Universidad de las Ciencias Informáticas
Facultad de Tecnologías Interactivas

Sistema para el control de acceso mediante plataforma Arduino.

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor: Diacnis del Rosario Ramos Jerez

Tutor: Ing. Julio Alberto Leyva Durán.

La Habana, 2024

Security is not a product, but a process
Bruce Schneier

Dedicatoria

A mis queridos abuelos, Roselia Quesada y Antonio Jerez Guevara, por su amor incondicional y por ser la fuente de inspiración en cada paso que doy. A mi tía, Elita Ramos García, quien desde donde esté, siempre estará en mi corazón y en mis pensamientos; su amor y apoyo han dejado una huella imborrable en mi vida. Y a mi maestra, Mirna, por su dedicación y por inculcar en mí el valor del conocimiento y la perseverancia. Esta tesis es un reflejo de todo lo que he aprendido de ustedes.

Agradecimientos

Quiero expresar mi más sincero agradecimiento a todas las personas que han sido parte fundamental en la realización de esta tesis.

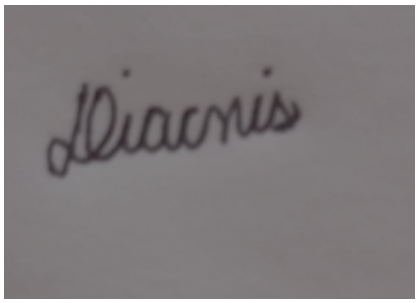
En primer lugar, agradezco a mis maestros y profesores de la universidad, quienes han compartido su conocimiento y sabiduría a lo largo de mi formación académica. Su dedicación y pasión por la enseñanza han sido una fuente constante de inspiración. Gracias por sus valiosos consejos, su apoyo incondicional y por motivarme a superar mis límites. Cada uno de ustedes ha dejado una huella imborrable en mi vida.

A mis padres, les debo un agradecimiento especial. Mamá y papá, su amor, apoyo y sacrificio han sido el pilar sobre el cual he construido mis sueños. Gracias por creer en mí incluso en los momentos más difíciles y por enseñarme la importancia del esfuerzo y la perseverancia. Sin su aliento constante y su confianza en mis capacidades, este logro no habría sido posible.

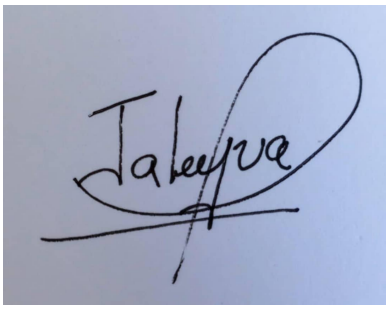
Declaración de autoría

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales sobre esta, con carácter exclusivo.

Para que así conste firmamos la presente a los 03 días del mes de diciembre del año 2024.



Diacnis del Rosario Ramos Jerez
Autor



Julio Alberto Leyva Durán
Tutor

Resumen

Los controles de acceso en áreas fiscas son sistemas de mucha importancia en cualquier organización, puesto que es el sistema encargado de restringir de forma eficaz el acceso a las zonas restringidas de cualquier entidad pública o privada. La investigación se enfoca en el diseño de un modelo de control de acceso para áreas físicas que integra tecnologías RFID y reconocimiento de huella dactilar, junto con un software de monitoreo desarrollado en plataformas Arduino y Raspberry Pi. Esta propuesta surge ante la creciente necesidad de mejorar la seguridad en espacios restringidos, donde los sistemas tradicionales han mostrado vulnerabilidades. El objetivo es implementar un sistema robusto que no solo restrinja el acceso a personas no autorizadas, sino que también facilite el monitoreo eficiente de entradas y salidas. La metodología utilizada es la programación extrema (XP), que permite una adaptación continua del sistema a las necesidades del usuario. Entre los hallazgos más relevantes, se destaca que la combinación de RFID con biometría proporciona un acceso más seguro y rápido, minimizando el riesgo de suplantación de identidad. Los resultados indican que el sistema propuesto mejora significativamente la seguridad operativa y permite una gestión más eficaz del acceso a áreas críticas. En conclusión, este modelo no solo fortalece la protección de las instalaciones, sino que también optimiza los procesos operativos, ofreciendo una solución integral para organizaciones que buscan salvaguardar sus activos y personal.

Palabras clave: RFID, Rasberry pi, Arduino, XP, Seguridad..

Índice general

Índice de figuras

Índice de tablas

Introducción

En la actualidad la sociedad está viviendo una época de grandes avances tecnológicos. Uno de los elementos más relevantes de este fenómeno tiene que ver con su carácter global, pues el progreso tecnológico no se limita a países del primer mundo, sino que se extiende a países emergentes.

En este marco se han podido identificar experiencias innovadoras y relevantes que suponen la utilización de las Tecnologías de la Información y las Comunicaciones (TIC) para la resolución de problemas o necesidades sociales de poblaciones de bajos recursos. Estas incluyen la electrónica como la tecnología base que soporta el desarrollo de las telecomunicaciones, la informática y el audiovisual (Muñoz, 2015).

A lo largo de todo este desarrollo tecnológico que está viviendo la sociedad, la búsqueda de un eficiente manejo de la información ha sido objetivo primordial. Teniendo en cuenta que la forma de acceder a la información, servicios y aplicaciones, se extiende cada vez más y también aumentan los riesgos de amenazas para los usuarios; surge la necesidad de crear sistemas de control de acceso que brinden técnicas seguras para limitar el libre acceso del público en general a la información, lugar o recurso que deban ser protegidos. El control de acceso es la parte principal de la pirámide de criticidad de sistemas de seguridad. Así que la elección de una solución fiable es crítica. El uso de tecnologías para ello ha crecido enormemente en los últimos años. Sin embargo, existen en el mercado muchas soluciones que no cumplen con los requisitos mínimos de seguridad, confiabilidad y estabilidad que estos sistemas requieren (Silva, 2016).

Con el control de acceso se puede restringir fácilmente días y horarios de acceso y emitir informes detallados de la actividad del usuario. Una sola tarjeta puede abrir todas las puertas desde que, obviamente, tenga los permisos para hacerlo.

Inicialmente para garantizar la seguridad se recurría a las alarmas antirrobo, las cuales tenían una función única: emitir un sonido ante la detección de una persona o intruso; estas alarmas sonoras no fueron lo suficientemente efectivas para poder asegurar la tranquilidad de las viviendas o locales. Esto se debe a que dichas alarmas tenían una alta probabilidad para que su mecanismo fallara o que simplemente el sensor se activara por su sensibilidad en cualquier momento o circunstancia no deseada (A.Palacios, 2017).

En sus comienzos los sistemas de control de acceso usaron con frecuencia como tecnología los teclados por

Número de Identificación Personal (PIN) llamados así por sus siglas en inglés, los cuales fueron reemplazados lentamente por sistemas con tarjetas magnéticas y código de barra. Con los avances tecnológicos se hizo presente la tarjeta de proximidad y al mismo tiempo aparecieron los lectores biométricos. Los usos de estas tecnologías brindan la posibilidad de aumentar el nivel de seguridad en las instalaciones militares, centros de investigación, centros escolares y demás instituciones que así lo requieran (Muñoz, 2015).

Una de las tecnologías que está cobrando auge a nivel mundial es la identificación por radiofrecuencia (RFID por sus siglas en inglés), esta no es más que un sistema para la comunicación inalámbrica entre dos o más objetos, donde uno emite señales de radio y el otro responde en función de la señal recibida; la comunicación se realiza a través de una antena con un transponder (también conocido como tag en inglés o etiqueta). Asimismo esta es útil donde tengan que realizarse continuos registros de datos, sin contacto y sin la necesidad de campo visual, ya sea para identificar, rastrear y gestionar productos, documentos, objetos, personas, animales (A.Palacios, 2017).

Un sistema de control de acceso RFID provee una solución sencilla y eficiente, esto se debe a que las Identificaciones RFID pueden ser leídas a distancias mucho mayores en comparación con medios tradicionales y la información contenida en cada credencial puede ser sobre escrita repetidamente. Además, los aumentos de las distancias de lectura permiten habilitar otras tecnologías como el activar cámaras de vigilancia cuando un empleado entra a una determinada área. Más aun, múltiples credenciales pueden ser leídas al mismo tiempo. También la información sobre horas de entrada, asistencia y labores realizadas puede ser fácilmente monitoreada y almacenada en una base de datos (Assessment y Emergency Responders, 2015).

Las organizaciones hoy en día requieren un sistema de control de acceso funcional y eficiente debido a que deben mejorar la seguridad para limitar el acceso a áreas restringidas, monitorear la entrada y salida de los empleados y realizar mejoras para prevenir pérdidas en los locales de trabajo.

Actualmente, existen grandes deficiencias en los sistemas tradicionales de control de acceso basados en la identificación por tarjetas, tales como los códigos de barra y cintas magnéticas; estas tecnologías dependen del contacto directo con el dispositivo lector o bien de colocar las identificaciones cerca del lector, generando con el tiempo desgaste físico de las tarjetas. Además, los códigos de barra solamente pueden ser utilizados una vez y la información contenida en ellos no puede ser actualizada, y las cintas magnéticas al rayarse o ser expuestas a campos magnéticos pierden la información que contienen. Estas limitaciones resultan engorrosas para mantener un control de acceso eficiente y además provocan pérdidas de tiempo.

Cuba no está exenta de los problemas mencionados anteriormente, en la actualidad hay locales que tienen un control de acceso con tecnología muy obsoleta o se basan en el contrato de personal para regular dicho acceso, lo cual provoca deficiencias en la seguridad de las áreas; además de que se utilizan mecanismos que generan retrasos y control insuficiente en el monitoreo del cumplimiento de la jornada laboral.

La Universidad de Ciencias Informáticas (UCI) se encuentra en una etapa crítica de su desarrollo, don-

de la innovación y la investigación son pilares fundamentales para su misión educativa y tecnológica. Sin embargo, la seguridad de sus laboratorios y centros de desarrollo, que albergan proyectos de investigación sensibles y recursos valiosos, es un aspecto que requiere atención inmediata. Actualmente, muchos de estos espacios utilizan sistemas de control de acceso obsoletos que no satisfacen las demandas de seguridad contemporáneas.

La implementación de un sistema moderno basado en tecnologías como RFID, reconocimiento de huella dactilar y reconocimiento facial es esencial para abordar estas deficiencias. Este sistema permitirá:

- Mejorar la Seguridad: Al restringir el acceso a áreas sensibles solo a personal autorizado.
- Optimizar el Monitoreo: Facilitar el seguimiento en tiempo real del acceso y salida de los usuarios, lo que contribuirá a una gestión más eficiente.
- Reducir Errores Humanos: Minimizar las vulnerabilidades asociadas con métodos tradicionales, como el uso de llaves o tarjetas magnéticas que pueden ser fácilmente perdidas o duplicadas.
- Aumentar la Eficiencia: Proporcionar un proceso ágil y sin contacto para el acceso a las instalaciones, lo que mejorará la experiencia del usuario.

En este contexto, la UCI necesita un sistema robusto y eficiente para el control de acceso a sus laboratorios y centros de desarrollo. Este sistema no solo mejorará la seguridad física y protegerá los recursos valiosos, sino que también optimizará los procesos operativos y fomentará un ambiente propicio para la investigación e innovación. La adopción de tecnologías avanzadas garantizará que la UCI esté a la vanguardia en materia de seguridad institucional, permitiendo así un desarrollo académico y científico más seguro y efectivo.

A partir de la situación problemática anteriormente expuesta, se plantea como **problema de investigación**: ¿Cómo controlar el acceso a un local?, teniendo como **objeto de estudio**: Sistemas de control de acceso. Enmarcado en el **campo de acción** : Sistemas de control de acceso basado en RFID y biometría. El **objetivo general**: es: Desarrollar un sistema que permita controlar el acceso mediante una plataforma arduino. Para dar solución al objetivo planteado se proponen como **tareas de la investigación**:

- Análisis los referentes teóricos acerca de portales digitales, la gestión de contenido y el uso de las herramientas innovadoras en el aprendizaje universitario para la construcción del marco teórico conceptual.
- Estudio del estado del arte del tema en cuestión valorando las funcionalidades de sistemas de control de acceso existentes (soluciones homólogas).
- Identificación de los requisitos funcionales para el desarrollo de la solución propuesta.
- Generación de los artefactos relacionados con el análisis y diseño de la solución propuesta.
- Diseño e implementación de la solución propuesta.
- Validación del sistema a través de la realización de pruebas.

Métodos teóricos:

Analítico-Sintético: Es un enfoque que implica descomponer un problema o sistema en sus componentes más simples (análisis) para entender su funcionamiento, y luego recombinar esos elementos para formar una visión integral (síntesis). Este método es útil para abordar problemas complejos, permitiendo una comprensión profunda de cada parte antes de integrarlas en un todo cohesivo.

El método analítico-sintético es fundamental para descomponer el sistema de control de acceso en sus componentes esenciales, lo que permite un análisis detallado de cada parte del sistema, como los métodos de autenticación (huella dactilar, RFID y reconocimiento facial). Al identificar y entender cómo cada componente funciona de manera independiente, se pueden optimizar sus interacciones y mejorar la eficiencia general del sistema. Este enfoque también facilita la identificación de posibles fallos o áreas de mejora antes de la integración final, asegurando que el sistema sea robusto y cumpla con los requisitos funcionales establecidos.

Histórico-Lógico: Se basa en el estudio de la evolución de un fenómeno a lo largo del tiempo (histórico) y la identificación de las relaciones causales que han influido en su desarrollo (lógico). Este enfoque permite comprender cómo se han formado las estructuras actuales a partir de procesos pasados.

El método histórico-lógico permite contextualizar el desarrollo del sistema dentro de la evolución de tecnologías de control de acceso. Al estudiar cómo han cambiado y mejorado estos sistemas a lo largo del tiempo, se pueden identificar tendencias, innovaciones y errores comunes en implementaciones anteriores. Esta comprensión histórica no solo ayuda a justificar las decisiones tecnológicas tomadas en el proyecto, sino que también proporciona lecciones valiosas que pueden aplicarse para evitar problemas similares en el futuro. Así, este método contribuye a construir un sistema más eficiente y alineado con las mejores prácticas del sector.

La aplicación de estos métodos teóricos en tu proyecto de control de acceso no solo enriquecerá tu análisis y diseño, sino que también proporcionará una base sólida para la toma de decisiones informadas.

Métodos empíricos:

Observación: Implica la recopilación de datos a través de la observación directa de un fenómeno o proceso en su entorno natural. Este enfoque permite a los investigadores y desarrolladores obtener información valiosa sobre el comportamiento de los usuarios y el funcionamiento de un sistema sin interferir en su operación normal.

En el contexto del sistema de control de acceso, la observación se utilizó para analizar cómo los usuarios interactúan con los diferentes métodos de autenticación (como huellas dactilares, RFID y reconocimiento facial). Al observar a los usuarios en situaciones reales, se pueden identificar problemas de usabilidad, como dificultades en la interacción con los dispositivos o confusiones sobre el proceso de acceso. Esta información es crucial para realizar ajustes en el diseño del sistema y mejorar la experiencia del usuario, asegurando que el sistema sea intuitivo y eficiente.

Pruebas: Este método implica la ejecución de un conjunto planificado de actividades para evaluar el rendimiento, la funcionalidad y la seguridad de un sistema. Este enfoque permite verificar si el sistema cumple con los requisitos establecidos y si funciona correctamente bajo diferentes condiciones.

En el desarrollo del sistema de control de acceso, las pruebas son esenciales para garantizar que cada método de autenticación funcione correctamente. Esto incluye pruebas unitarias para evaluar componentes individuales y pruebas de aceptación para confirmar que el sistema cumple con las expectativas del usuario final. Realizar pruebas exhaustivas ayuda a identificar y corregir errores antes de la implementación completa del sistema, lo que reduce riesgos y mejora la calidad general del producto. Además, las pruebas pueden incluir simulaciones de escenarios reales para evaluar cómo responde el sistema ante intentos no autorizados o condiciones adversas, asegurando así su robustez y fiabilidad.

El presente trabajo de diploma está compuesto por tres capítulos, estructurados de la siguiente forma:

- **Capítulo 1 Fundamentos y Referentes teórico-metodológicos sobre Sistema de control de acceso:** En él se describen los conceptos fundamentales asociados al proyecto, se realiza el análisis de los sistemas homólogos existentes para realizar funciones similares a las requiridas en el sistema a desarrollar, así como la selección de las tecnologías y metodología a emplear.
- **Capítulo 2 Diseño de la solución propuesta al problema científico:** En este capítulo se describe la propuesta de solución, se definen los requisitos del sistema , así como la arquitectura y los patrones empleados para el desarrollo del sistema a través del uso de historias de usuario y tarjetas CRC (Clase Responsabilidad y Colaboración).
- **Capítulo 3 Validación de la propuesta de solución:** En este capítulo se describen las tareas ingenieriles, el plan de pruebas y la realización de las mismas así como su resultado para verificar si lo realizado cumple con las especificaciones definidas por el cliente.

CAPÍTULO 1

FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE SISTEMA DE CONTROL DE ACCESO.

El capítulo tiene como objetivo principal ofrecer una comprensión integral de la seguridad física en las organizaciones, abordando sus componentes, importancia y técnicas de identificación y autenticación. Se busca resaltar cómo estas medidas son esenciales para proteger los activos tangibles y garantizar la seguridad del personal frente a diversas amenazas. En el mismo se abordan los conceptos fundamentales asociados a un sistema para el control de acceso, así como sistemas homólogos además de las herramientas y tecnologías necesarias para el desarrollo del sistema.

1.1. Control de acceso

El control de acceso es un sistema de seguridad que se utiliza para regular y gestionar el acceso a un determinado recurso, ya sea físico (como un edificio, una habitación o un vehículo) o digital (como un sistema informático, una red o una aplicación). El objetivo principal del control de acceso es garantizar que solo las personas autorizadas puedan acceder a los recursos protegidos (Koot, 2022).

Aunque mucha gente piensa que un sistema de control de acceso físico es para controlar puertas y ascensores, los sistemas de control de acceso físico van mucho más allá. También pueden utilizarse para gestionar el acceso a oficinas privadas, zonas de almacenamiento, maquinarias, cámaras acorazadas, laboratorios, aparcamientos, armarios, zonas de almacenamiento de documentos y cualquier otro espacio que deba tener un acceso limitado por motivos de seguridad. Para el presente proyecto en particular, el control de acceso es la habilidad de conceder o denegar el acceso a un espacio físico.

Para lograr desarrollar un sistema de control de acceso funcional, eficiente y robusto, es vital examinar una

serie de conceptos que son fundamentales para su implementación efectiva. En un entorno donde los riesgos son inevitables, es fundamental identificar y comprender las amenazas potenciales que pueden comprometer la integridad y la confidencialidad de la información y los activos físicos. Para esto la evaluación del concepto de **seguridad** es un paso crucial .

Seguridad:

En términos generales, la seguridad se define como .[el] estado de bienestar que el ser humano percibe y disfruta". Según el Diccionario de la lengua española 'seguridad' tiene varias acepciones. Una de ellas es ausencia de peligro, invulnerabilidad, inmunidad, protección, indestructibilidad, amparo, defensa. La seguridad consiste en hacer que el riesgo situacional se reduzca a niveles aceptables, debido a que el riesgo es inherente a cualquier actividad y nunca puede ser eliminado (Española, 2023).

Seguridad Informática:

La seguridad informática es el conjunto de tecnologías, procesos y prácticas diseñadas para proteger sistemas informáticos, redes, dispositivos y datos de accesos no autorizados, ciberataques y daños. Su objetivo principal es garantizar la confidencialidad, integridad y disponibilidad de la información.teclab, 2024

Seguridad Física:

La seguridad física se refiere a las medidas y controles implementados para proteger los activos físicos de una organización, como edificios, equipos y personal, de amenazas como robos, vandalismo, desastres naturales y accesos no autorizados (360, 2023).

Componentes de la seguridad física:

- Controles de acceso: Sistemas que regulan quién puede entrar a un área específica, como cerraduras, tarjetas de acceso y sistemas biométricos.
- Vigilancia: Uso de cámaras de seguridad y personal de seguridad para monitorear áreas críticas y detectar actividades sospechosas.
- Alarmas: Sistemas que alertan sobre intrusiones o situaciones de emergencia.
- Diseño ambiental: Estrategias que incluyen la planificación del espacio físico para maximizar la seguridad, como la iluminación adecuada y la ubicación estratégica de barreras.

Importancia de la seguridad física:

La seguridad física es fundamental para proteger los activos tangibles de una organización, prevenir pérdidas y garantizar la seguridad del personal. La combinación de medidas de seguridad física y cibernética crea un enfoque integral que ayuda a mitigar riesgos y proteger la organización en su totalidad.

Evaluar el concepto de seguridad es esencial para el desarrollo de un sistema de control de acceso, ya que este sistema debe ser capaz de reducir los riesgos inherentes a la gestión de espacios sensibles. La seguridad

implica no solo la protección física de las instalaciones, sino también la implementación de medidas que aseguren el acceso restringido a personal autorizado. Un enfoque integral en la seguridad garantizará que el sistema de control de acceso no solo sea efectivo en la protección de los recursos, sino que también fomente un entorno seguro y confiable para todos los usuarios.

Dado que el riesgo es una constante en cualquier actividad, es fundamental establecer protocolos y tecnologías que permitan mitigar estos riesgos a niveles aceptables. Por lo que se hace necesario analizar las técnicas de identificación y autenticación existentes, además de determinar las que se van a emplear en el sistema.

Técnicas de identificación y autenticación:

Identificación es la acción por parte de un usuario de presentar su identidad a un sistema, usualmente se usa un identificador de usuario. Establece que el usuario es responsable de las acciones que lleve a cabo en el sistema. Esto está relacionado con los registros de auditorías que permiten guardar las acciones realizadas dentro del sistema y rastrearlas hasta el usuario autenticado (EcuRed, n.d.).

Autenticación es la verificación de que el usuario que trate de identificarse es válido, usualmente se implementa con una contraseña en el momento de iniciar una sección. Existen 4 tipos de técnicas que permiten realizar la autenticación de la identidad del usuario, las cuales pueden ser utilizadas individualmente o combinadas (Autenticación de varios factores):

- Algo que solamente el individuo conoce: ejemplo una contraseña.
- Algo que una persona posee: ejemplo una tarjeta magnética.
- Algo que el individuo es y que lo identifica unívocamente: ejemplo las huellas digitales.
- Algo que solamente el individuo es capaz de hacer: ejemplo los patrones de escritura (EcuRed, n.d.).

Autorización:

La autorización sucede después de la autenticación y usa atributos o derechos, asociados con la identidad digital para determinar a qué recursos puede acceder dicha identidad digital. Está centrada en políticas de control de acceso (reglas para especificar quién puede acceder y a qué recursos), modelos (formalismos para describir las políticas) y mecanismos (traslado de la solicitud de acceso de un usuario a una tabla para conceder o denegar el acceso) (ibíd.).

Luego de analizar estas técnicas se han seleccionado las técnicas de autenticación que implican algo que una persona posee y algo que el individuo es y que lo identifica unívocamente debido a su capacidad para ofrecer un equilibrio óptimo entre seguridad y usabilidad. La autenticación basada en la posesión, como el uso de tarjetas magnéticas o tokens, proporciona una capa adicional de protección al requerir un objeto físico que solo el usuario autorizado debe tener. Esto reduce significativamente el riesgo de acceso no autorizado, ya que incluso si se compromete la contraseña, el atacante necesitaría también el dispositivo físico para obtener

acceso.

Por otro lado, la autenticación biométrica, que se basa en características únicas del individuo, como las huellas dactilares o el reconocimiento facial, ofrece un nivel de seguridad superior. Estas características son inherentes a cada persona y son extremadamente difíciles de replicar, lo que minimiza el riesgo de suplantación de identidad. Al combinar estas dos técnicas, se crea un sistema robusto que no solo verifica la identidad del usuario de manera efectiva, sino que también mejora la experiencia del usuario al facilitar un acceso rápido y seguro a los recursos protegidos. Esta dualidad en las técnicas elegidas asegura que el sistema de control de acceso sea tanto seguro como conveniente, alineándose con las necesidades contemporáneas de seguridad en entornos físicos y digitales.

El análisis de los **tipos de tecnologías de control de acceso que se basan en algo que una persona posee** es fundamental para diseñar un sistema de seguridad efectivo y confiable. Estas tecnologías incluyen dispositivos como tarjetas magnéticas, tokens de seguridad y llaves electrónicas, que requieren que el usuario tenga en su posesión un objeto físico para poder acceder a áreas restringidas o sistemas protegidos. Al evaluar estas opciones, es crucial considerar factores como la facilidad de uso, la seguridad frente a la clonación o pérdida, y la integración con otros sistemas de seguridad existentes. La elección adecuada de la tecnología no solo garantiza que solo las personas autorizadas puedan acceder a los recursos, sino que también mejora la experiencia del usuario al proporcionar un método ágil y eficiente para el control de acceso. Este análisis permitirá identificar las soluciones más adecuadas para satisfacer las necesidades específicas del entorno y asegurar una protección robusta contra accesos no autorizados. A continuación se definen dichas tecnologías.

1.1.1. Tipos de tecnologías para el control de acceso basadas en posesión

Tarjetas de código de barras:

La tarjeta de código de barras es una tarjeta física que incorpora un código de barras, que es un código legible por máquina que contiene información relacionada con el titular de la tarjeta o entidad relacionada. Estas tarjetas juegan un papel vital en diversas operaciones, desde simplificar la interacción con el cliente hasta reforzar las medidas de Seguridad.

