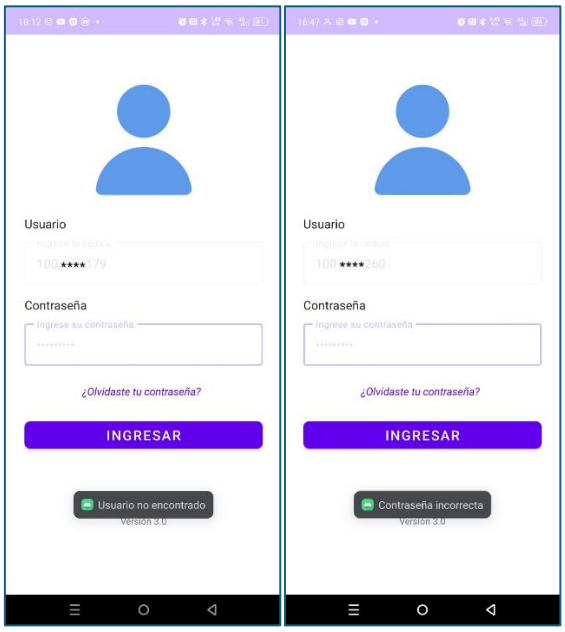


Figura 86

Inicio de sesión- usuario no registrado y contraseña incorrecta.

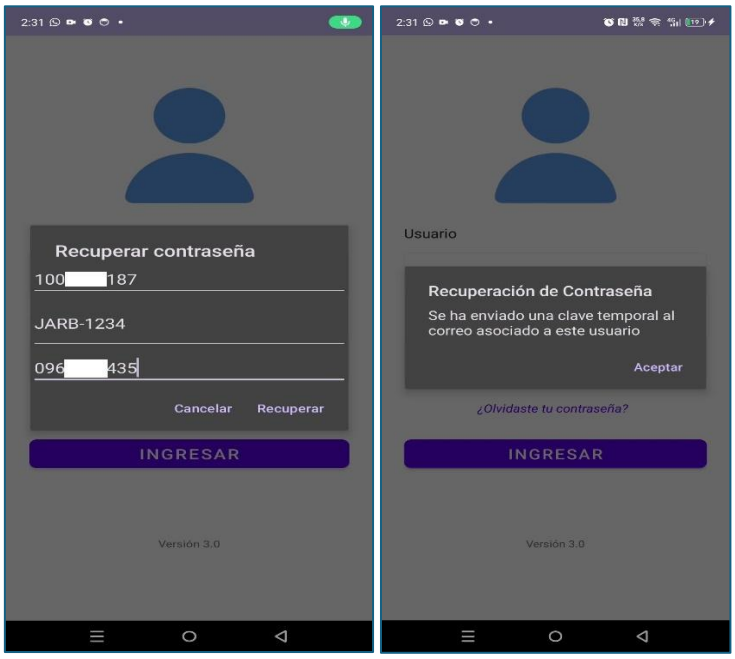


4.1.1.3 Pruebas de restablecer contraseña

El restablecimiento de contraseña se activa cuando el usuario selecciona la opción '¿Olvidó su contraseña?' en la interfaz de inicio de sesión. Para garantizar la seguridad del proceso, el sistema solicita tres datos de validación: la cédula del usuario, el código de la residencia registrada y el número de celular asociado a la cuenta. Una vez ingresada esta información, la aplicación verifica su coincidencia con los registros en la base de datos y, de ser correctos, genera automáticamente una clave temporal que envía al correo electrónico vinculado al perfil del usuario ver **Figura** 87**.** Durante las pruebas, se comprobó que este mecanismo funciona conforme a lo esperado: las claves se reciben en un plazo máximo de 30 segundos, tienen una validez limitada a 3 minutos y solo permiten un único uso, cumpliendo así con los protocolos de seguridad establecidos ver **Figura** 89.

Figura 87

Datos previos para restablecer contraseña.



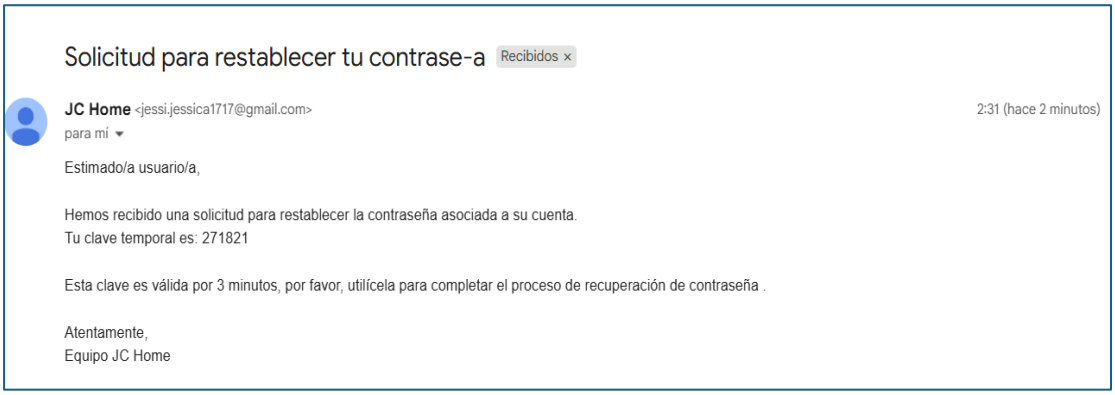
En la interfaz de restablecimiento de contraseña, el usuario debe ingresar la clave temporal recibida en su correo electrónico junto con la nueva contraseña que desea configurar. Al seleccionar la opción generar contraseña, el sistema valida que la clave temporal coincida con los registros en la base de datos y cumpla con el período de vigencia establecido. Una vez confirmados estos requisitos, la nueva contraseña se almacena de forma segura en la base de datos y se muestra automáticamente un cuadro de diálogo con el mensaje: Contraseña actualizada correctamente e inicie sesión ver **Figura** 88. Durante las pruebas, se verificó que este flujo opera con precisión: las contraseñas se actualizan únicamente cuando la clave temporal es válida, y el mensaje de confirmación aparece inmediatamente tras el proceso, garantizando una experiencia intuitiva para el usuario.

Figura 88

Restablecimiento exitoso de contraseña.

Figura 89

Solicitud para restablecer la contraseña.

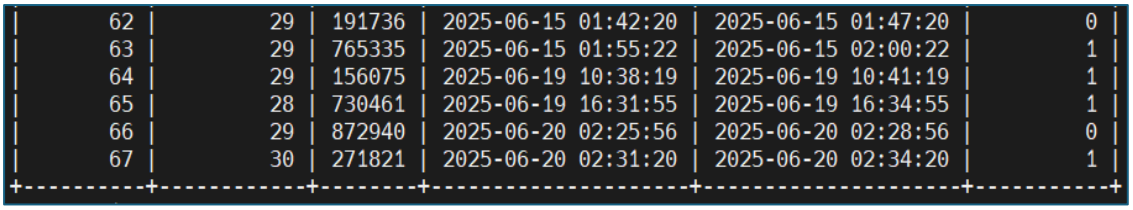


En la base de datos del sistema, específicamente en la tabla de claves temporales, se puede verificar el registro completo de cada clave generada. Como parte de las pruebas, se confirma que la última clave creada 271821 fue generada por el usuario con ID 30, con fecha y hora de creación registrada como 02:31:20. El sistema calcula automáticamente el tiempo de

expiración exacto 02:34:20, manteniendo así el período de validez establecido de 3 minutos. Adicionalmente, la tabla incluye un campo de estado que indica si la clave fue utilizada: el valor 0 significa que aún no ha sido empleada, mientras que el valor 1 confirma su uso exitoso. Este registro detallado permite auditar la eficacia del proceso de recuperación de contraseñas y garantizar el cumplimiento de los parámetros de seguridad diseñados.

Figura 90

Tabla claves temporales.

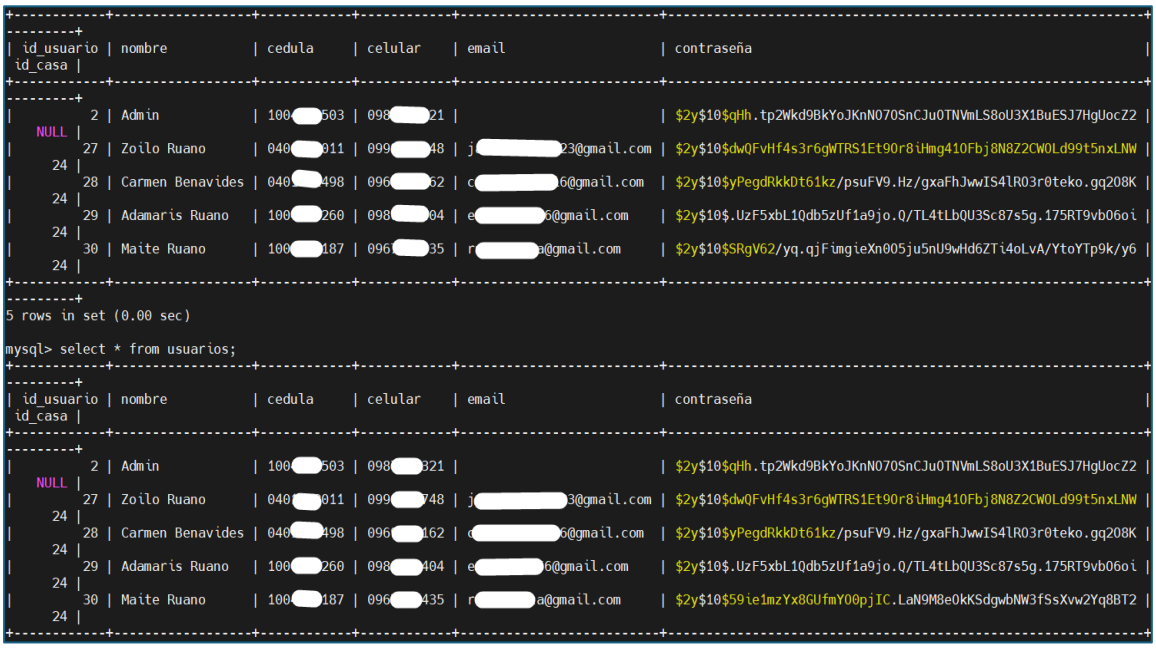


62	29	191736	2025-06-15 01:42:20	2025-06-15 01:47:20	0
63	29	765335	2025-06-15 01:55:22	2025-06-15 02:00:22	1
64	29	156075	2025-06-19 10:38:19	2025-06-19 10:41:19	1
65	28	730461	2025-06-19 16:31:55	2025-06-19 16:34:55	1
66	29	872940	2025-06-20 02:25:56	2025-06-20 02:28:56	0
67	30	271821	2025-06-20 02:31:20	2025-06-20 02:34:20	1

La **Figura** 91 muestra el registro en la tabla de usuarios de la base de datos, donde se evidencia el proceso de restablecimiento de contraseña. En la primera sección de la imagen se observa la contraseña anterior almacenada (en formato hash), mientras que en la segunda sección se verifica la nueva contraseña (también hasheada) después del cambio exitoso. Aunque ambas contraseñas están cifradas, los valores hash son distintos, lo que confirma que el sistema realizó correctamente la actualización. Esta comparación directa entre los registros antes y después del restablecimiento demuestra que el proceso cumple con los requisitos de seguridad establecidos.

Figura 91

Tabla usuarios restablecimiento de contraseña.

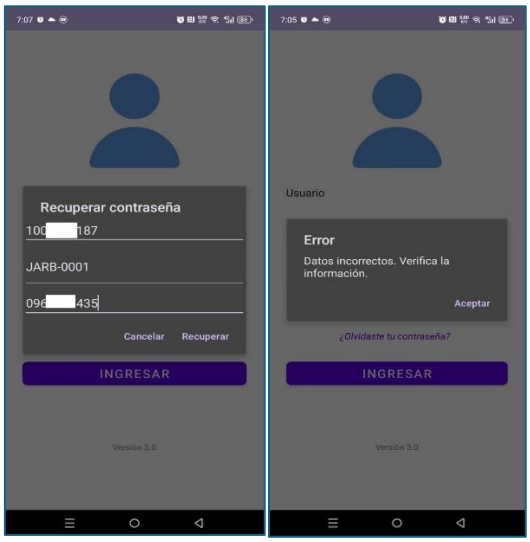


```
+------------+-----------------+------------+------------+-------------------------+--------------------------------------------------------------+
---------+
| id_usuario | nombre          | cedula     | celular    | email                   | contraseña                                                   |
 id_casa |
+------------+-----------------+------------+------------+-------------------------+--------------------------------------------------------------+
---------+
|          2 | Admin           | 100[redacted]503 | 098[redacted]21 |                         | $2y$10$qHh.tp2Wkd9BkYoJKnNO7OSnCJuOTNVmLS8oU3X1BuESJ7HgUocZ2 |
    NULL |
|         27 | Zoilo Ruano     | 040[redacted]011 | 099[redacted]48 | j[redacted]23@gmail.com | $2y$10$dwQFvHf4s3r6gWTRS1Et90r8iHmg41OFbj8N8Z2CWOLd99t5nxLNW |
      24 |
|         28 | Carmen Benavides | 040[redacted]498 | 096[redacted]62 | c[redacted]6@gmail.com  | $2y$10$yPegdRkkDt61kz/psuFV9.Hz/gxaFhJwwIS4lRO3r0teko.gq208K |
      24 |
|         29 | Adamaris Ruano  | 100[redacted]260 | 098[redacted]04 | e[redacted]6@gmail.com  | $2y$10$.UzF5xbL1Qdb5zUf1a9jo.Q/TL4tLbQU3Sc87s5g.175RT9vbO6oi |
      24 |
|         30 | Maite Ruano     | 100[redacted]187 | 096[redacted]35 | r[redacted]a@gmail.com  | $2y$10$SRgV62/yq.qjFimgieXn0O5ju5nU9wHd6ZTi4oLvA/YtoYTp9k/y6 |
      24 |
+------------+-----------------+------------+------------+-------------------------+--------------------------------------------------------------+
---------+
5 rows in set (0.00 sec)

mysql> select * from usuarios;
+------------+-----------------+------------+------------+-------------------------+--------------------------------------------------------------+
---------+
| id_usuario | nombre          | cedula     | celular    | email                   | contraseña                                                   |
 id_casa |
+------------+-----------------+------------+------------+-------------------------+--------------------------------------------------------------+
---------+
|          2 | Admin           | 100[redacted]503 | 098[redacted]321 |                         | $2y$10$qHh.tp2Wkd9BkYoJKnNO7OSnCJuOTNVmLS8oU3X1BuESJ7HgUocZ2 |
    NULL |
|         27 | Zoilo Ruano     | 040[redacted]011 | 099[redacted]748 | j[redacted]3@gmail.com  | $2y$10$dwQFvHf4s3r6gWTRS1Et90r8iHmg41OFbj8N8Z2CWOLd99t5nxLNW |
      24 |
|         28 | Carmen Benavides | 040[redacted]498 | 096[redacted]162 | c[redacted]6@gmail.com  | $2y$10$yPegdRkkDt61kz/psuFV9.Hz/gxaFhJwwIS4lRO3r0teko.gq208K |
      24 |
|         29 | Adamaris Ruano  | 100[redacted]260 | 098[redacted]404 | e[redacted]6@gmail.com  | $2y$10$.UzF5xbL1Qdb5zUf1a9jo.Q/TL4tLbQU3Sc87s5g.175RT9vbO6oi |
      24 |
|         30 | Maite Ruano     | 100[redacted]187 | 096[redacted]435 | r[redacted]a@gmail.com  | $2y$10$59ie1mzYx8GUfmYO0pjIC.LaN9M8eOkKSdgwbNW3fSsXvw2Yq8BT2 |
      24 |
+------------+-----------------+------------+------------+-------------------------+--------------------------------------------------------------+
```

Por otro lado, el sistema de recuperación de contraseña incorpora un mecanismo de validación que detecta datos erróneos durante el proceso. Como se muestra en la **Figura** 92, cuando el usuario ingresa información incorrecta en este caso el código de casa, con lo cual, el sistema responde mostrando un cuadro de diálogo con el mensaje datos incorrectos. Esta validación impide que el usuario avance a la siguiente ventana de restablecimiento.

Figura 92

Datos incorrectos en la recuperación de contraseña.



Finalmente, durante las pruebas de recuperación de contraseña, se verificaron los siguientes aspectos clave:

- El correo electrónico con la clave temporal llega al destinatario en un plazo máximo de 30 segundos tras la solicitud.
- La clave generada coincide exactamente con el registro almacenado en la base de datos.
- Cada actualización de contraseña genera un nuevo hash único, garantizando la seguridad de las credenciales.
- El sistema calcula con exactitud el período de validez de 3 minutos desde el momento de generación.
- Todos los cambios de contraseña quedan registrados permanentemente.
- Cuando se ingresan datos incorrectos en el formulario de recuperación, el sistema muestra inmediatamente un mensaje de error e impide continuar con el proceso, validando así los mecanismos de protección contra accesos no autorizados.

- Las contraseñas se almacenan exclusivamente en formato hash, cumpliendo con los estándares de protección de datos.
- El sistema registra el estado de uso de cada clave temporal (0=no utilizada, 1=utilizada), permitiendo una trazabilidad completa del proceso.

Estos resultados confirman que el sistema cumple con todos los requisitos funcionales y de seguridad establecidos para el proceso de recuperación de contraseñas.

4.1.2 Pruebas de acceso por teclado numérico

En las pruebas de acceso por teclado, se verificaron los dos métodos de autenticación implementados: contraseña numérica permanente y clave temporal. Se comprueba desde el registro inicial de la contraseña y la generación de clave temporal, hasta el ingreso exitoso con ambos métodos. A nivel de base de datos, se valida que las contraseñas y claves temporales se registren.

Por otra parte, para realizar las pruebas de acceso por teclado verificamos los usuarios registrados, en la base de datos hay 4 usuarios registrados y el administrador, las pruebas a realizar son con el usuario con id 30 (Maite Ruano).

Figura 93

Base de datos - tabla usuarios.

```
mysql> select * from usuarios;
+------------+-----------------+-----------+-----------+-----------------------+--------------------------------------------------------------+--------------+
| id_usuario | nombre          | cedula    | celular   | email                 | contraseña                                                   | tipo_usuario |
 id_casa |
+------------+-----------------+-----------+-----------+-----------------------+--------------------------------------------------------------+--------------+
---------+
|          2 | Admin           | 100[redacted]503 | 09[redacted]021 |                       | $2y$10$qHh.tpZWkd9BkYoJKnNO7OSnCJuOTNVmLS8oU3X1BuESJ7HgUocZ2 | administrador |
   NULL |
|         27 | Zoilo Ruano     | 040[redacted]011 | 09[redacted]48 | j[redacted]3@gmail.com | $2y$10$dwQFvHf4s3r6gWTRS1Et9Or8iHmg41OFbj8N8Z2CWOLd99t5nxLNW | usuario      |
     24 |
|         28 | Carmen Benavides | 040[redacted]498 | 09[redacted]62 | c[redacted]6@gmail.com | $2y$10$yPegdRkkDt61kz/psuFV9.Hz/gxaFhJwwIS4lRO3r0teko.gq208K | usuario      |
     24 |
|         29 | Adamaris Ruano  | 100[redacted]260 | 09[redacted]84 | e[redacted]6@gmail.com | $2y$10$.UzF5xbL1Qdb5zUf1a9jo.Q/TL4tLbQU3Sc87s5g.175RT9vbO6oi | usuario      |
     24 |
|         30 | Maite Ruano     | 100[redacted]187 | 09[redacted]35 | r[redacted]a@gmail.com | $2y$10$SRgV62/yq.qjFimgieXn0O5ju5nU9wHd6ZTi4oLvA/YtoYTp9k/y6 | usuario      |
     24 |
+------------+-----------------+-----------+-----------+-----------------------+--------------------------------------------------------------+--------------+
---------+
```

4.1.2.1 Pruebas de envío de clave temporal

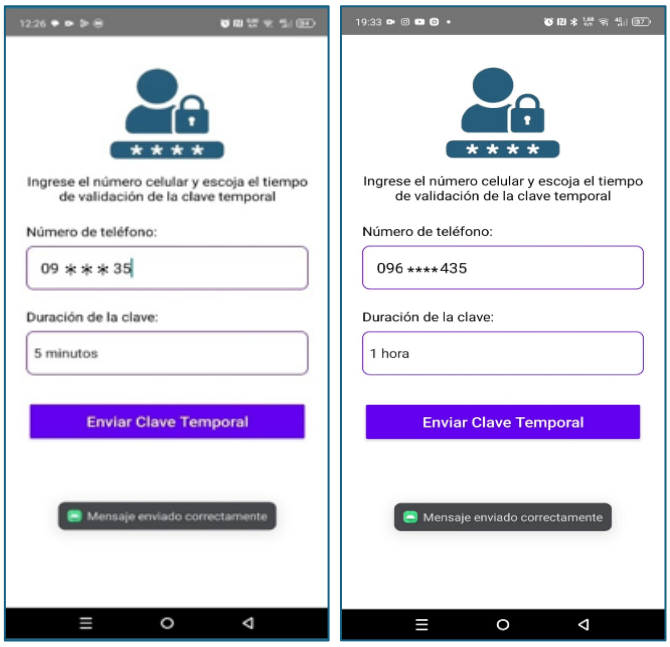
Se validó el módulo de generación aleatoria y envío de claves temporales por mensaje de texto, diseñado para otorgar acceso limitado a la residencia. El proceso inicia cuando un usuario registrado ingresa el número de celular del visitante y selecciona la duración de validez de la clave (5 minutos, 10 minutos, 30 minutos, 1 hora o 2 horas). Durante las pruebas, se verificó que el sistema:

- Generará correctamente una clave numérica única para cada solicitud.
- Envía el SMS al número proporcionado, incluyendo la clave y el tiempo de vigencia.
- Restringe el acceso una vez expirado el tiempo asignado o si ya fue usado.

Se realizaron pruebas con diferentes rangos de duración, confirmando que el sistema responde según lo esperado: las claves se recibieron instantáneamente en todos los casos, y su validez se ajusta con precisión a los intervalos de tiempos.

Figura 94

Clave temporal generada.