El núcleo de la tarjeta de código de barras es almacenar y transmitir información a través del Código de barras impreso. Al escanear, el Código de barras transmite los datos al sistema para lograr una identificación o seguimiento rápido. Las tarjetas de código de barras a menudo existen en los planes de membresía, la atención médica y los sistemas de seguridad, y se valoran por su fiabilidad, rentabilidad y facilidad de uso (Onlinetoolcenter, n.d).

Figura 1.1. Tarjetas de código de barras. Fuente:(Onlinetoolcenter, n.d.)

Tarjetas Magnéticas: Son tarjetas que contienen una banda magnética que posee un código que permite identificarse rápidamente. Este sistema utiliza señales electromagnéticas para registrar y codificar la información. Una de las aplicaciones más comunes de esta tecnología son las tarjetas de crédito. Las tarjetas magnéticas cuentan con una alta difusión y popularidad, entre otras cosas por su bajo costo. Sin embargo, su uso continuo las deteriora físicamente debido a la fricción en el momento de la lectura; también si la tarjeta es acercada a una fuente electromagnética, relativamente fuerte, la información contenida en ella puede ser modificada, con lo cual pierde su utilidad (Tejada, 2012).

Figura 1.2. Tarjetas Magnéticas. Fuente: (D. S. Morales Tejada, 2012)

Tecnología RFID:

La tecnología RFID (Radio Frequency Identification, Identificación por Radiofrecuencia) es un sistema de identificación de objetos que utiliza ondas de radio para transmitir información. Estas ondas de radio se utilizan para comunicarse con un microchip, que se puede montar en etiquetas, tarjetas o transpondedores. La información se almacena en el microchip y se puede leer y escribir utilizando un lector RFID. Esta tecnología es similar a los códigos de barras tradicionales, pero ofrece varias ventajas significativas (Tecnipesa, 2024).

Figura 1.3. Tecnología RFID. Fuente: (Tecnipesa, 2024).

Componentes de Hardware:

Etiqueta RFID (Transpondedor): Es el elemento central del sistema RFID. Consta de una antena, un transductor de radio y un material encapsulado que contiene la información de identificación del objeto al que está adherida. Existen diferentes tipos de etiquetas RFID: pasivas, activas y semiactivas.

Lector RFID: Es el dispositivo que se encarga de leer y escribir la información contenida en las etiquetas RFID. Consta de una antena, un transceptor y un decodificador.

Antena RFID: Permite la comunicación entre el lector y las etiquetas RFID. Transmite y recibe las señales de radiofrecuencia.

Componentes de Software:

Middleware RFID: Es el software que gestiona la comunicación entre los lectores RFID y los sistemas de información de la empresa, procesando los datos recibidos.

Sistema de Información: Es la aplicación o sistema informático que utiliza los datos recopilados por el sistema RFID para realizar diversas funciones, como control de inventario, trazabilidad, etc.

Ventajas de la tecnología RFID:

- Flexibilidad: No requiere una línea de visión directa para la lectura, lo que permite identificar objetos empaquetados o ubicados en cajas de transporte.
- Velocidad: Permite leer y escribir información en etiquetas RFID a velocidades muy altas.
- Precisión: Permite identificar objetos con precisión, sin necesidad de alineación específica.
- Seguridad: La información se almacena en el microchip y se puede cifrar para protegerla.
- Trazabilidad: Permite seguir el rastro de los productos a lo largo de todo el proceso de producción y distribución (Tecnipesa, 2024).

Etiqueta RFID:

La etiqueta RFID (Radio Frequency Identification, Identificación por Radiofrecuencia) es un dispositivo inteligente y de pequeñas dimensiones que almacena datos y es capaz de transmitirlos a través de señales de radiofrecuencia (Services, 2023).

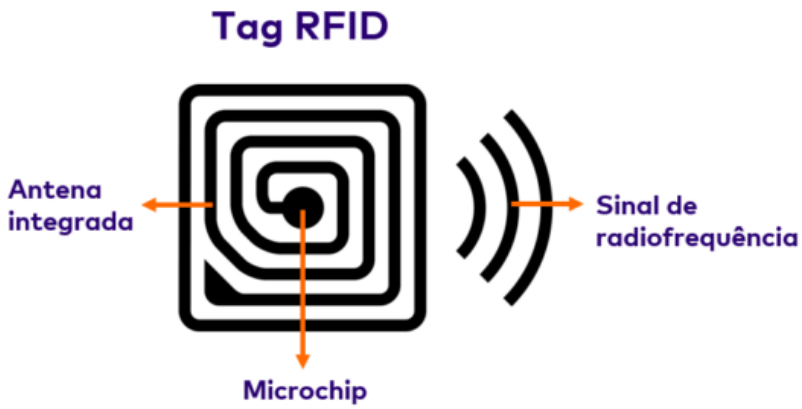


Figura 1.4. Etiqueta RFID Fuente: (TRBL Services, 2023).

Estas etiquetas son fundamentales para el sistema RFID y se clasifican en tres tipos principales según su fuente de alimentación:

- Etiquetas Pasivas:
  Las etiquetas pasivas no requieren de ninguna fuente de alimentación interna para poder cumplir la función. Estas etiquetas se activan cuando se les aplica una señal de radiofrecuencia emitida por un lector RFID. Su rango de acción es limitado debido a que no tienen una fuente de energía propia, pero son más baratas y compactas que las etiquetas activas (ibíd.).
- Etiquetas Activas:
  Las etiquetas activas tienen una batería interna que les permite emitir señales de manera autónoma

y ser detectadas desde distancias considerables. Estas etiquetas son más costosas y tienen una vida útil limitada debido a la batería, pero ofrecen un mayor rango de acción y pueden ser utilizadas en aplicaciones que requieren una mayor distancia de comunicación (Services, 2023).

- Etiquetas Semiactivas:
  Las etiquetas semiactivas incluyen una batería para que el chip siempre tenga suficiente energía para encenderse, pero no tienen un transmisor activo. Estas etiquetas tienen un mayor rango que las etiquetas pasivas, aunque a un mayor costo y una vida útil limitada debido a la batería (ibíd.).

Lector RFID:

Un lector RFID (Radio Frequency Identification, Identificación por Radiofrecuencia) es un dispositivo que se utiliza para leer y escribir información en etiquetas RFID (Radio Frequency Identification, Identificación por Radiofrecuencia). Estas etiquetas contienen un microchip y una antena que permiten la comunicación con el lector a través de ondas de radio.

Figura 1.5. Lector RFID Fuente: (TRBL Services, 2023).

Se ha seleccionado la tecnología **RFID** (Identificación por Radiofrecuencia) como parte fundamental del sistema de control de acceso, debido a sus múltiples ventajas que la hacen especialmente útil para este proyecto. RFID permite una identificación rápida y eficiente, ya que los usuarios pueden acceder a áreas restringidas simplemente acercando su tarjeta o etiqueta a un lector, eliminando la necesidad de contacto físico o inserción manual de credenciales. Esto no solo mejora la comodidad del usuario, sino que también acelera el proceso de acceso, lo que es crucial en entornos con alto tráfico de personas. Además, la tecnología RFID ofrece un alto nivel de seguridad al utilizar códigos de identificación únicos que son difíciles de replicar o falsificar, lo que reduce significativamente el riesgo de accesos no autorizados. La capacidad de rastrear en tiempo real el movimiento de personas y activos dentro de las instalaciones también proporciona información valiosa para optimizar los protocolos de seguridad y gestión de recursos. En resumen, la implementación de RFID en el sistema de control de acceso no solo garantiza una mayor seguridad, sino que también mejora la eficiencia operativa y la experiencia del usuario.

Realizar una evaluación de las **tecnologías de control de acceso que se basan en algo que el individuo**

es y que lo identifica unívocamente es fundamental para garantizar un sistema de seguridad eficaz y confiable. Este enfoque se centra en la **autenticación biométrica**, que utiliza características biológicas únicas, como huellas dactilares, reconocimiento facial, patrones de iris o voz, para verificar la identidad de una persona. La singularidad de estos rasgos biológicos proporciona un nivel de seguridad superior, ya que es extremadamente difícil replicar o falsificar estas características. Además, la autenticación biométrica mejora la experiencia del usuario al eliminar la necesidad de recordar contraseñas o utilizar dispositivos adicionales para acceder a sistemas o áreas restringidas. Este método no solo asegura que solo las personas autorizadas puedan acceder a recursos críticos, sino que también permite un proceso de autenticación rápido y eficiente, lo que es esencial en entornos donde el tiempo y la seguridad son primordiales. Al integrar tecnologías biométricas en el sistema de control de acceso, se puede crear una solución robusta que maximice tanto la seguridad como la conveniencia para los usuarios.

1.1.2. Tipos de tecnologías para el control de acceso basadas en autenticación biométrica

Sistemas Biométricos:

Estos sistemas están dotados de la capacidad de reconocer una característica personal, donde los lectores reconocen automáticamente la característica física del usuario, eliminando por completo el uso de tarjetas electrónicas o magnéticas. Principalmente se trabaja en el reconocimiento de huellas dactilares, reconocimiento de iris, reflexión de retina, geometría de la mano, geometría facial, termografía mano-facial y patrón de voz. La biometría ofrece una ventaja significativa: el alto grado de seguridad, dado que sólo identifica la característica de la persona autorizada por lo que es difícil la suplantación de información; los rasgos físicos son únicos e intransferibles (Tejada, 2012).

Figura 1.6. Sistemas Biométrico Fuente: (D. S. Morales Tejada, 2012)

Reconocimiento de Huellas Dactilares: Este es uno de los métodos más comunes y consiste en escanear las impresiones digitales de una persona. Cada huella dactilar es única, lo que permite una identificación precisa. Funcionamiento: Se utiliza un sensor para capturar la imagen de la huella, que luego se procesa y almacena en una base de datos. Durante la autenticación, la huella escaneada se compara con las al-

macenadas. Aplicaciones: Control de acceso en dispositivos móviles, sistemas de seguridad en edificios y autenticación en servicios financieros (Protección de Datos, 2023).

Reconocimiento Facial: Esta tecnología utiliza algoritmos para identificar a una persona a partir de su rostro. Analiza características como la distancia entre ojos, forma de la mandíbula y contornos faciales. Funcionamiento: Las imágenes del rostro se capturan y se procesan para crear un modelo facial. Este modelo se compara con los almacenados en una base de datos durante el proceso de verificación. Aplicaciones: Seguridad pública, desbloqueo de teléfonos móviles, y sistemas de control de acceso en instalaciones sensibles (ibíd.).

Biometría del Iris: El reconocimiento del iris se basa en los patrones únicos del iris del ojo humano. Esta tecnología es altamente precisa debido a la singularidad del iris. Funcionamiento: Se captura una imagen del iris utilizando cámaras infrarrojas para evitar la contracción de la pupila. Los patrones se convierten en un código matemático que se almacena y se compara durante el reconocimiento. Aplicaciones: Control de acceso en aeropuertos, sistemas gubernamentales y dispositivos móviles avanzados (ibíd.).

Biometría de Voz: La biometría de voz utiliza las características vocales únicas de una persona para autenticar su identidad. Esto incluye el tono, timbre y patrones de habla. Funcionamiento: Se graba una muestra de voz que se analiza para extraer características únicas. Durante la autenticación, la voz del usuario se compara con la muestra almacenada. Aplicaciones: Sistemas telefónicos automatizados, atención al cliente y seguridad bancaria (ibíd.).

Reconocimiento de Firmas: Este sistema biométrico analiza la firma manuscrita de una persona, considerando aspectos como la presión aplicada y la velocidad al firmar. Funcionamiento: Se captura la firma a través de dispositivos especializados que registran las características dinámicas al momento de firmar. Aplicaciones: Autenticación en documentos legales, contratos y transacciones financieras (ibíd.).

Se han seleccionado las tecnologías de reconocimiento de huella dactilar y reconocimiento facial como métodos clave para el sistema de control de acceso, debido a su alta efectividad y precisión en la autenticación de usuarios. El reconocimiento de huella dactilar es una opción ampliamente utilizada que ofrece un nivel de seguridad elevado, ya que cada huella es única e intransferible, lo que minimiza el riesgo de suplantación de identidad. Además, este método permite un acceso rápido y conveniente, ya que los usuarios pueden ser autenticados en cuestión de segundos. Por otro lado, el reconocimiento facial complementa esta seguridad al utilizar características biométricas del rostro, lo que permite la identificación sin contacto físico. Esta tecnología es especialmente útil en entornos donde la rapidez es esencial, ya que puede autenticar a múltiples usuarios simultáneamente y sin necesidad de interacción directa. La combinación de estas dos tecnologías proporciona un enfoque robusto y versátil para el control de acceso, garantizando que solo las personas autorizadas puedan acceder a áreas sensibles y mejorando la experiencia general del usuario al hacer el proceso más ágil y seguro.

Debido a la necesidad planteada por el cliente de q el sistema debe ser desarrollado con arduino y raspberry pi se examinarán a continuación dichos conceptos.

1.1.3. Arduino

Arduino es una plataforma de código abierto de prototipos electrónicos que se basa en hardware y software flexibles y fáciles de usar que ponen al alcance de cualquier persona la construcción de circuitos electrónicos/robots. En lo referente a hardware, se basa en placas que se pueden ensamblar a mano o que se pueden comprar directamente preensambladas. Cada una de las placas lleva un microcontrolador en el que se carga el programa software que es necesario desarrollar para darle vida a la placa (D.Lozano, 2022).

Arduino es una plataforma de hardware libre que permite a los usuarios crear proyectos interactivos mediante la combinación de software y hardware. Está diseñada para facilitar el uso de la electrónica en proyectos de todo tipo, desde simples hasta complejos. La plataforma incluye una serie de placas de circuito (hardware) y un entorno de desarrollo integrado (IDE) que permite programar las placas utilizando un lenguaje basado en C/C++ (ibíd.).

Componentes Electrónicos:

Placas Arduino: Dispositivos que contienen un microcontrolador y permiten la conexión con otros componentes.

Sensores: Dispositivos que detectan cambios en el entorno (como temperatura, luz o movimiento) y envían esa información a la placa. Actuadores: Componentes que realizan acciones físicas, como motores o LEDs, controlados por la placa Arduino.

Resistencias, capacitores y diodos: Elementos básicos de circuitos electrónicos que permiten controlar el flujo de corriente.

Placa Arduino:

Una placa Arduino es un microcontrolador programable que actúa como el cerebro del proyecto. Las placas más comunes son Arduino Uno, Mega y Nano. Cada placa tiene diferentes características en términos de pines de entrada/salida, memoria y capacidad de procesamiento (ibíd.).

Funcionamiento:

Entradas y Salidas: Las placas Arduino tienen pines digitales y analógicos que permiten la conexión con sensores (entradas) y actuadores (salidas). Los pines digitales pueden leer o escribir valores binarios (0 o 1), mientras que los pines analógicos pueden leer valores continuos.

Programación: Los usuarios programan la placa utilizando el IDE de Arduino. Un programa típico consta de dos funciones principales:

setup(): Se ejecuta una vez al inicio para configurar los pines y otras configuraciones iniciales.

loop(): Se ejecuta continuamente, permitiendo al programa responder a entradas y controlar salidas en tiempo real.

Interacción con el Entorno: La placa puede interactuar con el entorno mediante sensores que leen datos (como temperatura o luz) y actuadores que responden a esos datos (como encender un LED o mover un motor).

Comunicación: Arduino puede comunicarse con otros dispositivos a través de protocolos como I2C, SPI o UART, lo que permite la integración con otros módulos o sistemas (D.Lozano, 2022).

Arduino Uno: El Arduino Uno es una de las placas más populares de la familia Arduino, basada en el microcontrolador ATmega328P. Tiene 14 pines digitales de entrada/salida, de los cuales 6 pueden ser utilizados como salidas PWM, y 6 pines analógicos. La placa cuenta con 32 KB de memoria flash para almacenamiento de programas, 2 KB de SRAM y 1 KB de EEPROM. Su velocidad de reloj es de 16 MHz, lo que la hace adecuada para una amplia variedad de proyectos. Se conecta a una computadora mediante un cable USB y se programa utilizando el entorno de desarrollo Arduino IDE. El Uno es ideal para principiantes debido a su facilidad de uso y la gran cantidad de recursos disponibles en línea (JavaTpoint, n.d).

Figura 1.7. Arduino UNO Fuente: (JavaTpoint, n.d.)

Arduino Mega: El Arduino Mega es una versión más avanzada que el Uno, diseñada para proyectos que requieren más recursos y capacidades. Está equipado con el microcontrolador ATmega2560, que ofrece 54 pines digitales (15 de los cuales pueden ser utilizados como PWM) y 16 pines analógicos. Además, cuenta con 256 KB de memoria flash, 8 KB de SRAM y 4 KB de EEPROM. Su mayor número de pines y memoria lo hace ideal para aplicaciones complejas como controladores de robots o proyectos que requieren múltiples

sensores y actuadores. El Mega también se programa a través del Arduino IDE, pero su tamaño más grande puede requerir un manejo diferente en términos de diseño físico (Arduino, n.d).

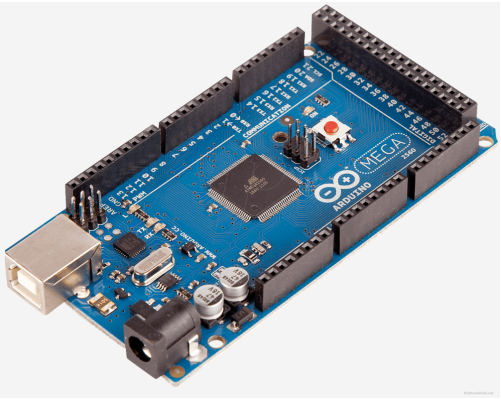


Figura 1.8. Arduino Mega Fuente: (Arduino, n.d.)

Arduino Nano: El Arduino Nano es una placa compacta y versátil que utiliza el mismo microcontrolador ATmega328P que el Uno, pero en un formato más pequeño. Tiene 14 pines digitales (6 PWM) y 8 pines analógicos, con características similares en cuanto a memoria: 32 KB de flash, 2 KB de SRAM y 1 KB de EEPROM. El Nano se alimenta a través de USB o mediante un conector externo y es particularmente útil para proyectos donde el espacio es limitado. Su tamaño compacto lo hace ideal para prototipos o integraciones en dispositivos más pequeños (Blikai, 2024).

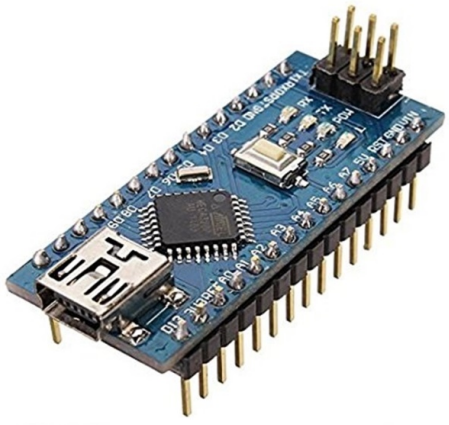


Figura 1.9. Arduino Nano Fuente: (Blikai, 2024)

1.1.4. Raspberry Pi

Raspberry Pi es lo que se conoce como ordenador de una sola placa, que es exactamente lo que su nombre indica: como un ordenador de sobremesa, un portátil o un smartphone, pero construido sobre una única placa de circuito impreso. Como la mayoría de los ordenadores de una sola placa, Raspberry Pi es pequeño más o menos del tamaño de una tarjeta de crédito pero eso no impide que sea potente: un Raspberry Pi puede hacer cualquier cosa que haga un ordenador más grande y de mayor consumo, aunque puede que no lo haga tan rápido (Halfacree, 2024).

Figura 1.10. Raspberry Pi Fuente: (Halfacree, 2024)

La familia Raspberry Pi nació del deseo de fomentar la educación informática práctica en todo el mundo. Sus creadores unieron sus esfuerzos para establecer la Raspberry Pi Foundation, organización sin ánimo de lucro, sin saber lo popular que iba a hacerse: los pocos miles de unidades fabricadas en 2012 para tantear el terreno se vendieron inmediatamente, y desde entonces se han despachado millones de unidades en todo el mundo. Estas placas se han adoptado en hogares, aulas, oficinas, centros de datos, fábricas, e incluso en embarcaciones dirigidas automáticamente y en globos espaciales (ibíd.).

Desde el Model B original se han lanzado varios modelos de Raspberry Pi, cada uno con especificaciones mejoradas o características específicas para un tipo de uso particular. La familia Raspberry Pi Zero, por ejemplo, es una versión diminuta de la Raspberry Pi de tamaño completo que omite algunas características ?en concreto los múltiples puertos USB y el puerto de red con cable? para reducir el formato y los requisitos energéticos (ibíd.).

Pero todos los modelos de Raspberry Pi tienen una cosa en común: son compatibles, lo que significa que el software escrito para un modelo funcionará en cualquier otro modelo. Incluso es posible utilizar la versión más reciente del sistema operativo de Raspberry Pi y ejecutarla en un prototipo original del Model B prelanzamiento. Si bien es cierto que el funcionamiento será más lento, el caso es que podrá ejecutarse (ibíd.).

- A Entrada de alimentación USB tipo C
- B Puerto de pantalla DSI
- C Conexión inalámbrica / Bluetooth
- D Micro-HDMI 0
- E Micro-HDMI 1
- F Sistema en chip
- G GPIO
- H RAM
- I Puerto de cámara CSI
- J AV de 3,5 mm
- K PoE
- L USB 2.0
- M USB 3.0
- N Puerto Ethernet (Halfacree, 2024).

Figura 1.11. Raspberry Pi Componentes. Fuente: (Halfacree, 2024).

La evaluación de los tipos de conexiones posibles entre Arduino y Raspberry Pi es un paso crucial para

garantizar la efectividad y funcionalidad del sistema que se está desarrollando. Dado que ambos dispositivos tienen características y capacidades distintas, es fundamental analizar las diversas opciones de comunicación, como la conexión a través de puertos serie, USB, SPI o tecnologías inalámbricas como WiFi y Bluetooth. Cada método de conexión presenta ventajas y desventajas en términos de velocidad, distancia, complejidad de implementación y consumo de energía. Por ejemplo, las conexiones por cable pueden ofrecer una mayor estabilidad y velocidad, mientras que las opciones inalámbricas proporcionan flexibilidad y movilidad. Al evaluar estas conexiones, se podrá seleccionar la más adecuada para el proyecto, asegurando una integración fluida entre los dispositivos y optimizando la transferencia de datos necesarios para el funcionamiento del sistema de control de acceso. Esta evaluación no solo facilitará la comunicación entre Arduino y Raspberry Pi, sino que también contribuirá a la creación de un sistema más robusto y eficiente.

1.1.5. Tipos de conexiones

Wi-Fi:

La conexión por Wi-Fi es una tecnología de comunicación inalámbrica que permite la interconexión de dispositivos electrónicos y el acceso a Internet sin necesidad de cables. A continuación, se detallan sus características, funcionamiento y tipos.

Wi-Fi es un término que se refiere a un conjunto de estándares de comunicación inalámbrica que permiten la transferencia de datos mediante ondas de radio, específicamente bajo el estándar IEEE 802.11. Esta tecnología facilita la conexión de dispositivos como computadoras, teléfonos móviles, tabletas y otros equipos electrónicos a redes locales e Internet.

Características del Wi-Fi:

- Conectividad Inalámbrica: Permite la conexión a redes sin cables físicos, lo que proporciona mayor movilidad y flexibilidad.
- Velocidad y Ancho de Banda: Dependiendo del estándar utilizado (como 802.11n, 802.11ac, o 802.11ax), las velocidades pueden variar significativamente, alcanzando hasta varios gigabits por segundo en los estándares más recientes.
- Rango de Cobertura: Generalmente, el alcance efectivo de una red Wi-Fi es de aproximadamente 100 metros en espacios abiertos, aunque puede ser menor en entornos con obstáculos.
- Seguridad: Las redes Wi-Fi pueden protegerse mediante cifrado (como WPA2 o WPA3) y contraseñas para evitar accesos no autorizados.

Funcionamiento del Wi-Fi:

El funcionamiento del Wi-Fi implica varios componentes clave:

Punto de Acceso (Access Point): Un dispositivo (como un router) que recibe la señal de Internet a través de un cable y la convierte en ondas de radio que pueden ser transmitidas a los dispositivos cercanos.

Dispositivos Conectados: Equipos como laptops, teléfonos y tabletas que cuentan con adaptadores Wi-Fi para recibir y enviar datos a través de estas ondas.

Transmisión de Datos: Los datos se envían en forma de paquetes a través del aire. Cuando un dispositivo envía información, esta se convierte en señales radioeléctricas que el punto de acceso puede interpretar y viceversa.

Ethernet:

La conexión Ethernet es una tecnología de red que permite la comunicación entre dispositivos a través de un cable, formando parte de redes de área local (LAN) o redes de área amplia (WAN). Esta conexión se basa en el estándar IEEE 802.3 y se utiliza ampliamente en entornos domésticos y empresariales debido a su fiabilidad, velocidad y seguridad.

Características de la Conexión Ethernet:

- Transmisión por Cable: La conexión Ethernet utiliza cables para transmitir datos, lo que proporciona una conexión más estable y rápida en comparación con las conexiones inalámbricas.
- Seguridad: Al ser una conexión física, es menos vulnerable a accesos no autorizados, lo que mejora la seguridad de la red.
- Velocidad: Las conexiones Ethernet ofrecen velocidades que varían según el tipo de cable utilizado, desde 100 Mbps hasta 40 Gbps con los cables más avanzados (como Cat 8).
- Baja Latencia: Proporciona tiempos de respuesta más rápidos, lo que es crucial para aplicaciones que requieren alta disponibilidad, como juegos en línea o videoconferencias.

Serial:

La conexión serial es un método de comunicación que permite la transmisión de datos entre dispositivos de manera secuencial, enviando un bit a la vez a través de un solo canal. Este tipo de conexión es ampliamente utilizado en microcontroladores como Arduino y en computadoras como la Raspberry Pi, facilitando la interacción entre ambos.

Características de la Conexión Serial:

- Transmisión Secuencial: Los datos se envían uno tras otro, lo que simplifica el diseño del hardware y el software.
- Bidireccional: Permite que los dispositivos se comuniquen en ambas direcciones, es decir, pueden enviar y recibir datos.

- Simplicidad: Es un método estándar y fácil de implementar, lo que lo hace accesible para proyectos de electrónica y programación.
- Velocidad: La velocidad de transmisión se mide en baudios (baud rate), que indica cuántos bits se pueden enviar por segundo. Comúnmente se utilizan tasas como 9600 bps.

Se ha seleccionado la conexión serial como método de comunicación entre Arduino y Raspberry Pi debido a su simplicidad y eficacia en la transferencia de datos. Este tipo de conexión permite establecer un enlace directo y confiable entre ambos dispositivos, facilitando el intercambio de información en tiempo real. La conexión serial es particularmente útil porque utiliza un solo par de cables para transmitir datos, lo que simplifica el cableado y reduce el riesgo de errores en la conexión. Además, la configuración de la comunicación serial es relativamente sencilla, ya que se puede implementar utilizando bibliotecas estándar en ambos entornos, lo que acelera el proceso de desarrollo. Al establecer una comunicación a través del puerto serie, se pueden enviar comandos y recibir datos de manera eficiente, lo que es crucial para el funcionamiento coordinado del sistema de control de acceso. Esta elección no solo optimiza la interacción entre Arduino y Raspberry Pi, sino que también garantiza una respuesta rápida y precisa en las operaciones del sistema.

La necesidad de definir y analizar la conexión a un servidor web en Raspberry Pi surge de la decisión de utilizar este dispositivo como un servidor web para el sistema de control de acceso. Configurar la Raspberry Pi como un servidor web implica establecer una infraestructura que permita gestionar y servir las solicitudes de acceso a través de una interfaz web. Esto requiere no solo la instalación de software adecuado, como Apache para manejar las peticiones HTTP, sino también la configuración de la red para garantizar que el servidor sea accesible tanto localmente como desde Internet. Además, es esencial considerar aspectos como la seguridad de las conexiones, la gestión de dominios y el redireccionamiento de puertos en el router, lo que permitirá que las solicitudes externas sean dirigidas correctamente a la Raspberry Pi. Al realizar este análisis, se podrá optimizar el rendimiento del servidor y asegurar que el sistema funcione de manera eficiente y segura, facilitando así el acceso controlado a los recursos protegidos.

Conexión a un Servidor Web en Raspberry Pi:

Cuando hablamos de un servidor web en el contexto de una Raspberry Pi, nos referimos a un sistema que permite alojar y servir contenido web (como páginas HTML, imágenes y aplicaciones) a través de Internet o una red local. La Raspberry Pi actúa como un servidor que puede ser accedido por otros dispositivos, como computadoras y teléfonos móviles, mediante una conexión Wi-Fi.

Funcionamiento de la Conexión:

Dirección IP: Cada dispositivo conectado a una red tiene una dirección IP única. Para acceder al servidor web en la Raspberry Pi, el dispositivo cliente (móvil o PC) necesita conocer esta dirección IP.

Protocolo HTTP/HTTPS: La comunicación entre el cliente y el servidor se realiza utilizando protocolos

como HTTP (Hypertext Transfer Protocol) o HTTPS (HTTP Secure). Estos protocolos son responsables de la transferencia de datos entre el servidor y el navegador del cliente.

Navegador Web: Cuando un usuario ingresa la dirección IP del servidor en su navegador, este envía una solicitud al servidor web alojado en la Raspberry Pi. El servidor procesa esta solicitud y devuelve la información solicitada, que se muestra en el navegador del cliente.

Ventajas de Usar Raspberry Pi como Servidor Web:

- Costo Efectivo: La Raspberry Pi es un dispositivo económico que puede funcionar como un servidor ligero para proyectos personales o educativos.
- Bajo Consumo Energético: Consume poca energía en comparación con servidores tradicionales, lo que lo hace ideal para proyectos de larga duración.
- Flexibilidad: Permite experimentar con diferentes tecnologías y lenguajes de programación para el desarrollo web.
- Portabilidad: Su tamaño compacto permite llevarla fácilmente a diferentes ubicaciones.

1.2. Análisis de sistemas homólogos

En la actualidad, los sistemas de control de acceso se están orientado en la implementación de la tecnología RFID ya que es un sistema de autoidentificación inalámbrico, que ha tenido mucho auge en los últimos años debido a la relativa reducción de precios en el mercado. Además, el mundo se acerca cada vez más a la adopción de entornos de infraestructura basados en ciudades inteligentes, en los que la mayoría de las actividades implican conectividad tecnológica innovadora. Por consiguiente, la implementación de esta ciencia se irá fortaleciendo (TECNOSeguro, 2024).

Los autores (Wang y Wang), han propuesto un sistema de control de acceso basado en técnicas RFID que está enfocado principalmente para ser implementado en oficinas pequeñas. Así, el sistema pasa a ser un complemento de seguridad que funciona de manera conjunta con los controles de acceso ya establecidos, haciendo que la organización sea más completa en relación con el sistema de seguridad, por consiguiente, más eficiente en la gestión de personal y aportando mayor confiabilidad. Al mismo tiempo, las tarifas de acceso inteligente RFID son bajas, más adecuadas para espacios de oficinas pequeños y sistemas de control de acceso de la comunidad doméstica.

Por otro lado, la tecnología RFID no es la única disponible y capaz de suplir las falencias de seguridad de áreas restringidas. Por lo tanto, existen otras tecnologías biométricas que se han implementado para mitigar fallos de seguridad en acceso físico, e incluso tecnologías ya existentes que no se les había dado un enfoque como el de control de acceso.

Las tendencias actuales en sistemas de control de acceso se centran en la integración de tecnologías móviles, automatización y IoT, inteligencia artificial y análisis de datos, y la seguridad en general. A continuación, se presentan algunas de las tendencias más destacadas (TECNOSeguro, 2024):

- Integración con Tecnología Móvil
  Uso de dispositivos móviles como credenciales: Los usuarios pueden utilizar directamente sus dispositivos móviles como credenciales de acceso, simplificando el proceso de autenticación y ofreciendo opciones adicionales como la autenticación biométrica y la encriptación de datos.
- Automatización y IoT
  Comunicación entre dispositivos: Los dispositivos de control de acceso pueden comunicarse entre sí y con otros sistemas dentro de un edificio o conjunto de espacios, actuando de manera coordinada para ajustar la climatización, iluminación y alarmas según la ocupación real.
  Adaptación dinámica: La automatización, impulsada por datos recogidos por dispositivos IoT, permite a los sistemas de control de acceso adaptarse y responder de forma dinámica a las necesidades específicas del momento, creando entornos inteligentes y seguros.
- Inteligencia Artificial y Análisis de Datos
  Análisis de datos: La integración de inteligencia artificial y análisis de datos en los sistemas de control de acceso permite una mejor comprensión de las necesidades y comportamientos de los usuarios, lo que puede mejorar la eficiencia y seguridad.
- Seguridad
  Mejora generalizada en la seguridad: Los sistemas de control de acceso modernos ofrecen una mayor seguridad al permitir el uso de tecnologías que facilitan el control de visitas, tanto puntuales como recurrentes, y reducir el peso del factor humano.
  Tecnologías de credenciales seguras: La adopción de tecnologías como OSDP Versión 2.0 o superiores, que notifican cualquier novedad o interferencia en la comunicación con el lector, y credenciales móviles como MiFARE Ev2 o Ev3, que no han sido vulneradas, garantizan la seguridad de la infraestructura.
- Crecimiento de las Credenciales Móviles
  Consolidación de credenciales móviles: Las credenciales móviles, gracias a su practicidad, economía y sostenibilidad, consolidan su posición como el sistema definitivo de ingreso y registro en el Control de Acceso.

Estas tendencias en control de acceso buscan mejorar la seguridad, eficiencia y comodidad en la gestión de entradas y salidas de personas y recursos, adaptándose a las necesidades específicas de cada empresa o espacio (SEGURDOMA, 2024).

Sistemas homólogos Internacional:

Moxa es un sistema de seguridad de red que se utiliza para proteger redes y dispositivos contra ataques

malintencionados. Moxa se enfoca en la seguridad de la infraestructura crítica, como la energía, la agua, la salud y la seguridad, y ofrece soluciones para proteger contra amenazas como la intrusion, la exfiltración de datos y la paralización de sistemas. La empresa se fundó en 1987 en Taiwán y desde entonces ha crecido para convertirse en un líder en la industria de la seguridad industrial.

Moxa se especializa en la creación de soluciones de seguridad para la infraestructura crítica, incluyendo sistemas de autenticación, detección de intrusos, recuperación y otros productos y servicios relacionados con la seguridad. La empresa se enfoca en la creación de soluciones que sean fáciles de implementar, escalables y compatibles con una amplia variedad de sistemas y tecnologías.Moxa, 2024

Sencon SRL se creó en Bolivia en 2022, específicamente el 5 de diciembre de ese año. Sencon SRL ofrece soluciones de seguridad industrial que incluyen:

Monitoreo de la línea de producción: Sencon SRL proporciona sistemas de monitoreo de la línea de producción que permiten a los supervisores ver lo que está sucediendo en una fábrica en tiempo real.

Fabricantes de equipos originales de máquinas (OEM): Sencon SRL ayuda a los fabricantes de equipos originales a conectar equipos desconectados utilizando Internet de las cosas (IoT).

Sistemas de seguridad industrial: Sencon SRL ofrece sistemas de seguridad industrial que incluyen tecnologías como circuitos cerrados de televisión, control de accesos y detección de intrusos para proteger la integridad física del personal, el equipamiento y las instalaciones.

Detección de gas y fuego: Sencon SRL utiliza la última tecnología para detectar gas y fuego en las instalaciones industriales.

Sistemas de seguridad industrial: Sencon SRL selecciona la tecnología precisa para cumplir con las normas de seguridad más exigentes según las particularidades de cada industria y su grado de complejidad específico.

Estos sistemas de seguridad industrial de Sencon SRL se enfocan en preservar la integridad física del personal, el equipamiento y las instalaciones de la organización, minimizando los riesgos y protegiendo la vida útil de los recursos más importantes.SRL, 2024

Schneider Electric: Schneider Electric ofrece soluciones integrales de seguridad para la infraestructura crítica, incluyendo sistemas de control y monitoreo que aseguran la protección de redes eléctricas y plantas industriales. Su enfoque se centra en la automatización y la eficiencia energética, integrando medidas de seguridad cibernética para proteger contra amenazas. Electric, 2024

Siemens: Siemens proporciona una amplia gama de soluciones de seguridad industrial que abarcan desde la protección física hasta la ciberseguridad. Sus sistemas están diseñados para proteger las instalaciones industriales y garantizar la continuidad operativa mediante tecnologías avanzadas de detección y respuesta

ante incidentes (Siemens, 2024).

Honeywell: Honeywell es un líder en soluciones de seguridad industrial, ofreciendo sistemas que incluyen monitoreo de procesos, control de acceso y detección de intrusos. Su tecnología se utiliza en diversas industrias para proteger activos críticos y mejorar la seguridad operativa.

Tabla 1.1. Comparativa de Sistemas homólogos internacionales. Fuente: Elaboración propia.

Sistema	**Enfoque Principal**	**Características Clave**
Moxa	Seguridad de redes y dispositivos en infraestructura crítica	Soluciones para autenticación, detección de intrusos, y recuperación ante desastres
Sencon SRL	Seguridad industrial y monitoreo en tiempo real	Sistemas de CCTV, control de accesos, detección de gas y fuego
Schneider Electric	Automatización y eficiencia energética en infraestructura crítica	Integración de ciberseguridad con sistemas de control y monitoreo
Siemens	Soluciones integrales para la protección física y ciberseguridad	Sistemas avanzados de detección y respuesta ante incidentes
Honeywell	Monitoreo de procesos y control de acceso en diversas industrias	Tecnología para protección de activos críticos y mejora de la seguridad operativa

Sistemas homólogos Nacional:

Sistema de Identificación: Este sistema brinda un servicio de certificación de identidad a otros sistemas informáticos, como los destinados al control del acceso. Tiene almacenados los datos de todo el personal que labora y estudia en la UCI: estudiantes y todo tipo de trabajadores.

Sistema de Control de Acceso a Comedores: Mediante este sistema se controla en los comedores de las diferentes edificaciones donde se brinda el servicio de alimentación; el acceso de los estudiantes, profesores y trabajadores durante las tres sesiones de servicio: desayuno, almuerzo y comida. El mismo se divide en dos partes: el control de acceso y la gestión de comensales. El acceso se controla registrando el código de barras, que se encuentra en la identificación de cada persona, en cada una de las puertas de los comedores. La gestión de comensales permite a los directivos la asignación de los comedores y puertas a los mismos, además de ofrecer reportes como la cantidad de comensales desglosado por puerta o tipo.

Tabla 1.2. Comparativa de Sistemas homólogos nacionales. Fuente: Elaboración propia.

Sistema	**Enfoque Principal**	**Características Clave**
Sistema de Identificación	Certificación de identidad para sistemas informáticos	Almacena datos de personal de la UCI, incluyendo estudiantes y trabajadores
Sistema de Control de Acceso a Comedores	Control de acceso en comedores universitarios	Registro mediante código de barras, gestión de comensales y reportes de asistencia

Valoración de los sistemas estudiados

Luego del estudio realizado a los sistemas de control de acceso descritos anteriormente se logró identificar varias de las funcionalidades y características comunes de estos software:

El control de horarios y los permisos concedidos.

La capacidad de generar informes de eventos de identificación, administración y estaciones de trabajo, esto es sumamente útil a la hora de extraer históricos sobre cualquier evento ocurrido en el local de trabajo.

Positivos:

Seguridad Mejorada: Tanto Moxa como Sencon SRL y los sistemas nacionales ofrecen soluciones que integran tecnologías avanzadas para proteger infraestructuras críticas y garantizar la seguridad física de los usuarios y activos. La implementación de sistemas de control de acceso, como RFID y biometría, proporciona un nivel de seguridad superior al permitir un control preciso sobre quién puede acceder a áreas restringidas.

Adaptabilidad y Escalabilidad: Los sistemas analizados son escalables, lo que permite su implementación en diferentes entornos, desde oficinas pequeñas hasta grandes instalaciones industriales. Esto es crucial para organizaciones que pueden expandirse o que requieren ajustes en sus sistemas de acceso.

Integración Tecnológica: La capacidad de integrar tecnologías móviles, IoT y análisis de datos en los sistemas de control de acceso mejora la eficiencia operativa y permite una gestión más inteligente del acceso. Esto se refleja en la automatización de procesos y la comunicación entre dispositivos.

Costos Asequibles: La reducción en los precios de tecnologías como RFID hace que estas soluciones sean accesibles para una variedad de empresas, facilitando su adopción en entornos donde antes no eran viables.

Negativos:

Preocupaciones sobre Privacidad: La recopilación y almacenamiento de datos biométricos y otros datos personales pueden generar preocupaciones sobre la privacidad y el uso indebido de la información.

Dependencia Tecnológica: La implementación de estos sistemas requiere una infraestructura tecnológica sólida. Cualquier fallo en el sistema puede interrumpir el acceso y las operaciones normales.

Costos Iniciales: Aunque los costos han disminuido, la inversión inicial en tecnología avanzada y capacitación del personal puede ser significativa, especialmente para pequeñas empresas.

Falsos Positivos/Negativos: Los sistemas biométricos pueden tener tasas de error, lo que podría resultar en accesos no autorizados o bloqueos indebidos a usuarios legítimos.

Resumen de Utilidad para este Proyecto de Control de Acceso:

La información sobre los sistemas homólogos proporciona valiosas lecciones para el desarrollo de un sistema de control de acceso basado en RFID, huellas dactilares y reconocimiento facial:

- Integración Multitecnológica: La combinación de RFID con tecnologías biométricas (huellas dactilares y reconocimiento facial) puede ofrecer un enfoque más robusto para el control de acceso, mejorando la seguridad general al requerir múltiples formas de autenticación.
- Adaptabilidad a Entornos Inteligentes: Con el auge de las ciudades inteligentes y la automatización, un sistema que incorpore RFID y biometría puede adaptarse fácilmente a cambios en las necesidades operativas y mejorar la gestión del acceso a través del análisis de datos.
- Eficiencia Operativa: Implementar un sistema que permita el uso de dispositivos móviles como credenciales junto con RFID puede simplificar el proceso de autenticación, aumentando la comodidad para los usuarios y reduciendo tiempos de espera.
- Seguridad Proactiva: Al adoptar tecnologías que permiten una supervisión constante (como las ofrecidas por Moxa o Sencon), se puede implementar un sistema que no solo controle el acceso, sino que también responda a incidentes en tiempo real, mejorando la seguridad general del local.
- Costos Asequibles: La tendencia hacia la reducción de costos en tecnologías RFID hace viable su implementación en entornos pequeños o medianos, lo cual es esencial para maximizar el retorno sobre la inversión en seguridad.

En conclusión, al estudiar estos sistemas homólogos se pueden identificar estrategias efectivas para desarrollar un sistema integral que combine RFID, biometría y otras tecnologías emergentes, garantizando así un control de acceso eficiente y seguro para cualquier local físico.

1.3. Metodología de desarrollo de software

La metodología XP (Extreme Programming) es un enfoque ágil para el desarrollo de software que se centra en la mejora continua, la flexibilidad y la calidad del producto final. A continuación, se presenta una descripción detallada de sus características, fases, valores y su utilidad en un proyecto de control de acceso

(Ginzo, 2022).

Características de la Metodología XP:

Adaptabilidad: XP se basa en la idea de que los requisitos pueden cambiar durante el desarrollo. Esto permite al equipo adaptarse a nuevas necesidades o cambios en las prioridades del cliente.

Enfoque en la Calidad: La metodología promueve prácticas de ingeniería que aseguran un software de alta calidad, utilizando pruebas automatizadas y revisiones constantes del código.

Desarrollo Iterativo: Se trabaja en ciclos cortos (iteraciones), lo que permite entregar versiones funcionales del software regularmente y recibir retroalimentación constante.

Comunicación Abierta: Fomenta una comunicación constante entre todos los miembros del equipo y con los clientes, eliminando barreras entre desarrollo y negocio.

Simplicidad: Se prioriza la solución más simple que funcione, evitando complicaciones innecesarias en el diseño y desarrollo.

Fases de la Metodología XP:

Planificación: Se identifican las historias de usuario (requisitos del cliente) y se priorizan. Se descomponen en versiones más pequeñas que se revisan regularmente.

Diseño: Se crea un diseño simple y funcional. Se utilizan tarjetas CRC (Clase-Responsabilidad-Colaboración) para definir las interacciones entre componentes.

Codificación: La programación se realiza en parejas (programación en pareja), lo que mejora la calidad del código y facilita el aprendizaje entre los desarrolladores.

Pruebas: Las pruebas son continuas y automáticas, asegurando que cada parte del software funcione correctamente antes de añadir nuevas funcionalidades.

Lanzamiento: Se entrega el software al cliente tras probar todas las historias de usuario, garantizando que cumple con los requisitos establecidos (ibíd.).

Utilidad de XP para este Proyecto de Control de Acceso

Implementar la metodología XP en un proyecto de control de acceso, como uno que utilice tecnologías RFID, huellas dactilares y reconocimiento facial, puede ofrecer varias ventajas:

- Adaptación a Cambios Rápidos: En un entorno donde las necesidades pueden cambiar rápidamente (por ejemplo, nuevas regulaciones o requisitos de seguridad), XP permite ajustar el desarrollo para satisfacer estas demandas sin grandes retrasos.

- Mejora Continua del Producto: Con ciclos cortos de desarrollo e iteraciones regulares, el sistema puede evolucionar constantemente basándose en la retroalimentación del usuario final, asegurando que el producto final sea realmente útil y eficiente.
- Alta Calidad del Software: La incorporación de pruebas continuas garantiza que cada componente del sistema funcione correctamente antes de ser implementado, lo cual es crucial para sistemas críticos como el control de acceso.
- Colaboración Efectiva: La comunicación abierta entre desarrolladores y clientes facilita una comprensión clara de los requisitos del sistema, lo que resulta en un producto más alineado con las expectativas del usuario.
- Simplicidad en el Diseño: Al enfocarse en soluciones simples, el sistema puede ser más fácil de mantener y escalar a medida que crecen las necesidades del negocio o se añaden nuevas funcionalidades.

En resumen, aplicar la metodología XP a un proyecto de control de acceso puede resultar en un desarrollo más ágil, eficiente y centrado en el usuario, asegurando que el sistema sea seguro, funcional y capaz de adaptarse a futuros cambios o requerimientos tecnológicos.

La metodología Extreme Programming (XP) se ha seleccionado para este proyecto debido a su enfoque ágil y adaptabilidad, características que son esenciales en el desarrollo de un sistema de control de acceso. En un entorno donde los requisitos pueden cambiar rápidamente, XP permite al equipo responder de manera efectiva a nuevas demandas y prioridades del cliente, asegurando que el producto final cumpla con sus expectativas. Además, la metodología promueve la calidad del software a través de prácticas como las pruebas continuas y la programación en pareja, lo que resulta en un código más robusto y menos propenso a errores. La comunicación abierta entre los miembros del equipo y con los clientes facilita una comprensión clara de los requisitos, lo que contribuye a un desarrollo más alineado con las necesidades reales del usuario. En resumen, la implementación de XP no solo optimiza el proceso de desarrollo al hacerlo más ágil y eficiente, sino que también garantiza un sistema de control de acceso seguro y funcional que puede adaptarse a futuros cambios tecnológicos o normativos.

1.4. Herramientas y tecnologías

A través del soporte tecnológico se proporciona una serie de recursos a emplear con la finalidad de lograr un producto final que garantice un buen proceso de gestión de la información. Es por ello que se necesita realizar un análisis previo para identificar todos los posibles recursos a estar implícitos en la solución final.

Herramienta CASE

Se puede definir a las herramientas de Ingeniería de Software Asistida por Computadora (CASE por sus siglas en inglés) como cualquier herramienta que se emplea para automatizar alguna actividad.

Visual Paradigm es la herramienta CASE utilizada para generar los modelos y diagramas ingenieriles del proyecto. Esta herramienta soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, implementación y pruebas. Ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite construir diagramas de diversos tipos, código inverso, generar código desde diagramas y generar documentación (Cevallos, 2015).

Lenguaje de modelado

El Lenguaje Unificado de Modelado (UML) es un lenguaje gráfico para visualizar, especifi-car y documentar cada una de las partes que comprende el desarrollo de software. UML en-trega una forma de modelar casos conceptuales como lo son procesos de negocio y funcio-nes de sistema, además de casos concretos como lo son escribir clases en un lenguaje de-terminado, esquemas de base de datos y componentes de software reusables. El lenguaje unificado de modelado es una de las herramientas más empleadas en el mundo actual del desarrollo de sistemas. Esto es debido a que permite a los creadores de sistemas generar diseños que capturen sus ideas en una forma convencional fácil de comprender para comu-nicarlas a otras personas.

Entorno de desarrollo integrado

Visual Studio Code(versión 1.78.2):

Visual Studio Code (VS Code) es un editor de código fuente desarrollado por Microsoft. Es conocido por su ligereza y potencia, permitiendo a los desarrolladores trabajar en múltiples lenguajes de programación gracias a su amplia gama de extensiones. Algunas de sus características incluyen:

Soporte para múltiples lenguajes: Incluye soporte incorporado para JavaScript, TypeScript, Python, C++, y más.

Extensiones: Permite la instalación de extensiones para agregar funcionalidades adicionales.

Integración con Git: Ofrece control de versiones integrado.

Depuración: Herramientas para depurar aplicaciones directamente desde el editor.

Framework

Django(versión 5.1):

Django es un framework web de alto nivel para Python que fomenta el desarrollo rápido y limpio. Se basa en el patrón arquitectónico Modelo-Vista-Controlador (MVC), lo que facilita la creación de aplicaciones web robustas. Sus características incluyen:

Desarrollo rápido: Permite construir aplicaciones complejas rápidamente.

Seguridad: Proporciona herramientas integradas para proteger aplicaciones contra ataques comunes.

Escalabilidad: Ideal para proyectos que requieren crecer con el tiempo.

Lenguaje de programación

Python(versión 3.12.5)

Python es un lenguaje de programación interpretado, conocido por su legibilidad y simplicidad. Es ampliamente utilizado en diversas áreas como desarrollo web, análisis de datos y aprendizaje automático. Algunas características son:

Sintaxis clara y concisa: Facilita el aprendizaje para principiantes.