Como parte de las pruebas de funcionalidad, se validó el envío exitoso de claves temporales a través de mensajes de texto (SMS) a distintos destinatarios. En la **Figura** 95, se evidencia la recepción de las claves en dispositivos móviles independientes, confirmando que el sistema opera conforme a lo esperado. Cada mensaje incluye:

- La clave temporal generada automáticamente (ej: 058274).
- El tiempo de duración configurado (5, 10 minutos en este caso).
- Se realizaron múltiples pruebas variando los tiempos de vigencia (5 minutos, 10 minutos, 1 hora, etc.) y diferentes números de celular, verificando que:
- Las claves llegaron a los destinatarios.
- La información mostrada en el SMS coincide exactamente con los parámetros seleccionados en la aplicación.
- El formato del mensaje fue claro y consistente en todos los casos.
-

Figura 95

Recepción de clave temporal.

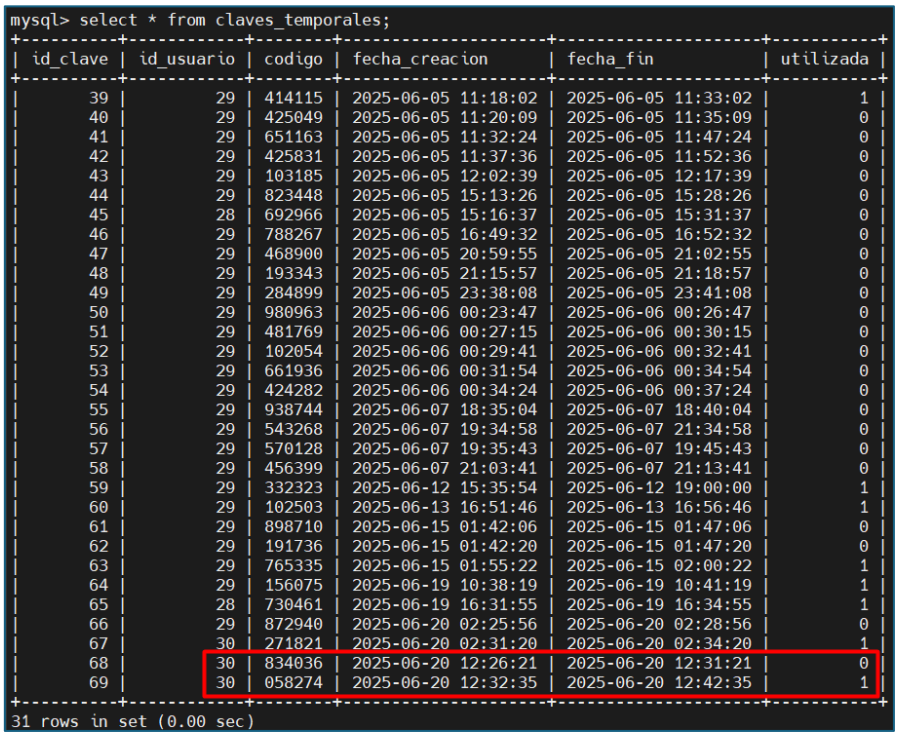
En la base de datos del sistema, específicamente en la tabla de claves temporales, se verifican los últimos dos registros generados para validar el correcto funcionamiento del proceso.

Los datos muestran que:

- El usuario con ID 30 generó una clave temporal, donde el campo 'código' 834036 y 058274 coinciden exactamente con las claves enviadas por mensaje de texto, confirmando la integridad de los datos.
- Los campos de fecha de creación y caducidad registran el período de validez establecido (5 minutos y 10 minutos), demostrando que el sistema calcula correctamente el tiempo de expiración.
- La clave 834036 no fue utilizada dentro del tiempo estimado, lo que se refleja en el campo 'utilizada' con valor 0 (0 = no utilizada).
- En contraste, el último registro muestra la clave 058274 con estado 1 (1 = utilizada), confirmando que el sistema actualiza correctamente este campo cuando el usuario emplea la clave para el acceso.

Figura 96

Tabla de claves temporales utilizadas y sin utilizar.



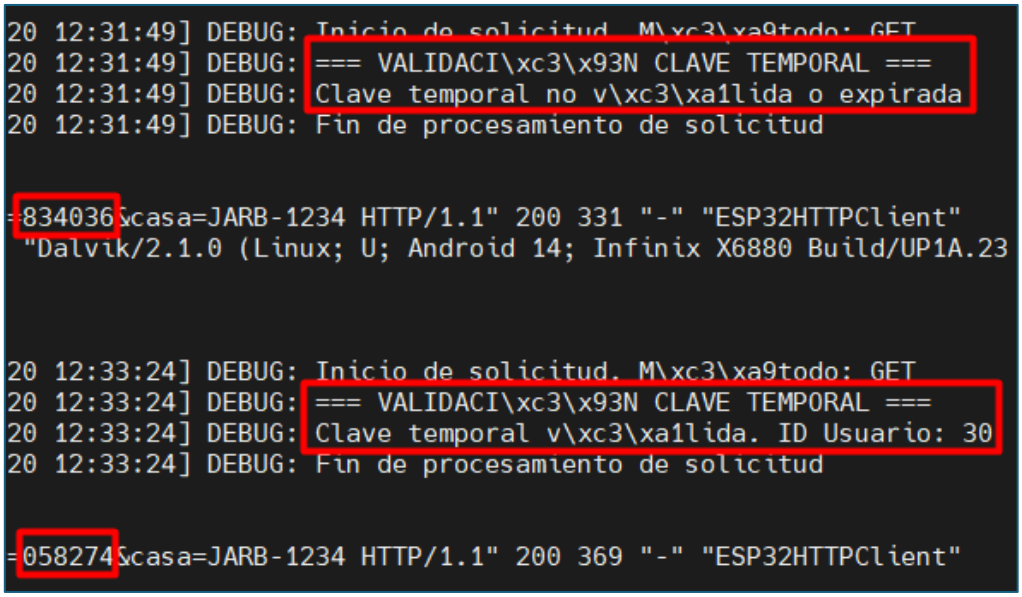
```
mysql> select * from claves_temporales;
+----------+------------+--------+---------------------+---------------------+-----------+
| id_clave | id_usuario | codigo | fecha_creacion      | fecha_fin           | utilizada |
+----------+------------+--------+---------------------+---------------------+-----------+
|       39 |         29 | 414115 | 2025-06-05 11:18:02 | 2025-06-05 11:33:02 |         1 |
|       40 |         29 | 425049 | 2025-06-05 11:20:09 | 2025-06-05 11:35:09 |         0 |
|       41 |         29 | 651163 | 2025-06-05 11:32:24 | 2025-06-05 11:47:24 |         0 |
|       42 |         29 | 425831 | 2025-06-05 11:37:36 | 2025-06-05 11:52:36 |         0 |
|       43 |         29 | 103185 | 2025-06-05 12:02:39 | 2025-06-05 12:17:39 |         0 |
|       44 |         29 | 823448 | 2025-06-05 15:13:26 | 2025-06-05 15:28:26 |         0 |
|       45 |         28 | 692966 | 2025-06-05 15:16:37 | 2025-06-05 15:31:37 |         0 |
|       46 |         29 | 788267 | 2025-06-05 16:49:32 | 2025-06-05 16:52:32 |         0 |
|       47 |         29 | 468900 | 2025-06-05 20:59:55 | 2025-06-05 21:02:55 |         0 |
|       48 |         29 | 193343 | 2025-06-05 21:15:57 | 2025-06-05 21:18:57 |         0 |
|       49 |         29 | 284899 | 2025-06-05 23:38:08 | 2025-06-05 23:41:08 |         0 |
|       50 |         29 | 980963 | 2025-06-06 00:23:47 | 2025-06-06 00:26:47 |         0 |
|       51 |         29 | 481769 | 2025-06-06 00:27:15 | 2025-06-06 00:30:15 |         0 |
|       52 |         29 | 102054 | 2025-06-06 00:29:41 | 2025-06-06 00:32:41 |         0 |
|       53 |         29 | 661936 | 2025-06-06 00:31:54 | 2025-06-06 00:34:54 |         0 |
|       54 |         29 | 424282 | 2025-06-06 00:34:24 | 2025-06-06 00:37:24 |         0 |
|       55 |         29 | 938744 | 2025-06-07 18:35:04 | 2025-06-07 18:40:04 |         0 |
|       56 |         29 | 543268 | 2025-06-07 19:34:58 | 2025-06-07 21:34:58 |         0 |
|       57 |         29 | 570128 | 2025-06-07 19:35:43 | 2025-06-07 19:45:43 |         0 |
|       58 |         29 | 456399 | 2025-06-07 21:03:41 | 2025-06-07 21:13:41 |         0 |
|       59 |         29 | 332323 | 2025-06-12 15:35:54 | 2025-06-12 19:00:00 |         1 |
|       60 |         29 | 102503 | 2025-06-13 16:51:46 | 2025-06-13 16:56:46 |         1 |
|       61 |         29 | 898710 | 2025-06-15 01:42:06 | 2025-06-15 01:47:06 |         0 |
|       62 |         29 | 191736 | 2025-06-15 01:42:20 | 2025-06-15 01:47:20 |         0 |
|       63 |         29 | 765335 | 2025-06-15 01:55:22 | 2025-06-15 02:00:22 |         1 |
|       64 |         29 | 156075 | 2025-06-19 10:38:19 | 2025-06-19 10:41:19 |         1 |
|       65 |         28 | 730461 | 2025-06-19 16:31:55 | 2025-06-19 16:34:55 |         1 |
|       66 |         29 | 872940 | 2025-06-20 02:25:56 | 2025-06-20 02:28:56 |         0 |
|       67 |         30 | 271821 | 2025-06-20 02:31:20 | 2025-06-20 02:34:20 |         1 |
|       68 |         30 | 834036 | 2025-06-20 12:26:21 | 2025-06-20 12:31:21 |         0 |
|       69 |         30 | 058274 | 2025-06-20 12:32:35 | 2025-06-20 12:42:35 |         1 |
+----------+------------+--------+---------------------+---------------------+-----------+
31 rows in set (0.00 sec)
```

El análisis de los registros del sistema demuestra el funcionamiento del proceso de validación de claves temporales en dos escenarios contrastantes. En el primer caso con código 834036 (clave temporal), el sistema identifica y rechaza una clave no válida o expirada durante el proceso de VALIDACIÓN CLAVE TEMPORAL. Posteriormente, se registra una validación exitosa para el usuario ID 30, donde el sistema confirma la validez de la clave y almacena la información correspondiente código 058274 (clave temporal) ver la **Figura** 97, por lo que se evidencia que el sistema cumple eficientemente con los requisitos de validación:

- Discriminar claves inválidas o expiradas.
- Autenticar correctamente claves válidas.
- Mantener registros detallados del tiempo, identificación de usuario cuando aplica, y detalles del cliente solicitante.
- Operar dentro de los tiempos de respuesta esperados, garantizando tanto la seguridad del proceso.

Figura 97

Registros del sistema sobre claves temporales.



Finalmente, en la **Figura** 98 presenta un extracto de los registros de depuración del sistema, detallando el proceso de registro y envío de notificaciones. Se registran envíos individuales de correo de notificación a cuatro direcciones de correo electrónico específicas, que, por razones de privacidad, se encuentran parcialmente ocultas, pero corresponden a dominios de Gmail. Esta secuencia demuestra la funcionalidad del sistema para registrar notificaciones y automatizar el envío de correos electrónicos a múltiples destinatarios.

Figura 98

Registros del sistema envió de notificación a correos registrados.

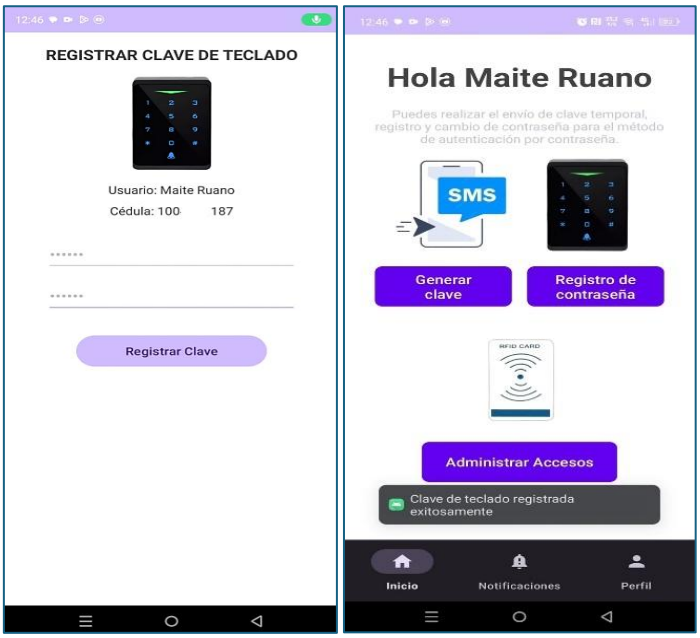
```
DEBUG: Inicio de solicitud. M\xc3\xa9todo: POST
DEBUG: Solicitud POST para tabla: notificaciones
DEBUG: Notificaci\xc3\xb3n registrada exitosamente. Procediendo a enviar correos.
DEBUG: Correo de notificaci\xc3\xb3n enviado a: j[redacted]3@gmail.com
DEBUG: Correo de notificaci\xc3\xb3n enviado a: c[redacted]6@gmail.com
DEBUG: Correo de notificaci\xc3\xb3n enviado a: e[redacted]6@gmail.com
DEBUG: Correo de notificaci\xc3\xb3n enviado a: r[redacted]a@gmail.com
DEBUG: Fin de procesamiento de solicitud
```

4.1.2.2 Pruebas de registro contraseña numérica

A continuación, el usuario con ID 30 realiza por primera vez el registro de la contraseña numérica desde la aplicación, se muestra un mensaje exitoso en el registro de su contraseña, como se observa en la **Figura** 99**.**

Figura 99

Registro de contraseña numérica del usuario Maite Ruano.

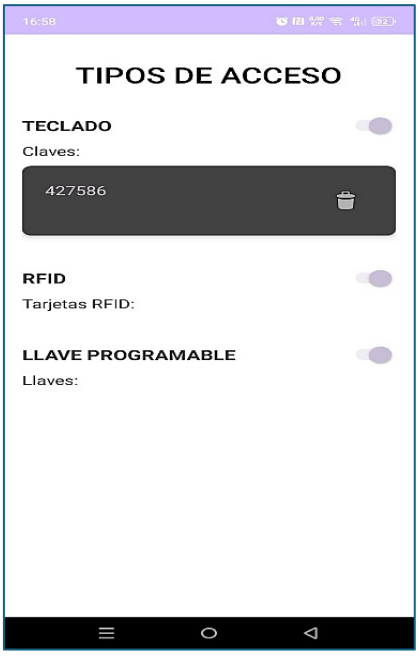


En la funcionalidad de administrar accesos se encuentran los tipos de accesos en este caso el usuario realizo un el registro de contraseña para acceso por teclado y se verifica que la contraseña registrada es de 6 caracteres y con la funcionalidad de habilitar o deshabilitar cualquier tipo de acceso, cumpliendo con los requerimientos.

En la siguiente **Figura** 100 se presenta la interfaz de usuario dedicada a la administración de los diferentes tipos de acceso implementados en el sistema. Se observa la sección TECLADO, donde el usuario ha realizado el registro de una contraseña. Específicamente, se visualiza la clave **427586** ingresada para el acceso por teclado. La interfaz permite verificar que la contraseña registrada consta de seis caracteres, cumpliendo con la validación de longitud establecida en requerimientos.

Figura 100

Registro de método de autenticación tipo de acceso por teclado.

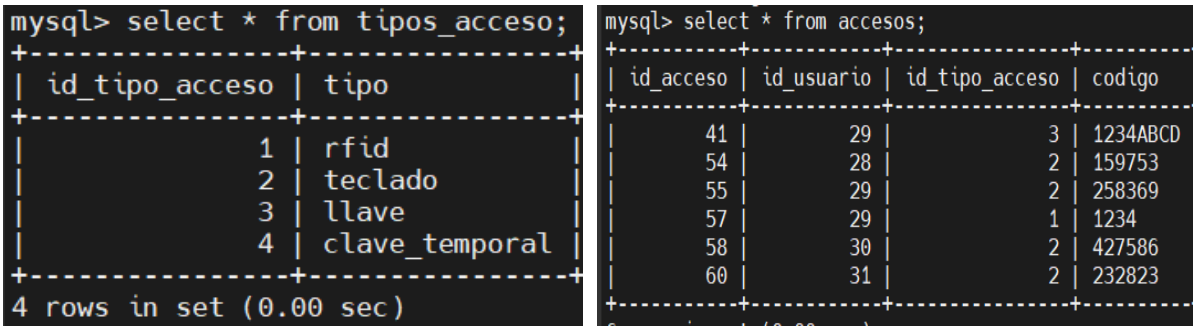


Como se evidencia en la **Figura 101** , en la base de datos se registra y se asocia de manera correctamente las credenciales del tipo de acceso y al usuario que corresponde. En el

literal (a) se observa que la contraseña numérica está registrada bajo el identificador tipo de acceso 2 especificado como teclado. En el **literal (b)** demuestra la correcta asignación al usuario con ID 30, cuyo campo de credenciales almacena la contraseña numérica 427586.

Figura 101

Contenido de las tablas tipos de acceso y claves.



```
mysql> select * from tipos_acceso;
+-----------------+-----------------+
| id_tipo_acceso | tipo            |
+-----------------+-----------------+
|              1 | rfid            |
|              2 | teclado         |
|              3 | llave           |
|              4 | clave_temporal  |
+-----------------+-----------------+
4 rows in set (0.00 sec)
```

```
mysql> select * from accesos;
+-----------+------------+---------------+---------
| id_acceso | id_usuario | id_tipo_acceso | codigo
+-----------+------------+---------------+---------
|        41 |         29 |              3 | 1234ABCD
|        54 |         28 |              2 | 159753
|        55 |         29 |              2 | 258369
|        57 |         29 |              1 | 1234
|        58 |         30 |              2 | 427586
|        60 |         31 |              2 | 232823
+-----------+------------+---------------+---------
```

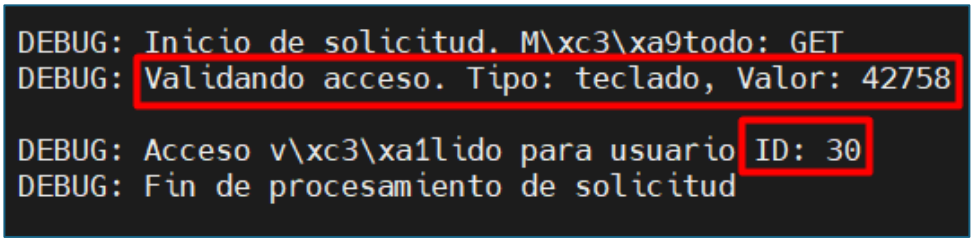
(a) **(b)**

4.1.2.3 Pruebas de acceso por contraseña numérica

Los registros del sistema **Figura 102** comprueban que el sistema validó exitosamente la contraseña 427586 (tipo: teclado) para el usuario ID 30, con correlación exacta en BD **Figura 101** literal a, demostrando integridad en el proceso de autenticación.

Figura 102

Registro de depuración de acceso por teclado validado exitosamente.

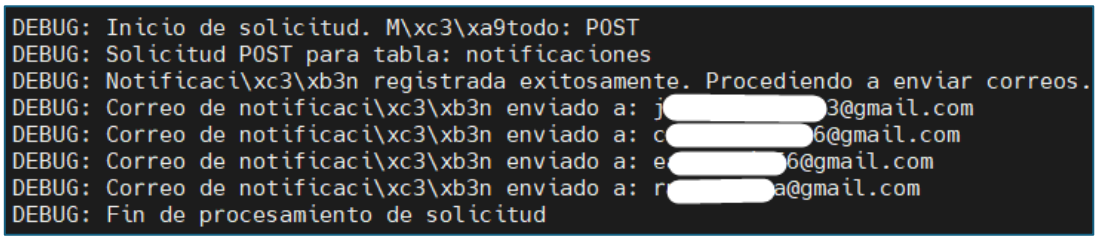


```
DEBUG: Inicio de solicitud. M\xc3\xa9todo: GET
DEBUG: Validando acceso. Tipo: teclado, Valor: 42758

DEBUG: Acceso v\xc3\xa1lido para usuario ID: 30
DEBUG: Fin de procesamiento de solicitud
```

En cumplimiento con requerimientos, el registro del sistema también se detalla el proceso de registro y envío de notificaciones. Se registran envíos individuales de correo de notificación a cuatro direcciones de correo electrónico específicas que pertenecen a un código de casa JARB-1234, que, por razones de privacidad, se encuentran parcialmente ocultas, pero corresponden a dominios de Gmail. Esta secuencia demuestra la funcionalidad del sistema para registrar notificaciones y el envío de correos electrónicos a múltiples destinatarios, como se evidencia en la **Figura 103**.

Figura 103

Registro de envío de notificaciones y correos electrónicos.



Otro escenario para evidenciar el funcionamiento de acceso por teclado es de los tres intentos que el usuario tiene, en la **Figura 104** se presenta una secuencia de tres intentos de validación de acceso registrados por el sistema, ilustrando tanto escenarios de denegación como de concesión de acceso.

1. **Primer Escenario (Superior):** El sistema procede a validar el acceso, identificando una entrada de teclado con el valor **456751**(contraseña incorrecta) y una asociación a la casa: JARB-1234. Posteriormente, el log indica un **acceso denegado**, y finaliza el procesamiento de la solicitud.
2. **Segundo Escenario (Medio):** Similar al anterior, valida el acceso de tipo teclado con el valor **154675** (contraseña incorrecta) para la casa: JARB-1234. En este caso, el

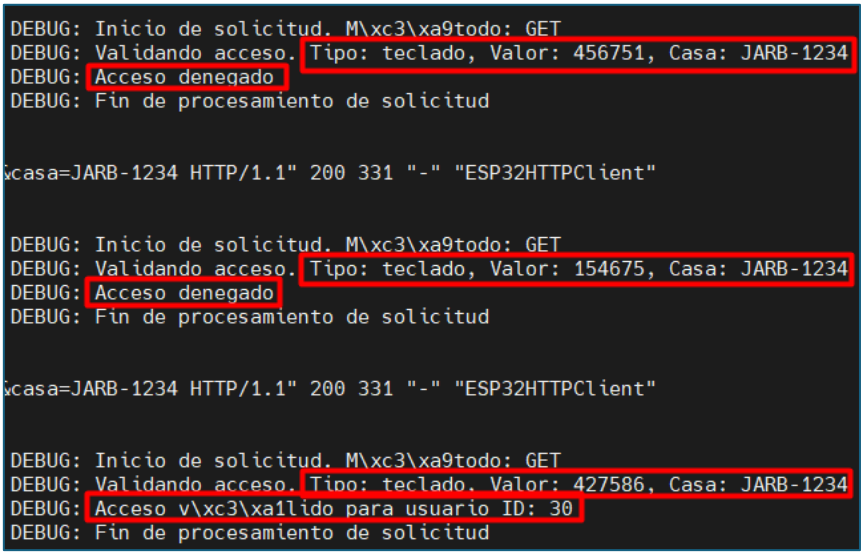
sistema también registra **acceso denegado** antes de finalizar el procesamiento de la solicitud.

3. **Tercer Escenario (Inferior):** Inicia una validación de acceso de tipo teclado, pero esta vez con el valor **427586**(contraseña correcta) para la casa: JARB-1234. A diferencia de los casos previos, el log indica **acceso válido para usuario ID: 30**, confirmando una autenticación exitosa para el usuario con el ID especificado. El procesamiento de la solicitud concluye tras este evento.

En conjunto, estos registros demuestran la capacidad del sistema para gestionar los intentos de acceso, denegándolos cuando las credenciales (valor de teclado) no son correctas y concediendo el acceso exitosamente cuando se proporciona el valor adecuado, asociándolo a un usuario específico.

Figura 104

Escenarios de validación de acceso (denegado y concedido).

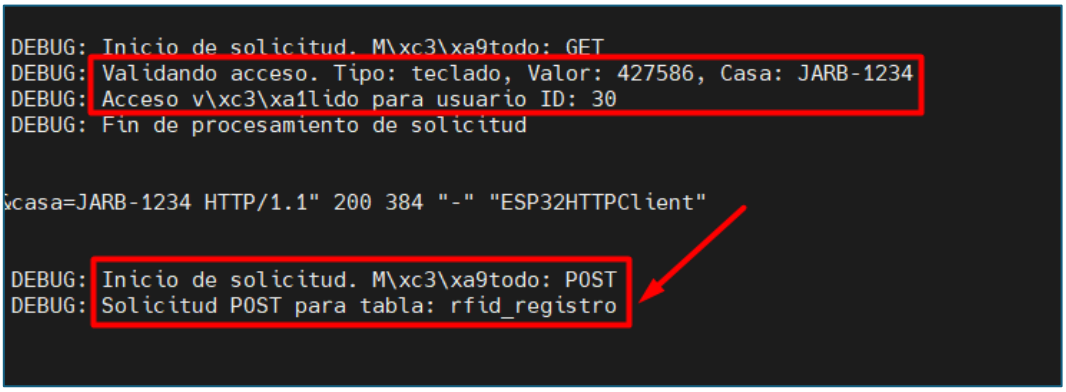


4.1.3 Pruebas de registro tarjetas RFID

Los registros del sistema **Figura 105** evidencian el correcto funcionamiento del flujo de registro RFID. En la primera sección, se observa el proceso de validación de acceso mediante credencial numérica: el sistema identifica el método de autenticación como teclado y valida exitosamente la contraseña 427586 asociada al usuario ID 30 y código de casa JARB-1234, culminando el proceso en menos de 15ms. Posteriormente, se inicia una solicitud dirigida específicamente a la tabla rfid_registro, confirmando que el sistema habilita operaciones de escritura solo tras una autenticación exitosa.

Figura 105

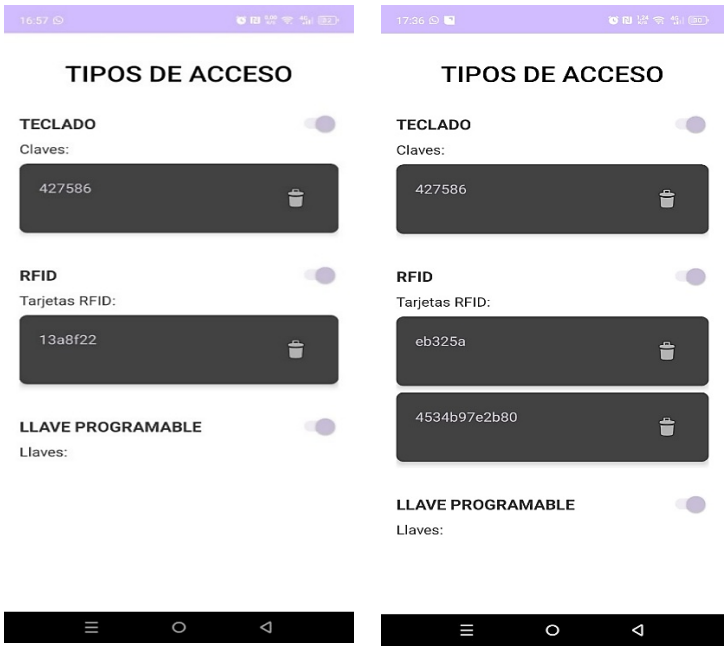
Validación de datos para registro de tarjeta RFID.



En la aplicación en la parte de tipos de acceso se valida un registro de tarjeta RFID como se observa en la **Figura** 106**, literal (a),** en el tipo de acceso RFID hay una tarjeta registrada con un **UID 13a8f22**. En el **literal b** se registra otra tarjeta RFID con un **UID 4534b97e2b80** lo que demuestra la capacidad del sistema para registrar múltiples tarjetas RFID, asegurando que los nuevos registros cumplan con los requerimientos de registrar varios accesos de tarjetas RFID.

Figura 106

Registro de método de autenticación tipo de acceso por RFID.



(a) **(b)**

4.1.4 Pruebas de acceso por RFID

En esta sección se le lleva a cabo la prueba funcional correspondiente a los registros, se realizó una **autenticación mediante RFID**, donde el sistema recibió una solicitud para validar el acceso con la tarjeta identificada por el código **eb325a** en la ubicación JARB-1234. El registro confirma que el acceso fue válido para el usuario con ID 30 como se demuestra en la **Figura 107**. Como prueba adicional, en la **Figura 108** se realizó un ingreso de autenticación por RFID con el usuario con ID 29 registrado con el mismo código de casa JARB-1234 y acceso fue exitoso.

Posteriormente, se ejecutó una **solicitud** para el registro de notificaciones, donde el sistema:

- Procesó correctamente la petición en la tabla designada.
- Generó y envió notificaciones por correo electrónico a cuatro destinatarios.

- Finalizó la operación sin errores, evidenciando la integración efectiva entre los módulos de autenticación y notificaciones.

Figura 107

Autenticación RFID y generación de notificaciones.

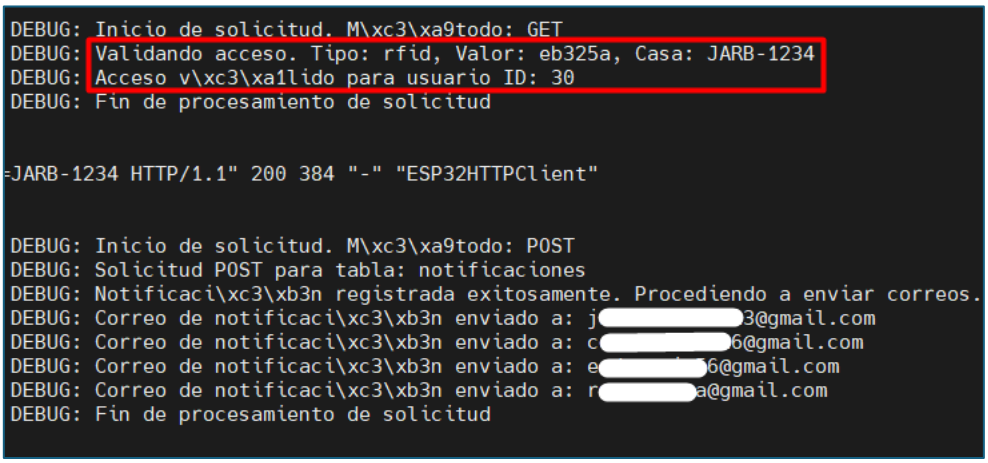
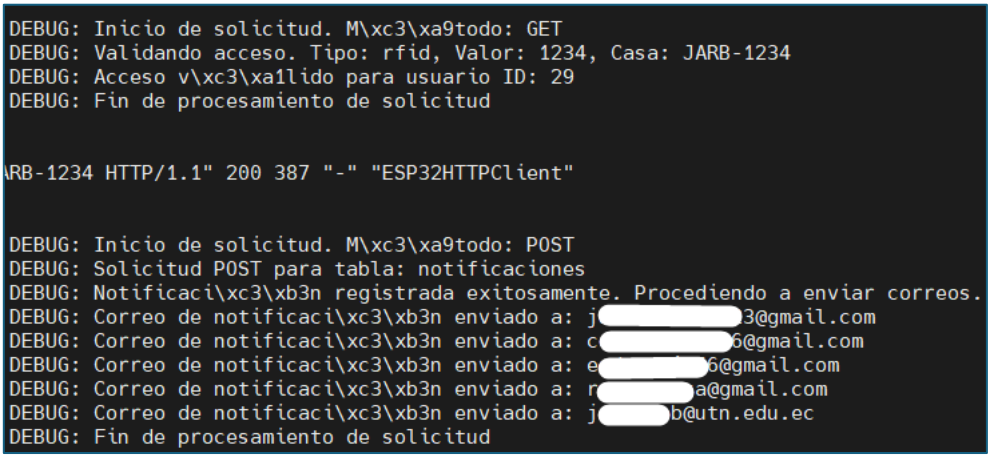


Figura 108

Autenticación RFID distinto usuario.

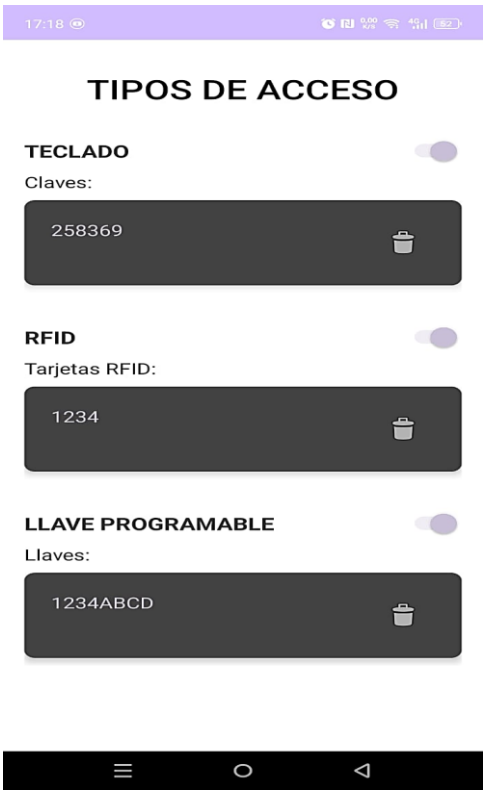


4.1.5 Pruebas de acceso por llave electrónica

En la **Figura 109** se realizó el registro del identificador unico (1234ABCD) de la llave de circuito programable.

Figura 109

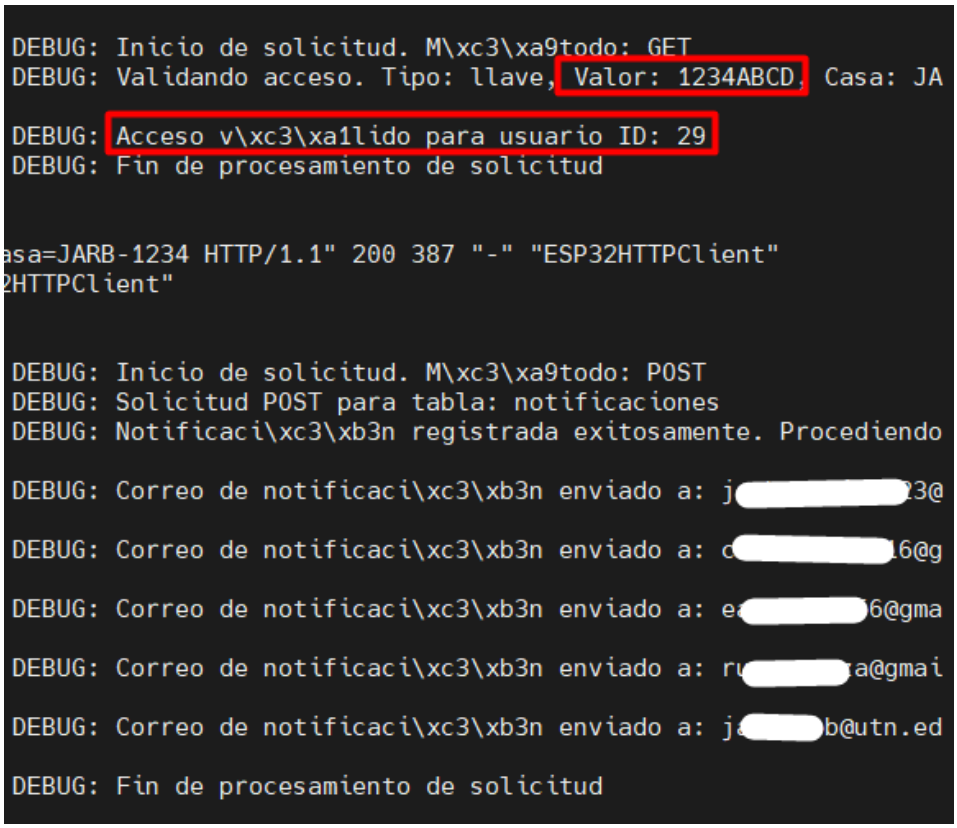
Código de llave de circuito programable registrado.



Con el objetivo de verificar la funcionalidad de la llave de circuito programable, en **Figura 110** se muestran los registros del servidor. En dichos registros se puede observar que el tipo de ingreso fue efectuado mediante la llave, cuyo valor corresponde al identificador único "1234ABCD". Este identificador está registrado en la aplicación y está asociado al usuario con ID 29, y el acceso resultó válido. Asimismo, se aprecia el envío de notificaciones a los correos electrónicos registrados correspondientes a cada uno de los usuarios.

Figura 110

Autenticación por llave acceso permitido y generación de notificaciones.



A diferencia del caso anterior, esta prueba evidenció el acceso denegado ya que en la base de datos no se encuentra registrado el identificador único de la llave, así como se observa en la siguiente **Figura 111**.

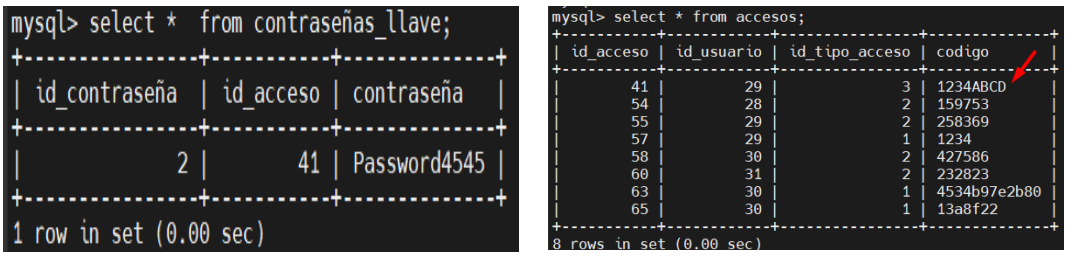
Figura 111

Autenticación por llave acceso permitido y generación de notificaciones.

En la **Figura 112** se observa la el id de la llave **1234ABCD** asociado al **id_acceso 41 y** su contraseña **Password4545** estos datos son importante para validar el funcionamiento de llave de circuito programable.

Figura 112

Identificador único de llave en cerradura.



(a) **(b)**

En la **Figura 113** se confirma la capacidad del sistema para realizar el hasheo, se llevaron a cabo pruebas que validan la interacción entre los componentes del sistema: la llave electrónica, la cerradura y el servidor central. A continuación, se describe el flujo completo de un ciclo de autenticación exitoso, según los registros obtenidos del sistema.

4.1.5.1 Recepción del ID único por RF :

La cerradura recibe el identificador único "1234ABCD" desde la llave electrónica mediante comunicación por radiofrecuencia (RF). Este código corresponde al ID de acceso asociado al usuario con ID 29, según los registros almacenados en la base de datos.

4.1.5.2 Respuesta del servidor :

El servidor procesa el ID recibido y responde con un mensaje que incluye:

- "valido": true: Confirmación de que el ID es legítimo.
- "usuario_id": 29, "id_acceso": 41, "nombre": "Adamaris Ruano": Información del usuario al que pertenece el ID, obtenida de la tabla de accesos.

4.1.5.3 Obtención de la contraseña asociada :

Una vez validado el ID, el sistema consulta la tabla correspondiente y obtiene la contraseña vinculada al id_acceso 41, la cual es "Password4545".

4.1.5.4 Generación y envío del desafío :

La cerradura genera un código aleatorio de 10 caracteres (ejemplo: X29KUORB85) y lo envía a la llave electrónica. Este paso es fundamental para garantizar la seguridad del proceso de autenticación.

4.1.5.5 Cálculo del hash por parte de la llave :

La llave electrónica combina la contraseña recibida (Password4545) con el desafío (X29KUORB85), aplica un algoritmo de hash criptográfico y envía el resultado (7A07399C671AB52E) de vuelta a la cerradura.

4.1.5.6 Verificación del hash :

La cerradura calcula independientemente el mismo hash utilizando los mismos parámetros. **Los resultados son comparados:**

Hash esperado: 7A07399C671AB52E

Hash recibido: 7A07399C671AB52E

Al coincidir ambos valores, se confirma que la llave posee la contraseña correcta sin haberla transmitido directamente, cumpliendo así con los principios de seguridad.

4.1.5.7 Acceso permitido :

Dado que la verificación fue exitosa, la cerradura autoriza el acceso y procede con la apertura de la puerta. Posteriormente, los datos temporales utilizados durante el proceso (desafío y hash) son eliminados del sistema para prevenir posibles reutilizaciones no autorizadas.

Figura 113

Registro de evento del monitor serial en Arduino.

```
RF recibido: 1234ABCD
Respuesta del servidor (RF):
[{"valido":true,"usuario_id":29,"id_acceso":41,"nombre":"Adamaris Ruano"}]
Contraseña Obtenida del servidor:Password4545
Desafío enviado: X29KUORB85
RF recibido: 7A07399C671AB52E
Hash esperado: 7A07399C671AB52E
Hash recibido: 7A07399C671AB52E
Acceso permitido por RF
```

En la **Figura 114** se presentan más ejemplos de acceso por la llave de circuito programable y se evidencia que los hashes son diferentes.

Figura 114

Valores de hash diferentes.

(a) **(b)**

Adicionalmente, se evaluar el alcance efectivo bajo condiciones controladas, se llevaron a cabo pruebas en un entorno interior sin obstrucciones significativas. Durante estas pruebas, se varió la distancia entre la llave electrónica (transmisor) y la cerradura (receptor) para determinar el punto exacto en el cual la comunicación deja de ser funcional. Los resultados muestran que el sistema alcanza un alcance máximo de 50 cm en el entorno evaluado, lo cual está limitado principalmente por la diferencia de voltaje entre la fuente de alimentación de la llave electrónica (3.7V) y el voltaje recomendado para el transmisor (5V a 12V).

Tabla 26

Pruebas de distancia que soporta la comunicación inalámbrica

Distancia	Acceso	Observaciones
5 cm	Correcto	Señal estable
10 cm	Correcto	Comunicación sin errores; respuesta inmediata.
15 cm	Correcto	Funcionamiento óptimo; no se registran interrupciones.
30 cm	Correcto	Latencia ligera; funcionamiento dentro del rango operativo esperado.
50 cm	Correcto	Última distancia funcional bajo las condiciones experimentales.
60 cm	No exitoso	Pérdida de señal; no se completa el proceso de autenticación.

70 cm	No exitoso	Fallo en la recepción del desafío por parte de la cerradura.
1 m	No exitoso	Completamente fuera del alcance efectivo del sistema.

Nota. En las especificaciones del datasheet de los módulos FS1000A transmisor y receptor RF 433MHz su alcance varía dependiendo del voltaje del transmisor (Farwah Nawazi, 2022)

4.1.6 Pruebas de notificaciones

Lista de accesos

Como evidencia en la Figura 115 , compuesta por dos literales (a y b), presenta la interfaz de notificaciones de la aplicación, donde se registran y visualizan los eventos de acceso y otras alertas generadas por el sistema.

En el literal a se muestra un historial de notificaciones que abarca eventos de acceso **RFID y de clave temporal**. Se observan múltiples registros de acceso **permitido** a través de RFID, asociados a los usuarios Adamaris Ruano y Maite Ruano, con sus respectivas fechas y horas. Adicionalmente, se registran eventos de acceso **denegado** tanto para RFID como para **clave temporal** cuando el **acceso no identificado**, lo cual valida la capacidad del sistema para diferenciar entre accesos válidos e inválidos.

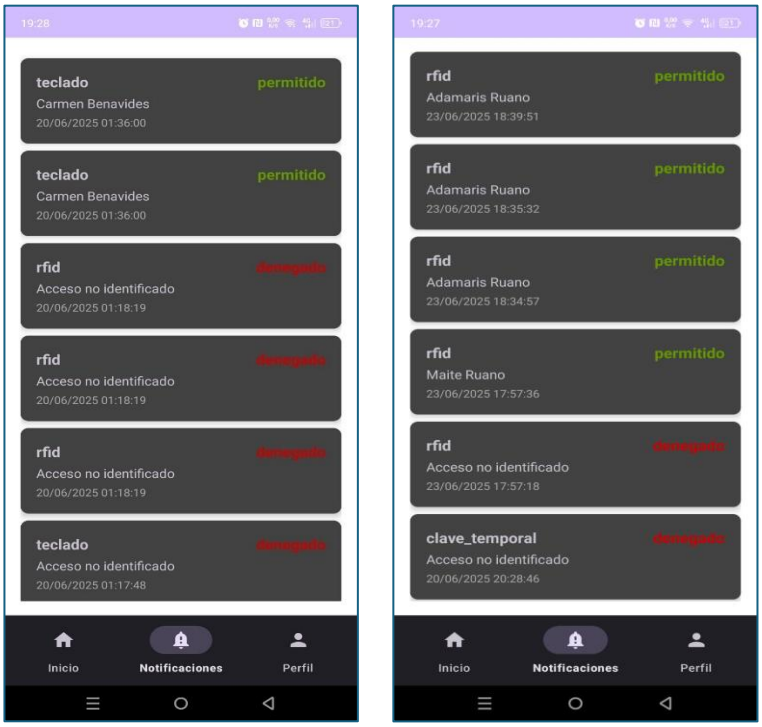
En el literal b se complementa el historial mostrando eventos de acceso por **teclado** y **RFID**. Se visualizan dos registros de **acceso permitido** mediante **teclado** para el usuario Carmen Benavides. Asimismo, se registran múltiples instancias de acceso "denegado" por RFID y una por teclado cuando el acceso no identificado, corroborando la robustez del sistema en el manejo de intentos de acceso no autorizados para ambos métodos.