Bibliotecas extensas: Ofrece una amplia gama de bibliotecas para diferentes aplicaciones.

Comunidad activa: Gran soporte y recursos disponibles.

C++

C++ es un lenguaje de programación multipropósito que se utiliza ampliamente en el desarrollo de software que requiere alto rendimiento, como juegos y sistemas operativos. Sus características principales incluyen:

Orientación a objetos: Facilita la creación de programas modulares y reutilizables.

Control sobre recursos del sistema: Permite optimizar el uso del hardware.

Compatibilidad con C: Facilita la transición para programadores familiarizados con C.

JavaScript

JavaScript es un lenguaje de programación esencial para el desarrollo web, permitiendo la creación de contenido interactivo en las páginas web. Sus características incluyen:

Interactividad en el navegador: Permite actualizar contenido sin recargar la página.

Compatibilidad con todos los navegadores modernos: Se ejecuta en cualquier navegador que soporte JavaScript.

Ecosistema rico: Amplia variedad de bibliotecas y frameworks como React y Angular.

HTML5

HTML (HyperText Markup Language) es el estándar para crear páginas web. Define la estructura del contenido web mediante etiquetas. Sus características son:

Estructura básica del contenido web: Utiliza etiquetas para organizar texto, imágenes y otros elementos multimedia.

Facilidad de uso: Fácilmente entendible incluso para principiantes en programación.

Compatibilidad universal: Todos los navegadores pueden interpretar HTML.

Gestor de base de datos

PostgreSQL14

PostgreSQL es un sistema de gestión de bases de datos relacional objeto que se destaca por su robustez y flexibilidad. Sus características incluyen:

Soporte para SQL avanzado: Permite consultas complejas y manejo eficiente de datos.

Extensibilidad: Se pueden crear tipos de datos personalizados, funciones y más.

Alta disponibilidad y escalabilidad: Ideal para aplicaciones empresariales que requieren fiabilidad.

Hadware

La selección del hadware adecuado es un pilar fundamental para el proyecto, luego del estudio a las tecnologías de control de acceso además de a Raspberry pi y Arduino se realizó la elección de los componentes a emplear.

Escáner de huellas dactilares TTL (GT-511C3)

Se trata de un escáner de huellas dactilares con comunicación TTL. El módulo lee e identifica las huellas dactilares mediante un sensor óptico y una CPU integrados.

Figura 1.12. Escáner de huellas dactilares
Fuente: (Circuito.io, n.d.)

Arduino Uno - R3

La UNO es la mejor placa para iniciarse en la electrónica y la codificación. Si esta es tu primera experiencia trasteando con la plataforma, la UNO es la placa más robusta con la que puedes empezar a jugar. La UNO es la placa más utilizada y documentada de toda la familia Arduino y Genuino. Arduino/Genuino Uno es una placa de microcontrolador basada en el ATmega328P (hoja de datos) (Circuito.io, n.d.).

Tiene 14 pines de entrada/salida digitales (de los cuales 6 se pueden utilizar como salidas PWM), 6 entradas analógicas, un cristal de cuarzo de 16 MHz, una conexión USB, un conector de alimentación, un conector ICSP y un botón de reinicio.

Contiene todo lo necesario para soportar el microcontrolador; simplemente conéctalo a una computadora con un cable USB o aliméntalo con un adaptador de CA a CC o una batería para comenzar.

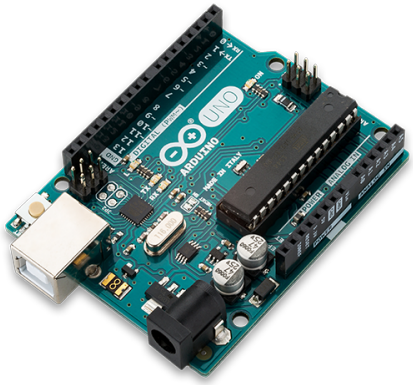


Figura 1.13. Arduino Uno - R3 Fuente: (Circuito.io, n.d.)

ESP32-CAM:

La cámara ESP32-CAM es un módulo de desarrollo que combina un microcontrolador ESP32 con una cámara OV2640, permitiendo la creación de aplicaciones que requieren captura de imágenes y transmisión de video. Este dispositivo es popular en proyectos de Internet de las Cosas (IoT) debido a su bajo costo y versatilidad.

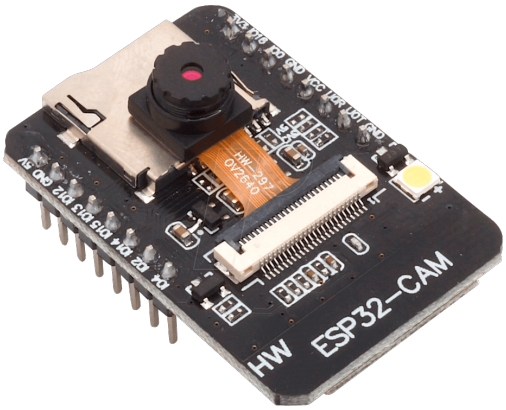


Figura 1.14. ESP32-CAM Fuente: (Circuito.io, n.d.)

Módulo de Lector RFID RC522:

El módulo de lector RFID RC522 es un dispositivo que permite leer y escribir etiquetas RFID utilizando la tecnología MIFARE. Es un módulo popular entre los makers y es compatible con Arduino y otros microcontroladores que utilizan la interfaz SPI, I2C, o UART. A continuación, se presentan sus características clave (Valle Hernández, 2019):

- Voltaje de operación: 2.5V a 3.3V.
- Corriente de operación: 13-26 mA.
- Corriente en reposo: <80 uA.
- Corriente pico: <30 mA.
- Frecuencia de operación: 13.56 MHz.
- Interfaz: SPI, I2C, UART.
- Alcance: 5 cm.
- Niveles lógicos: 5V y 3V3.
- Dimensiones: 40 mm x 60 mm.
- Temperatura de operación: -20°C a 80°C.
- Humedad relativa: 5
- Tipos de tarjetas compatibles: Mifare1 S50, S70 Mifare1, MIFARE Ultralight, Mifare Pro, Mifare DESFire.

Figura 1.15. Lector RFID RC522 Fuente: (Circuito.io, n.d.)

Este módulo es ampliamente utilizado en proyectos de control de acceso, seguridad electrónica y trazabilidad, y es compatible con una amplia variedad de microcontroladores y sistemas (Valle Hernández, 2019).

Solenoide tipo cerradura de 12 V CC

Este solenoide tiene una babosa con un corte inclinado y un soporte de montaje. Es una cerradura electrónica para gabinetes, cajas fuertes o puertas. Le permite mantener una puerta cerrada sin electricidad y tira de la babosa cuando se aplica corriente.Circuito.io, n.d.

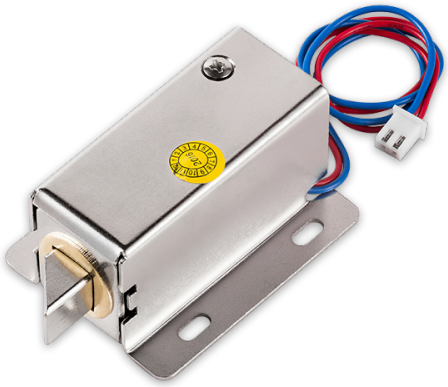


Figura 1.16. Solenoide Fuente: (Circuito.io, n.d.)

Raspberry Pi 3

La Raspberry Pi 3 es un modelo de computadora de placa única (SBC) que fue lanzado en febrero de 2016. Se caracteriza por su procesador ARM Cortex-A53 de cuatro núcleos a 1.2 GHz, lo que le proporciona un rendimiento mejorado en comparación con sus predecesores. Este dispositivo compacto, del tamaño de una tarjeta de crédito, incluye conectividad Wi-Fi y Bluetooth integradas, lo que facilita su uso en proyectos que requieren comunicación inalámbrica. La Raspberry Pi 3 es compatible con una amplia variedad de sistemas operativos, siendo Raspberry Pi OS (anteriormente conocido como Raspbian) el más popular, aunque también puede ejecutar otros sistemas basados en Linux y Windows 10 IoT Core (Circuito.io, n.d.).

Figura 1.17. Raspberry Pi 3 Fuente: (Circuito.io, n.d.)

Conclusiones del capítulo:

La implementación de sistemas de control de acceso avanzados, como RFID y biometría, no solo mejora la protección de los activos, sino que también optimiza la gestión operativa al permitir un control preciso sobre quién puede acceder a áreas restringidas.

Sin embargo, es crucial considerar los desafíos asociados con estas tecnologías, incluyendo preocupaciones sobre privacidad y dependencia tecnológica.

La combinación de RFID y biometría proporciona un nivel de seguridad superior, ya que ambas tecnologías son difíciles de falsificar. La biometría utiliza características únicas del individuo, mientras que RFID permite un acceso rápido y eficiente a áreas restringidas.

Los sistemas RFID permiten llevar un registro detallado de quién accede a qué áreas y cuándo, lo que facilita la auditoría y el cumplimiento normativo. Esta trazabilidad es crucial para la seguridad y la gestión de riesgos.

CAPÍTULO 2

DISEÑO DE LA SOLUCIÓN PROPUESTA AL PROBLEMA CIENTÍFICO

El objetivo de este capítulo es presentar una propuesta de solución integral para el control de acceso físico en áreas restringidas, utilizando tecnologías avanzadas como RFID, reconocimiento de huella dactilar y reconocimiento facial. Se abordarán los métodos de autenticación disponibles, el proceso de acceso y validación, así como la importancia del sistema propuesto en términos de seguridad y eficiencia. La estructura del capítulo se divide en secciones que incluyen la descripción del sistema, las funcionalidades específicas, los requisitos necesarios para su implementación, el diseño arquitectónico y un plan de entregas.

2.1. Propuesta de solución

Como solución a la problemática se propone la implementación de un sistema que permitirá informatizar el proceso de control de acceso físico utilizándose la tecnología RFID, reconocimiento de huella dactilar y reconocimiento facial empleando múltiples métodos de autenticación para garantizar la seguridad en áreas restringidas. Este sistema tiene como principal objetivo establecer un control del personal que accede a un determinado local. Este enfoque proporciona un sistema robusto de control de acceso, permitiendo una gran flexibilidad y seguridad en la gestión de áreas restringidas.

Funcionalidad:

Métodos de Autenticación: El administrador del sistema determinará qué método o combinación de métodos de autenticación se aplicará en cada área restringida. Las opciones incluyen:

- Huella Dactilar: Los usuarios deberán escanear su huella en un sensor especializado.
- Tarjeta RFID: Los usuarios presentarán una tarjeta o llavero RFID ante el lector.
- Reconocimiento Facial: Se capturará una imagen del rostro del usuario a través de una cámara.

Proceso de Acceso:

Cuando un usuario intente acceder a un área restringida, el sistema activará el método de autenticación definido por el administrador para esa ubicación.

El usuario presentará su método de acceso (huella, tarjeta o rostro) al sensor correspondiente.

Validación:

Para la huella dactilar: El sensor escaneará la huella y la comparará con la base de datos.

Para RFID: El lector captará el número de la tarjeta y buscará una coincidencia en la base de datos.

Para reconocimiento facial: La cámara capturará la imagen del rostro y la comparará con las imágenes almacenadas en el sistema.

Decisión de Acceso:

Si el método de autenticación seleccionado por el administrador es exitoso, el sistema permitirá el acceso al usuario.

En caso de que la autenticación falle, se le negará el acceso y se registrará el intento en la base de datos.

Registro de Eventos:

Todas las sesiones de acceso (tanto exitosas como fallidas) serán registradas en una base de datos, incluyendo detalles como la fecha, hora, identificador del usuario y el método utilizado. El sistema podrá generar reportes o informes sobre el uso del sistema.

Interfaz Web:

Los administradores podrán acceder a una interfaz web alojada en la Raspberry Pi, lo que les permitirá:

Ver los registros de acceso y salida en tiempo real.

Gestionar los usuarios, incluyendo la adición y eliminación de cuentas (podrán agregar, editar o eliminar usuarios y sus permisos de acceso).

Configurar el método de autenticación para diferentes áreas según las necesidades de seguridad.

Conexión a la Raspberry Pi:

La Raspberry Pi actuará como un servidor web al que se podrá acceder desde cualquier dispositivo conectado a la misma red Wi-Fi.

Los usuarios podrán ingresar a la documentación web utilizando una dirección IP estática asignada a la Raspberry Pi.

Importancia del Sistema Propuesto:

Ante los desafíos actuales en cuanto a la seguridad, la implementación de un sistema moderno de control de acceso basado en tecnologías avanzadas como RFID, reconocimiento de huella dactilar y reconocimiento facial se convierte en una necesidad urgente para la UCI. Este sistema no solo abordará las deficiencias actuales, sino que también ofrecerá beneficios significativos:

Seguridad Aumentada: Al restringir el acceso a laboratorios y centros de desarrollo solo al personal autorizado, se protege la propiedad intelectual y los recursos críticos. La autenticación multifactorial asegura que solo aquellos con permisos específicos puedan acceder a áreas sensibles.

Monitoreo Efectivo: El sistema registrará detalladamente todas las entradas y salidas, proporcionando informes en tiempo real sobre quién accede a qué áreas y cuándo. Esto permitirá a los administradores identificar patrones inusuales o intentos no autorizados de acceso.

Agilidad en el Acceso: La utilización de tecnologías sin contacto como RFID y biometría permitirá un acceso rápido y eficiente a los laboratorios. Esto es especialmente importante en un entorno académico donde el tiempo es esencial para la investigación y el desarrollo.

Adaptabilidad: El sistema podrá ser configurado para diferentes niveles de seguridad según la sensibilidad del área. Por ejemplo, los laboratorios que manejan información crítica podrán tener métodos de autenticación más estrictos que otros espacios.

Integración Tecnológica: La implementación del sistema no solo moderniza el control de acceso, sino que también se alinea con la visión tecnológica de la UCI. Al adoptar soluciones innovadoras, la universidad se posiciona como un referente en el uso eficiente de tecnologías emergentes.

La definición de requisitos, análisis y modelado del sistema es una fase crucial en el desarrollo de proyectos, especialmente en el ámbito de la ingeniería de sistemas y el desarrollo de software. Esta etapa permite establecer una base sólida para el diseño y la implementación del sistema, asegurando que se satisfacen las necesidades del cliente y se cumplen los objetivos del proyecto.

2.2. Requisitos del sistema

Los requisitos de un sistema son especificaciones que definen las funcionalidades y características que debe cumplir dicho sistema para satisfacer las necesidades de los usuarios y las expectativas de las partes interesadas. Estos requisitos se dividen en dos categorías principales: requisitos funcionales y requisitos no funcionales.

Requisitos funcionales

Los requisitos funcionales son especificaciones detalladas que definen las acciones, comportamientos y funcionalidades que un sistema debe ser capaz de realizar para cumplir con su propósito previsto. En el contexto del desarrollo de software, estos requisitos son fundamentales para garantizar que el sistema satisfaga las necesidades de los usuarios y cumpla con las expectativas de las partes interesadas. A diferencia de los requisitos no funcionales, que se centran en aspectos como el rendimiento y la seguridad, los requisitos funcionales se enfocan en lo que el sistema debe hacer, incluyendo tareas específicas, interacciones entre el usuario y el sistema, y cómo debe responder a diversas entradas o eventos (Solutions, 2024).

Los requisitos funcionales pueden incluir una variedad de elementos, como la capacidad del sistema para procesar pagos, enviar correos electrónicos de confirmación o recuperar datos de usuario. Por ejemplo, un requisito funcional podría especificar que .[el] sistema enviará un correo electrónico de confirmación al usuario después de que haya realizado un pedido con éxito", donde la función es enviar el correo y el comportamiento es hacerlo tras una acción específica (PMOInformatica, 2018). Estos requisitos no solo describen las características del sistema, sino que también establecen un marco para el desarrollo y la implementación del software.

Es esencial que los requisitos funcionales sean claros, específicos y medibles. Esto asegura que todos los miembros del equipo de desarrollo comprendan lo que se espera del sistema y puede ayudar a evitar malentendidos que podrían llevar a fallos en el proyecto. La identificación y gestión adecuadas de estos requisitos son críticas para el éxito de cualquier proyecto de software, ya que una gestión deficiente puede ser una de las causas más comunes de fracaso en los mismos (ibíd.). En resumen, los requisitos funcionales son una parte integral del proceso de desarrollo de software. Proporcionan una descripción clara y concisa de lo que debe hacer el sistema, asegurando que se cumplan las necesidades del usuario y se logren los objetivos comerciales. Su correcta definición y gestión son fundamentales para desarrollar productos de software efectivos y satisfactorios.

Tabla 2.1. Requerimientos funcionales. Fuente: Elaboración propia

No.	Nombre	Descripción	Prioridad	Complejidad
RF1	Autenticar usuario	El sistema debe permitir autenticarse al administrador (usuario y contraseña).	Alta	Media
RF2	Cerrar sesión	El sistema debe permitir que el administrador cierre sesión.	Alta	Baja
RF3	Listar usuarios	El sistema debe permitir mostrar una lista de los usuarios (Nombre, apellidos, edad, categoría).	Alta	Baja
RF4	Agregar usuario	El sistema debe permitir registrar un usuario (Nombre, apellidos, edad, categoría).	Alta	Media
RF5	Eliminar usuario	El sistema debe permitir eliminar un usuario.	Alta	Baja
RF6	Modificar usuario	El sistema debe permitir modificar un usuario (Nombre, apellidos, edad, categoría).	Alta	Media

No.	Nombre	Descripción	Prioridad	Complejidad
RF7	Denegar permisos	El sistema debe permitir quitar los permisos de acceso al local de un usuario.	Alta	Media
RF8	Realizar búsquedas	El sistema debe permitir realizar búsquedas de usuarios por nombre y apellidos.	Media	Media
RF9	Conceder permisos	El sistema debe permitir autorizar el acceso a usuarios.	Alta	Alta
RF10	Agregar tarjeta	El sistema debe permitir generar una tarjeta (código RFID) a un usuario.	Alta	Alta
RF11	Eliminar tarjeta	El sistema debe permitir eliminar tarjeta (código RFID asignado).	Alta	Media
RF12	Procesar Tarjeta RFID	El sistema debe leer la información de las tarjetas (llaveros) y procesarla.	Alta	Alta
RF13	Asignar tarjeta	El sistema debe permitir asignar una tarjeta al usuario.	Alta	Media
RF14	Registrar acceso	El sistema debe guardar un registro del ingreso, salida o intento de acceso de un usuario al local (Nombre, apellidos, estado, categoría, hora, día, método).	Alta	Media
RF15	Listar Registros de acceso	El sistema debe permitir mostrar una lista de los registros (Nombre, apellidos, estado, categoría, hora, día, método).	Alta	Media
RF16	Configurar acceso	El sistema debe permitir configurar el método de autenticación a emplear.	Alta	Media
RF17	Visualizar usuario	El sistema debe permitir visualizar los datos de un usuario.	Media	Alta
RF18	Actualizar registros	El sistema debe eliminar los registros con un mes de antiguedad.	Media	Alta
RF19	Listar Tarjetas	El sistema debe mostrar una lista de las tarjetas.	Media	Media

Por otro lado, los requisitos no funcionales se refieren a las características de calidad del sistema, es decir, cómo debe funcionar el sistema en lugar de qué debe hacer. Estos requisitos abarcan aspectos como el rendimiento, la seguridad, la usabilidad, la escalabilidad y la fiabilidad. Por ejemplo, un requisito no funcional podría indicar que el sistema debe cargar todas las páginas en menos de 2 segundos o debe ser accesible para usuarios con discapacidades (ScopeMaster, 2024).

Requisitos no funcionales

Los requisitos no funcionales son importantes porque afectan la experiencia del usuario y la efectividad general del sistema. Si bien los requisitos funcionales aseguran que se implementen las características necesarias, los no funcionales garantizan que estas características se entreguen de manera eficiente y efectiva.

PMOInformatica, 2018 Además, estos requisitos pueden clasificarse en varias categorías, como:

Rendimiento: Rapidez y eficiencia del sistema.

Seguridad: Protección contra accesos no autorizados.

Usabilidad: Facilidad de uso del sistema.

Escalabilidad: Capacidad del sistema para manejar un aumento en la carga de trabajo.

Mantenibilidad: Facilidad con la que se puede modificar el sistema (Solutions, 2024).

Para obtener requisitos no funcionales en el desarrollo de sistemas, se pueden seguir diversas normas y guías que facilitan su identificación y especificación. Una de las normas más reconocidas es la ISO/IEC 25010, que proporciona un marco para la evaluación de la calidad del software y define los atributos de calidad que deben considerarse durante el desarrollo.

La norma ISO/IEC 25010 establece un modelo de calidad que incluye dos categorías principales: características de calidad del producto y características de calidad en uso. Esta norma ayuda a los equipos de desarrollo a identificar y definir requisitos no funcionales al proporcionar una lista clara de atributos que pueden ser medidos y evaluados.

Características de Calidad del Producto

Funcionalidad: Se refiere a las capacidades del sistema para cumplir con los requisitos funcionales.

Rendimiento: Incluye aspectos como la eficiencia, el tiempo de respuesta y la utilización de recursos.

Compatibilidad: La capacidad del sistema para interactuar con otros sistemas o componentes.

Usabilidad: La facilidad con la que los usuarios pueden aprender a utilizar el sistema y su satisfacción al hacerlo.

Fiabilidad: Incluye la disponibilidad, la tolerancia a fallos y la capacidad de recuperación.

Seguridad: La protección contra accesos no autorizados y la integridad de los datos.

Mantenibilidad: Facilidad con la que se puede modificar el sistema para corregir defectos o mejorar su rendimiento.

Portabilidad: La capacidad del sistema para ser transferido a diferentes entornos.

Características de Calidad en Uso

Eficiencia en el uso: Relación entre el rendimiento del sistema y los recursos utilizados.

Satisfacción del usuario: Grado en que los usuarios están satisfechos con el uso del sistema.

Accesibilidad: Facilidad con la que personas con discapacidades pueden utilizar el sistema.

RNF1 Requisitos de Rendimiento

RNF1.1 Tiempo de Respuesta: El sistema debe procesar la autenticación y proporcionar una respuesta (acceso permitido o denegado) en menos de 2 segundos por usuario para garantizar un flujo eficiente en áreas con alto tráfico.

RNF1.2 Latencia Mínima: La latencia del sistema, desde el momento en que un usuario presenta su método de autenticación hasta que se recibe la respuesta, no debe superar los 1.5 segundos en condiciones óptimas.

RNF1.3 Actualización en Tiempo Real: Los datos sobre el estado de acceso (como entradas y salidas) deben actualizarse en la base de datos en tiempo real, garantizando que la información esté siempre disponible para los administradores.

RNF1.4 Monitoreo y Reportes en Tiempo Real: Debe permitir a los administradores generar reportes y monitorear el estado del sistema en tiempo real, con actualizaciones que no excedan los 5 segundos entre cada visualización.

RNF2 Requisitos de Seguridad

RNF2.1 Cifrado de Datos: Todos los datos transmitidos entre los dispositivos (lectores y servidores) deben estar cifrados utilizando estándares como AES-256 para proteger la información sensible del usuario.

RNF2.2 Autenticación Multifactor: El sistema debe soportar múltiples métodos de autenticación (huella dactilar, RFID, reconocimiento facial) y permitir configuraciones personalizadas para diferentes áreas restringidas.

RNF2.3 Registro de Eventos: Cada intento de acceso debe ser registrado en una base de datos segura, incluyendo detalles como fecha, hora y método utilizado. Esta información debe ser inalterable y accesible solo por personal autorizado.

RNF3 Requisitos de Usabilidad RNF3.1 Interfaz Intuitiva: La interfaz web para administradores debe ser fácil de usar, permitiendo la gestión de usuarios y configuración de accesos sin necesidad de formación técnica avanzada.

RNF3.2 Accesibilidad: La interfaz debe ser accesible desde dispositivos móviles y desktops a través de una red Wi-Fi local, asegurando que los administradores puedan gestionar el sistema desde cualquier lugar dentro del área cubierta.

Estos requisitos ayudan a mitigar riesgos en el desarrollo al establecer criterios claros para evaluar la calidad del software antes de su implementación. Ignorar los requisitos no funcionales puede resultar en sistemas que, aunque cumplen con las funcionalidades requeridas, no son eficientes, seguros o agradables para el

usuario. Gluo, 2023

Utilizar normas como la ISO/IEC 25010 para definir requisitos no funcionales es fundamental porque proporciona un marco estructurado y ampliamente aceptado para evaluar la calidad del software. Esto ayuda a garantizar que todos los aspectos críticos sean considerados durante el desarrollo, lo cual es vital para el éxito del proyecto (ScopeMaster, 2024).

Diseño físico

El diseño físico del sistema se refiere a la representación tangible y estructural de cómo se implementarán los componentes de un sistema, en este caso, un sistema de control de acceso que utiliza un Arduino, un lector RFID, un sensor de huella dactilar, una cámara y una Raspberry Pi. Este diseño abarca la disposición física de los dispositivos, la interconexión entre ellos y la integración con el entorno en el que se desplegará el sistema.