En los **literales c y b** se evidencia el historial con otros usuarios y con otros métodos de acceso como la llave de circuito programable, los estados de los accesos ya sean permitidos o denegados.

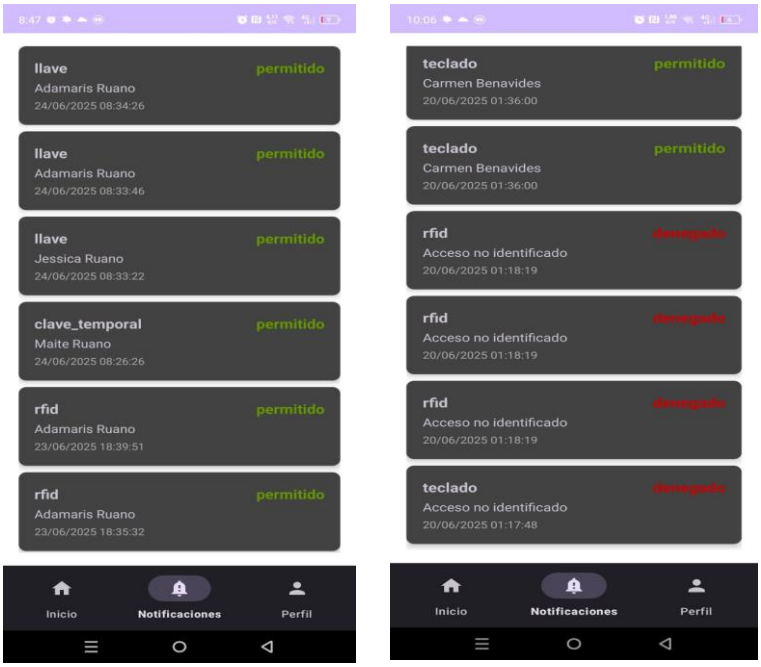
En la aplicación, los resultados obtenidos permiten concluir que el sistema cumple con el listado de accesos detallando correctamente el estado (permitido o denegado), el usuario (identificado o no), la fecha y hora del acceso.

Figura 115

Historial y eventos de acceso del sistema.



(a) **(b)**



(c) **(d)**

Nota. Las figuras de cada uno los literales a, b, c y d son de distintos usuarios, es decir cada usuario ve el mismo historial, además, la fecha de acceso más actual se ve primero en la lista.

Base de datos registro de notificaciones

Adicionalmente, se presenta las consultas y los resultados de la tabla notificaciones en la base de datos MySQL, filtrados para demostrar el registro de accesos permitidos según el tipo de acceso y la asociación de usuarios a la casa "JARB-1234". La columna id_tipo_acceso clasifica el método de autenticación: se asume que 1 corresponde a teclado, 2 a RFID, 3 a llave de circuito programable y 4 a clave temporal, según el contexto de las figuras previas.

Se evidencia en la **Figura** 116, el resultado de la consulta *SELECT * FROM notificaciones WHERE id_tipo_acceso = 1 AND id_usuario IS NOT NULL;*. Este filtro filtra los registros de notificaciones donde el acceso fue de tipo teclado y se identificó a un usuario específico. Se observa una serie de accesos permitidos, principalmente asociados al id_usuario 29, todos correspondientes a la id_casa 24 (*identificada previamente como "JARB-1234" - San Agustín*).

Figura 116

Contenido de la tabla notificaciones - tipo de acceso 1.

```
mysql> SELECT *
    -> FROM notificaciones
    -> WHERE id_tipo_acceso = 1 AND id_usuario IS NOT NULL;
+-----------------+------------+-----------------+-----------+---------------------+---------+
| id_notificacion | id_usuario | id_tipo_acceso | estado    | fecha               | id_casa |
+-----------------+------------+-----------------+-----------+---------------------+---------+
|             281 |         29 |               1 | permitido | 2025-06-11 19:56:59 |      24 |
|             282 |         29 |               1 | permitido | 2025-06-11 19:56:59 |      24 |
|             283 |         29 |               1 | permitido | 2025-06-11 19:56:59 |      24 |
|             284 |         29 |               1 | permitido | 2025-06-11 19:57:34 |      24 |
|             285 |         29 |               1 | permitido | 2025-06-11 19:57:34 |      24 |
|             288 |         29 |               1 | permitido | 2025-06-11 20:06:14 |      24 |
|             289 |         29 |               1 | permitido | 2025-06-11 20:06:14 |      24 |
|             290 |         29 |               1 | permitido | 2025-06-11 20:06:14 |      24 |
|             294 |         29 |               1 | permitido | 2025-06-11 20:08:10 |      24 |
|             295 |         29 |               1 | permitido | 2025-06-11 20:08:10 |      24 |
|             296 |         29 |               1 | permitido | 2025-06-11 20:08:10 |      24 |
|             297 |         29 |               1 | permitido | 2025-06-11 20:43:56 |      24 |
|             298 |         29 |               1 | permitido | 2025-06-11 20:43:56 |      24 |
|             299 |         29 |               1 | permitido | 2025-06-11 20:43:56 |      24 |
|             300 |         29 |               1 | permitido | 2025-06-11 20:45:40 |      24 |
|             301 |         29 |               1 | permitido | 2025-06-11 20:45:40 |      24 |
|             303 |         29 |               1 | permitido | 2025-06-11 20:47:54 |      24 |
|             304 |         29 |               1 | permitido | 2025-06-11 20:47:54 |      24 |
|             305 |         29 |               1 | permitido | 2025-06-11 20:47:54 |      24 |
|             309 |         29 |               1 | permitido | 2025-06-11 20:50:26 |      24 |
|             310 |         29 |               1 | permitido | 2025-06-11 20:50:26 |      24 |
|             311 |         29 |               1 | permitido | 2025-06-11 20:50:26 |      24 |
|             314 |         29 |               1 | permitido | 2025-06-11 20:55:46 |      24 |
|             315 |         29 |               1 | permitido | 2025-06-11 20:55:46 |      24 |
|             317 |         29 |               1 | permitido | 2025-06-11 20:58:06 |      24 |
|             319 |         29 |               1 | permitido | 2025-06-11 21:04:16 |      24 |
|             322 |         29 |               1 | permitido | 2025-06-12 14:01:39 |      24 |
|             326 |         29 |               1 | permitido | 2025-06-12 14:47:01 |      24 |
|             330 |         29 |               1 | permitido | 2025-06-12 14:50:57 |      24 |
|             332 |         29 |               1 | permitido | 2025-06-12 14:52:12 |      24 |
|             333 |         29 |               1 | permitido | 2025-06-12 14:54:32 |      24 |
|             336 |         29 |               1 | permitido | 2025-06-12 14:58:48 |      24 |
|             337 |         29 |               1 | permitido | 2025-06-12 15:00:17 |      24 |
|             341 |         29 |               1 | permitido | 2025-06-12 18:39:29 |      24 |
|             344 |         29 |               1 | permitido | 2025-06-12 18:41:50 |      24 |
```

En la **Figura 117** se realiza la consulta SELECT * FROM notificaciones WHERE id_tipo_acceso = 2 AND id_usuario IS NOT NULL;. Aquí se presentan los registros de accesos "permitidos" mediante "RFID", también con un id_usuario no nulo. Predominan los accesos del id_usuario "29" que es con el usuario que más se realizó pruebas, junto con registros para los id_usuario "28" y "30", todos asociados a la id_casa "24". Las fechas de estos eventos varían desde el 03 hasta el 20 de junio de 2025.

Figura 117

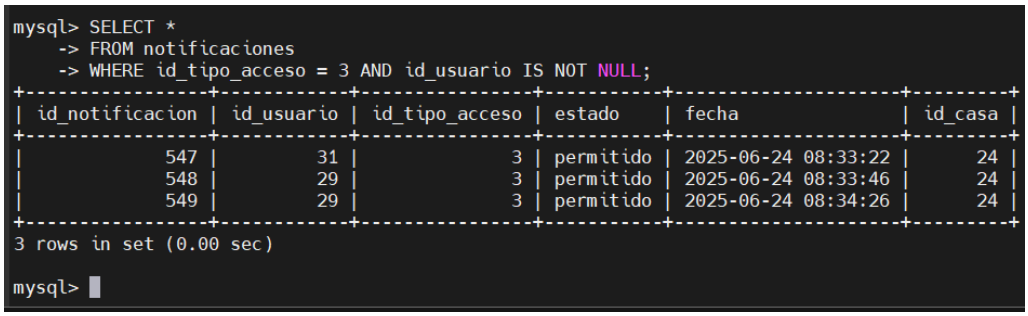
Contenido de la tabla notificaciones - tipo de acceso 2.

```
mysql> SELECT *
    -> FROM notificaciones
    -> WHERE id_tipo_acceso = 2 AND id_usuario IS NOT NULL;
+-----------------+------------+----------------+----------+---------------------+---------+
| id_notificacion | id_usuario | id_tipo_acceso | estado   | fecha               | id_casa |
+-----------------+------------+----------------+----------+---------------------+---------+
|             271 |         29 |              2 | permitido | 2025-06-03 23:43:18 |      24 |
|             275 |         29 |              2 | permitido | 2025-06-11 19:55:40 |      24 |
|             276 |         29 |              2 | permitido | 2025-06-11 19:55:40 |      24 |
|             277 |         29 |              2 | permitido | 2025-06-11 19:55:40 |      24 |
|             286 |         29 |              2 | permitido | 2025-06-11 19:58:05 |      24 |
|             287 |         29 |              2 | permitido | 2025-06-11 19:58:05 |      24 |
|             291 |         29 |              2 | permitido | 2025-06-11 20:06:55 |      24 |
|             292 |         29 |              2 | permitido | 2025-06-11 20:06:55 |      24 |
|             293 |         29 |              2 | permitido | 2025-06-11 20:06:55 |      24 |
|             302 |         29 |              2 | permitido | 2025-06-11 20:46:43 |      24 |
|             316 |         29 |              2 | permitido | 2025-06-11 20:57:27 |      24 |
|             318 |         29 |              2 | permitido | 2025-06-11 21:03:42 |      24 |
|             343 |         29 |              2 | permitido | 2025-06-12 18:41:09 |      24 |
|             368 |         29 |              2 | permitido | 2025-06-13 15:01:38 |      24 |
|             369 |         29 |              2 | permitido | 2025-06-13 15:07:44 |      24 |
|             370 |         29 |              2 | permitido | 2025-06-13 15:07:44 |      24 |
|             374 |         29 |              2 | permitido | 2025-06-13 15:14:37 |      24 |
|             376 |         29 |              2 | permitido | 2025-06-13 15:19:07 |      24 |
|             382 |         29 |              2 | permitido | 2025-06-13 16:41:49 |      24 |
|             383 |         29 |              2 | permitido | 2025-06-13 16:45:18 |      24 |
|             411 |         29 |              2 | permitido | 2025-06-15 01:10:08 |      24 |
|             412 |         29 |              2 | permitido | 2025-06-15 01:13:21 |      24 |
|             422 |         29 |              2 | permitido | 2025-06-15 01:40:20 |      24 |
|             426 |         29 |              2 | permitido | 2025-06-15 01:53:03 |      24 |
|             427 |         30 |              2 | permitido | 2025-06-15 01:53:50 |      24 |
|             442 |         28 |              2 | permitido | 2025-06-20 00:56:12 |      24 |
|             453 |         28 |              2 | permitido | 2025-06-20 01:36:00 |      24 |
|             454 |         28 |              2 | permitido | 2025-06-20 01:36:00 |      24 |
|             462 |         28 |              2 | permitido | 2025-06-20 01:41:30 |      24 |
|             468 |         30 |              2 | permitido | 2025-06-20 12:58:10 |      24 |
|             469 |         30 |              2 | permitido | 2025-06-20 13:06:01 |      24 |
|             470 |         30 |              2 | permitido | 2025-06-20 13:10:16 |      24 |
|             475 |         30 |              2 | permitido | 2025-06-20 13:53:21 |      24 |
|             476 |         30 |              2 | permitido | 2025-06-20 13:55:35 |      24 |
```

En la **Figura 118** se filtra *SELECT * FROM notificaciones WHERE id_tipo_acceso = 3 AND id_usuario IS NOT NULL;*. Aquí se presentan los registros de accesos permitidos mediante llave de circuito programable. Los accesos son de los usuarios de id_usuario 29 y 31 asociados a la id_casa 24.

Figura 118

Contenido de la tabla notificaciones - tipo de acceso 3.

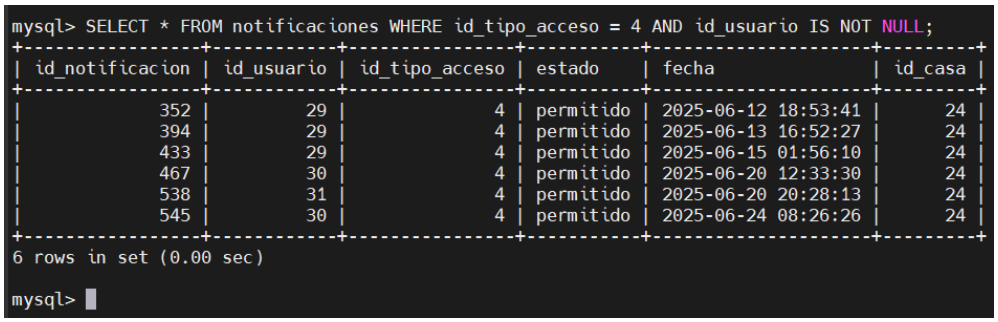


```
mysql> SELECT *
    -> FROM notificaciones
    -> WHERE id_tipo_acceso = 3 AND id_usuario IS NOT NULL;
+-----------------+------------+----------------+-----------+---------------------+---------+
| id_notificacion | id_usuario | id_tipo_acceso | estado    | fecha               | id_casa |
+-----------------+------------+----------------+-----------+---------------------+---------+
|             547 |         31 |              3 | permitido | 2025-06-24 08:33:22 |      24 |
|             548 |         29 |              3 | permitido | 2025-06-24 08:33:46 |      24 |
|             549 |         29 |              3 | permitido | 2025-06-24 08:34:26 |      24 |
+-----------------+------------+----------------+-----------+---------------------+---------+
3 rows in set (0.00 sec)

mysql>
```

El tipo de acceso 4 al filtrar *SELECT * FROM notificaciones WHERE id_tipo_acceso = 4 AND id_usuario IS NOT NULL;,* muestra los accesos permitidos utilizando clave temporal (id_tipo_acceso = 4), con usuarios identificados. Se incluyen accesos para los id_usuario 29, 30, y 31, todos asociados a la id_casa 24, como se observa en la **Figura** 119.

Figura 119

Contenido de la tabla notificaciones - tipo de acceso 4.



```
mysql> SELECT * FROM notificaciones WHERE id_tipo_acceso = 4 AND id_usuario IS NOT NULL;
+-----------------+------------+----------------+-----------+---------------------+---------+
| id_notificacion | id_usuario | id_tipo_acceso | estado    | fecha               | id_casa |
+-----------------+------------+----------------+-----------+---------------------+---------+
|             352 |         29 |              4 | permitido | 2025-06-12 18:53:41 |      24 |
|             394 |         29 |              4 | permitido | 2025-06-13 16:52:27 |      24 |
|             433 |         29 |              4 | permitido | 2025-06-15 01:56:10 |      24 |
|             467 |         30 |              4 | permitido | 2025-06-20 12:33:30 |      24 |
|             538 |         31 |              4 | permitido | 2025-06-20 20:28:13 |      24 |
|             545 |         30 |              4 | permitido | 2025-06-24 08:26:26 |      24 |
+-----------------+------------+----------------+-----------+---------------------+---------+
6 rows in set (0.00 sec)

mysql>
```

En la **Figura 120 y Figura 121**, se muestran eventos de acceso denegados o que no pudieron ser asociados completamente a un usuario o casa. Estos registros son fundamentales para la auditoría de seguridad y la detección de intentos de acceso no autorizados.

El resultado de tabla notificaciones filtra registros donde el id_usuario es nulo o el id_casa es nulo, y el estado es denegado. Se observa un extenso listado de eventos donde el acceso fue "denegado" para varios id_tipo_acceso (1 y 2, correspondientes a Teclado y RFID

respectivamente), y donde el id_usuario se presenta como NULL (usuario no identificado). Todos estos eventos están asociados a la id_casa 24 (San Agustín).

Figura 120

Contenido de la tabla notificaciones - tipo de acceso NULL.

```
mysql> SELECT * FROM notificaciones
    -> WHERE id_usuario IS NULL OR id_casa IS NULL;
+-----------------+------------+----------------+----------+---------------------+---------+
| id_notificacion | id_usuario | id_tipo_acceso | estado   | fecha               | id_casa |
+-----------------+------------+----------------+----------+---------------------+---------+
|             242 |       NULL |              1 | denegado | 2025-05-17 20:20:26 |      24 |
|             243 |       NULL |              1 | denegado | 2025-05-17 20:20:36 |      24 |
|             245 |       NULL |              2 | denegado | 2025-05-17 20:40:44 |      24 |
|             247 |       NULL |              1 | denegado | 2025-05-17 21:06:13 |      24 |
|             248 |       NULL |              1 | denegado | 2025-05-17 23:50:08 |      24 |
|             249 |       NULL |              2 | denegado | 2025-05-17 23:51:51 |      24 |
|             250 |       NULL |              2 | denegado | 2025-05-17 23:57:55 |      24 |
|             251 |       NULL |              2 | denegado | 2025-05-18 00:40:20 |      24 |
|             252 |       NULL |              1 | denegado | 2025-05-18 00:40:58 |      24 |
|             253 |       NULL |              2 | denegado | 2025-05-18 00:45:34 |      24 |
|             254 |       NULL |              2 | denegado | 2025-05-18 00:45:34 |      24 |
|             255 |       NULL |              2 | denegado | 2025-05-18 01:21:02 |      24 |
|             256 |       NULL |              1 | denegado | 2025-05-18 01:31:30 |      24 |
|             257 |       NULL |              1 | denegado | 2025-05-18 01:31:30 |      24 |
|             258 |       NULL |              1 | denegado | 2025-05-18 02:28:48 |      24 |
|             260 |       NULL |              1 | denegado | 2025-05-20 15:16:09 |      24 |
|             261 |       NULL |              1 | denegado | 2025-05-20 15:16:09 |      24 |
|             262 |       NULL |              1 | denegado | 2025-05-20 15:16:30 |      24 |
|             264 |       NULL |              1 | denegado | 2025-05-20 15:21:04 |      24 |
|             267 |       NULL |              1 | denegado | 2025-05-20 15:46:42 |      24 |
|             268 |       NULL |              1 | denegado | 2025-06-03 23:41:24 |      24 |
|             269 |       NULL |              1 | denegado | 2025-06-03 23:41:32 |      24 |
|             270 |       NULL |              2 | denegado | 2025-06-03 23:42:15 |      24 |
|             272 |       NULL |              2 | denegado | 2025-06-11 19:55:00 |      24 |
|             273 |       NULL |              2 | denegado | 2025-06-11 19:55:00 |      24 |
|             274 |       NULL |              2 | denegado | 2025-06-11 19:55:00 |      24 |
|             278 |       NULL |              1 | denegado | 2025-06-11 19:56:02 |      24 |
|             279 |       NULL |              1 | denegado | 2025-06-11 19:56:02 |      24 |
|             280 |       NULL |              1 | denegado | 2025-06-11 19:56:02 |      24 |
|             306 |       NULL |              2 | denegado | 2025-06-11 20:49:24 |      24 |
|             307 |       NULL |              2 | denegado | 2025-06-11 20:49:24 |      24 |
|             308 |       NULL |              2 | denegado | 2025-06-11 20:49:24 |      24 |
```

En la siguiente figura se visualiza los registros filtrados por fecha con denegaciones y accesos válidos. Se observan varios eventos de acceso denegado con id_usuario nulo para el id_tipo_acceso 1 (Teclado). Sin embargo, también se incluyen eventos de acceso permitido para id_usuario 29 y 30, abarcando id_tipo_acceso 1 (Teclado), 2 (RFID) y 4 (Clave Temporal). Todos estos registros corresponden a la id_casa 24.

Figura 121

Contenido de la tabla notificaciones.

```
Database changed
mysql> use jc_home2;
Database changed
mysql> SELECT * FROM notificaciones
    -> WHERE fecha >= '2025-06-13 16:50:25';
+-----------------+------------+---------------+----------+---------------------+---------+
| id_notificacion | id_usuario | id_tipo_acceso | estado   | fecha               | id_casa |
+-----------------+------------+---------------+----------+---------------------+---------+
|             389 |       NULL |             1 | denegado | 2025-06-13 16:50:25 |      24 |
|             390 |       NULL |             1 | denegado | 2025-06-13 16:50:31 |      24 |
|             391 |       NULL |             1 | denegado | 2025-06-13 16:50:35 |      24 |
|             392 |       NULL |             1 | denegado | 2025-06-13 16:50:45 |      24 |
|             393 |         29 |             1 | permitido | 2025-06-13 16:51:03 |      24 |
|             394 |         29 |             4 | permitido | 2025-06-13 16:52:27 |      24 |
|             395 |       NULL |             4 | denegado | 2025-06-13 16:53:08 |      24 |
|             396 |       NULL |             1 | denegado | 2025-06-13 17:02:04 |      24 |
|             397 |         29 |             1 | permitido | 2025-06-13 17:02:25 |      24 |
|             398 |         29 |             1 | permitido | 2025-06-13 18:07:46 |      24 |
|             399 |         29 |             1 | permitido | 2025-06-13 18:07:59 |      24 |
|             400 |       NULL |             1 | denegado | 2025-06-13 18:08:02 |      24 |
|             401 |       NULL |             1 | denegado | 2025-06-13 18:08:07 |      24 |
|             402 |       NULL |             1 | denegado | 2025-06-13 18:08:13 |      24 |
|             403 |         29 |             1 | permitido | 2025-06-13 18:08:26 |      24 |
|             404 |       NULL |             4 | denegado | 2025-06-13 18:09:18 |      24 |
|             405 |         29 |             1 | permitido | 2025-06-13 18:09:33 |      24 |
|             406 |         29 |             1 | permitido | 2025-06-14 21:05:02 |      24 |
|             407 |         29 |             1 | permitido | 2025-06-14 21:05:13 |      24 |
|             408 |       NULL |             2 | denegado | 2025-06-14 21:12:25 |      24 |
|             409 |         29 |             1 | permitido | 2025-06-15 00:05:52 |      24 |
|             410 |         29 |             1 | permitido | 2025-06-15 00:06:05 |      24 |
|             411 |         29 |             2 | permitido | 2025-06-15 01:10:08 |      24 |
|             412 |         29 |             2 | permitido | 2025-06-15 01:13:21 |      24 |
|             413 |         29 |             1 | permitido | 2025-06-15 01:13:51 |      24 |
|             414 |         29 |             1 | permitido | 2025-06-15 01:14:04 |      24 |
|             415 |         29 |             1 | permitido | 2025-06-15 01:14:15 |      24 |
|             416 |       NULL |             1 | denegado | 2025-06-15 01:14:19 |      24 |
|             417 |         29 |             1 | permitido | 2025-06-15 01:14:48 |      24 |
|             418 |       NULL |             1 | denegado | 2025-06-15 01:15:06 |      24 |
|             419 |       NULL |             1 | denegado | 2025-06-15 01:15:14 |      24 |
|             420 |         30 |             1 | permitido | 2025-06-15 01:17:44 |      24 |
```

Esta evidencia confirma la capacidad del sistema para registrar los eventos de acceso según su tipo, identidad del usuario que realizó el acceso y asociar correctamente a un (id_casa: 24). También se demuestra la coexistencia de intentos fallidos y exitosos en el log de notificaciones, y la capacidad de la base de datos para registrar ambos escenarios de forma detallada, lo cual es vital para el análisis y la depuración del sistema.

Notificaciones enviadas al correo electrónico

Se presentan ejemplos de las notificaciones automáticas enviadas por correo electrónico a los usuarios del sistema, detallando eventos de acceso relevantes en sus residencias como se

evidencia en la **Figura** 122 y **Figura** 123. Estas notificaciones son un componente crucial para mantener a los usuarios informados y para la seguridad general del sistema.

Figura 122

Notificaciones por correo electrónico (dominio Gmail).

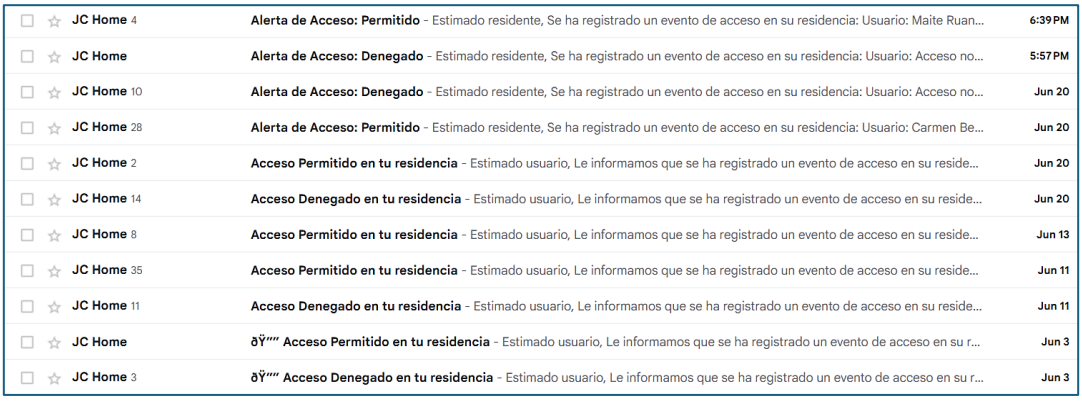


Figura 123

Notificaciones por correo electrónico (dominio institucional).

En la **Figura** 124 se muestra el correo electrónico generado por el sistema JC Home. La notificación informa al usuario que se ha registrado un evento de **acceso permitido** el

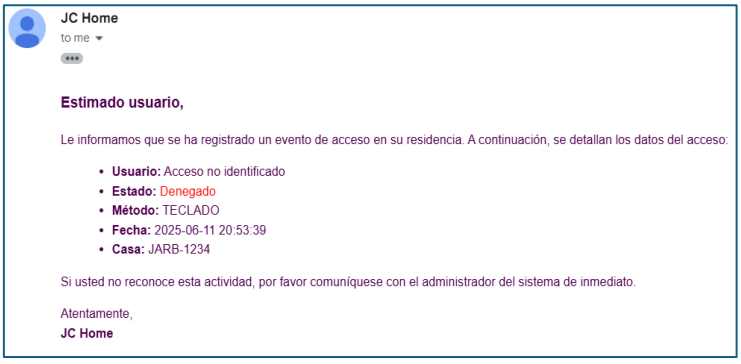
usuario identificado es Maite Ruano. En la **Figura** 125 se evidencia un **acceso denegado** en la residencia, con un "**usuario: Acceso no identificado**" y un "Método: TECLADO". Este correo incluye una advertencia para que el usuario se comunique con el administrador si no reconoce la actividad, lo cual valida la funcionalidad de alerta ante intentos de acceso no autorizados y promueve la acción del usuario.

Figura 124

Notificación de acceso permitido por teclado.

Figura 125

Notificación de acceso denegado por teclado.



Se presenta el correo electrónico de JC Home que notifica un evento de acceso Permitido como se observa en la **Figura** 126. Se especifica que el usuario es "Adamaris", el estado es "Permitido" y el "Método" de acceso es RFID. Esta notificación confirma la capacidad del sistema para informar a los usuarios sobre accesos exitosos mediante

identificación RFID, proporcionando transparencia y tranquilidad. Adicionalmente, en la **Figura** 127 se evidencia un **acceso denegado** en la residencia, con un "**usuario: Acceso no identificado**" y un "Método: RFID".

Figura 126

Notificación de acceso permitido por RFID.

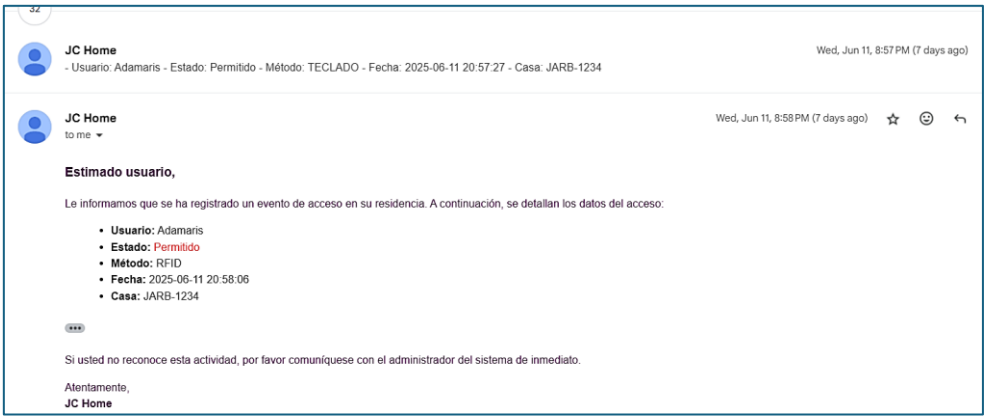
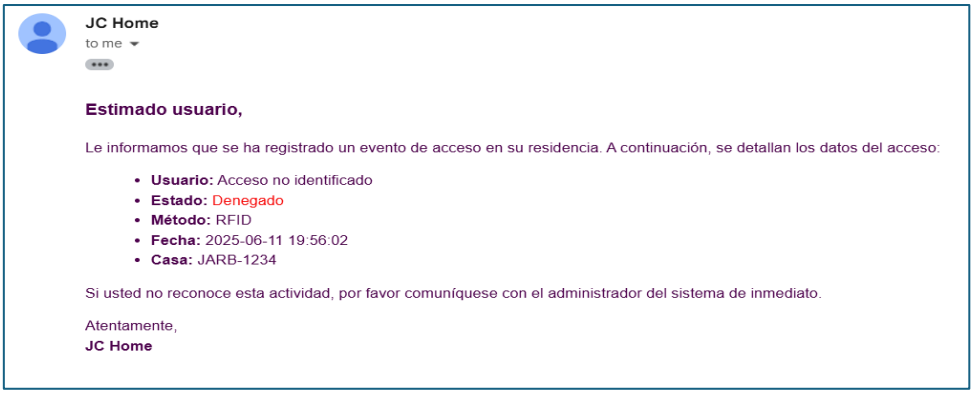


Figura 127

Notificación de acceso denegado por RFID.



Se verifica que, para el acceso por llave de circuito programable, notifica un evento de acceso "Permitido". Se especifica que el "Usuario" es "Adamaris", el "Estado" es "Permitido" y el "Método" de acceso es "llave". También, en la **Figura** 129 el sistema envía notificaciones de **accesos denegados**.

Figura 128

Notificación de acceso permitido por llave.

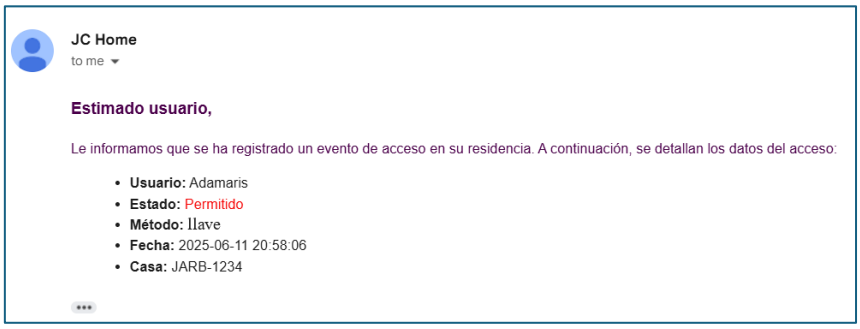
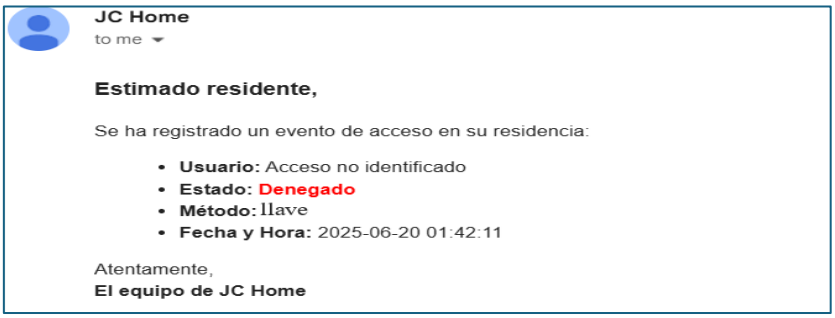


Figura 129

Notificación de acceso denegado por llave.



Se muestra un correo electrónico enviado desde la dirección jessi.jessica1717@gmail.com dirigido a "RUANO BENAVIDES JESSICA ARACELY". El contenido del correo informa sobre un evento de acceso "Denegado" con un "Usuario: Acceso no identificado" y un "Método: CLAVE_TEMPORAL". A diferencia del caso anterior en la **Figura** 131 se corrobora la notificación cuando el acceso es permitido.

Figura 130

Notificación de acceso denegado por clave temporal.

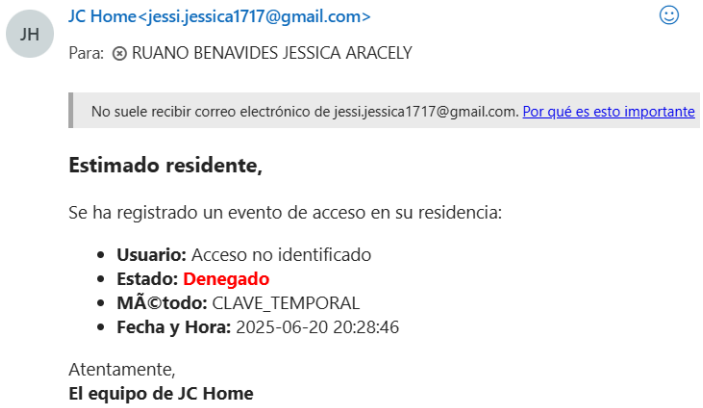


Figura 131

Notificación de acceso permitido por clave temporal.

Los resultados obtenidos permiten concluir que el sistema cumple en su totalidad con él envió de notificaciones por correo electrónico. Además, se observa la capacidad para generar alertas detalladas, diferenciando entre accesos permitidos y denegados, e indicando el método de acceso utilizado (teclado, RFID, clave temporal). Esto asegura que los usuarios reciban

información oportuna y relevante sobre la actividad de acceso en sus residencias, cumpliendo con los requisitos de comunicación y seguridad del sistema.

4.1.7 Discusión de resultados

El sistema desarrollado demostró ser funcional a través de un conjunto de pruebas, abarcó todos los aspectos esenciales: registro de usuarios, autenticación mediante tres métodos diferentes, por **teclado** (contraseña numérica y clave temporal), **tarjeta RFID** y **llave de circuito programable**. Además, del envío de notificaciones por correo electrónico. Los resultados obtenidos durante las pruebas validaron no solo la viabilidad técnica del proyecto, sino también su capacidad para cumplir con los requisitos funcionales y de seguridad establecidos desde el diseño inicial.

El proceso de registro y autenticación de usuarios demostró una alta eficiencia. La aplicación posibilitó el registro exitoso de múltiples usuarios, asegurando la validez de los datos ingresados mediante validaciones obligatorias y restricciones en campos sensibles como la cédula. Adicionalmente, se implementó una política de restablecimiento de contraseñas segura, fundamentada en claves temporales generadas dinámicamente y enviadas por correo electrónico. Estas claves poseían un tiempo limitado de uso, lo cual minimizaba el riesgo de interceptación o reutilización no autorizada, aspecto crucial en sistemas de control de acceso. No obstante, se podría contemplar la incorporación de alternativas adicionales, como la autenticación de dos factores (2FA) mediante aplicaciones móviles, para optimizar aún más la protección contra accesos no autorizados.

Por otro lado, los tres métodos de acceso evaluados por teclado, por RFID y por llave de circuito programable funcionaron según lo esperado, aunque cada uno presentó características propias. El método por teclado resultó intuitivo y fácil de usar, y demostraron que el sistema puede validar tanto contraseñas permanentes como claves temporales con

precisión. La contraseña numérica registrada (6 dígitos) cumplió con el criterio de longitud mínima. En cuanto a las claves temporales, se observa que estas llegan al destinatario en menos de 30 segundos y son válidas durante los periodos configurados (5, 10, 30 minutos, 1 hora y 2 horas), según la necesidad del usuario. En cuanto al acceso por RFID, fue rápido y efectivo, durante las pruebas, se validó que el sistema podía registrar múltiples tarjetas RFID asociadas a usuarios diferentes, garantizando la capacidad de gestionar varios accesos. Además, se comprobó que el tiempo de respuesta del sistema al validar las tarjetas fue menos de 1 minuto , lo cual cumple con los requisitos de rapidez y eficiencia esperados para aplicaciones de control de acceso. Sin embargo, es importante destacar que este método presenta un riesgo medio-bajo de clonación de tarjetas. Este riesgo se debe a que, aunque técnicamente es posible clonar una tarjeta RFID utilizando dispositivos especializados (clonadores RFID), dicho proceso requiere estar físicamente cerca de la tarjeta original (aproximadamente 1 cm) y contar con conocimientos técnicos específicos.

Finalmente, la llave de circuito programable destacó por el alto nivel de seguridad, empleando un protocolo de desafíos aleatorios y hashes criptográficos. Como observación, su alcance inalámbrico es de 50 cm aproximadamente, y para un mayor alcance es necesario realizar cambios en la alimentación de voltaje de los módulos RF-433MHz, específicamente en los transmisores tanto de la cerradura como de la llave de circuito programable.

El sistema de notificaciones también demostró ser una herramienta clave para mantener informados a los usuarios sobre eventos de acceso. Las alertas se enviaron correctamente a los correos registrados, incluyendo información detallada como nombre del usuario, tipo de acceso y hora exacta del evento. Esto permite una mayor transparencia y facilita la auditoría de accesos.

En cuanto al manejo de intentos fallidos de acceso, el sistema permite hasta tres intentos, de ser incorrectos envía una notificación de acceso no identificado, esto mejora la seguridad y reduce el riesgo de ataques de fuerza bruta.

Finalmente, el sistema logró satisfacer los objetivos planteados, demostrando solidez técnica y viabilidad operativa. Su arquitectura modular permite extenderse a otros escenarios, como edificios residenciales, oficinas corporativas o instalaciones industriales. Además, el uso de componentes económicos y ampliamente disponibles hace que el sistema sea replicable y accesible para diversos contextos.

4.2 Costo beneficio

En esta sección se detallan los costos de los componentes electrónicos utilizados en la construcción del prototipo, junto con sus respectivos precios unitarios y costos totales. Los materiales fueron adquiridos directamente desde distribuidores autorizados, considerando compras al por mayor con el fin de obtener precios más competitivos. A partir de estos datos se calcularon los costos unitarios promedio, permitiendo estimar el costo total del hardware necesario para la operación del sistema.

4.2.1 Costos de componentes electrónicos (hardware)

En la **Tabla 27** se detallan la cantidad de componentes utilizados y su precio. Entre los componentes principales destacan: un microcontrolador ESP32, un teclado matricial de 4x4, un lector RFID RC522, dos tarjetas RFID, una pantalla LCD con interfaz I2C, un módulo transmisor y receptor RF de 433 MHz, además de diversos elementos pasivos como condensadores, resistencias, LEDs, interruptores, entre otros.

Tabla 27

Costos de componentes electrónicos para cerradura eléctrica.

Componentes	Cantidad	Costo unitario	Total
Microcontrolador ESP32	1	$3.20	$3.20
Teclado Matricial	1	$1.84	$1.84
Lector RFID	1	$1.74	$1.74
Tarjetas RFID RC522	2	$0.16	$0.32
Pantalla LCD con interfaz I2C	1	$1.27	$1.27
Transmisor y Receptor de RF 433MHz	1	$1.69	$1.69
Borneras	2	$0.10	$0.20
Módulo Relé	1	$1.80	$1.80
LEDs	4	$0.05	$0.20
Condensadores 4700uF-25v	1	$0.10	$0.10
Condensadores 1000uF-16v	1	$0.01	$0.01
Capacitor cerámico	2	$0.01	$0.02
Resistencias	6	$0.03	$0.20
Fusible 2A	1	$0.05	$0.05
Porta fusible	1	$0.31	$0.31
Interruptores	1	$0.10	$0.10
Total			$13.05

Nota. El costo total de los componentes electrónicos asciende a $13.05, lo cual representa una inversión relativamente baja teniendo en cuenta las funcionalidades avanzadas que incorpora el prototipo. Esta característica refuerza la viabilidad del proyecto tanto desde el punto de vista

técnico como económico, especialmente si se considera la posibilidad de escalamiento y producción en serie.

A continuación, se detallan la cantidad de componentes utilizados en la fabricación de la llave electrónica programable. Los componentes seleccionados responden a criterios de funcionalidad. Cabe destacar que las antenas utilizadas en los módulos transmisores y receptores RF, tanto para la llave como para la cerradura, fueron elaboradas manualmente.

A continuación, se presenta una **Tabla 28** con el detalle de los costos asociados a cada componente utilizado en la elaboración del prototipo de la llave electrónica.

Tabla 28

Costos de componentes electrónicos para llave electrónica.

Componentes	Cantidad	Costo unitario	Total
Arduino Nano	1	$1.28	$1.28
Pulsador de 4 pines	1	$0.10	$0.10
Bornera	1	$0.10	$0.10
Resistencias	4	$0.03	$0.12
LEDs	3	$0.05	$0.15
Batería de litio 3.7v	1	$6.20	$6.20
Módulo de carga	1	$0.28	$0.28
Transmisor y Receptor de RF 433MHz	1	$1.69	$1.69
Total			$9.92

Nota. Cabe destacar que las antenas utilizadas en los módulos transmisores y receptores RF, tanto para la llave como para la cerradura, fueron elaboradas manualmente. El costo total de los componentes para la llave electrónica fue de $9.92 USD.

4.2.2 Costo de herramientas adicionales para el desarrollo del sistema

Para el correcto desarrollo e implementación del prototipo del sistema de cerradura eléctrica con llave programable, fue necesario contar con un conjunto de herramientas y materiales auxiliares que facilitan la construcción, soldadura y prueba de las placas electrónicas tanto de la cerradura como de la llave. Aunque parte de algunas herramientas básicas ya se encontraban disponibles, como multímetro, estación de soldadura (cautín), corta cables, pasta de soldar, termo contráctil, sierra, alcohol isopropílico, esmalte protector para las placas (para evitar oxidación), taladro de mano y brocas, fue necesario adquirir ciertos componentes adicionales específicos para el proyecto como se detalla en la ¡Error! No se encuentra el origen d e la referencia..

Tabla 29

Componentes adicionales y herramientas.

Componentes	Costo total
Cerradura Eléctrica	$22
Placa PCB (prototipo)	$0.65
Transformador 12V	$11.40
Cableado y conectores	$4.20
Acido férrico	$2.35
Estaño	$1.13

Total	$41.73

4.2.3 Software y servicios

Para el desarrollo del prototipo se utilizaron diferentes herramientas de software y plataformas en línea que apoyaron tanto el diseño del circuito como la operación del sistema. A continuación, en la **Tabla 30** se presenta un resumen de los elementos de software y servicios utilizados durante el proyecto, junto con sus respectivos costos.

Tabla 30

Software y servicios.

Elemento	**Costo**
Plataforma de Hosting (plan básico mensual)	$14.61
Software de diseño (Proteus)	$0.00
Android Studio	$0.00
Total	$14.61

4.2.4 Resumen general de costos

A fin de tener una visión integral del presupuesto invertido en el desarrollo del prototipo del sistema de cerradura eléctrica con llave programable, se presenta a continuación un resumen general de los costos incurridos en cada una de las categorías analizadas.

Tabla 31

Costos totales de todo el sistema.

Categoría	Subtotal
Hardware Cerradura	$13.05
Hardware Llave	$9.92
Componentes Adicionales	$41.73
Software y Servicios	$14.61
Total	$79.31

4.2.5 Análisis de costo – beneficio

El presente análisis tiene como objetivo evaluar los costos asociados a la materialización del prototipo del sistema de cerradura eléctrica con llave programable, así como identificar los beneficios esperados derivados de su implementación. Este estudio permite obtener una visión clara sobre la viabilidad económica y funcional de este proyecto, lo cual resulta fundamental para justificar la inversión realizada y el potencial retorno que ofrece la propuesta tecnológica.