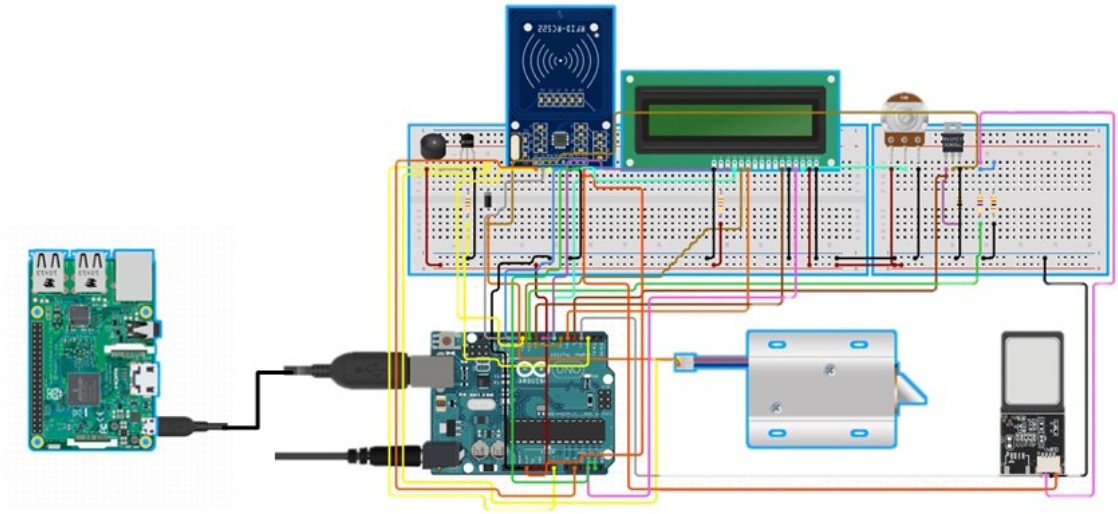


Figura 2.1. Diseño físico. Fuente: Elaboración propia

El diseño físico fue dividido en partes para una mejor apreciación de las conexiones y componentes además se debe tener en cuenta q este va conectado a la Raspberry pi de manera serial.

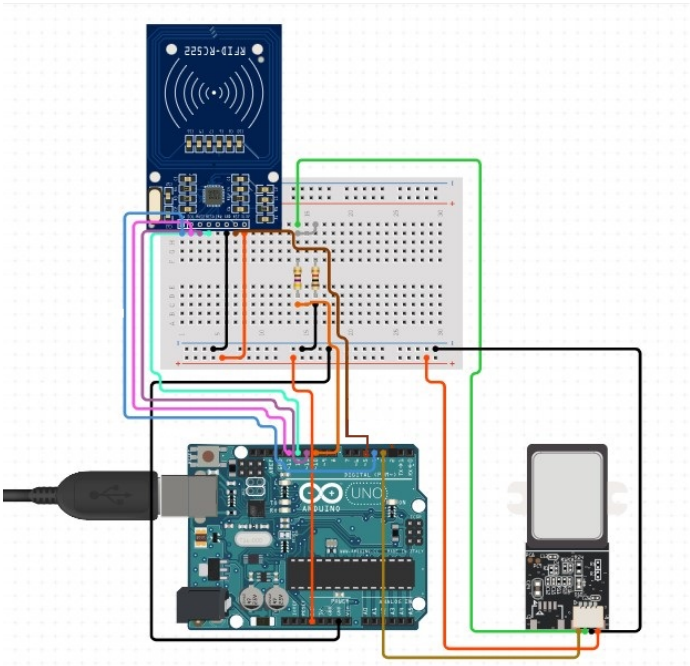


Figura 2.2. Arduino con lectores. Fuente: Elaboración propia

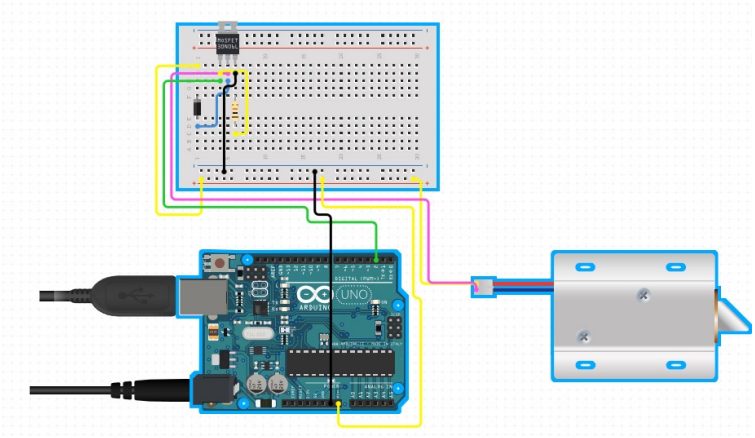


Figura 2.3. Arduino con solenoide. Fuente: Elaboración propia

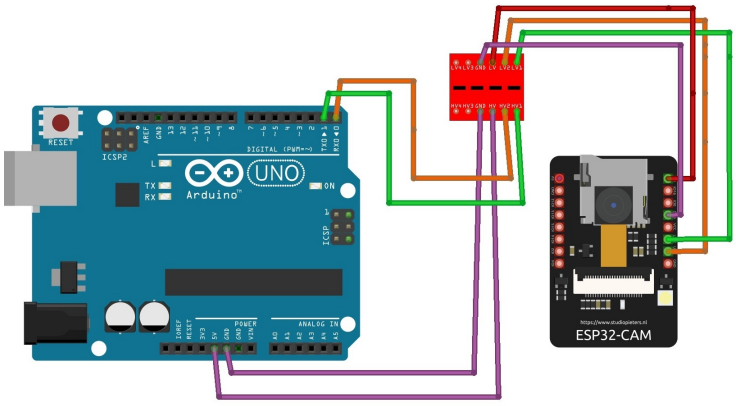


Figura 2.4. Arduino con cámara. Fuente: Elaboración propia

Figura 2.5. Raspberry pi. Fuente: Elaboración propia

2.3. Análisis del sistema

El análisis del sistema implica evaluar los requisitos recopilados para identificar problemas existentes y oportunidades de mejora. Este proceso permite a los desarrolladores comprender mejor cómo interactúan los diferentes componentes del sistema y cómo se pueden optimizar para mejorar la eficiencia y efectividad (Galileo, 2024). Durante esta fase, se utilizan diversas técnicas de análisis, como diagramas de flujo, modelos de datos y análisis de costos, para evaluar las opciones disponibles y seleccionar la mejor solución técnica Información, 2024.

El análisis también incluye la identificación de las necesidades del cliente, la evaluación de la viabilidad técnica y económica del sistema propuesto, así como la asignación de funciones a los diferentes elementos del sistema (Oposinet, 2024). Esta evaluación rigurosa ayuda a mitigar riesgos potenciales y a garantizar que el proyecto se mantenga dentro de los límites de tiempo y presupuesto establecidos.

2.3.1. Historias de usuario

Las historias de usuario son descripciones breves y sencillas de las funcionalidades que un usuario desea obtener de un sistema, redactadas desde su perspectiva. Estas historias se utilizan en metodologías ágiles, como Scrum, para captar los requisitos del usuario de manera que se centren en sus necesidades y el valor que esperan recibir. Generalmente, siguen el formato: Cómo [tipo de usuario], quiero [objetivo] para [beneficio]. Este enfoque ayuda a los equipos de desarrollo a comprender mejor lo que se necesita implementar y por qué, facilitando la comunicación y la priorización de tareas en el proceso de desarrollo.

Tabla 2.2. Historia de usuario # 1

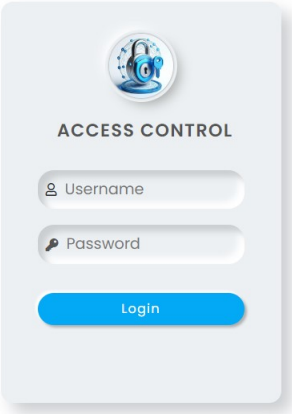
Historia de usuario	
Número: 1	**Nombre:** Autenticar administrador
Usuario: Administrador	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Medio
Puntos estimados: 1	**Iteración asignada:** 1
Programador responsable: Diacnis Ramos Jerez	
Descripción: El sistema debe permitir que solo el administrador se autentiquen en el mismo, ingresando usuario y contraseña.	
Observaciones: No aplica.	
Interfaz: 	

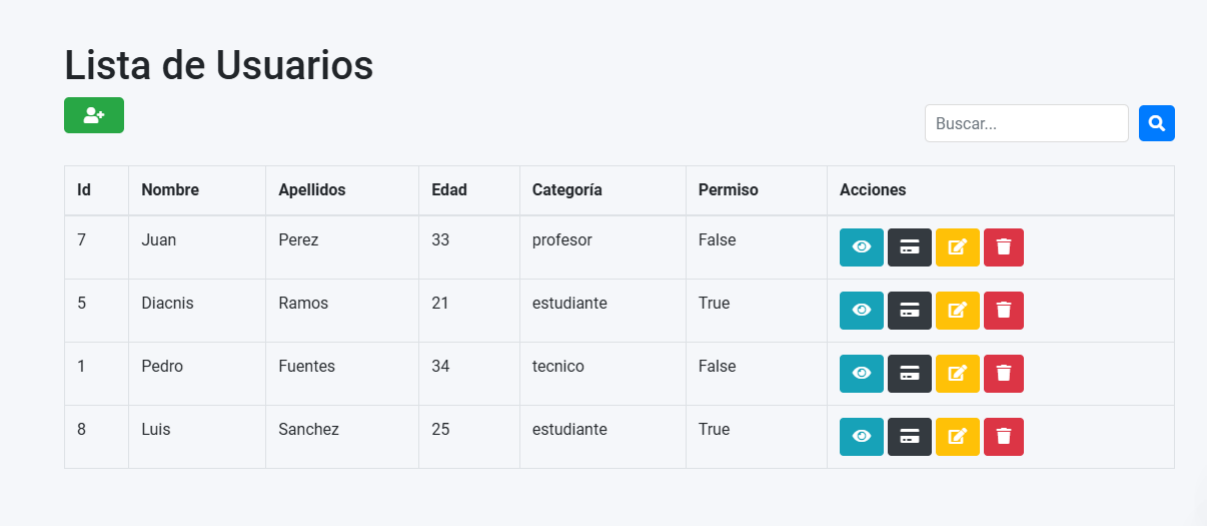
Tabla 2.3. Historia de usuario # 2

Historia de usuario	
Número: 2	**Nombre:** Cerrar sesión
Usuario: Administrador	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Medio
Puntos estimados: 0.2	**Iteración asignada:** 1
Programador responsable: Diacnis Ramos Jerez	
Descripción: El sistema debe permitir que el administrador cierre sesión.	
Observaciones: Debe estar ya autenticado.	

Tabla 2.4. Historia de usuario # 3

Historia de usuario	
Número: 3	**Nombre:** Listar usuarios
Usuario: Administrador	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Medio
Puntos estimados: 0.5	**Iteración asignada:** 1
Programador responsable: Diacnis Ramos Jerez	
Descripción: El sistema debe mostrar un listado de todos los usuarios.	
Observaciones: El administrador debe estar ya autenticado.	

Interfaz:



2.3.2. Tarjetas Clase, Responsabilidad y Colaboración (CRC)

Las Tarjetas de Clase, Responsabilidad y Colaboración (CRC) son una herramienta utilizada en la metodología de Programación Extrema (XP) para facilitar el diseño y la comprensión del software orientado a objetos.

Cada tarjeta incluye información sobre:

Clase: El nombre de la clase que representa.

Responsabilidades: Las funciones o tareas que la clase debe realizar.

Colaboradores: Otras clases con las que esta clase interactúa para cumplir sus responsabilidades.

Tabla 2.5. Tarjeta CRC # 1

Tarjeta CRC	
Clase: Registro	
Responsabilidad	**Colaboración**
• Listar Registros de acceso • Actualizar registros • Registrar acceso • Eliminar Registros antiguos	Usuario Local

Tabla 2.6. Tarjeta CRC # 2

Tarjeta CRC	
Clase: Usuario	
Responsabilidad	**Colaboración**
• Agregar usuarios • Modificar usuarios • Listar usuarios • Eliminar usuarios	Local Registro

Tabla 2.7. Tarjeta CRC # 3

Tarjeta CRC	
Clase: Local	
Responsabilidad	**Colaboración**
• Configurar acceso	Usuario Registro

El uso de tarjetas CRC en este proyecto de control de acceso radica en su capacidad para mejorar la claridad del diseño, facilita la definición de responsabilidades de cada componente del sistema y contribuye a un diseño más cohesivo y eficiente. Al utilizar tarjetas CRC, se pueden identificar claramente las clases que representan los diferentes elementos del sistema de control de acceso, como los lectores RFID, los sensores biométricos y las bases de datos. Además sirven como una forma efectiva de documentación visual, que puede ser referenciada a lo largo del ciclo de vida del desarrollo del sistema, facilitando la adaptación a cambios futuros en los requisitos o el diseño.

2.3.3. Estimación de esfuerzo por historia de usuario

El objetivo principal de la estimación por historia de usuario es ayudar a los equipos a planificar su trabajo y determinar cuánto pueden completar en un sprint o ciclo de desarrollo. En lugar de centrarse en el tiempo exacto que tomará completar una tarea, los equipos asignan puntos a las historias basándose en tres factores clave: esfuerzo, complejidad y riesgo (ClickUp, 2024)(Asana, 2024).

Tabla 2.8. Estimación de esfuerzo por historia de usuario

Iteración	**Historias de usuario**		**Puntos estimados (semanas)**
1	1	Autenticar administrador	0.52
	2	Cerrar sesión	0.2
	3	Listar usuarios	0.6
	4	Agregar usuario	1.2
	5	Realizar búsquedas	0.3
	6	Conceder permisos	0.4
	7	Denegar permisos	0.4
	8	Agregar tarjeta	0.7
2	9	Modificar usuario	0.4
	10	Eliminar usuario	0.3
	11	Visualizar Usuario	0.4
3	12	Eliminar tarjeta	0.4
	13	Procesar tarjeta rfid	0.8
	14	Asignar tarjeta	0.5
	15	Configurar acceso	0.2
4	16	Registrar acceso	1.4
	17	Listar Registros de Acceso	0.8
	18	Actualizar registros	0.6
	19	Listar Tarjetas	0.2
Total			**10.3**

La estimación de esfuerzo por historia de usuario es una herramienta valiosa dentro del marco ágil que

permite a los equipos planificar eficazmente su trabajo y gestionar expectativas. Al centrarse en puntos de historia en lugar de tiempo absoluto, los equipos pueden abordar las incertidumbres inherentes al desarrollo ágil y mejorar su capacidad para entregar valor al cliente.

La técnica empleada es la estimación por analogía, dicha técnica utiliza datos históricos de proyectos previos para hacer estimaciones sobre el esfuerzo y el tiempo requeridos para nuevas tareas o historias de usuario. Este enfoque se basa en la premisa de que si una historia de usuario anterior fue completada con éxito y su contexto es similar al de la nueva historia, se puede inferir que el esfuerzo requerido será comparable.

2.3.4. Plan de iteraciones

El Plan de Iteraciones es un enfoque utilizado en metodologías ágiles, como XP, para gestionar el desarrollo de proyectos de manera eficiente y flexible. Se basa en la repetición de ciclos cortos de trabajo, conocidos como iteraciones o sprints, que permiten a los equipos planificar, ejecutar y revisar su trabajo de forma continua.

Tabla 2.9. Plan de iteraciones. Fuente: Elaboración propia.

Iteración	HU a implementar	Duración Total (Semanas)
1	1,2,3,4,7,8,9,10	4
2	5,6,18	1
3	11,12,13,16	2
4	14,15,17,19	3

Plan de entregas:

El Plan de Entregas es un documento o estrategia que se utiliza en la gestión de proyectos para definir cómo se llevarán a cabo las entregas de productos o servicios a lo largo del ciclo de vida de un proyecto. Este plan es fundamental para asegurar que todas las partes interesadas estén alineadas con los objetivos y expectativas del proyecto.

Tabla 2.10. Plan de entregas. Fuente: Elaboración propia.

Entregable	Duración (semanas)	Fecha de entrega
Autenticar administrador, Cerrar sesión, Listar usuarios, Realizar búsquedas, Conceder permisos, Denegar permisos, Agregar tarjeta, Crear usuario.	4	Julio 2024

Tabla 2.10. Plan de entregas. Fuente: Elaboración propia.

Entregable	Duración (semanas)	Fecha de entrega
Modificar usuario, Eliminar usuario, Visualizar Usuario.	1	Agosto 2024
Eliminar tarjeta, Procesar tarjeta rfid, Asignar tarjeta, Configurar acceso.	2	Septiembre 2024
Registrar acceso, Listar Registros de Acceso , Actualizar registros, Listar Tarjetas.	3	Octubre 2024

2.4. Modelado del Sistema

El modelado del sistema es la fase donde se crean representaciones visuales o matemáticas del sistema propuesto. Esto puede incluir modelos arquitectónicos, diagramas de casos de uso o prototipos funcionales. El modelado permite a los desarrolladores visualizar cómo funcionará el sistema en su conjunto y cómo interactuarán sus componentes(Geocities, 2024). Además, facilita la comunicación entre los miembros del equipo y con las partes interesadas al proporcionar una representación tangible del sistema.

Este proceso no solo ayuda a identificar problemas potenciales antes de que se inicie la implementación, sino que también permite realizar ajustes en el diseño basándose en la retroalimentación obtenida durante las revisiones con los interesados. Al final de esta fase, se espera tener un modelo claro que sirva como guía para el desarrollo posterior.

2.4.1. Diseño Arquitectónico

El Diseño Arquitectónico es un proceso complejo que implica la conceptualización y planificación de espacios y estructuras, teniendo en cuenta tanto aspectos estéticos como funcionales. Este proceso no solo se centra en la creación de edificios, sino también en la organización del espacio, la interacción entre los elementos y su entorno, así como en la satisfacción de las necesidades de los usuarios.

El diseño arquitectónico es crucial porque determina cómo interactúan las personas con el espacio construido. Un buen diseño puede mejorar la calidad de vida, fomentar la sostenibilidad y contribuir al bienestar general de la comunidad. Además, debe cumplir con normativas legales y estándares de seguridad.

Arquitectura del software

La arquitectura cliente-servidor es un modelo de diseño de software ampliamente utilizado en el desarrollo

de aplicaciones y sistemas informáticos. Este enfoque se basa en la división de tareas y responsabilidades entre dos entidades principales: el cliente y el servidor.

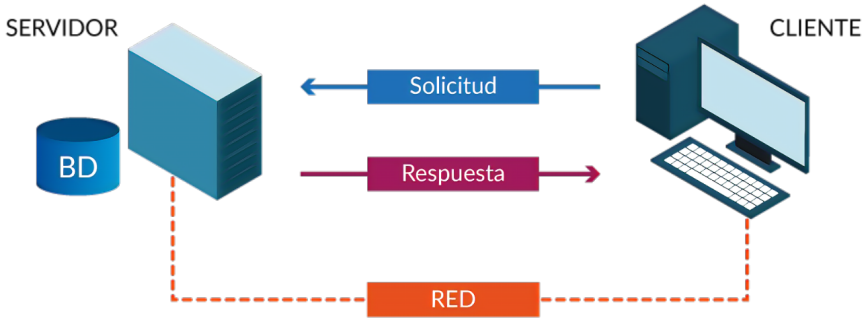


Figura 2.6. Arquitectura cliente-servidor. Fuente: Elaboración propia.

Patrones de arquitectura

Los patrones de arquitectura son soluciones generales y reutilizables a problemas comunes en el diseño de software. Actúan como guías que ayudan a los arquitectos de software a estructurar sistemas complejos, definiendo la organización de componentes y sus interacciones. Estos patrones son cruciales para crear sistemas que sean escalables, mantenibles y eficientes.

El patrón de arquitectura de software Modelo-Vista-Plantilla (MVT) es una variación del patrón Modelo-Vista-Controlador (MVC) y es utilizado principalmente en el desarrollo de aplicaciones web, especialmente en el framework Django de Python. A continuación, se describen sus componentes, funcionamiento y ventajas.

Componentes del Patrón MVT

Modelo:

El modelo representa la lógica de negocio y los datos de la aplicación. Se encarga de la interacción con la base de datos, definiendo las estructuras de datos y las operaciones que se pueden realizar sobre ellos.

En Django, los modelos son clases que se traducen en tablas de base de datos y permiten realizar operaciones como crear, leer, actualizar y eliminar registros.

Vista:

La vista es responsable de la lógica que determina qué datos se mostrarán al usuario. En lugar de ser una representación visual directa, la vista en MVT actúa como un intermediario que procesa las solicitudes del usuario y decide qué información enviar al template.

En Django, las vistas son funciones o clases que reciben solicitudes HTTP y devuelven respuestas HTTP.

Plantilla (Template):

La plantilla es el componente que se encarga de la presentación visual de los datos. Define cómo se mostrarán los datos al usuario final.

En Django, las plantillas son archivos HTML que pueden contener etiquetas especiales para insertar dinámicamente contenido proveniente del modelo.

Funcionamiento del Patrón MVT

Solicitud del Usuario: Cuando un usuario realiza una solicitud (por ejemplo, accediendo a una URL), esta solicitud es manejada por una vista en el servidor.

Procesamiento en la Vista: La vista recibe la solicitud y utiliza el modelo para obtener los datos necesarios. Puede realizar operaciones sobre la base de datos para recuperar o modificar información.

Renderización en Plantilla: Una vez que la vista tiene los datos requeridos, utiliza una plantilla para generar una respuesta HTML. La plantilla se llena con los datos obtenidos del modelo y se prepara para ser enviada al cliente.

Respuesta al Cliente: Finalmente, el servidor envía la respuesta HTML generada por la plantilla al navegador del usuario, donde se muestra la información.

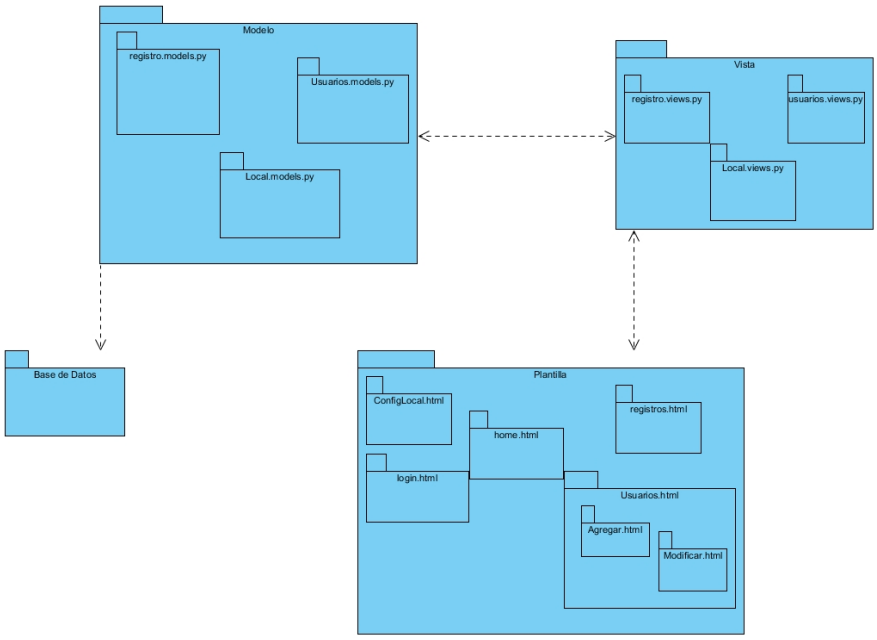


Figura 2.7. Modelo Vista Plantilla. Fuente: Elaboración propia.

Patrones de diseño

Los patrones de diseño son soluciones estandarizadas y reutilizables a problemas comunes que surgen en el desarrollo de software. Estas soluciones han sido probadas a lo largo del tiempo y son utilizadas por desarrolladores para crear sistemas más eficientes, mantenibles y escalables. El concepto se popularizó con la publicación del libro "Design Patterns: Elements of Reusable Object-Oriented Software.[en] 1994, escrito por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides, conocidos como el Gang of Four (GoF) (Estudio, 2023).

Patrones GRASP:

Los patrones GRASP (General Responsibility Assignment Software Patterns) son un conjunto de principios que ayudan a asignar responsabilidades a las clases y objetos en el diseño orientado a objetos. Estos patrones son fundamentales para crear sistemas bien estructurados y mantenibles.

Alta Cohesión (High Cohesion): Este patrón sugiere que las clases deben tener responsabilidades bien definidas y relacionadas entre sí. En este proyecto, cada clase debe encargarse de una sola responsabilidad, como GestorTarjetasRFID para manejar todo lo relacionado con las tarjetas RFID, lo cual mejora la claridad y mantenibilidad del código.

```python
class Registro(models.Model):
    ACCESO = [
        ('permitido', 'Permitido'),
        ('denegado', 'Denegado'),

    ]
    usuario = models.ForeignKey(Usuario, on_delete=models.CASCADE)
    local = models.ForeignKey(Local, on_delete=models.CASCADE)
    fecha_hora = models.DateTimeField(auto_now_add=True)
    metodo_autenticacion = models.CharField(max_length=50)
    estado_acceso = models.CharField(max_length=20, choices=ACCESO)

    def __str__(self):
        return f"{self.usuario} - {self.local} - {self.estado_acceso} - {self.fecha_hora}"
```

Figura 2.8. Patrón Alta Cohesión en la clase Registro. Fuente: Elaboración propia.

La alta cohesión se evidencia aquí porque la clase Registro se ocupa únicamente de los aspectos relacionados con el registro de accesos. Esto significa que todos los métodos y atributos dentro de esta clase están centrados en su función principal, lo que facilita la comprensión del código y su mantenimiento.