El costo total estimado para el desarrollo del prototipo asciende a $79.31 USD. Este monto se distribuye en las siguientes categorías: hardware de la cerradura ($13.05), hardware de la llave electrónica ($9.92), componentes adicionales ($41.73) y software y servicios ($14.61). La mayor parte del presupuesto se destina a los componentes adicionales, los cuales representan aproximadamente el 53% del costo total. Este porcentaje se justifica principalmente por la inversión en materiales necesarios para la fabricación de placas PCB, así como para la alimentación y conexión física del sistema.

En cuanto al hardware de la cerradura, el microcontrolador ESP32 destaca como uno de los elementos más significativos, representando el 24% del costo total de esta categoría. Su versatilidad y capacidad de integración permiten soportar múltiples modos de autenticación, comunicación inalámbrica y registro de eventos, lo cual incrementa considerablemente el valor funcional del dispositivo sin un elevado costo adicional.

El costo de la llave electrónica es moderado, con un total de $9.92. La batería de litio de 3.7 V constituye el 62% de este valor, debido a las especificaciones técnicas necesarias para asegurar una autonomía adecuada. Sin embargo, los demás componentes como el Arduino Nano, las resistencias, los LEDs y el módulo RF son económicos y fáciles de conseguir, lo que valida la viabilidad económica del diseño. Además, la fabricación manual de las antenas RF contribuyó a disminuir los gastos, optimizando el presupuesto general del prototipo.

En cuanto a los costos de herramientas y materiales auxiliares, se requirió la adquisición de ciertos elementos para el desarrollo e integración del sistema, como la placa PCB, un transformador, cableado y ácido férrico. Estos insumos totalizaron $41.73, cubriendo tanto materiales básicos de construcción como componentes funcionales indispensables para el correcto funcionamiento del prototipo.

Finalmente, en términos de software y servicios, se incurrió en un costo de $14.61, correspondiente exclusivamente a un plan básico de hosting mensual. El resto de las herramientas utilizadas, como el software de simulación Proteus y posibles APIs o servicios adicionales, fueron utilizados sin costo alguno gracias a versiones gratuitas.

En conjunto, el bajo costo unitario de los componentes empleados, combinado con las avanzadas funcionalidades ofrecidas por el sistema, posiciona al prototipo como una solución viable, escalable y económicamente rentable. Esta característica permite considerar escenarios futuros de producción a gran escala, donde se podría lograr una disminución adicional en el

precio final, facilitando su adopción en sectores residenciales, industriales o institucionales que demandan soluciones de seguridad inteligente y personalizable.

5 CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- El diseño de la llave de circuito programable cumple con los requisitos establecidos y garantiza un alto nivel de seguridad mediante la generación de hashes para el envío seguro de datos.
- En las pruebas de acceso con teclado, se confirmó la necesidad de usar una tecla de activación antes de la clave: la tecla A para la contraseña fija y la C para la temporal. Este sistema incluye una medida de seguridad que le da al usuario tres intentos para ingresar; si falla, se registra el acceso como fallido y se envía una notificación.
- Se pudo ver que los módulos de radiofrecuencia (RF) comunican la llave y la cerradura de manera efectiva en un rango de 1 a 50 cm. Esto supera por mucho la limitación de las tarjetas RFID, que apenas llegan a 1 cm. También se determinó que esta distancia podría ser incluso mayor si se mejora el diseño del circuito y se le da más voltaje al transmisor.
- Se pudo comprobar que el proceso para registrar nuevos dispositivos es seguro. Para vincular una tarjeta RFID o una llave programable con un usuario, primero se debe presionar una tecla de activación (la **B** para RFID o la **D** para la llave) y, justo después, introducir la contraseña numérica de esa persona. De esta forma, se garantiza que solo el dueño de la cuenta pueda agregar nuevos métodos de acceso.
- El sistema de la cerradura demostró una comunicación estable y robusta con el servidor a través de internet. Esta conectividad constante fue fundamental para realizar consultas en tiempo real a la base de datos, validar la información de

acceso y, en consecuencia, autorizar o denegar la apertura de puerta de manera fiable.

- Para vigilar la actividad mientras se probaban los diferentes métodos de acceso, se usó el comando tail -f en los archivos de registro de Apache (error.log y access.log). Esta acción permitió supervisar de manera eficaz y en tiempo real todo lo que estaba sucediendo en el servidor.

5.2 Recomendaciones

- Durante la etapa de diseño, es muy recomendable revisar las hojas de datos (datasheets) de cada uno de los componentes electrónicos. Resulta fundamental comprobar los distintos niveles de voltaje con los que operan para así asegurar la compatibilidad entre ellos y, si hiciera falta, usar los reguladores de voltaje necesarios. Esta buena práctica es crucial para proteger los componentes y garantizar que el circuito funcione de forma estable.
- Antes de diseñar y fabricar la placa de circuito impreso (PCB), es fundamental que el circuito se ensamble y se ponga a prueba en un protoboard. Esta etapa de prototipado es precisamente la que permite hacer ajustes y mejoras de una manera flexible y económica, evitando así los costos extra y los retrasos que aparecerían si se tuviera que modificar una placa ya fabricada.
- Para futuras versiones de la llave electrónica, se sugiere incrementar el voltaje de alimentación del transmisor RF. Para aumentar la potencia de salida del módulo, y extender el alcance de la comunicación con la cerradura, así mejorando la experiencia del usuario.
- En el desarrollo del backend, se debe implementar un archivo de configuración centralizado, ejemplo config.php. Para que en este archivo se deben almacenen

las credenciales de la base de datos y otras variables globales. Esto con el fin de mejorar la seguridad y reducir la redundancia de código, al evitar reescribir información sensible en múltiples archivos.

- Desde la fase inicial del proyecto, es fundamental definir si la aplicación de software será nativa o multiplataforma, para evitar cambiar la compatibilidad después del desarrollo.

6 REFERENCIAS

ABRA. (2025). *ESP-WROOM-32 Micro-contrôleur IOT WiFi+Bluetooth BLE, 30 pin, usb micro connection, CP2102*. https://abra-electronics.com/robotics-embedded-electronics/esp-series/geeekus-wifi-esp32-esp32-wroom-32-wi-fi-bluetooth-ble-low-power-iot-microcontroller.html?sl=fr

Alfonso Pagán, I. M. (2019). *Sistema de autenticación robusto* [Trabajo fin de máster]. Universidad de Alicante.

Alvarez-Marin, A., & Castillo-Vergara, M. (2015). Estrategias para acercar la tecnología de identificación por radiofrecuencia a la formación de futuros ingenieros industriales. *Formacion Universitaria*, *8*(1), 23–34. https://doi.org/10.4067/S0718-50062015000100004

ANDROID STUDIO. (2025). *Cómo descargar Android Studio y App Tools - Android Developers*. https://developer.android.com/studio?hl=es-419

Arduino. (2025). *Arduino® Nano*. *Maker, Security, Environmental, Robotics and Control Systems*. https://docs.arduino.cc/resources/datasheets/A000005-datasheet.pdf

Consejo Nacional para la Igualdad de Discapacidades. (2016). *LEY DE SEGURIDAD PÚBLICA Y DEL ESTADO*.

DIGITAL TALENT AGENCY. (2018). *METODOLOGÍAS DE GESTIÓN DE PROYECTOS TEMA 1 MODELO WATERFALL O EN CASCADA*. https://www.dtagency.tech/cursos/metodologias_gestion_proyectos/tema_1-ModeloWaterfall.pdf

Enrique, A. :, Garrote, M., Tutores, S., Portilla, J., Teresa, B., & Alcaide, R. (2017). *CERRADURA ELECTRÓNICA CON SISTEMA DE ALIMENTACIÓN INTEGRADO EN LLAVE*.

Espressif Systems. (2023). *ESP32WROOM32 Datasheet*. https://www.espressif.com/en/support/download/documents.

Farwah Nawazi. (2022, March 1). *Módulos transmisores y receptores de RF FS1000A de 433 MHz*. https://www.circuits-diy.com/fs1000a-433mhz-rf-transmitter-receiver-modules/

Gonzalo, A. :, & Salmerón, M. (2021). *Herramientas para la Ruptura del Secreto de Contraseñas*.

Grupo de Regulación de AUTELSI. (2021). *SUPLANTACIÓN DE IDENTIDAD DIGITAL Grupo de Regulación de AUTELSI*.

HOSTINGER. (2025). *Parámetros y límites de los planes de hosting | Centro de ayuda de Hostinger*. https://support.hostinger.com/es/articles/6976044-parametros-y-limites-de-los-planes-de-hosting

Huidobro, J. M. (2010). *La tecnología RFID*.

INCIBE. (2016). *Tecnologías biométricas aplicadas a la ciberseguridad*.

Infante, J. (2021, October 31). *Comunicación de datos*. https://es.scribd.com/document/541665322/Actividad-1-Teleprocesos

ISO/IEC/IEEE 42010:2018. (2024). *ISO/IEC/IEEE 29148:2018 - Systems and software engineering — Life cycle processes — Requirements engineering*.

Https://Www.Iso.Org/Es/Contents/Data/Standard/07/20/72089.Html. https://www.iso.org/es/contents/data/standard/07/20/72089.html

ISO/IEC/IEEE 42010:2022. (2022). *ISO/IEC/IEEE 42010:2022 - Software, systems and enterprise — Architecture description*. Https://Www.Iso.Org/Standard/74393.Html. https://www.iso.org/standard/74393.html

La Hora. (2023, April 12). *Drones que sobrevuelan conjuntos residenciales podrían ser usados por delincuentes*. 12 de Abril 2023. https://www.lahora.com.ec/archivo/Drones-que-sobrevuelan-conjuntos-residenciales-podrian-ser-usados-por-delincuentes-20230412-0042.html

Lozano, A. V. (2016). *Impacto de la seguridad física en los condominios residenciales de la ciudad de Cali*.

Mendoza Gómez, M. Á. (2017). *UNA MIRADA A LA COMPLEJIDAD COMPUTACIONAL Y SEGURIDAD EN LA PRÁCTICA DE LOS ALGORITMOS MD5 Y DES*. UNIVERSIDAD TECNOLÓGICA DE PEREIRA.

Mojica Francisco. (2014). *Evolución Histórica de la Seguridad INVESTIGACIÓN CRIMINAL*.

Ortega Chulde, C. A. (2023). *IMPLEMENTACIÓN DE MECANISMO DE SEGURIDAD PARA REDES DE SENSORES INALÁMBRICOS BASADO EN CRIPTOGRAFÍA DE CURVA ELÍPTICA*. Universidad Técnica del Norte.

Pastrana, S. A., Vidal, P. J., & Lasso, M. G. (2010). *Sistema de Control de Acceso y Permanencia-PASA*.

Portillo García, J., Bermejo Nieto, A., & Bernardos Barbolla. (2008). *Tecnología de identificación por radiofrecuencia (RFID): aplicaciones en el ámbito de la salud*. 82–84.

Red de Bibliotecas Universitarias (REBIUN). (2019). *APRENDIZAJE OBJETOS DE LÍNEA 2*.

Tarazona, C. H. (2023). *AMENAZAS INFORMÁTICAS Y SEGURIDAD DE LA INFORMACIÓN*.

UNAM. (2020). *Criptografía*. https://www.matem.unam.mx/~rajsbaum/cursos/web/presentacion_seguridad_1.pdf

UNED. (n.d.). *2 CONTRASEÑAS gestión segura*.

Veriddica. (2022). *El presente y futuro de la autenticación*. https://blog.portinos.com/el-dato/como-nos-resguarda-el-segundo-factor-de-autenticacion-de-identidad

ABRA. (2025). *ESP-WROOM-32 Micro-contrôleur IOT WiFi+Bluetooth BLE, 30 pin, usb micro connection, CP2102*. https://abra-electronics.com/robotics-embedded-electronics/esp-series/geeekus-wifi-esp32-esp32-wroom-32-wi-fi-bluetooth-ble-low-power-iot-microcontroller.html?sl=fr

Alfonso Pagán, I. M. (2019). *Sistema de autenticación robusto* [Trabajo fin de máster]. Universidad de Alicante.

Alvarez-Marin, A., & Castillo-Vergara, M. (2015). Estrategias para acercar la tecnología de identificación por radiofrecuencia a la formación de futuros ingenieros industriales. *Formacion Universitaria*, *8*(1), 23–34. https://doi.org/10.4067/S0718-50062015000100004

ANDROID STUDIO. (2025). *Cómo descargar Android Studio y App Tools - Android Developers*. https://developer.android.com/studio?hl=es-419

Arduino. (2025). *Arduino® Nano*. *Maker, Security, Environmental, Robotics and Control Systems*. https://docs.arduino.cc/resources/datasheets/A000005-datasheet.pdf

Consejo Nacional para la Igualdad de Discapacidades. (2016). *LEY DE SEGURIDAD PÚBLICA Y DEL ESTADO*.

DIGITAL TALENT AGENCY. (2018). *METODOLOGÍAS DE GESTIÓN DE PROYECTOS TEMA 1 MODELO WATERFALL O EN CASCADA*. https://www.dtagency.tech/cursos/metodologias_gestion_proyectos/tema_1-ModeloWaterfall.pdf

Enrique, A. :, Garrote, M., Tutores, S., Portilla, J., Teresa, B., & Alcaide, R. (2017). *CERRADURA ELECTRÓNICA CON SISTEMA DE ALIMENTACIÓN INTEGRADO EN LLAVE*.

Espressif Systems. (2023). *ESP32WROOM32 Datasheet*. https://www.espressif.com/en/support/download/documents.

Farwah Nawazi. (2022, March 1). *Módulos transmisores y receptores de RF FS1000A de 433 MHz*. https://www.circuits-diy.com/fs1000a-433mhz-rf-transmitter-receiver-modules/

Gonzalo, A. :, & Salmerón, M. (2021). *Herramientas para la Ruptura del Secreto de Contraseñas*.

Grupo de Regulación de AUTELSI. (2021). *SUPLANTACIÓN DE IDENTIDAD DIGITAL Grupo de Regulación de AUTELSI*.

HOSTINGER. (2025). *Parámetros y límites de los planes de hosting | Centro de ayuda de Hostinger*. https://support.hostinger.com/es/articles/6976044-parametros-y-limites-de-los-planes-de-hosting

Huidobro, J. M. (2010). *La tecnología RFID*.

INCIBE. (2016). *Tecnologías biométricas aplicadas a la ciberseguridad*.

Infante, J. (2021, October 31). *Comunicación de datos*. https://es.scribd.com/document/541665322/Actividad-1-Teleprocesos

ISO/IEC/IEEE 42010:2018. (2024). *ISO/IEC/IEEE 29148:2018 - Systems and software engineering — Life cycle processes — Requirements engineering*. Https://Www.Iso.Org/Es/Contents/Data/Standard/07/20/72089.Html. https://www.iso.org/es/contents/data/standard/07/20/72089.html

ISO/IEC/IEEE 42010:2022. (2022). *ISO/IEC/IEEE 42010:2022 - Software, systems and enterprise — Architecture description*. Https://Www.Iso.Org/Standard/74393.Html. https://www.iso.org/standard/74393.html

La Hora. (2023, April 12). *Drones que sobrevuelan conjuntos residenciales podrían ser usados por delincuentes*. 12 de Abril 2023. https://www.lahora.com.ec/archivo/Drones-que-sobrevuelan-conjuntos-residenciales-podrian-ser-usados-por-delincuentes-20230412-0042.html

Lozano, A. V. (2016). *Impacto de la seguridad física en los condominios residenciales de la ciudad de Cali*.

Mendoza Gómez, M. Á. (2017). *UNA MIRADA A LA COMPLEJIDAD COMPUTACIONAL Y SEGURIDAD EN LA PRÁCTICA DE LOS ALGORITMOS MD5 Y DES*. UNIVERSIDAD TECNOLÓGICA DE PEREIRA.

Mojica Francisco. (2014). *Evolución Histórica de la Seguridad INVESTIGACIÓN CRIMINAL*.

Ortega Chulde, C. A. (2023). *IMPLEMENTACIÓN DE MECANISMO DE SEGURIDAD PARA REDES DE SENSORES INALÁMBRICOS BASADO EN CRIPTOGRAFÍA DE CURVA ELÍPTICA*. Universidad Técnica del Norte.

Pastrana, S. A., Vidal, P. J., & Lasso, M. G. (2010). *Sistema de Control de Acceso y Permanencia-PASA*.

Portillo García, J., Bermejo Nieto, A., & Bernardos Barbolla. (2008). *Tecnología de identificación por radiofrecuencia (RFID): aplicaciones en el ámbito de la salud*. 82–84.

Red de Bibliotecas Universitarias (REBIUN). (2019). *APRENDIZAJE OBJETOS DE LÍNEA 2*.

Tarazona, C. H. (2023). *AMENAZAS INFORMÁTICAS Y SEGURIDAD DE LA INFORMACIÓN*.

UNAM. (2020). *Criptografía*. https://www.matem.unam.mx/~rajsbaum/cursos/web/presentacion_seguridad_1.pdf

UNED. (n.d.). *2 CONTRASEÑAS gestión segura*.

Veriddica. (2022). *El presente y futuro de la autenticación*. https://blog.portinos.com/el-dato/como-nos-resguarda-el-segundo-factor-de-autenticacion-de-identidad

Cepal. (26 de abril de 2019). *ODS 11: Lograr que las ciudades y los asentamientos humanos sean inclusivos, seguros, resilientes y sostenibles en América Latina y el Caribe*. Obtenido de https://www.cepal.org/sites/default/files/static/files/ods11_c1900717_press.pdf

DIGITAL TALENT AGENCY. (2018). *Modelos de gestion de proyectos | Digital Talent Agency*. Obtenido de Modelos de gestion de proyectos | Digital Talent Agency: https://www.dtagency.tech/cursos/metodologias_gestion_proyectos/tema_1-ModeloWaterfall.pdf

Duarte, J. A. (2013). *Sistema de control para una cerradura electronica con microcontrolador.* Cartagena: Universidad Politécnica de Cartagena .

El Comercio. (14 de junio de 2023). *El 97% de los ecuatorianos priorizan la seguridad al buscar una vivienda*. Obtenido de https://www.elcomercio.com/actualidad/negocios/ecuatorianos-priorizan-seguridad-encontrar-vivienda.html

Garcia, C. J., & Mesa, C. D. (2022). *Sistema de cerradura electrónica para el control de acceso por reconocimiento digital de huella dactilar.* Bogota: Universidad Católica de Colombia .

Lozano, A. V. (2016). *Impacto de la seguridad física en los condominios residenciales de la ciudad de Cali.* Santiago de Cali: Universidad San Buenaventura .

Martínez, V. A. (enero de 2023). *Diseño y montaje de una cerradura electrónica con sensor de huella AS608*. Obtenido de CEUTEC : https://repositorio.unitec.edu/xmlui/bitstream/handle/123456789/12419/Dise%c3%b1o%20y%20montaje%20de%20una%20cerradura%20electr%c3%b3nica%20con%20sensor%20de%20huella%20AS608.pdf?sequence=1

Miño, H. P. (2017). *Diseño e implementación de un prototipo de cerradura con registro de acceso, supervisión remota y vigilancia.* Quito: Escuela Politécnica Nacional.

Nossa Jiménez, B. S. (2021). *Diseño de un sistema con datos sobre el estado de las cerraduras electromecánicas Abloy por medio de un dispositivo móvil android.* Universidad Dsitral Francisco Jose de Caldas.

Revelo, J. C., & Andrade, J. C. (2020). *Diseño de cerradura inteligente para el hostal Armonía Lodge* . Guayaquil: Escuela Superior Politécnica del Litoral.

Vaca, M. (2021). *Diseño e implementación de un prototipo de cerradura con registro de acceso, supervisión remota y videovigilancia.* Quito: Escuela Politecnica Nacional.