Experto en Información (Información Expert): Este patrón sugiere que la responsabilidad debe ser asignada a la clase que tiene la información necesaria para cumplirla. La clase Local tiene la responsabilidad de gestionar su propio nombre y los métodos de acceso asociados a él. Esto significa que cualquier lógica relacionada con los métodos de acceso (como modificar métodos) puede residir dentro de esta clase. Además, al utilizar un campo ManyToManyField para los métodos de acceso, Local puede manejar internamente las relaciones con esos métodos.

```python
class Local(models.Model):
    nombre = models.CharField(max_length=100)

    metodos_acceso = models.ManyToManyField(MetodoAcceso)

    def __str__(self):
        return f"{self.nombre} - Métodos: {', '.join(str(m) for m in self.metodos_acceso.all())}"
```

Figura 2.9. Patrón Experto en Información en la clase Local. Fuente: Elaboración propia.

Controlador(Controller): En este proyecto, el patrón Controller se puede observar en las clases o funciones que manejan las interacciones del usuario con el sistema. Por ejemplo, un controlador que maneja la lógica para agregar un nuevo usuario y gestionar las solicitudes del formulario (como el manejo de huellas dactilares o tarjetas RFID) es un buen ejemplo de este patrón.

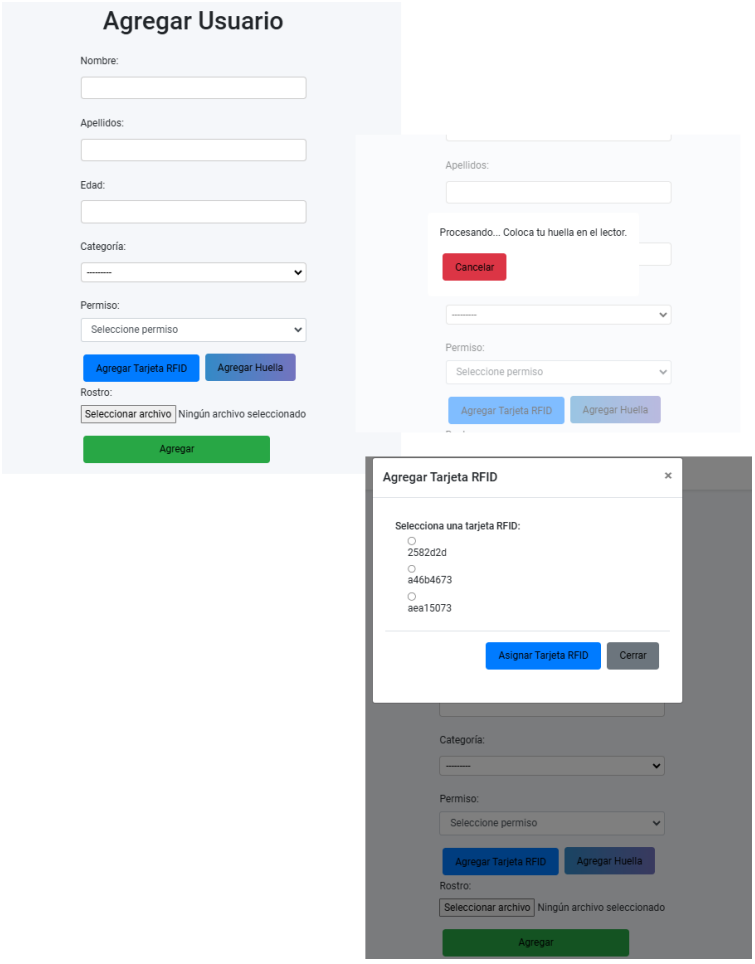


Figura 2.10. Patrón Controlador en formulario Agregar Usuario. Fuente: Elaboración propia.

Bajo Acoplamiento(Low Coupling): Este patrón se evidencia en la forma en que las clases están diseñadas para minimizar las dependencias entre ellas. Por ejemplo, las clases Registro, Usuario, y TarjetaRFID están estructuradas para interactuar a través de relaciones bien definidas (como claves foráneas), lo que permite que cada clase funcione independientemente sin depender excesivamente de otras.

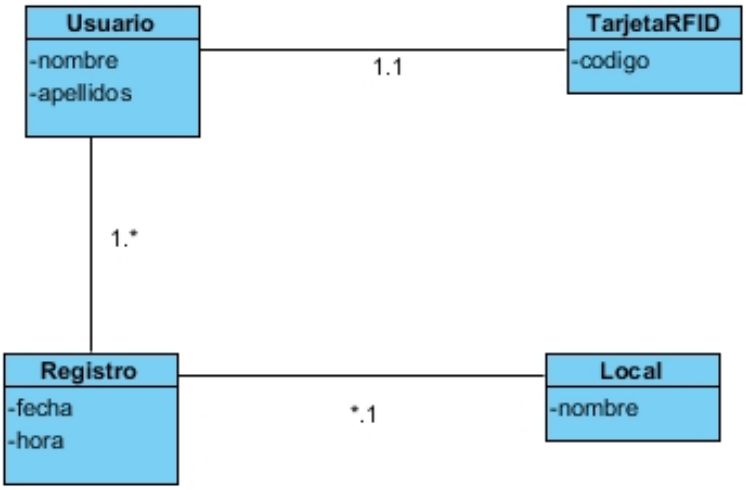


Figura 2.11. Patrón Bajo Acoplamiento Diagrama de Clases. Fuente: Elaboración propia.

Descripción del Diagrama:

Clases Independientes: Cada clase (Usuario, TarjetaRFID, Registro, Local) tiene su propia responsabilidad y no depende directamente de las implementaciones internas de las otras clases. Por ejemplo, Usuario y TarjetaRFID están conectados a través de una relación (OneToOne), pero no necesitan conocer los detalles internos de cada uno para funcionar.

Relaciones Claras: Las relaciones entre las clases son claras y bien definidas. Por ejemplo, Registro tiene una relación con Usuario y Local, pero no está acoplado a la lógica interna de estas clases. Esto significa que si cambias la implementación de Usuario o Local, no afectará directamente a Registro.

Facilidad de Mantenimiento: Este diseño permite que cada clase se mantenga y se modifique de manera independiente, lo que facilita la evolución del sistema sin crear un efecto dominó en otras partes del código.

Patrones GoF:

Los patrones de diseño de software son un conjunto de artefactos que encapsulan el conocimiento de los problemas de diseño que ocurren en un contexto particular. Se han propuesto varios patrones de software para proporcionar soluciones a problemas de diseño recurrentes.

Dominio(Command): Es un patrón de diseño de comportamiento que convierte una solicitud en un objeto

independiente que contiene toda la información sobre la solicitud. Esta transformación le permite pasar solicitudes como argumentos de un método, retrasar o poner en cola la ejecución de una solicitud y admitir operaciones que se pueden deshacer (Guru, 2024).

El patrón de Dominio se manifiesta a través de las clases que representan conceptos del negocio, como Usuario, TarjetaRFID, Registro, y Local. Estas clases encapsulan la lógica relacionada con sus respectivos dominios. Clases: Usuario: Maneja la información del usuario, como nombre, apellidos, edad, etc. TarjetaRFID: Representa una tarjeta RFID única y su relación con un usuario. Registro: Almacena información sobre los accesos realizados por los usuarios. Local: Representa un lugar donde se puede controlar el acceso.

Mediador(Mediator): es un patrón de diseño de comportamiento que permite reducir las dependencias caóticas entre objetos. El patrón restringe las comunicaciones directas entre los objetos y los obliga a colaborar únicamente a través de un objeto mediador (ibíd.).

El patrón Mediador se observa en cómo las vistas actúan como intermediarias entre los modelos y la interfaz de usuario. Por ejemplo, al agregar un usuario, la vista maneja tanto la entrada del formulario como la lógica para asignar una tarjeta RFID. Ejemplo en HTML: En el código HTML proporcionado para agregar un usuario, el formulario actúa como mediador entre el usuario y la lógica del backend.

```
<h1>Agregar Usuario</h1>
<form method="post">
    {% csrf_token %}

        <!-- Nombre -->
        <div class="form-group">
            <label for="{{ form.nombre.id_for_label }}">Nombre:</label>
            {{ form.nombre }}
        </div>

        <!-- Apellidos -->
        <div class="form-group">
            <label for="{{ form.apellidos.id_for_label }}">Apellidos:</label>
            {{ form.apellidos }}
        </div>

        <!-- Edad -->
        <div class="form-group">
            <label for="{{ form.edad.id_for_label }}">Edad:</label>
            {{ form.edad }}
        </div>

        <!-- Categoría -->
        <div class="form-group">
            <label for="{{ form.categoria.id_for_label }}">Categoría:</label>
            {{ form.categoria }}  <!-- Dropdown para seleccionar la categoría -->
        </div>
```

Figura 2.12. Patrón Mediador. Fuente: Elaboración propia.

Estrategia: La estrategia es un patrón de diseño de comportamiento que le permite definir una familia de algoritmos, colocar cada uno de ellos en una clase separada y hacer que sus objetos sean intercambiables.(Refactoring Guru, n.d.).

El patrón Estrategia se puede observar en cómo se pueden implementar diferentes métodos de autenticación

(por ejemplo, RFID, huella dactilar) sin cambiar el código que maneja el acceso. Esto permite intercambiar algoritmos de autenticación según las necesidades.

Fachada(Facade): es un patrón de diseño estructural que proporciona una interfaz simplificada para una biblioteca, un marco o cualquier otro conjunto complejo de clases (Guru, 2024).

El patrón Fachada se puede observar en cómo las vistas simplifican la interacción con múltiples modelos. Por ejemplo, al agregar un usuario, la vista maneja la creación del usuario y la asignación de una tarjeta RFID a través de una interfaz simple.

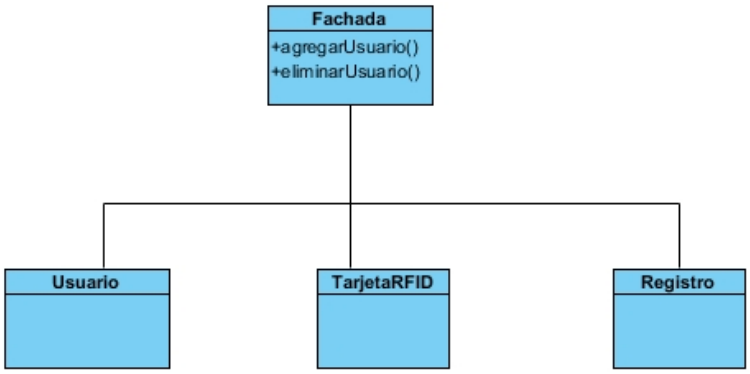


Figura 2.13. Patrón Fachada. Fuente: Elaboración propia.

Mapa de navegación

Un mapa de navegación es una representación visual o esquemática de la estructura y organización de un sitio web o sistema de información. Su objetivo principal es facilitar la navegación del usuario al mostrar cómo están conectadas las diferentes páginas o secciones del sitio, permitiendo una comprensión clara de la jerarquía y las relaciones entre los contenidos.

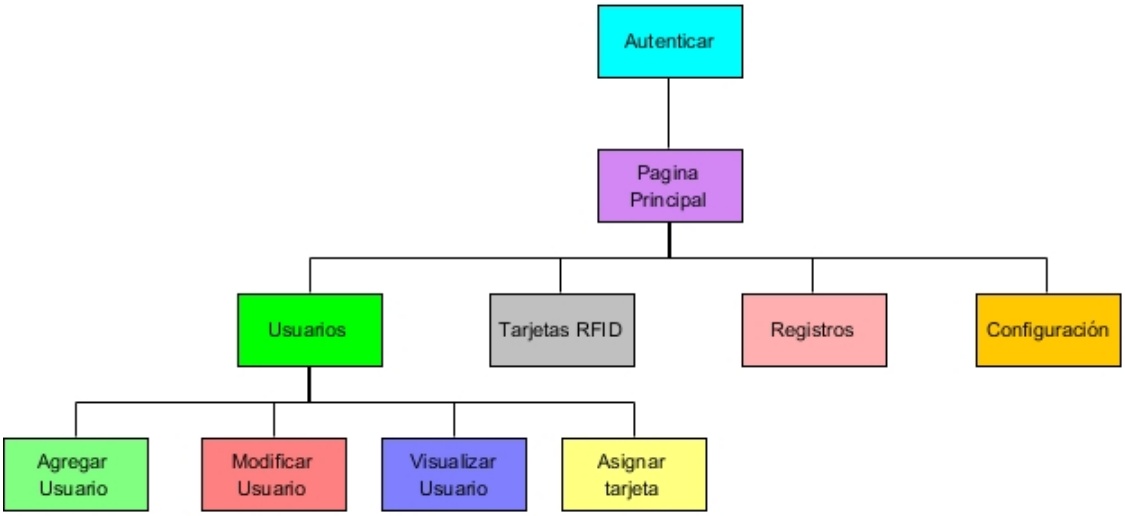


Figura 2.14. Mapa de navegación. Fuente: Elaboración propia.

Utilidad de Modelar un Mapa de Navegación:

Mejora la Usabilidad: Un mapa de navegación bien diseñado ayuda a los usuarios a encontrar rápidamente la información que buscan, lo que mejora su experiencia en el sitio.

Estructuración del Contenido: Permite organizar el contenido de manera lógica y coherente, facilitando la identificación de categorías y subcategorías importantes. Esto es especialmente útil en sitios grandes donde la cantidad de información puede ser abrumadora.

Facilita la Planificación: Durante la fase de desarrollo, un mapa de navegación actúa como una herramienta de planificación que ayuda a los diseñadores y desarrolladores a visualizar la estructura del sitio antes de su implementación. Esto puede prevenir problemas futuros relacionados con la organización del contenido.

Optimización para Motores de Búsqueda (SEO): Un mapa de navegación también puede ser útil para crear un mapa XML que facilite a los motores de búsqueda indexar el contenido del sitio, mejorando así su visibilidad en los resultados de búsqueda.

Auditorías y Mantenimiento: Proporciona una referencia clara para realizar auditorías web y mantenimiento del contenido, ayudando a identificar áreas que necesitan actualizaciones o mejoras.

Guía para el Usuario: Actúa como una guía visual que puede ser utilizada por los usuarios para navegar por el sitio, similar a un índice en un libro. Esto es especialmente útil en sitios complejos donde los usuarios pueden perderse fácilmente.

Modelo de datos

Un modelo de datos es una representación estructurada de la información que se utiliza para organizar,

almacenar y gestionar datos en sistemas de bases de datos. Su propósito principal es proporcionar un marco que facilite la comprensión de cómo se relacionan los diferentes elementos de información, permitiendo así su manipulación y recuperación efectiva.

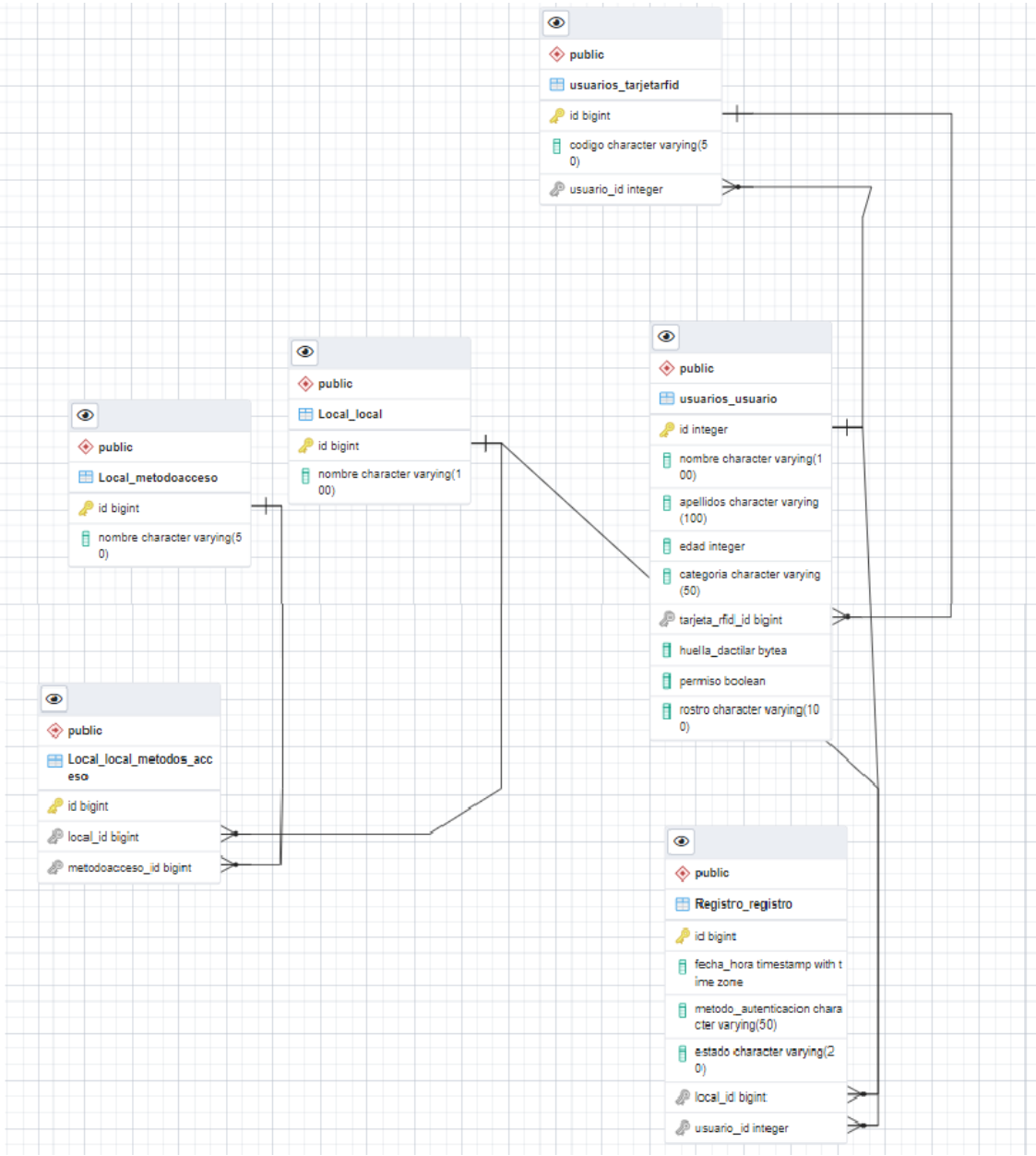


Figura 2.15. Modelo de datos. Fuente: Elaboración propia.

Diagrama de despliegue

Un diagrama de despliegue es un tipo de diagrama utilizado en la modelación de sistemas que representa la

arquitectura física del sistema, mostrando los componentes de hardware y cómo se interconectan. Este tipo de diagrama es parte del Lenguaje Unificado de Modelado (UML) y es especialmente útil para visualizar la distribución de los componentes en un entorno físico, como servidores, dispositivos, y conexiones de red.

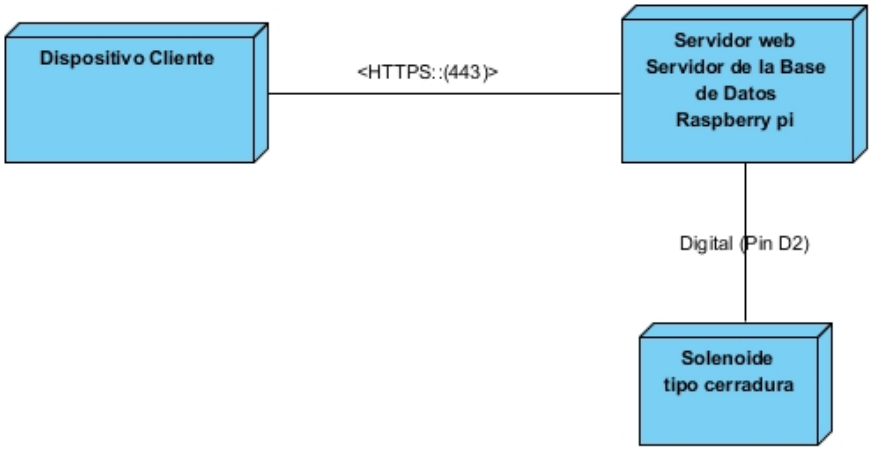


Figura 2.16. Diagrama de despliegue. Fuente: Elaboración propia.

Componentes del diagrama de despliegue:

- Dispositivo Cliente: PC, móvil o dispositivo desde el q se acceda a la página web.
- Servidor Web, Servidor de la Base de Datos: Dicho servidor es la Raspberry pi la cual es un componente de la tarjeta genérica desarrolada.
- Solenoide tipo cerradura: Este está conectado a la tarjeta genérica específicamente al Arduino mediante conexión digital al Pin D2

La utilidad del diagrama de despliegue en este proyecto de control de acceso es fundamental, ya que proporciona una representación visual clara de la arquitectura física del sistema. Este diagrama permite identificar y organizar los diferentes componentes, como servidores, dispositivos de control (lectores RFID, huellas dactilares) y la base de datos, así como sus interconexiones. Facilita la comprensión del flujo de datos y las comunicaciones entre los elementos, lo que es crucial para asegurar que el sistema funcione de manera eficiente y segura. Además, al visualizar cómo se distribuyen los componentes en el entorno físico, puedes planificar mejor la infraestructura necesaria, prever posibles problemas y realizar un mantenimiento más efectivo a largo plazo. En resumen, el diagrama de despliegue no solo mejora la claridad del diseño del sistema, sino que también contribuye a su implementación y gestión exitosa.

Conclusiones del capítulo

La propuesta de un sistema de control de acceso que integra tecnologías como RFID, reconocimiento de huella dactilar y reconocimiento facial representa un avance significativo en la seguridad de áreas restringidas. La combinación de múltiples métodos de autenticación no solo mejora la seguridad, sino que también

proporciona flexibilidad para adaptarse a diferentes necesidades de acceso.

El sistema permite un monitoreo efectivo y un registro detallado de todas las actividades de acceso. Esto no solo facilita la identificación de patrones inusuales o intentos no autorizados, sino que también proporciona a los administradores herramientas para gestionar usuarios y permisos de manera eficiente.

La identificación clara de requisitos funcionales y no funcionales es esencial para el éxito del desarrollo del sistema. Los requisitos funcionales aseguran que se implementen las características necesarias, mientras que los no funcionales garantizan que estas características se entreguen con eficiencia y efectividad, afectando directamente la experiencia del usuario.

La fase de modelado del sistema es crucial para establecer una base sólida en el desarrollo del proyecto. Un diseño bien estructurado no solo ayuda a cumplir con las expectativas del cliente, sino que también minimiza riesgos asociados a malentendidos o deficiencias en la implementación.

Finalmente, la implementación del sistema propuesto alinea a la universidad con tendencias tecnológicas emergentes, posicionándola como un referente en el uso eficiente de nuevas tecnologías en el ámbito académico y administrativo.

CAPÍTULO 3

Validación de la propuesta de solución

En este capítulo, se abordarán las tareas ingenieriles esenciales para la implementación y validación del sistema de control de acceso propuesto. La ingeniería de software no solo implica el desarrollo de soluciones tecnológicas, sino también la planificación y ejecución de pruebas rigurosas que aseguren la calidad y funcionalidad del sistema. Para ello, se presentará un plan de pruebas que incluirá tanto pruebas unitarias como pruebas de aceptación, cada una desempeñando un papel crucial en el ciclo de vida del desarrollo del software.

3.1. Tareas ingenieriles

Las tareas ingenieriles son actividades específicas que los ingenieros realizan para resolver problemas, diseñar sistemas, o implementar soluciones tecnológicas. Estas tareas pueden abarcar desde el análisis de requisitos hasta la implementación y prueba de sistemas, y son esenciales en el desarrollo de proyectos técnicos.

Tabla 3.1. Tarea de ingeniería # 1

Tarea	
Número de tarea: 1	**Número de Historia de usuario:** 1
Nombre de la tarea: Autenticar Administrador	
Tipo de tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 1 de julio de 2024	**Fecha de fin:** 4 de julio de 2024
Programador responsable: Diacnis Ramos Jerez	
Descripción: Implementar la funcionalidad de autenticarse el administrador al sistema.	

Tabla 3.2. Tarea de ingeniería # 2

Tarea	
Número de tarea: 2	**Número de Historia de usuario:** 3
Nombre de la tarea: Listar usuarios	
Tipo de tarea: Desarrollo	**Puntos estimados:** 0.5
Fecha de inicio: 6 de julio de 2024	**Fecha de fin:** 11 de julio de 2024
Programador responsable: Diacnis Ramos Jerez	
Descripción: Implementar la funcionalidad de Listar usuarios	

Tabla 3.3. Tarea de ingeniería # 3

Tarea	
Número de tarea: 3	**Número de Historia de usuario:** 4
Nombre de la tarea: Agregar usuario	
Tipo de tarea: Desarrollo	**Puntos estimados:** 0.5
Fecha de inicio: 12 de julio de 2024	**Fecha de fin:** 20 de julio de 2024
Programador responsable: Diacnis Ramos Jerez	
Descripción: Implementar la funcionalidad de Agregar usuario	

Las tareas ingenieriles son fundamentales en el desarrollo de este sistema de control de acceso, ya que permiten estructurar y organizar el proceso de diseño e implementación. Estas tareas incluyen el análisis de requisitos, diseño del sistema, programación, pruebas y documentación, lo que asegura que el sistema sea funcional y cumpla con las expectativas de seguridad. Además garantizan la integración efectiva de tecnologías como RFID, reconocimiento de huella dactilar y reconocimiento facial, facilitando un control de acceso robusto y eficiente.