7 ANEXOS

7.1 Anexo I – Diseño entidad-relación para la base de datos

```
Table usuarios {
  id_usuario int [pk, not null]
  nombre varchar(100) [not null]
  cedula varchar(20) [not null, unique]
  celular varchar(20) [not null]
  email varchar(100) [not null]
  contraseña varchar(255) [not null]
  tipo_usuario enum('usuario', 'administrador') [not null]
  id_casa int [ref: > casas.id_casa]
}

Table casas {
  id_casa int [pk, not null]
  codigo varchar(50) [not null, unique]
  descripcion varchar(255)
}

Table tipos_acceso {
  id_tipo_acceso int [pk, not null]
  tipo enum('rfid', 'teclado', 'llave', 'clave_temporal') [not null, unique]
}

Table accesos {
  id_acceso int [pk, not null]
  id_usuario int [not null, ref: > usuarios.id_usuario]
  id_tipo_acceso int [not null, ref: > tipos_acceso.id_tipo_acceso]
  codigo varchar(50) [not null]
}

Table contraseñas_llave {
  id_contraseña int [pk, not null]
  id_acceso int [not null, ref: > accesos.id_acceso]
  contraseña varchar(255) [not null]
}

Table claves_temporales {
  id_clave int [pk, not null]
  id_usuario int [not null, ref: > usuarios.id_usuario]
  codigo varchar(50) [not null]
  fecha_creacion datetime [not null]
  fecha_fin datetime [not null]
  utilizada boolean [default: false]
}

Table permisos_acceso {
  id_permiso int [pk, not null]
  id_usuario int [not null, ref: > usuarios.id_usuario]
  id_tipo_acceso int [not null, ref: > tipos_acceso.id_tipo_acceso]
  activo boolean [not null, default: true]
}

Table notificaciones {
  id_notificacion int [pk, not null]
  id_usuario int [ref: > usuarios.id_usuario]
  id_tipo_acceso int [not null, ref: > tipos_acceso.id_tipo_acceso]
  estado enum('permitido', 'denegado') [not null]
```

```
  fecha datetime [not null]
  id_casa int [not null, ref: > casas.id_casa]
}
```

7.2 Anexo II - Código para funcionamiento de la llave electrónica

```
#include <RH_ASK.h>
#include <SPI.h>

RH_ASK driver(2000, 12, 11);
const int button_pin = 2;
int ledg = 4;
int ledr = 3;

const char* id_llave = "1234ABCD";
const char* contrasena_local = "Password4545";

String codigo_aleatorio = "";
bool esperando_codigo = false;
unsigned long tiempo_inicio_espera = 0;
const unsigned long tiempo_espera_maximo = 3000; // 3 segundos

void simpleMD5_16(const char *str, char *output) {
  unsigned long hash1 = 475381;
  unsigned long hash2 = 0x12345678;
  int c;
  while ((c = *str++)) {
    hash1 = (hash1 * 31) ^ c;
    hash2 = (hash2 * 33) ^ c;
    hash1 = ((hash1 << 5) + hash1) + c;
    hash2 = ((hash2 << 5) + hash2) + c;
    hash1 ^= (hash1 >> 13);
    hash2 ^= (hash2 >> 13);
    hash1 += (hash1 << 17);
    hash2 += (hash2 << 17);
  }
  sprintf(output, "%08lX%08lX", hash1 & 0xFFFFFFFF, hash2 & 0xFFFFFFFF);
}

void setup() {
  Serial.begin(9600);
  pinMode(button_pin, INPUT);
  driver.init();
  Serial.println("NANO LISTO");
  pinMode(ledg, OUTPUT);
```

```
  pinMode(ledr, OUTPUT);

  digitalWrite(ledg, HIGH);
  digitalWrite(ledr, LOW);
  for (int i = 0; i < 4; i++) {
    digitalWrite(ledg, LOW);
    digitalWrite(ledr, HIGH);
    delay(100);
    digitalWrite(ledg, HIGH);
    digitalWrite(ledr, LOW);
    delay(100);
  }
  digitalWrite(ledg, LOW);
  digitalWrite(ledr, LOW);

}

void loop() {
  // Enviar ID
  if (digitalRead(button_pin) == HIGH && !esperando_codigo) {
    Serial.println("Solicitando acceso...");
    driver.send((uint8_t*)id_llave, strlen(id_llave));
    driver.waitPacketSent();
    esperando_codigo = true;
    tiempo_inicio_espera = millis(); // Guardar el momento en que empezamos
a esperar

    digitalWrite(ledg, HIGH);
    for (int i = 0; i < 4; i++) {
      digitalWrite(ledg, LOW);
      delay(50);
      digitalWrite(ledg, HIGH);
      delay(50);
    }
    digitalWrite(ledg, LOW);
    delay(100);
  }

  // Verificar timeout
  if (esperando_codigo && (millis() - tiempo_inicio_espera) >
tiempo_espera_maximo) {
    Serial.println("Tiempo de espera agotado. Volviendo al estado
inicial.");
    esperando_codigo = false;

    digitalWrite(ledr, HIGH);
    for (int i = 0; i < 4; i++) {
      digitalWrite(ledr, LOW);
```

```
      delay(50);
      digitalWrite(ledr, HIGH);
      delay(50);
    }
    digitalWrite(ledr, LOW);

  }

  // Recibir código aleatorio
  uint8_t buf[RH_ASK_MAX_MESSAGE_LEN];
  uint8_t buflen = sizeof(buf);
  if (driver.recv(buf, &buflen) && esperando_codigo) {
    codigo_aleatorio = "";
    for (int i = 0; i < buflen; i++) {
      codigo_aleatorio += (char)buf[i];
    }
    Serial.print("Código recibido: ");
    Serial.println(codigo_aleatorio);

    // Generar hash
    String combinado = codigo_aleatorio + contrasena_local;
    char resumen[17];
    simpleMD5_16(combinado.c_str(), resumen);

    Serial.print("Hash generado: ");
    Serial.println(resumen);

    digitalWrite(ledg, HIGH);
    for (int i = 0; i < 4; i++) {
      digitalWrite(ledg, LOW);
      delay(50);
      digitalWrite(ledg, HIGH);
      delay(50);
    }
    digitalWrite(ledg, LOW);

    delay(100);

    // Enviar hash
    Serial.println("ENVIANDO HASH... ");
    driver.send((uint8_t*)resumen, strlen(resumen));
    driver.waitPacketSent();
    esperando_codigo = false;
  }
}
```

7.3 Anexo III - Código para la cerradura eléctrica

```
#include <Wire.h>              //LCD
#include <LiquidCrystal_I2C.h>//LCD
#include <SPI.h>               //LECTOR RFID - RADIOFRECUENCIA
#include <MFRC522.h>           //LECTOR RFID
#include <Keypad.h>            //TECLADO MATRICIAL
#include <WiFi.h>              //CONEXION A WIFI
#include <HTTPClient.h>        //CONSULTAS BDD
#include <ArduinoJson.h>       //CONSULTAS BDD
#include <RH_ASK.h>            //CONEXION DE RADIOFRECUENCIA

//----------------------METODOS----------------------

// Prototipos de función
void insertarAccesoResidencial(String metodo, String estado, String fecha,
String usuario_id = "");
void insertarRFIDRegistro(String refid_valor, String usuario_id);
bool validarClaveIngresada(String claveIngresada, String &usuario_id, String
&nombre_usuario);
bool validarRfid(String rfid, String &usuario_id, String &nombre_usuario);
bool validarClaveTemporal(String claveTemporal, String &usuario_id, String
&nombre_usuario);
void enviarGET(String url);
void enviarPOST(String postData);
String obtenerFechaHora();
String obtenerHoraActual(bool mostrarPuntos = true);
void menu();
void regresarAlMenu();
void triggerDoorUnlock();
void consultarTablaCompleta(String tabla);
void consultarColumna(String tabla, String columna);
void consultarEspecifico(String tabla, String columna, String id);
// Prototipos para radiofrecuencia
void simpleMD5_16(const char *str, char *output);
String generarCodigoAleatorio(int longitud);
String obtenerContraseñaDesdeAcceso(String id_acceso);
bool validarLlaveRF(String llave, String &usuario_id, String &id_acceso);
void accesoDenegadoRF();



//----------------------RFID-------------------------
// Configuración de pines para RFID RC522
#define SS_PIN 5
#define RST_PIN 15

MFRC522 mfrc522(SS_PIN, RST_PIN);
```

```
//----------------------Teclado-------------------------
// Configuración de pines para teclado matricial 4x4
const byte ROWS = 4;
const byte COLS = 4;
char keys[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'F', '0', 'E', 'D'}
};

// Definir pines del ESP32 conectados al teclado
byte rowPins[ROWS] = {32, 33, 25, 26}; // Pines de filas
byte colPins[COLS] = {27, 14, 12, 13}; // Pines de columnas

Keypad customKeypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS,
COLS);

String cifra;
int customKey;

int opcion = 0, presion = 3, contpass = 0;
int verpass[6];  //variable ingreso de contrasena



//----------------------LCD-------------------------
LiquidCrystal_I2C lcd(0x27, 16, 2);


// ---------------------- RADIOFRECUENCIA ----------------------
RH_ASK driver(2000, 35, 4); // Configuración del módulo RF (2000 baudios,
RX:35, TX:4)
String ultimo_codigo_enviado = "";
String contraseña_real = "";
bool esperando_hash = false;
unsigned long tiempo_inicio_espera = 0;
const unsigned long tiempo_espera_maximo = 3000; // 3 segundos para recibir
el hash


//---------------------CONEXION BDD--------------------

const char* ssid = "***********";
const char* password = "@@@@@@@@@@";
const char* serverUrl = "http://82.25.91.138/nuevo_datos.php";  //IP del
servidor
```

```
const String codigo = "JARB-1234";

//---------- Configuración del servidor NTP-----------
const char* ntpServer = "pool.ntp.org";
const long gmtOffset_sec = -18000;  // Ajuste para zona horaria (-5 horas
para Ecuador)
const int daylightOffset_sec = 0;


//----------------------LEDS-------------------------
const int doorPin = 2; //LED PUERTA
const int redLED = 16; // LED ROJO
const int greenLED = 17; // LED VERDE

int errores = 0;

const int lockTurnTime = 3000; //Segunos para mantener abirto la puerta


void setup() {

  //Inicializar Leds
  pinMode(doorPin, OUTPUT);
  pinMode(redLED, OUTPUT);
  pinMode(greenLED, OUTPUT);

  digitalWrite(greenLED, HIGH);
  digitalWrite(redLED, LOW);
  digitalWrite(doorPin, LOW);

  //iniciar comunicacion serial
  Serial.begin(115200);

  // Inicializar LCD
  lcd.init();
  lcd.backlight();

  Wire.begin();

  String punto = ".";

  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(250);
    lcd.setCursor(0, 0);
    lcd.print("   CONECTANDO   ");
    lcd.setCursor(0, 1);
    lcd.print(punto);
    punto = punto + ".";
```

```
    Serial.println("Conectando a WiFi...");
  }
  Serial.println("Conectado a WiFi");

  lcd.setCursor(0, 0);
  lcd.print("    CONECTADO   ");
  lcd.setCursor(0, 1);
  lcd.print("!!!!!!!!!!!!!!!!");


  // Configurar y sincronizar la hora con NTP
  configTime(gmtOffset_sec, daylightOffset_sec, ntpServer);
  Serial.println("Esperando sincronización NTP...");
  delay(2000);
  Serial.println(obtenerFechaHora());

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("   SMART DOOR   ");
  lcd.setCursor(4, 1);
  lcd.print(obtenerHoraActual());


  // Inicializar SPI y RFID
  SPI.begin();
  mfrc522.PCD_Init();

  // Inicializar comunicación RF
  if (!driver.init()) {
    Serial.println("Error al inicializar RF");
  } else {
    Serial.println("RF inicializado correctamente");
  }

  randomSeed(analogRead(0));
}

void loop() {
  //----------------------Leer teclado--------------------------
  customKey = customKeypad.getKey();  //almacena el dato ingresado por
teclado
  //Serial.println(customKey);

  static bool puntosVisibles = true;
  static unsigned long lastBlink = 0;

  //----------------------------MENU-----------------------------
  if (opcion == 0) {
    if (millis() - lastBlink >= 500) { // Cambia cada 500ms
```

```
      puntosVisibles = !puntosVisibles;
      lcd.setCursor(4, 1);
      lcd.print(obtenerHoraActual(puntosVisibles));
      lastBlink = millis();
    }
    menu();
  }

  //---------------------Abrir Puerta con telacdo-------------------------

  if (opcion == 1) {


    // VALIDAR TECLAS
    if (customKey == '0' || customKey == '1' || customKey == '2' ||
customKey == '3' || customKey == '4' ||
        customKey == '5' || customKey == '6' || customKey == '7' ||
customKey == '8' || customKey == '9') {
      customKey = customKey - 48;
      verpass[contpass] = customKey; //almacena el dato ingresado en cada
posicion
      contpass++; //aumenta contador
      presion++;

      lcd.setCursor(0, 0);
      lcd.print(" INGRESE PASS: ");
      lcd.setCursor(presion, 1);
      lcd.print('*');
    }


    if (customKey == 69 && contpass > 0) { // Tecla E (69) y hay caracteres
para borrar
      lcd.setCursor(presion, 1);
      lcd.print(" "); // Borra el asterisco
      lcd.setCursor(presion, 1); // Vuelve a posicionar el cursor
      contpass--; // Reduce el contador
      presion--; // Retrocede la posición del cursor
    }

    if (contpass == 6) {

      // Convertir la contraseña ingresada a String
      String claveIngresada = "";
      for (int i = 0; i < 6; i++) {
        claveIngresada += String(verpass[i]);
      }

      String usuario_id = "";
```

```
      String nombre_usuario = "";

      if (validarClaveIngresada(claveIngresada, usuario_id,
nombre_usuario))  {
        //lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("    CORRECTO    ");
        lcd.setCursor(0, 1);
        lcd.print("++++++++++++++++");

        delay(400);

        triggerDoorUnlock();

        // 📌 INSERTAR datos---
        String fechaHora = obtenerFechaHora();  // Obtener la hora actual
        insertarAccesoResidencial("TECLADO", "Permitido", fechaHora,
usuario_id);

        contpass = 0;
        presion = 3;
        opcion = 0;

      }
      else {
        errores = errores + 1;

        if (errores == 3) {
          lcd.clear();
          lcd.print("   INCORRECTO   ");
          lcd.setCursor(0, 1);
          lcd.print("XXXXXXXXXXXXXXXX");

          String fechaHora = obtenerFechaHora();  // Obtener la hora actual
          insertarAccesoResidencial("TECLADO", "Denegado", fechaHora,
"NULL");

          delay(1000);

          lcd.setCursor(0, 0);
          lcd.print("   SMART DOOR   ");
          lcd.setCursor(0, 1);
          lcd.print("      LOCK      ");

          opcion = 0;
          errores = 0;
          contpass = 0;
          presion = 3;
```

```
        digitalWrite(greenLED, LOW);
        for (int i = 0; i < 4; i++) {
          digitalWrite(redLED, HIGH);
          delay(100);
          digitalWrite(redLED, LOW);
          delay(100);
        }
        delay(200);
      }
      else {
        lcd.clear();
        lcd.print("   INCORRECTO   ");
        lcd.setCursor(0, 1);
        lcd.print("XXXXXXXXXXXXXXXX");

        delay(1000);

        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("  Intenta otra  ");
        lcd.setCursor(0, 1);
        lcd.print("      vez       ");
        delay(500);
        lcd.setCursor(0, 1);
        lcd.print("                ");
        opcion = 1;
        contpass = 0;
        presion = 3;
      }

      digitalWrite(greenLED, LOW);
      for (int i = 0; i < 4; i++) {
        digitalWrite(redLED, HIGH);
        delay(100);
        digitalWrite(redLED, LOW);
        delay(100);
      }
      digitalWrite(greenLED, HIGH);

    }
  }
  delay(100);

  //---------salir de 1-------------
  if (customKey == 70 && opcion == 1) {
    regresarAlMenu();
  }
```

```
  }


  //----------------------Registrar RFID-------------------------
  if (opcion == 2) {

    // VALIDAR TECLAS
    if (customKey == '0' || customKey == '1' || customKey == '2' ||
customKey == '3' || customKey == '4' ||
        customKey == '5' || customKey == '6' || customKey == '7' ||
customKey == '8' || customKey == '9') {
      customKey = customKey - 48;
      verpass[contpass] = customKey; //almacena el dato ingresado en cada
posicion
      contpass++; //aumenta contador
      presion++;

      lcd.setCursor(0, 0);
      lcd.print(" INGRESE PASS: ");
      lcd.setCursor(presion, 1);
      lcd.print('*');
    }
    if (customKey == 69 && contpass > 0) { // Tecla E (69) y hay caracteres
para borrar
      lcd.setCursor(presion, 1);
      lcd.print(" "); // Borra el asterisco
      lcd.setCursor(presion, 1); // Vuelve a posicionar el cursor
      contpass--; // Reduce el contador
      presion--; // Retrocede la posición del cursor
    }

    if (contpass == 6) {

      // Convertir la contraseña ingresada a String
      String claveIngresada = "";
      for (int i = 0; i < 6; i++) {
        claveIngresada += String(verpass[i]);
      }
      String usuario_id = "";
      String nombre_usuario = "";
      if (validarClaveIngresada(claveIngresada, usuario_id,
nombre_usuario))  {
        //lcd.clear();
        //        lcd.setCursor(0, 0);
        //        lcd.print("    CORRECTO    ");
        //        lcd.setCursor(0, 1);
        //        lcd.print("++++++++++++++++");
```

```
        lcd.setCursor(0, 0);
        lcd.print(" ACERQUE UN RFID ");
        lcd.setCursor(0, 1);
        lcd.print("  PARA REGISTRAR ");

        delay(200);
        // Leer RFID
        if (mfrc522.PICC_IsNewCardPresent() &&
mfrc522.PICC_ReadCardSerial()) {
          // Convertir el UID en una cadena sin espacios y en minúsculas
          String uidString = "";
          for (byte i = 0; i < mfrc522.uid.size; i++) {
            uidString += String(mfrc522.uid.uidByte[i], HEX); // Convertir a
hexadecimal
          }
          uidString.toLowerCase(); // Asegurar formato minúscula

          Serial.print("UID registrado: ");
          Serial.println(uidString);

          // Enviar el RFID junto con el ID del usuario
          insertarRFIDRegistro(uidString, usuario_id);
          lcd.clear();
          lcd.setCursor(0, 0);
          lcd.print(" RFID REGISTRADO ");
          lcd.setCursor(0, 1);
          lcd.print("  EXITOSAMENTE  ");
          delay(1000);
          regresarAlMenu();

        }

      }
      else {
        errores = errores + 1;

        if (errores == 3) {
          lcd.clear();
          lcd.print("   INCORRECTO   ");
          lcd.setCursor(0, 1);
          lcd.print("XXXXXXXXXXXXXXXX");

          delay(1000);

          lcd.setCursor(0, 0);
          lcd.print("   SMART DOOR   ");
          lcd.setCursor(0, 1);
```

```
          lcd.print("      LOCK      ");

          opcion = 0;
          errores = 0;
          contpass = 0;
          presion = 3;


          digitalWrite(greenLED, LOW);
          for (int i = 0; i < 4; i++) {
            digitalWrite(redLED, HIGH);
            delay(100);
            digitalWrite(redLED, LOW);
            delay(100);
          }
          delay(200);
        }
        else {
          lcd.clear();
          lcd.print("   INCORRECTO   ");
          lcd.setCursor(0, 1);
          lcd.print("XXXXXXXXXXXXXXXX");

          delay(1000);

          lcd.clear();
          lcd.setCursor(0, 0);
          lcd.print("  Intenta otra  ");
          lcd.setCursor(0, 1);
          lcd.print("      vez       ");
          delay(500);
          lcd.setCursor(0, 1);
          lcd.print("                ");
          opcion = 2;
          contpass = 0;
          presion = 3;
        }

        digitalWrite(greenLED, LOW);
        for (int i = 0; i < 4; i++) {
          digitalWrite(redLED, HIGH);
          delay(100);
          digitalWrite(redLED, LOW);
          delay(100);
        }
        digitalWrite(greenLED, HIGH);

      }
    }
```

```
    //---------salir de 2-------------
    if (customKey == 70 && opcion == 2) {
      regresarAlMenu();
    }
  }


  //---------------------- CLAVE TEMPORAL ---------------------------------
  // Opción 3: Clave temporal
  if (opcion == 3) {
    // VALIDAR TECLAS
    if (customKey == '0' || customKey == '1' || customKey == '2' ||
customKey == '3' || customKey == '4' ||
        customKey == '5' || customKey == '6' || customKey == '7' ||
customKey == '8' || customKey == '9') {
      customKey = customKey - 48;
      verpass[contpass] = customKey; // Almacena el dato ingresado en cada
posición
      contpass++; // Aumenta contador
      presion++;

      lcd.setCursor(0, 0);
      lcd.print(" INGRESE CLAVE: ");
      lcd.setCursor(presion, 1);
      lcd.print('*');
    }
    if (customKey == 69 && contpass > 0) { // Tecla E (69) y hay caracteres
para borrar
      lcd.setCursor(presion, 1);
      lcd.print(" "); // Borra el asterisco
      lcd.setCursor(presion, 1); // Vuelve a posicionar el cursor
      contpass--; // Reduce el contador
      presion--; // Retrocede la posición del cursor
    }

    if (contpass == 6) {
      // Convertir la clave ingresada a String
      String claveTemporal = "";
      for (int i = 0; i < 6; i++) {
        claveTemporal += String(verpass[i]);
      }
      String usuario_id = "";
      String nombre_usuario = "";
      // Validar la clave temporal
      if (validarClaveTemporal(claveTemporal, usuario_id, nombre_usuario)) {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("    CORRECTO    ");
```

```
        lcd.setCursor(0, 1);
        lcd.print("++++++++++++++++");

        delay(400);

        triggerDoorUnlock();

        // INSERTAR datos---
        String fechaHora = obtenerFechaHora();  // Obtener la hora actual
        insertarAccesoResidencial("CLAVE_TEMPORAL", "Permitido", fechaHora,
usuario_id);

        contpass = 0;
        presion = 3;
        opcion = 0;
      } else {
        errores = errores + 1;

        if (errores == 3) {
          lcd.clear();
          lcd.print("   INCORRECTO   ");
          lcd.setCursor(0, 1);
          lcd.print("XXXXXXXXXXXXXXXX");

          delay(1000);
          String fechaHora = obtenerFechaHora();  // Obtener la hora actual
          insertarAccesoResidencial("CLAVE_TEMPORAL", "Denegado", fechaHora,
"NULL");

          lcd.setCursor(0, 0);
          lcd.print("   SMART DOOR   ");
          lcd.setCursor(0, 1);
          lcd.print("      LOCK      ");

          opcion = 0;
          errores = 0;
          contpass = 0;
          presion = 3;


          digitalWrite(greenLED, LOW);
          for (int i = 0; i < 4; i++) {
            digitalWrite(redLED, HIGH);
            delay(100);
            digitalWrite(redLED, LOW);
            delay(100);
          }
          delay(200);
        }
```

```
        else {
          lcd.clear();
          lcd.print("   INCORRECTO   ");
          lcd.setCursor(0, 1);
          lcd.print("XXXXXXXXXXXXXXXX");

          delay(1000);

          lcd.clear();
          lcd.setCursor(0, 0);
          lcd.print("  Intenta otra  ");
          lcd.setCursor(0, 1);
          lcd.print("      vez       ");
          delay(500);
          lcd.setCursor(0, 1);
          lcd.print("                ");
          opcion = 3;
          contpass = 0;
          presion = 3;
        }

        digitalWrite(greenLED, LOW);
        for (int i = 0; i < 4; i++) {
          digitalWrite(redLED, HIGH);
          delay(100);
          digitalWrite(redLED, LOW);
          delay(100);
        }
        digitalWrite(greenLED, HIGH);



      }
    }

    // Regresar al menú con la tecla *
    if (customKey == 70) {
      regresarAlMenu();
    }
  }
  //---------------------- REGISTRO LLAVE ELECTRÓNICA  ----------------------
------------

  if (opcion == 4) {

    // VALIDAR TECLAS
    if (customKey == '0' || customKey == '1' || customKey == '2' ||
customKey == '3' || customKey == '4' ||
```

```
        customKey == '5' || customKey == '6' || customKey == '7' ||
customKey == '8' || customKey == '9') {
      customKey = customKey - 48;
      verpass[contpass] = customKey; //almacena el dato ingresado en cada
posicion
      contpass++; //aumenta contador
      presion++;

      lcd.setCursor(0, 0);
      lcd.print(" INGRESE PASS: ");
      lcd.setCursor(presion, 1);
      lcd.print('*');
    }
    if (customKey == 69 && contpass > 0) { // Tecla E (69) y hay caracteres
para borrar
      lcd.setCursor(presion, 1);
      lcd.print(" "); // Borra el asterisco
      lcd.setCursor(presion, 1); // Vuelve a posicionar el cursor
      contpass--; // Reduce el contador
      presion--; // Retrocede la posición del cursor
    }

    if (contpass == 6) {

      // Convertir la contraseña ingresada a String
      String claveIngresada = "";
      for (int i = 0; i < 6; i++) {
        claveIngresada += String(verpass[i]);
      }
      String usuario_id = "";
      String nombre_usuario = "";
      if (validarClaveIngresada(claveIngresada, usuario_id,
nombre_usuario))  {
        //lcd.clear();
        //        lcd.setCursor(0, 0);
        //        lcd.print("    CORRECTO    ");
        //        lcd.setCursor(0, 1);
        //        lcd.print("++++++++++++++++");

        lcd.setCursor(0, 0);
        lcd.print(" PRESIONE BOTON ");
        lcd.setCursor(0, 1);
        lcd.print("   DE LA LLAVE  ");

        delay(200);
        // Leer LLAVE-------------------------------

        // Enviar el RFID junto con el ID del usuario
```

```
        delay(2000);

        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print(" LLAVE GUARDADA ");
        lcd.setCursor(0, 1);
        lcd.print("  EXITOSAMENTE  ");
        delay(1000);
        regresarAlMenu();

      }
      else {
        errores = errores + 1;

        if (errores == 3) {
          lcd.clear();
          lcd.print("   INCORRECTO   ");
          lcd.setCursor(0, 1);
          lcd.print("XXXXXXXXXXXXXXXX");

          delay(1000);

          lcd.setCursor(0, 0);
          lcd.print("   SMART DOOR   ");
          lcd.setCursor(0, 1);
          lcd.print("      LOCK      ");

          opcion = 0;
          errores = 0;
          contpass = 0;
          presion = 3;


          digitalWrite(greenLED, LOW);
          for (int i = 0; i < 4; i++) {
            digitalWrite(redLED, HIGH);
            delay(100);
            digitalWrite(redLED, LOW);
            delay(100);
          }
          delay(200);
        }
        else {
          lcd.clear();
          lcd.print("   INCORRECTO   ");
          lcd.setCursor(0, 1);
          lcd.print("XXXXXXXXXXXXXXXX");

          delay(1000);
```

```
          lcd.clear();
          lcd.setCursor(0, 0);
          lcd.print("  Intenta otra  ");
          lcd.setCursor(0, 1);
          lcd.print("      vez       ");
          delay(500);
          lcd.setCursor(0, 1);
          lcd.print("                ");
          opcion = 4;
          contpass = 0;
          presion = 3;
        }

        digitalWrite(greenLED, LOW);
        for (int i = 0; i < 4; i++) {
          digitalWrite(redLED, HIGH);
          delay(100);
          digitalWrite(redLED, LOW);
          delay(100);
        }
        digitalWrite(greenLED, HIGH);

      }
    }

    //---------salir de 4-------------
    if (customKey == 70 && opcion == 4) {
      regresarAlMenu();
    }

  }









  //------------------------ Leer tarjeta RFID-----------------------------
--

  // Leer tarjeta RFID (solo si no está en modo de registro)
  if (opcion != 2 && (mfrc522.PICC_IsNewCardPresent() &&
mfrc522.PICC_ReadCardSerial())) {
    // Convertir el UID en una cadena sin espacios y en minúsculas
    String uidString = "";
    for (byte i = 0; i < mfrc522.uid.size; i++) {
      uidString += String(mfrc522.uid.uidByte[i], HEX); // Convertir a
hexadecimal
```

```
    }
    uidString.toLowerCase(); // Asegurar formato minúscula

    String usuario_id = "";
    String nombre_usuario = "";

    Serial.print("UID: ");
    Serial.println(uidString);

    // Validar el RFID escaneado
    if (validarRfid(uidString, usuario_id, nombre_usuario)) {
      Serial.println("Acceso permitido");
      triggerDoorUnlock();
      String fechaHora = obtenerFechaHora();  // Obtener la hora actual
      insertarAccesoResidencial("RFID", "Permitido", fechaHora, usuario_id);
    } else {
      Serial.println("Acceso denegado");
      String fechaHora = obtenerFechaHora();  // Obtener la hora actual
      insertarAccesoResidencial("RFID", "Denegado", fechaHora, "NULL");
      for (int i = 0; i < 4; i++) {
        digitalWrite(redLED, HIGH);
        delay(100);
        digitalWrite(redLED, LOW);
        delay(100);
      }
    }

    delay(1000); // Esperar antes de leer otra tarjeta
  }


  //------------------------ Leer radiofrecuencia ------------------------
------
  if (esperando_hash && (millis() - tiempo_inicio_espera) >
tiempo_espera_maximo) {
    Serial.println("Tiempo de espera agotado para HASH");
    esperando_hash = false;
  }
  // Leer mensajes RF entrantes
  uint8_t buf[RH_ASK_MAX_MESSAGE_LEN];
  uint8_t buflen = sizeof(buf);

  // Declarar las variables fuera del bloque condicional
  static String usuario_id = "";
  static String id_acceso = "";

  if (driver.recv(buf, &buflen) && opcion != 4) { // Solo procesar RF si no
estamos en modo registro
    String recibido = "";
```

```
    for (int i = 0; i < buflen; i++) recibido += (char)buf[i];

    Serial.print("RF recibido: ");
    Serial.println(recibido);

    if (!esperando_hash) {
      // Fase 1: Recibimos el ID de la llave
      // String usuario_id, id_acceso;
      if (validarLlaveRF(recibido, usuario_id, id_acceso)) {
        contraseña_real = obtenerContraseñaDesdeAcceso(id_acceso);
        Serial.println("Contraseña Obtenida del servidor:" +
String(contraseña_real));
        if (contraseña_real != "") {
          ultimo_codigo_enviado = generarCodigoAleatorio(10);

          // Enviar el código de desafío
          driver.send((uint8_t*)ultimo_codigo_enviado.c_str(),
ultimo_codigo_enviado.length());
          driver.waitPacketSent();

          esperando_hash = true;
          tiempo_inicio_espera = millis();

          Serial.println("Desafío enviado: " + ultimo_codigo_enviado);
        } else {
          Serial.println("No se encontró contraseña asociada");
          accesoDenegadoRF();
        }
      } else {
        Serial.println("Llave RF no válida");
        accesoDenegadoRF();
      }
    } else {
      // Fase 2: Recibimos el hash de respuesta
      char esperado[17];
      String combinado = ultimo_codigo_enviado + contraseña_real;
      simpleMD5_16(combinado.c_str(), esperado);

      Serial.print("Hash esperado: ");
      Serial.println(esperado);
      Serial.print("Hash recibido: ");
      Serial.println(recibido);

      if (recibido.equals(String(esperado))) {
        Serial.println("Acceso permitido por RF");

        // No necesitamos validar nuevamente, ya tenemos los datos de la
primera validación
```

```
        triggerDoorUnlock();
        String fechaHora = obtenerFechaHora();
        insertarAccesoResidencial("LLAVE", "Permitido", fechaHora,
usuario_id);
      } else {
        Serial.println("Acceso denegado por RF (hash inválido)");
        accesoDenegadoRF();
      }

      esperando_hash = false;
    }
  }




  //-------------FIN loop-------------------------
}


void menu() {
  switch (customKey) {
    case 65:
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("  ABRIR PUERTA  ");
      lcd.setCursor(0, 1);
      lcd.print("  USANDO PASS   ");

      delay(1500);

      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print(" INGRESE PASS: ");

      opcion = 1;
      break;

    case 66:
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("    REGISTRAR   ");
      lcd.setCursor(0, 1);
      lcd.print("   NUEVO RFID   ");

      delay(1500);
```

```
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print(" INGRESE PASS: ");



      opcion = 2;
      break;

    case 67:
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print(" CLAVE TEMPORAL ");
      lcd.setCursor(0, 1);
      lcd.print(" -------------- ");

      delay(1500);

      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print(" INGRESE PASS: ");

      opcion = 3;
      break;

    case 68:
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("    REGISTRAR   ");
      lcd.setCursor(0, 1);
      lcd.print("  NUEVA LLAVE   ");

      delay(1500);

      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print(" INGRESE PASS: ");

      opcion = 4;

      break;

    default:
      opcion = 0;
      break;
  }
}
```

```
void regresarAlMenu() {
  opcion = 0;
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(" Regresar... al ");
  lcd.setCursor(0, 1);
  lcd.print("      MENU      ");
  delay(500);
  // Mostrar el menú principal con la hora
  lcd.setCursor(0, 0);
  lcd.print("   SMART DOOR   ");
  lcd.setCursor(4, 1);
  lcd.print(obtenerHoraActual());
  errores = 0;
  contpass = 0;
  presion = 3;
}



void triggerDoorUnlock() {

  lcd.setCursor(0, 0);
  lcd.print(" Puerta Abierta ");
  lcd.setCursor(0, 1);
  lcd.print("                ");

  delay(1000);
  errores = 0;

  lcd.setCursor(0, 0);
  lcd.print("   SMART DOOR   ");
  lcd.setCursor(0, 1);
  lcd.print("      LOCK      ");


  int i = 0;

  digitalWrite(doorPin, HIGH);
  digitalWrite(greenLED, HIGH);

  delay (lockTurnTime);

  digitalWrite(doorPin, LOW);
  lcd.setCursor(0, 0);
  lcd.print(" Puerta Cerrada ");

  delay(1000);
```

```
  lcd.setCursor(0, 0);
  lcd.print("   SMART DOOR   ");
  lcd.setCursor(0, 1);
  lcd.print("      LOCK      ");

  opcion = 0;

  for (i = 0; i < 5; i++) {
    digitalWrite(greenLED, LOW);
    delay(100);
    digitalWrite(greenLED, HIGH);
    delay(100);
  }

}

//------------------CONSULTAS BASE DE DATOS------------------

//  ◇  FUNCIONES PARA CONSULTAR DATOS  ◇
void consultarTablaCompleta(String tabla) {
  String url = String(serverUrl) + "?tabla=" + tabla;
  enviarGET(url);
}

void consultarColumna(String tabla, String columna) {
  String url = String(serverUrl) + "?tabla=" + tabla + "&columna=" +
columna;
  enviarGET(url);
}

void consultarEspecifico(String tabla, String columna, String id) {
  String url = String(serverUrl) + "?tabla=" + tabla + "&columna=" + columna
+ "&id=" + id;
  enviarGET(url);
}


//------------------- Función para validar la clave ingresada---------------
-------
bool validarClaveIngresada(String claveIngresada, String &usuario_id, String
&nombre_usuario) {
  String url = String(serverUrl) + "?tabla=accesos&columna=codigo&valor=" +
claveIngresada + "&casa=" + codigo;
  String response = "";

  if (WiFi.status() == WL_CONNECTED) {
    HTTPClient http;
    int intentos = 0;
    int httpResponseCode = 0;
```

```
    while (intentos < 3) {
      http.begin(url);
      httpResponseCode = http.GET();

      if (httpResponseCode == 200) {
        response = http.getString();
        Serial.println("Respuesta del servidor:");
        Serial.println(response);

        DynamicJsonDocument doc(1024);
        deserializeJson(doc, response);

        if (doc.size() > 0 && doc[0]["valido"] == true) {
          usuario_id = doc[0]["usuario_id"].as<String>();
          nombre_usuario = doc[0]["nombre"].as<String>();
          http.end();
          return true;
        }

        http.end();
        return false;
      } else {
        Serial.print("Error en GET (Intento ");
        Serial.print(intentos + 1);
        Serial.print("): ");
        Serial.println(httpResponseCode);
      }

      http.end();
      intentos++;
      delay(1000);
    }

    Serial.println("Error persistente en GET. No se pudo validar la
clave.");
    return false;
  } else {
    Serial.println("No conectado a WiFi");
    return false;
  }
}

//------------------- Función para validar RFID----------------------
bool validarRfid(String rfid, String &usuario_id, String &nombre_usuario) {
  String url = String(serverUrl) + "?tabla=accesos&tipo=rfid&valor=" + rfid
+ "&casa=" + codigo;
  String response = "";
```

```
  if (WiFi.status() == WL_CONNECTED) {
    HTTPClient http;
    int intentos = 0;
    int httpResponseCode = 0;

    while (intentos < 3) {
      http.begin(url);
      httpResponseCode = http.GET();

      if (httpResponseCode == 200) {
        response = http.getString();
        Serial.println("Respuesta del servidor:");
        Serial.println(response);

        DynamicJsonDocument doc(1024);
        deserializeJson(doc, response);

        if (doc.size() > 0 && doc[0]["valido"] == true) {
          usuario_id = doc[0]["usuario_id"].as<String>();
          nombre_usuario = doc[0]["nombre"].as<String>();
          http.end();
          return true;
        }

        http.end();
        return false;
      } else {
        Serial.print("Error en GET (Intento ");
        Serial.print(intentos + 1);
        Serial.print("): ");
        Serial.println(httpResponseCode);
      }

      http.end();
      intentos++;
      delay(1000);
    }

    Serial.println("Error persistente en GET. No se pudo validar el RFID.");
    return false;
  } else {
    Serial.println("No conectado a WiFi");
    return false;
  }
}

//----------------------- FUNCION PARA CLAVE TEMPORAL-----------------
```

```
bool validarClaveTemporal(String claveTemporal, String &usuario_id, String
&nombre_usuario) {
  String url = String(serverUrl) + "?tabla=claves_temporales&valor=" +
claveTemporal + "&casa=" + codigo;
  Serial.println("URL de consulta: " + url); // Depuración

  String response = "";

  if (WiFi.status() == WL_CONNECTED) {
    HTTPClient http;
    int intentos = 0;
    int httpResponseCode = 0;

    while (intentos < 3) {
      http.begin(url);
      httpResponseCode = http.GET();

      if (httpResponseCode == 200) {
        response = http.getString();
        Serial.println("Respuesta del servidor:");
        Serial.println(response);

        DynamicJsonDocument doc(1024);
        deserializeJson(doc, response);

        if (doc.size() > 0 && doc[0]["valido"] == true) {
          usuario_id = doc[0]["usuario_id"].as<String>();
          nombre_usuario = doc[0]["nombre"].as<String>();
          http.end();
          return true;
        }

        http.end();
        return false;
      } else {
        Serial.print("Error en GET (Intento ");
        Serial.print(intentos + 1);
        Serial.print("): ");
        Serial.println(httpResponseCode);
        // Mostrar respuesta de error si existe
        if (httpResponseCode >= 400) {
          String errorResponse = http.getString();
          Serial.println("Detalles del error:");
          Serial.println(errorResponse);
        }
      }
```

```
      http.end();
      intentos++;
      delay(1000);
    }

    Serial.println("Error persistente en GET. No se pudo validar la clave
temporal.");
    return false;
  } else {
    Serial.println("No conectado a WiFi");
    return false;
  }
}

void enviarGET(String url) {
  if (WiFi.status() == WL_CONNECTED) {
    HTTPClient http;
    int intentos = 0;
    int httpResponseCode = 0;

    while (intentos < 3) {  // Intentar hasta 3 veces
      http.begin(url);
      httpResponseCode = http.GET();

      if (httpResponseCode == 200) {
        String response = http.getString();
        Serial.println("Respuesta del servidor:");
        Serial.println(response);
        http.end();
        return;  // Salir de la función si la solicitud fue exitosa
      } else {
        Serial.print("Error en GET (Intento ");
        Serial.print(intentos + 1);
        Serial.print("): ");
        Serial.println(httpResponseCode);
      }

      http.end();
      intentos++;
      delay(2000);  // Esperar 2 segundos antes de reintentar
    }

    Serial.println("Error 404 persistente. No se pudo recuperar la
información.");
  } else {
    Serial.println("No conectado a WiFi");
  }
}
```

```
//------------------------------------------------------------------------
-------


void insertarAccesoResidencial(String metodo, String estado, String fecha,
String usuario_id) {
  // Solo enviar si no es un duplicado reciente
  static String lastMethod = "";
  static String lastEstado = "";
  static String lastUsuario = "";
  static unsigned long lastTime = 0;

  unsigned long currentTime = millis();

  // Verificar si es un duplicado (mismo método, estado y usuario en menos
de 5 segundos)
  if (metodo == lastMethod && estado == lastEstado && usuario_id ==
lastUsuario &&
      (currentTime - lastTime) < 5000) {
    Serial.println("Notificación duplicada, no se enviará");
    return;
  }

  // Actualizar últimos valores
  lastMethod = metodo;
  lastEstado = estado;
  lastUsuario = usuario_id;
  lastTime = currentTime;

  String postData = "tabla=notificaciones&metodo=" + metodo + "&estado=" +
estado + "&fecha=" + fecha + "&usuario_id=" + usuario_id + "&casa=" +
codigo;
  enviarPOST(postData);
}




void insertarRFIDRegistro(String refid_valor, String usuario_id) {
  String postData = "tabla=rfid_registro&refid_valor=" + refid_valor +
"&usuario_id=" + usuario_id + "&casa=" + codigo;
  enviarPOST(postData);
}



//VERSION 1
```

```
void enviarPOST(String postData) {
  if (WiFi.status() == WL_CONNECTED) {
    HTTPClient http;
    http.begin(serverUrl);
    http.addHeader("Content-Type", "application/x-www-form-urlencoded");

    // Configurar timeout (3 segundos máximo)
    http.setTimeout(3000);

    int httpResponseCode = http.POST(postData);

    // No esperar respuesta larga, solo verificar si se inició
    if (httpResponseCode > 0) {
      Serial.print("POST iniciado, código: ");
      Serial.println(httpResponseCode);
    } else {
      Serial.print("Error en POST: ");
      Serial.println(httpResponseCode);
    }

    // No procesar la respuesta para no demorar
    http.end();
  } else {
    Serial.println("No conectado a WiFi");
  }
}


// Función para generar hash MD5 simplificado (igual que en tu código)
void simpleMD5_16(const char *str, char *output) {
  unsigned long hash1 = 475381;
  unsigned long hash2 = 0x12345678;
  int c;
  while ((c = *str++)) {
    hash1 = (hash1 * 31) ^ c;
    hash2 = (hash2 * 33) ^ c;
    hash1 = ((hash1 << 5) + hash1) + c;
    hash2 = ((hash2 << 5) + hash2) + c;
    hash1 ^= (hash1 >> 13);
    hash2 ^= (hash2 >> 13);
    hash1 += (hash1 << 17);
    hash2 += (hash2 << 17);
  }
  sprintf(output, "%08lX%08lX", hash1 & 0xFFFFFFFF, hash2 & 0xFFFFFFFF);
}

// Función para generar código aleatorio
String generarCodigoAleatorio(int longitud) {
```

```
  const char caracteres[] = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";
  String codigo = "";
  for (int i = 0; i < longitud; i++) {
    int index = random(0, sizeof(caracteres) - 1);
    codigo += caracteres[index];
  }
  return codigo;
}

// Función para obtener contraseña desde la base de datos
String obtenerContraseñaDesdeAcceso(String id_acceso) {
  if (WiFi.status() != WL_CONNECTED) return "";

  String url = "http://82.25.91.138/get_password.php?id_acceso=" +
id_acceso;
  HTTPClient http;
  http.begin(url);
  int res = http.GET();
  if (res == 200) {
    DynamicJsonDocument doc(512);
    deserializeJson(doc, http.getString());
    if (doc["contraseña"]) {
      http.end();
      return doc["contraseña"].as<String>();
    }
  }
  http.end();
  return "";
}

// Función para validar la llave RF ORIGINAL
bool validarLlaveRF(String llave, String &usuario_id, String &id_acceso) {
  static String last_valid_usuario_id = "";
  static String last_valid_id_acceso = "";

  String url = String(serverUrl) + "?tabla=accesos&tipo=llave&valor=" +
llave + "&casa=" + codigo;
  String response = "";

  if (WiFi.status() == WL_CONNECTED) {
    HTTPClient http;
    int intentos = 0;
    int httpResponseCode = 0;

    while (intentos < 3) {
      http.begin(url);
      httpResponseCode = http.GET();
```

```
      if (httpResponseCode == 200) {
        response = http.getString();
        Serial.println("Respuesta del servidor (RF):");
        Serial.println(response);

        DynamicJsonDocument doc(1024);
        deserializeJson(doc, response);

        if (doc.size() > 0 && doc[0]["valido"] == true) {
          last_valid_usuario_id = doc[0]["usuario_id"].as<String>();
          last_valid_id_acceso = doc[0]["id_acceso"].as<String>();
          usuario_id = last_valid_usuario_id;
          id_acceso = last_valid_id_acceso;
          http.end();
          return true;
        }
      }
      http.end();
      intentos++;
      delay(1000);
    }
  }

  // Si falla la validación, usar los últimos valores válidos
  usuario_id = last_valid_usuario_id;
  id_acceso = last_valid_id_acceso;
  return false;
}




// Función para manejar acceso denegado por RF
void accesoDenegadoRF() {
  String fechaHora = obtenerFechaHora();
  insertarAccesoResidencial("LLAVE", "Denegado", fechaHora, "NULL");

  lcd.clear();
  lcd.print(" ACCESO RF DENEG ");
  delay(2000);
  regresarAlMenu();

  for (int i = 0; i < 4; i++) {
    digitalWrite(redLED, HIGH);
    delay(100);
    digitalWrite(redLED, LOW);
    delay(100);
```

```
  }
}


//  ◇  FUNCIÓN PARA OBTENER FECHA Y HORA ACTUAL  ◇
String obtenerFechaHora() {
  struct tm timeinfo;
  if (!getLocalTime(&timeinfo)) {
    Serial.println("Error al obtener la hora");
    return "0000-00-00 00:00:00";  // Retorna un valor por defecto en caso
de error
  }

  char fechaHora[20];  // Formato YYYY-MM-DD HH:MM:SS
  strftime(fechaHora, sizeof(fechaHora), "%Y-%m-%d %H:%M:%S", &timeinfo);
  return String(fechaHora);
}

String obtenerHoraActual(bool mostrarPuntos) {
  struct tm timeinfo;
  if (!getLocalTime(&timeinfo)) {
    Serial.println("Error al obtener la hora");
    return "00:00:00";
  }

  char hora[9];
  if (mostrarPuntos) {
    strftime(hora, sizeof(hora), "%H:%M:%S", &timeinfo);
  } else {
    strftime(hora, sizeof(hora), "%H %M %S", &timeinfo);
  }


  return String(hora);
}
```