3.2. Plan de Pruebas

Un plan de pruebas es un documento que describe el enfoque, los recursos y la programación de las actividades de prueba. Es fundamental para garantizar que el sistema cumple con los requisitos y funciona como se espera. En el contexto de este proyecto de control de acceso, el plan de pruebas incluirá tanto pruebas unitarias como pruebas de aceptación. A continuación, se presenta un plan de pruebas desarrollado específicamente para este proyecto.

1. Objetivo del Plan de Pruebas

El objetivo de este plan es validar que el sistema de control de acceso funcione correctamente, asegurando que todas las funcionalidades operen según lo previsto.

2. Alcance

Este plan cubrirá las siguientes áreas:

Métodos de autenticación (huella dactilar, RFID, reconocimiento facial).

Gestión de usuarios (agregar, eliminar, modificar).

Registro y monitoreo de accesos.

Administración de tarjetas RFID(agregar, eliminar, asignar).

Interfaz web para administradores.

3. Tipos de Pruebas

3.1 Pruebas Unitarias

Estas pruebas se centrarán en validar individualmente cada componente del sistema. Se realizarán las siguientes pruebas unitarias:

Pruebas del Modelo Usuario: Verificar la creación y validación de un nuevo usuario. Comprobar que las relaciones con TarjetaRFID funcionan correctamente.

Pruebas del Modelo TarjetaRFID: Validar la creación y eliminación de tarjetas RFID. Asegurar que una tarjeta no se pueda eliminar si está asignada a un usuario.

Pruebas del Modelo Registro: Validar el registro de un evento de acceso al local por un usuario. Asegurar que un registro se asocie correctamente con un usuario y un local específico. Verificar que el estado del registro refleje correctamente el tipo de acceso (permitido, denegado) en función de lo permisos.

Pruebas del Modelo Local: Asegurar que el local pueda tener múltiples métodos de acceso asociados. Verificar que la representación del local incluya correctamente los métodos de acceso disponibles en su descripción.

Pruebas en Vistas: Probar las funciones asociadas a todas las vistas del sistema. Verificar que los mensajes de éxito o error se muestran adecuadamente.

3.2 Pruebas de Aceptación

Estas pruebas se realizarán para validar que el sistema cumple con los requisitos del usuario final. **4. Recursos Necesarios** Entorno de desarrollo configurado con Django. Base de datos para almacenar usuarios y registros. Herramientas para realizar pruebas unitarias (como pytest o unittest).

5. Cronograma

Las pruebas se llevarán a cabo en las siguientes fases:

Fase 1: Desarrollo y ejecución de pruebas unitarias (Semana 1).

Fase 2: Desarrollo y ejecución de pruebas de aceptación (Semana 2).

Fase 3: Revisión y ajustes basados en los resultados (Semana 3).

6. Criterios de Éxito

El sistema será considerado exitoso si:

Todas las pruebas unitarias pasan sin errores.

Las pruebas de aceptación cumplen con los requisitos establecidos por los usuarios.

No hay errores críticos durante la operación normal del sistema.

3.2.1. Pruebas unitarias

Las pruebas unitarias se enfocan en validar componentes individuales del sistema, garantizando que cada módulo funcione correctamente en aislamiento. Este enfoque permite identificar y corregir errores en etapas tempranas del desarrollo, lo que contribuye a una mayor estabilidad y confiabilidad del sistema en su conjunto. Por otro lado, las pruebas de aceptación se centran en evaluar si el sistema cumple con los requisitos y expectativas del usuario final. Estas pruebas son fundamentales para asegurar que el producto final no solo sea funcional, sino que también satisfaga las necesidades específicas para las cuales fue diseñado.

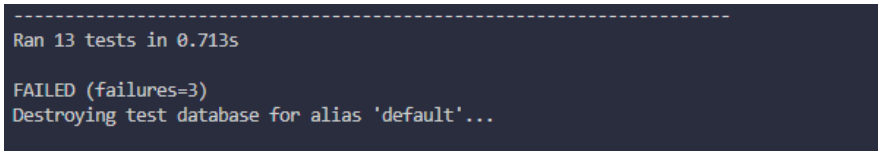


Figura 3.1. Resultados de las Pruebas unitarias v1

El resultado de las pruebas unitarias indica que de un total de 13 pruebas ejecutadas, 3 han fallado. Esto sugiere que existen errores en el sistema que necesitan ser abordados y mitigados. Es fundamental revisar los detalles de las pruebas fallidas para identificar las causas subyacentes de estos errores, lo que permitirá realizar las correcciones necesarias y garantizar que el sistema funcione correctamente antes de su implementación final.

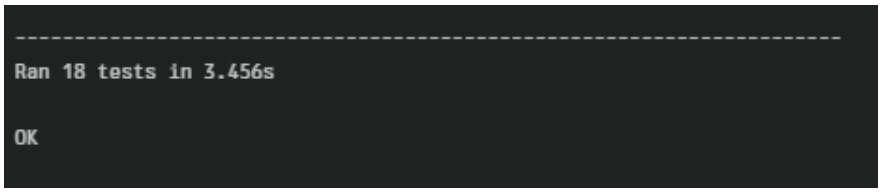


Figura 3.2. Resultados de las Pruebas unitarias v2.0

Al ejecutar las pruebas unitarias indica que se realizaron un total de 18 pruebas en un tiempo de 3.456 segundos. La nueva versión de la aplicación ha dado resultados satisfactorios, ya que todas las pruebas unitarias se ejecutaron correctamente y no se presentaron problemas. Esto indica que el sistema funciona de manera eficiente y cumple con los requisitos establecidos, proporcionando una experiencia confiable para los usuarios. La implementación de las funcionalidades ha sido exitosa, lo que refuerza la calidad y estabilidad del software antes de su despliegue en un entorno real.

3.2.2. Pruebas de aceptación

Las pruebas de aceptación son un tipo de evaluación que se realiza al final del ciclo de desarrollo de un software para determinar si el sistema cumple con los requisitos y expectativas del usuario final. Estas pruebas son generalmente realizadas por los usuarios o clientes y se centran en validar si el software es funcional, útil y cumple con las especificaciones acordadas.

Las pruebas de aceptación son esenciales para validar que el sistema cumple con los requisitos y expectativas del usuario final. En el contexto de este proyecto de control de acceso, estas pruebas permiten verificar que todas las funcionalidades, como la autenticación mediante huella dactilar, RFID y reconocimiento facial, se implementen correctamente y funcionen como se espera. Al involucrar a los usuarios finales en este proceso, se pueden identificar problemas de usabilidad y errores que podrían no haber sido detectados durante las pruebas unitarias. Esto asegura que el sistema no solo sea técnicamente sólido, sino también intuitivo y fácil de usar, lo que es fundamental para su éxito en un entorno real.

Tabla 3.4. Prueba de aceptación # 1

Caso de prueba de aceptación	
Código: P1_HU1	**Historia de usuario:** 1
Nombre: Autenticar usuario	
Descripción: Se debe probar la autenticación existosa	
Condiciones de ejecución: El usuario debe ser administrador del sistema.	

Continúa en la próxima página

Tabla 3.4. Continuación de la página anterior

Pasos de ejecución: • El usuario accede a la url del portal web • El usuario rellena el formulario de autenticación • El usuario se autentica en el sistema
Resultados esperados: Autenticación usuario exitosa

Tabla 3.5. Prueba de aceptación # 2

Caso de prueba de aceptación	
Código: P2_HU2	**Historia de usuario:** 2
Nombre: Cerrar sesión	
Descripción: Se debe probar que se pueda salir correctamente de la página	
Condiciones de ejecución: El usuario debe ser Administrador del sistema.	
Pasos de ejecución: • El usuario debe seleccionar la opción Perfil • Seleccionar salir • El sistema debe salir y mostrar el formulario le autenticar.	
Resultados esperados: Salió del sistema con éxito	

Tabla 3.6. Prueba de aceptación # 3

Caso de prueba de aceptación	
Código: P3_HU3	**Historia de usuario:** 3
Nombre: Listar usuarios	
Descripción: Se debe probar que se muestre de forma correcta la lista de usuarios	
Condiciones de ejecución: El usuario debe ser Administrador del sistema .	
Pasos de ejecución: • El usuario debe acceder a la opción Usuario • Se debe mostrar la lista de todos los usuarios.	
Resultados esperados: Muestra la lista de forma exitosa	

Resultado de las pruebas de aceptación

El resultado de las pruebas de aceptación ha sido completamente satisfactorio, ya que todas las pruebas se ejecutaron con éxito y no se presentaron problemas. Esto indica que la aplicación cumple con los requisitos y expectativas establecidos por los usuarios finales, asegurando que todas las funcionalidades, como la autenticación mediante huella dactilar, RFID y reconocimiento facial, funcionan correctamente. La implementación de esta nueva versión del sistema de control de acceso ha demostrado ser efectiva y confiable, lo

que refuerza la calidad del software y su preparación para su uso en un entorno real.

3.2.3. Resultados Generales de las Pruebas

Basado en los resultados de las pruebas unitarias y de aceptación, el resumen general de las pruebas indica que el sistema de control de acceso ha demostrado ser robusto y confiable. En las pruebas unitarias, se ejecutaron un total de 18 pruebas, todas las cuales fueron satisfactorias, lo que sugiere que cada componente del sistema, desde la gestión de usuarios hasta la asignación de tarjetas RFID, funciona como se esperaba sin errores. Además, las pruebas de aceptación confirmaron que el sistema cumple con los requisitos y expectativas establecidos por los usuarios finales. Todas las funcionalidades, incluyendo la autenticación mediante huella dactilar, RFID y reconocimiento facial, operaron sin inconvenientes. Este resultado positivo refuerza la calidad del software y su preparación para su implementación en un entorno real, garantizando así una experiencia segura y eficiente para los usuarios.

Conclusiones del Capítulo

Las tareas ingenieriles y el desarrollo de un plan de pruebas han permitido lograr una implementación estructurada y efectiva del sistema de control de acceso.

Gracias a la planificación meticulosa, se han definido claramente los requisitos y se han asignado tareas específicas, lo que ha facilitado un seguimiento eficiente del progreso del proyecto.

La ejecución del plan de pruebas ha validado la funcionalidad del sistema, asegurando que cada componente opere correctamente y cumpla con las expectativas de los usuarios. Este enfoque proactivo no solo ha identificado y corregido errores en etapas tempranas, sino que también ha garantizado que el sistema sea robusto y confiable antes de su despliegue.

En resumen, la combinación de tareas ingenieriles bien definidas y un plan de pruebas riguroso ha sido fundamental para el éxito del proyecto, proporcionando confianza en la calidad y seguridad del sistema de control de acceso.

Conclusiones

La revisión del estado del arte ha permitido identificar la importancia crítica de la seguridad física en las organizaciones, destacando sus componentes esenciales como controles de acceso, vigilancia, alarmas y diseño ambiental. Se ha evidenciado que la integración de la seguridad física con la seguridad informática es fundamental para crear un entorno seguro que proteja tanto los activos tangibles como la información sensible.

El análisis del estado actual del control de acceso revela que muchas organizaciones aún dependen de métodos tradicionales que pueden ser vulnerables a diversas amenazas. Las tecnologías emergentes, como RFID y sistemas biométricos, ofrecen soluciones más seguras y eficientes, pero su implementación varía significativamente entre diferentes entornos organizacionales.

En el análisis y diseño del sistema propuesto para el control de acceso, se ha considerado un enfoque integral que combina múltiples métodos de autenticación (huellas dactilares, tarjetas RFID y reconocimiento facial). Este enfoque no solo mejora la seguridad, sino que también permite una gestión flexible y adaptativa según las necesidades específicas de cada área restringida.

Las pruebas unitarias y de aceptación realizadas durante el desarrollo del sistema han mostrado resultados satisfactorios, validando su funcionalidad y efectividad. El sistema propuesto no solo cumple con los requisitos establecidos, sino que también proporciona un registro detallado de accesos, lo que facilita el monitoreo y la gestión de la seguridad en tiempo real.

En conclusión, la implementación de un sistema avanzado de control de acceso utilizando tecnologías modernas representa una solución robusta para mejorar la seguridad física en las organizaciones. La capacidad de adaptarse a diferentes niveles de seguridad y proporcionar un monitoreo efectivo establece un nuevo estándar en la gestión de accesos, alineándose con las mejores prácticas actuales en el campo de la seguridad.

Recomendaciones

Se recomienda desarrollar un modelo más abarcador donde se puedan gestionar varios locales, así como implementar que al registrar un acceso se determine si es entrada o salida.

Referencias bibliográficas

360, Revista Seguridad, 2023. *¿Qué es la seguridad física? Conceptos clave y su importancia*. Url: https://revistaseguridad360.com/noticias/que-es-la-seguridad-fisica-clave/ (vid. pág. 7).

A.PALACIOS, 2017. *Alarmas antirrobo para negocios: ¿Cómo funcionan?* Url: http://www.distribuidordealarmas.com/alarmas-antirrobo/. Distribuidor oficial Alarmas Tyco (vid. págs. 1, 2).

ARDUINO, n.d. *Mega 2560 Rev3*. Url: %7Bhttps://docs.arduino.cc/hardware/mega-2560%7D. Informe técnico (vid. pág. 18).

ASANA, 2024. *Puntos de historia: guía para estimar las historias de usuarios en Agile*. Url: https://asana.com/es/resources/story-points (vid. pág. 52).

ASSESSMENT, SPACE: System y EMERGENCY RESPONDERS, Validation for, 2015. *System Assessment and Validation for Emergency Responders*. Url: https://www.dhs.gov/sites/default/files/publications/ACT-HB_0915-508.pdf. Informe técnico (vid. pág. 2).

BLIKAI, 2024. *What Arduino Nano board is and how it works*. Url: %7Bhttps://www.blikai.com/blog/featured-products/what-arduino-nano-board-is-and-how-it-works%7D (vid. pág. 18).

CEVALLOS, K., 2015. UML: Diagrama de secuencia. *Ingeniería del Software*. Url: https://ingsotfwarekarlacevallo.wordpress.com/2015/07/07/uml-diagrama-de-secuencia/ (vid. pág. 32).

CIRCUITO.IO, n.d. *Circuito.io - Build your electronics projects easily*. Url: https://circuito.io/ (vid. págs. 35, 37, 38).

CLICKUP, 2024. *Cómo calcular los puntos de historia en Agile*. Url: https://clickup.com/es-ES/blog/11715/puntos-de-historia-agiles (vid. pág. 52).

D.LOZANO, 2022. *Arduino práctico*. Anaya Multimedia (vid. págs. 16, 17).

ECURED, n.d. *Control de acceso*. Url: https://www.ecured.cu/Control_de_acceso (vid. pág. 8).

ELECTRIC, Schneider, 2024. *Schneider Electric Global*. Url: https://www.se.com/ww/en/ (vid. pág. 26).

ESPAÑOLA, Real Academia, 2023. *Diccionario de la lengua española*. Edición del Tricentenario. Consultado el 12 de mayo de 2024 (vid. pág. 7).

ESTUDIO, Torresburriel, 2023. *Qué son los patrones de diseño y por qué utilizarlos*. Url: https://torresburriel.com/weblog/que-son-los-patrones-de-diseno-y-por-que-utilizarlos/ (vid. pág. 57).

GEOCITIES, 2024. *Tema 2 .- ANÁLISIS DE SISTEMAS 1. INTRODUCCIÓN*. Url: http://www.geocities.ws/xmezones/manuales/IntroduccionAnalisisSistemas.pdf (vid. pág. 54).

GINZO, 2022. *How XP methodology works for software development*. Url: %7Bhttps://ginzo.tech/como-funciona-metodologia-xp-desarrollo-software%7D (vid. pág. 30).

GLUO, 2023. *Requisitos no funcionales: ¿Por qué son importantes?* Url: https://www.gluo.mx/blog/requisitos-no-funcionales-por-que-son-importantes (vid. pág. 46).

GURU, Refactoring, 2024. *Catalog of Design Patterns*. Url: https://refactoring.guru/design-patterns/catalog Accedido el 28 de noviembre de 2024 (vid. págs. 60, 61).

HALFACREE, Gareth, 2024. *La guía oficial de Raspberry Pi para principiantes: Cómo usar tu nuevo ordenador*. No disponible (vid. págs. 19, 20).

INFORMACIÓN, Tecnologías, 2024. *Qué es el análisis de sistemas: Principios y procesos*. Url: https://www.tecnologias-informacion.com/analisis-sistemas.html (vid. pág. 48).

JAVATPOINT, n.d. *Arduino UNO*. Url: %7Bhttps://www.javatpoint.com/arduino-uno%7D (vid. pág. 17).

KOOT, André, 2022. *Introducción al control de acceso*. v4 (vid. pág. 6).

MOXA, 2024. *Moxa Global*. Url: https://www.moxa.com/en (vid. pág. 26).

MUÑOZ, E., 2015. Innovaciones disruptivas en el uso de las TIC con objetivos de desarrollo social (vid. págs. 1, 2).

ONLINETOOLCENTER, n.d. *Complete guide to barcode cards: Types, applications, and best practices*. Url: %7Bhttps://es.onlinetoolcenter.com/blog/Complete-Guide-to-Barcode-Cards-Types-Applications-and-Best-Practices.html%7D (vid. pág. 9).

PMOINFORMATICA, 2018. *¿Qué es un requerimiento funcional?* Url: https://www.pmoinformatica.com/2018/05/que-es-requerimiento-funcional.html (vid. págs. 42, 44).

PROTECCIÓN DE DATOS, Agencia Española de, 2023. *Control de presencia mediante sistemas biométricos*. Url: https://www.aepd.es/guias/guia-control-presencia-biometrico.pdf. Informe técnico (vid. pág. 15).

SCOPEMASTER, 2024. *Requisitos no funcionales: detectados automáticamente*. Url: https://www.scopemaster.com/es/caracteristicas/requisito-no-funcional/ (vid. págs. 43, 46).

SEGURDOMA, 2024. *Noticias y tendencias alternativas a la biometría para el control de acceso en empresas*. Url: https://segurdoma.es/blog/noticias-tendencias (vid. pág. 25).

SERVICES, TRBL, 2023. *Desarrollo de aplicaciones, sistema embebido, tecnologías*. Url: https://trbl-services.eu/blog-tecnologia-rfid-consiste-funciona-sirve/ (vid. págs. 12, 13).

SIEMENS, 2024. *Siemens Global*. Url: https://www.siemens.com/global/en.html (vid. pág. 27).

SILVA, R., 2016. Tecnologías de control de acceso: factores importantes en la elección de soluciones (vid. pág. 1).

SOLUTIONS, Visure, 2024. *Requisitos funcionales y no funcionales (con ejemplos)*. Url: https://visuresolutions.com/es/gu%C3%ADa-de-trazabilidad-de-gesti%C3%B3n-de-requisitos/requisitos-funcionales-vs-no-funcionales/ (vid. págs. 42, 44).

SRL, Sencon, 2024. *Soluciones de seguridad industrial y monitoreo en tiempo real*. Url: http://www.senconbolivia.com/index.php/our-programs/50-industria-4-1 Accedido el 28 de noviembre de 2024 (vid. pág. 26).

TECLAB, 2024. *¿Qué es la seguridad informática? Definición del concepto*. Url: https://teclab.edu.ar/tecnologia-y-desarrollo/que-es-la-seguridad-informatica/ (vid. pág. 7).

TECNIPESA, 2024. *Qué es y cómo funciona la tecnología RFID*. Url: https://www.tecnipesa.com/blog/69-tecnologia-rfid-que-ventajas-tiene (vid. págs. 11, 12).

TECNOSEGURO, 2024. *Tendencias de control de acceso en 2024*. Url: https://www.tecnoseguro.com/analisis/control-de-acceso/tendencias-control-acceso-2024 (vid. págs. 24, 25).

TEJADA, D. S. Morales, 2012. *Prototipo de control de acceso peatonal al campus de la Corporación Universitaria Lasallista*. Caldas (vid. págs. 10, 14).

VALLE HERNÁNDEZ, L. del, 2019. *Lector RFID RC522 control de acceso RFID con Arduino*. Url: https://programarfacil.com/blog/arduino-blog/lector-rfid-rc522-con-arduino/ (vid. págs. 36, 37).

Generado con LaTeX: 30 de noviembre de 2024: 9:54am

Apéndices

APÉNDICE A

Historias de Usuario

Tabla A.1. Historia de usuario # 4

Historia de usuario	
Número: 4	**Nombre:** Crear usuario
Usuario: Administrador	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Medio
Puntos estimados: 0.5	**Iteración asignada:** 1
Programador responsable: Diacnis Ramos Jerez	
Descripción: El sistema debe permitir registrar un nuevo usuario (Nombre, apellidos, edad, categoría).	
Observaciones: El administrador debe estar ya autenticado.	

Continúa en la próxima página

Tabla A.1. Continuación de la página anterior

Interfaz:

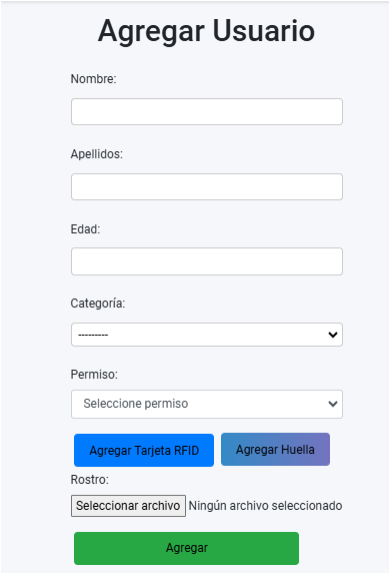


Tabla A.2. Historia de usuario # 5

Historia de usuario	
Número: 5	**Nombre:** Modificar usuario
Usuario: Administrador	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Medio
Puntos estimados: 0.4	**Iteración asignada:** 1
Programador responsable: Diacnis Ramos Jerez	
Descripción: El sistema debe permitir modificar un usuario (Nombre, apellidos, edad, categoría).	
Observaciones: El administrador debe estar ya autenticado.	

Continúa en la próxima página

Tabla A.2. Continuación de la página anterior

Interfaz: 

Tabla A.3. Historia de usuario # 6

Historia de usuario	
Número: 6	**Nombre:** Eliminar usuario
Usuario: Administrador	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Medio
Puntos estimados: 0.3	**Iteración asignada:** 1
Programador responsable: Diacnis Ramos Jerez	
Descripción: El sistema debe permitir que el administrador elimine usuarios del sistema.	
Observaciones: No aplica.	

Tabla A.4. Historia de usuario # 7

Historia de usuario	
Número: 7	**Nombre:** Realizar búsquedas
Usuario: Administrador	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Medio
Puntos estimados: 0.3	**Iteración asignada:** 1
Programador responsable: Diacnis Ramos Jerez	
Descripción: El sistema debe permitir que el administrador busque usuarios en la lista.	
Observaciones: No aplica.	

Tabla A.5. Historia de usuario # 8

Historia de usuario	
Número: 8	**Nombre:** Conceder permisos
Usuario: Administrador	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Medio
Puntos estimados: 0.4	**Iteración asignada:** 1
Programador responsable: Diacnis Ramos Jerez	
Descripción: El sistema debe permitir que el administrador conceda permisos de entrada al local a los usuarios.	
Observaciones: No aplica.	

Tabla A.6. Historia de usuario # 9

Historia de usuario	
Número: 9	**Nombre:** Denegar permisos
Usuario: Administrador	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Medio
Puntos estimados: 0.5	**Iteración asignada:** 1
Programador responsable: Diacnis Ramos Jerez	
Descripción: El sistema debe permitir que el administrador elimine los permisos otorgados a los usuarios.	
Observaciones: Deben estar otorgados dichos permisos para denegarlos.	

Tabla A.7. Historia de usuario # 10

Historia de usuario	
Número: 10	**Nombre:** Agregar tarjeta
Usuario: Administrador	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Medio
Puntos estimados: 0.5	**Iteración asignada:** 1
Programador responsable: Diacnis Ramos Jerez	
Descripción: El sistema debe permitir que el administrador agregue una tarjeta a un usuario.	
Observaciones: El usuario no debe tener otra tarjeta.	
Interfaz: 	

Tabla A.8. Historia de usuario # 11

Historia de usuario	
Número: 11	**Nombre:** Eliminar tarjeta
Usuario: Administrador	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Medio
Puntos estimados: 0.4	**Iteración asignada:** 1
Programador responsable: Diacnis Ramos Jerez	
Descripción: El sistema debe permitir que el administrador elimine la tarjeta de un usuario.	
Observaciones: No aplica.	

Tabla A.9. Historia de usuario # 12

Historia de usuario	
Número: 12	**Nombre:** Procesar tarjeta rfid
Usuario: Administrador	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Medio
Puntos estimados: 0.8	**Iteración asignada:** 1
Programador responsable: Diacnis Ramos Jerez	
Descripción: El sistema debe leer la información de las tarjetas (llaveros) y procesarla cuando un usuario intente entrar mediante esta y conceder permiso de acceso o no.	
Observaciones: No aplica.	

Tabla A.10. Historia de usuario # 13

Historia de usuario	
Número: 13	**Nombre:** Asignar tarjeta
Usuario: Administrador	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Medio
Puntos estimados: 0.5	**Iteración asignada:** 1
Programador responsable: Diacnis Ramos Jerez	
Descripción: El sistema debe permitir asignar una tarjeta al usuario.	
Observaciones: No aplica.	

Continúa en la próxima página

Tabla A.10. Continuación de la página anterior

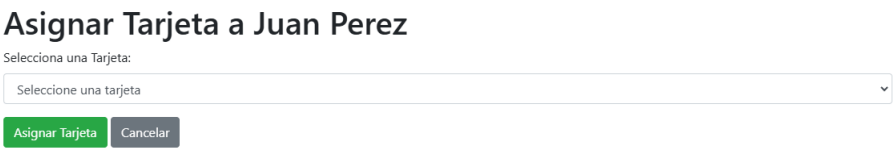
Interfaz: 

Tabla A.11. Historia de usuario # 14

Historia de usuario	
Número: 14	**Nombre:** Registrar acceso
Usuario: Administrador	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Medio
Puntos estimados: 1.4	**Iteración asignada:** 1
Programador responsable: Diacnis Ramos Jerez	
Descripción: El sistema debe guardar un registro del ingreso, salida o intento de acceso de un usuario al local (Nombre, apellidos, estado, categoría, hora, dia, método).	
Observaciones: No aplica.	

Tabla A.12. Historia de usuario # 15

Historia de usuario	
Número: 15	**Nombre:** Listar registros de acceso
Usuario: Administrador	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Medio
Puntos estimados: 0.8	**Iteración asignada:** 1
Programador responsable: Diacnis Ramos Jerez	
Descripción: El sistema debe permitir mostrar una lista de los registros (Nom-bre, apellidos, estado, categoría, hora, dia, método).	
Observaciones: No aplica.	

Continúa en la próxima página

Tabla A.12. Continuación de la página anterior

Interfaz:

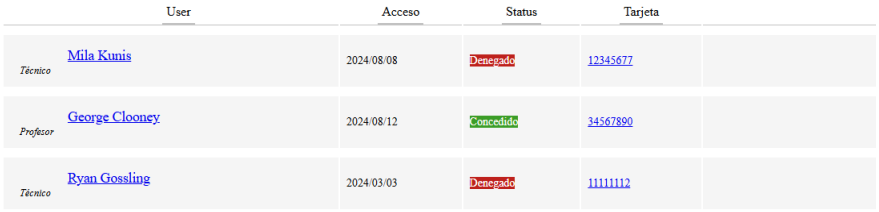


Tabla A.13. Historia de usuario # 16

Historia de usuario	
Número: 16	**Nombre:** Configurar acceso
Usuario: Administrador	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Medio
Puntos estimados: 0.2	**Iteración asignada:** 1
Programador responsable: Diacnis Ramos Jerez	
Descripción: El sistema debe permitir configurar el método de autenticación a emplear.	
Observaciones: No aplica.	
Interfaz: 	

Tabla A.14. Historia de usuario # 17

Historia de usuario	
Número: 17	**Nombre:** Actualizar registros
Usuario: Administrador	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Medio
Puntos estimados: 0.6	**Iteración asignada:** 1
Programador responsable: Diacnis Ramos Jerez	
Descripción: El sistema debe eliminar los registros con un mes de antiguedad.	

Continúa en la próxima página

Tabla A.14. Continuación de la página anterior

Observaciones: No aplica.

Tabla A.15. Historia de usuario # 18

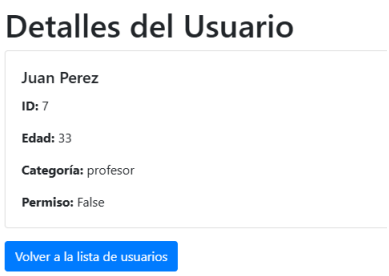
Historia de usuario	
Número: 18	**Nombre:** Visualizar Usuario
Usuario: Administrador	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Medio
Puntos estimados: 0.4	**Iteración asignada:** 1
Programador responsable: Diacnis Ramos Jerez	
Descripción: El sistema debe permitir visualizar los datos de un usuario.	
Observaciones: No aplica.	
Interfaz: 	

Tabla A.16. Historia de usuario # 19

Historia de usuario	
Número: 19	**Nombre:** Listar tarjeta
Usuario: Administrador	
Prioridad en negocio: Media	**Riesgo en desarrollo:** Medio
Puntos estimados: 0.2	**Iteración asignada:** 1
Programador responsable: Diacnis Ramos Jerez	
Descripción: El sistema debe mostrar una lista de todas las tarjetas.	
Observaciones: No aplica.	

Continúa en la próxima página

Tabla A.16. Continuación de la página anterior

Interfaz:

APÉNDICE B

Tareas Ingenieriles

Tabla B.1. Tarea de ingeniería # 4

Tarea	
Número de tarea: 4	**Número de Historia de usuario:** 2
Nombre de la tarea: Cerrar sesión	
Tipo de tarea: Desarrollo	**Puntos estimados:** 0.2
Fecha de inicio: 4 de julio de 2024	**Fecha de fin:** 5 de julio de 2024
Programador responsable: Diacnis Ramos Jerez	
Descripción: Implementar la funcionalidad de Cerrar sesión	

Tabla B.2. Tarea de ingeniería # 5

Tarea	
Número de tarea: 5	**Número de Historia de usuario:** 7
Nombre de la tarea: Realizar busquedas	
Tipo de tarea: Desarrollo	**Puntos estimados:** 0.3
Fecha de inicio: 21 de julio de 2024	**Fecha de fin:** 23 de agosto de 2024
Programador responsable: Diacnis Ramos Jerez	
Descripción: Implementar la funcionalidad de Realizar busquedas	

Tabla B.3. Tarea de ingeniería # 6

Tarea	
Número de tarea: 6	**Número de Historia de usuario:** 8
Nombre de la tarea: Conceder Permisos	

Continúa en la próxima página

Tabla B.3. Continuación de la página anterior

Tipo de tarea: Desarrollo	**Puntos estimados:** 0.4
Fecha de inicio: 23 de julio de 2024	**Fecha de fin:** 25 de agosto de 2024
Programador responsable: Diacnis Ramos Jerez	
Descripción: Implementar la funcionalidad de Conceder Permisos	

Tabla B.4. Tarea de ingeniería # 7

Tarea	
Número de tarea: 7	**Número de Historia de usuario:** 9
Nombre de la tarea: Denegar Permisos	
Tipo de tarea: Desarrollo	**Puntos estimados:** 0.5
Fecha de inicio: 26 de julio de 2024	**Fecha de fin:** 28 de agosto de 2024
Programador responsable: Diacnis Ramos Jerez	
Descripción: Implementar la funcionalidad de Denegar Permisos	

Tabla B.5. Tarea de ingeniería # 8

Tarea	
Número de tarea: 8	**Número de Historia de usuario:** 10
Nombre de la tarea: Agregar tarjeta	
Tipo de tarea: Desarrollo	**Puntos estimados:** 0.5
Fecha de inicio: 29 de julio de 2024	**Fecha de fin:** 2 de agosto de 2024
Programador responsable: Diacnis Ramos Jerez	
Descripción: Implementar la funcionalidad de Agregar tarjeta	

Tabla B.6. Tarea de ingeniería # 9

Tarea	
Número de tarea: 9	**Número de Historia de usuario:** 5
Nombre de la tarea: Modificar usuario	
Tipo de tarea: Desarrollo	**Puntos estimados:** 0.4
Fecha de inicio: 3 de agosto de 2024	**Fecha de fin:** 5 de agosto de 2024
Programador responsable: Diacnis Ramos Jerez	
Descripción: Implementar la funcionalidad de Modificar usuario	

Tabla B.7. Tarea de ingeniería # 10

Tarea

Continúa en la próxima página

Tabla B.7. Continuación de la página anterior

Número de tarea: 10	**Número de Historia de usuario:** 6
Nombre de la tarea: Eliminar usuario	
Tipo de tarea: Desarrollo	**Puntos estimados:** 0.3
Fecha de inicio: 6 de agosto de 2024	**Fecha de fin:** 8 de agosto de 2024
Programador responsable: Diacnis Ramos Jerez	
Descripción: Implementar la funcionalidad de Eliminar usuario	

Tabla B.8. Tarea de ingeniería # 11

Tarea	
Número de tarea: 11	**Número de Historia de usuario:** 18
Nombre de la tarea: Visualizar usuario	
Tipo de tarea: Desarrollo	**Puntos estimados:** 0.4
Fecha de inicio: 8 de agosto de 2024	**Fecha de fin:** 10 de agosto de 2024
Programador responsable: Diacnis Ramos Jerez	
Descripción: Implementar la funcionalidad de Visualizar usuario	

Tabla B.9. Tarea de ingeniería # 12

Tarea	
Número de tarea: 12	**Número de Historia de usuario:** 11
Nombre de la tarea: Eliminar tarjeta	
Tipo de tarea: Desarrollo	**Puntos estimados:** 0.4
Fecha de inicio: 11 de agosto de 2024	**Fecha de fin:** 13 de agosto de 2024
Programador responsable: Diacnis Ramos Jerez	
Descripción: Implementar la funcionalidad de Eliminar tarjeta	

Tabla B.10. Tarea de ingeniería # 13

Tarea	
Número de tarea: 13	**Número de Historia de usuario:** 12
Nombre de la tarea: Procesar tarjeta rfid	
Tipo de tarea: Desarrollo	**Puntos estimados:** 0.8
Fecha de inicio: 14 de agosto de 2024	**Fecha de fin:** 18 de agosto de 2024
Programador responsable: Diacnis Ramos Jerez	
Descripción: Implementar la funcionalidad de Procesar tarjeta rfid	

Tabla B.11. Tarea de ingeniería # 14

Tarea	
Número de tarea: 14	**Número de Historia de usuario:** 13
Nombre de la tarea: Asignar tarjeta	
Tipo de tarea: Desarrollo	**Puntos estimados:** 0.5
Fecha de inicio: 19 de agosto de 2024	**Fecha de fin:** 21 de agosto de 2024
Programador responsable: Diacnis Ramos Jerez	
Descripción: Implementar la funcionalidad de Asignar tarjeta	

Tabla B.12. Tarea de ingeniería # 15

Tarea	
Número de tarea: 15	**Número de Historia de usuario:** 16
Nombre de la tarea: Configurar Acceso	
Tipo de tarea: Desarrollo	**Puntos estimados:** 0.2
Fecha de inicio: 22 de agosto de 2024	**Fecha de fin:** 23 de agosto de 2024
Programador responsable: Diacnis Ramos Jerez	
Descripción: Implementar la funcionalidad de Configurar Acceso	

Tabla B.13. Tarea de ingeniería # 16

Tarea	
Número de tarea: 16	**Número de Historia de usuario:** 14
Nombre de la tarea: Registrar acceso	
Tipo de tarea: Desarrollo	**Puntos estimados:** 1.4
Fecha de inicio: 24 de agosto de 2024	**Fecha de fin:** 2 de septiembre de 2024
Programador responsable: Diacnis Ramos Jerez	
Descripción: Implementar la funcionalidad de Registrar acceso	

Tabla B.14. Tarea de ingeniería # 17

Tarea	
Número de tarea: 17	**Número de Historia de usuario:** 15
Nombre de la tarea: Listar Registros de acceso	
Tipo de tarea: Desarrollo	**Puntos estimados:** 0.8
Fecha de inicio: 3 de septiembre de 2024	**Fecha de fin:** 8 de septiembre de 2024
Programador responsable: Diacnis Ramos Jerez	
Descripción: Implementar la funcionalidad de Listar Registros de acceso	

Tabla B.15. Tarea de ingeniería # 18

Tarea	
Número de tarea: 18	**Número de Historia de usuario:** 17
Nombre de la tarea: Actualizar registros	
Tipo de tarea: Desarrollo	**Puntos estimados:** 0.6
Fecha de inicio: 8 de septiembre de 2024	**Fecha de fin:** 12 de septiembre de 2024
Programador responsable: Diacnis Ramos Jerez	
Descripción: Implementar la funcionalidad de Actualizar registros	

Tabla B.16. Tarea de ingeniería # 19

Tarea	
Número de tarea: 19	**Número de Historia de usuario:** 19
Nombre de la tarea: Listar tarjetas	
Tipo de tarea: Desarrollo	**Puntos estimados:** 0.2
Fecha de inicio: 13 de septiembre de 2024	**Fecha de fin:** 14 de septiembre de 2024
Programador responsable: Diacnis Ramos Jerez	
Descripción: Implementar la funcionalidad de Listar tarjetas	

APÉNDICE C

Pruebas unitarias

```python
from django.test import TestCase
from django.urls import reverse
from django.contrib.messages import get_messages
from usuarios.models import Usuario, TarjetaRFID
from usuarios.forms import UsuarioForm, TarjetaRFIDForm

class VistaTarjetaRFIDTests(TestCase):

    def setUp(self):
        self.tarjeta = TarjetaRFID.objects.create(codigo='123456')
        self.usuario = Usuario.objects.create(nombre='Juan', apellidos='Pérez', edad=30, categoria='estudiante')

    def test_agregar_tarjeta_post(self):
        response = self.client.post(reverse('agregar_tarjeta'), {'codigo': '654321'})
        self.assertEqual(TarjetaRFID.objects.count(), 2)
        self.assertRedirects(response, reverse('lista_tarjetas'))
        messages = [m.message for m in get_messages(response.wsgi_request)]
        self.assertIn("Tarjeta RFID agregada exitosamente.", messages)

    def test_agregar_tarjeta_get(self):
        response = self.client.get(reverse('agregar_tarjeta'))
        self.assertEqual(response.status_code, 200)
        self.assertTemplateUsed(response, 'agregar_tarjeta.html')

    def test_eliminar_tarjeta_asignada(self):
        self.tarjeta.usuario = self.usuario
        self.tarjeta.save()
        response = self.client.post(reverse('eliminar_tarjeta', args=[self.tarjeta.id]))
        messages = [m.message for m in get_messages(response.wsgi_request)]
        self.assertIn("No se puede eliminar la tarjeta porque está asignada a un usuario.", messages)
        self.assertEqual(TarjetaRFID.objects.count(), 1)
```

Figura C.1. Pruebas unitarias Fuente: Elaboración propia

```python
    def test_eliminar_tarjeta_no_asignada(self):
        response = self.client.post(reverse('eliminar_tarjeta', args=[self.tarjeta.id]))
        messages = [m.message for m in get_messages(response.wsgi_request)]
        self.assertIn("Tarjeta RFID eliminada exitosamente.", messages)
        self.assertEqual(TarjetaRFID.objects.count(), 0)

    def test_lista_tarjetas(self):
        response = self.client.get(reverse('lista_tarjetas'))
        self.assertEqual(response.status_code, 200)
        self.assertTemplateUsed(response, 'lista_tarjetas.html')
        self.assertContains(response, '123456')

class VistaUsuarioTests(TestCase):

    def setUp(self):
        self.usuario = Usuario.objects.create(nombre='Juan', apellidos='Pérez', edad=30, categoria='estudiante')

    def test_agregar_usuario_post(self):
        response = self.client.post(reverse('agregar_usuario'), {
            'nombre': 'Ana',
            'apellidos': 'García',
            'edad': 25,
            'categoria': 'profesor',
            'permiso': True
        })
        self.assertEqual(Usuario.objects.count(), 2)
        self.assertRedirects(response, reverse('lista_usuarios'))
        messages = [m.message for m in get_messages(response.wsgi_request)]
        self.assertIn("Usuario agregado exitosamente.", messages)
```

Figura C.2. Pruebas unitarias Fuente: Elaboración propia

```python
def test_agregar_usuario_get(self):
    response = self.client.get(reverse('agregar_usuario'))
    self.assertEqual(response.status_code, 200)
    self.assertTemplateUsed(response, 'usuarios/agregar_usuario.html')

def test_visualizar_usuario(self):
    response = self.client.get(reverse('visualizar_usuario', args=[self.usuario.id]))
    self.assertEqual(response.status_code, 200)
    self.assertTemplateUsed(response, 'usuarios/visualizar_usuario.html')
    self.assertContains(response, 'Juan')

def test_modificar_usuario_post(self):
    response = self.client.post(reverse('modificar_usuario', args=[self.usuario.id]), {
        'nombre': 'Juan Carlos',
        'apellidos': 'Pérez',
        'edad': 31,
        'categoria': 'estudiante',
        'permiso': True
    })
    self.usuario.refresh_from_db()
    self.assertEqual(self.usuario.nombre, 'Juan Carlos')
    self.assertRedirects(response, reverse('lista_usuarios'))

def test_eliminar_usuario_post(self):
    response = self.client.post(reverse('eliminar_usuario', args=[self.usuario.id]))
    messages = [m.message for m in get_messages(response.wsgi_request)]
    self.assertIn("Usuario eliminado exitosamente.", messages)
    self.assertRedirects(response, reverse('lista_usuarios'))
```

Figura C.3. Pruebas unitarias Fuente: Elaboración propia

```python
class VistaLocalTests(TestCase):

    def setUp(self):
        # Configura un local para las pruebas
        from .models import Local
        Local.objects.create(nombre="Laboratorio A")

    def test_configurar_local_post(self):
        response = self.client.post(reverse('configurar_local'), {
            'nombre': "Laboratorio B"
        })

        # Verifica que el local se haya actualizado correctamente
        from .models import Local
        local = Local.objects.first()

        # Asegúrate de que el nombre se haya cambiado
        local.refresh_from_db()

        # Verifica el mensaje de éxito y la redirección
        messages = [m.message for m in get_messages(response.wsgi_request)]

        # Verifica que el mensaje de éxito esté presente
        self.assertIn("Configuración guardada con éxito.", messages)

    def test_configurar_local_get(self):
        # Verifica que la vista de configuración del local se cargue correctamente.
        response = self.client.get(reverse('configurar_local'))
        self.assertEqual(response.status_code, 200)
        # Verifica que se utilice la plantilla correcta.
        self.assertTemplateUsed(response, 'configurar_local.html')
```

Figura C.4. Pruebas unitarias Fuente: Elaboración propia

```python
class VistaRegistroTests(TestCase):

    def setUp(self):
        # Configura registros para las pruebas
        from .models import Registro
        from usuarios.models import Usuario
        from Local.models import Local

        usuario = Usuario.objects.create(nombre='Juan', apellidos='Pérez', edad=30, categoria='estudiante')
        local = Local.objects.create(nombre="Laboratorio A")
        Registro.objects.create(usuario=usuario, local=local, metodo_autenticacion='RFID', estado='ingreso')

    def test_registro_acceso_get(self):
        # Verifica que la vista de registro de acceso se cargue correctamente.
        response = self.client.get(reverse('registro_acceso'))
        self.assertEqual(response.status_code, 200)
        # Verifica que se utilice la plantilla correcta.
        self.assertTemplateUsed(response, 'registro_acceso.html')
        # Verifica que los registros se muestren en la página.
        self.assertContains(response, "Juan")
```

Figura C.5. Pruebas unitarias Fuente: Elaboración propia

APÉNDICE D

Pruebas de aceptación

Tabla D.1. Prueba de aceptación # 4

Caso de prueba de aceptación	
Código: P4_HU4	**Historia de usuario:** 4
Nombre: Agregar Usuario	
Descripción: Se debe probar que se pueda agregar un usario correctamente.	
Condiciones de ejecución: El usuario debe ser Administrador del sistema .	
Pasos de ejecución: • El usuario debe acceder a la opción Usuario y selecionar el botón agregar usuario. • Se debe agregar un usuario.	
Resultados esperados: Se agregó el usuario exitosamente	

Tabla D.2. Prueba de aceptación # 5

Caso de prueba de aceptación	
Código: P5_HU5	**Historia de usuario:** 5
Nombre: Modificar usuario	
Descripción: Se debe probar que se modifique un usuario de forma correcta	
Condiciones de ejecución: El usuario debe ser Administrador del sistema .	
Pasos de ejecución: • El usuario debe acceder a la opción Usuario y seleccionar modificar del usuario deseado. • Se debe mostrar el formulario de modificar. • Se deben realizar los cambios correctamente.	
Resultados esperados: Usuario modificado de forma exitosa	

Tabla D.3. Prueba de aceptación # 6

Caso de prueba de aceptación	
Código: P6_HU6	**Historia de usuario:** 6
Nombre: Eliminar usuario	
Descripción: Se debe probar que se elimine de forma correcta un usuarios	
Condiciones de ejecución: El usuario debe ser Administrador del sistema .	
Pasos de ejecución: • El usuario debe acceder a la opción Usuario y seleccionar eliminar del usuario q desea eliminar • Se debe confirmar la elección. • Se debe mostrar la lista de todos los usuarios actualizada.	
Resultados esperados: Usuario eliminado de forma exitosa	

Tabla D.4. Prueba de aceptación # 7

Caso de prueba de aceptación	
Código: P7_HU7	**Historia de usuario:** 7
Nombre: Realizar búsquedas	
Descripción: Se debe probar que se puedan realizar búsquedas	
Condiciones de ejecución: El usuario debe ser Administrador del sistema .	
Pasos de ejecución: • El usuario debe acceder a la opción Usuario. • Se debe introducir el usuario a buscar.	
Resultados esperados: Usuario encontrado de forma exitosa	

Tabla D.5. Prueba de aceptación # 8

Caso de prueba de aceptación	
Código: P8_HU8	**Historia de usuario:** 8
Nombre: Conceder Permisos	
Descripción: Se debe probar que se puedan conceder permisos a un usuario de forma correcta	
Condiciones de ejecución: El usuario debe ser Administrador del sistema .	
Pasos de ejecución: • El usuario debe acceder a la opción Usuario • Se debe mostrar la lista de todos los usuarios.	
Resultados esperados: Permisos otorgados de forma exitosa	

Tabla D.6. Prueba de aceptación # 9

Caso de prueba de aceptación	
Código: P9_HU9	**Historia de usuario:** 9
Nombre: Denegar	
Descripción: Se debe probar que se puedan denegar permisos a un usuario de forma correcta	
Condiciones de ejecución: El usuario debe ser Administrador del sistema .	
Pasos de ejecución: • El usuario debe acceder a la opción Usuario • Se debe mostrar la lista de todos los usuarios.	
Resultados esperados: Permisos denegados de forma exitosa	

Tabla D.7. Prueba de aceptación # 10

Caso de prueba de aceptación	
Código: P10_HU10	**Historia de usuario:** 10
Nombre: Agregar tarjeta	
Descripción: Se debe probar que se pueda agregar una tarjeta de forma exitosa	
Condiciones de ejecución: El usuario debe ser Administrador del sistema .	
Pasos de ejecución: • El usuario debe acceder a la opción Tarjetas • Se debe mostrar la lista de todas las tarjetas y seleccionar la opción agregar tarjeta.	
Resultados esperados: Tarjeta agregada de forma exitosa	

Tabla D.8. Prueba de aceptación # 11

Caso de prueba de aceptación	
Código: P11_HU11	**Historia de usuario:** 11
Nombre: Eliminar tarjeta	
Descripción: Se debe probar que se elimine de forma correcta una tarjeta	
Condiciones de ejecución: El usuario debe ser Administrador del sistema .	
Pasos de ejecución: • El usuario debe acceder a la opción Tarjeta • Se debe mostrar la lista de todas las tarjetas y seleccionar la opción eliminar de la tarjeta q desea eliminar.	
Resultados esperados: Tarjeta eliminada de forma exitosa	

Tabla D.9. Prueba de aceptación # 12

Caso de prueba de aceptación	
Código: P12_HU13	**Historia de usuario:** 13
Nombre: Asignar tarjeta	
Descripción: Se debe probar que se pueda asignar una tarjeta a un usuario	
Condiciones de ejecución: El usuario debe ser Administrador del sistema .	
Pasos de ejecución: • El usuario debe acceder a la opción Usuario • Se debe mostrar la lista de todos los usuarios y seleccionar la opción asignar tarjeta al usuario deseado.	
Resultados esperados: Tarjeta asignada de forma exitosa	

Tabla D.10. Prueba de aceptación # 13

Caso de prueba de aceptación	
Código: P13_HU15	**Historia de usuario:** 15
Nombre: Listar Registros de acceso	
Descripción: Se debe probar que se muestre la lista de registros correctamente.	
Condiciones de ejecución: El usuario debe ser Administrador del sistema .	
Pasos de ejecución: • El usuario debe acceder a la opción Registros • Se debe mostrar la lista de todos los registros.	
Resultados esperados: Registros mostrados de forma exitosa	

Tabla D.11. Prueba de aceptación # 14

Caso de prueba de aceptación	
Código: P14_HU16	**Historia de usuario:** 16
Nombre: Configurar acceso	
Descripción: Se debe probar que seconfigure el local correctamente.	
Condiciones de ejecución: El usuario debe ser Administrador del sistema .	
Pasos de ejecución: • El usuario debe acceder a la opción Configuración • Se debe mostrar el formulario de configurar local.	
Resultados esperados: Configuración guardada de forma exitosa	

Tabla D.12. Prueba de aceptación # 15

Caso de prueba de aceptación	
Código: P15_HU18	**Historia de usuario:** 18
Nombre: Visualizar Usuario	
Descripción: Se debe probar que se muestre la un usuario correctamente.	
Condiciones de ejecución: El usuario debe ser Administrador del sistema .	
Pasos de ejecución: • El usuario debe acceder a la opción Usuarios • Se debe mostrar la lista de todos los usuarios y seleccionar la opción visualizar del usuario deseado.	
Resultados esperados: Usuario mostrado de forma exitosa	

Tabla D.13. Prueba de aceptación # 16

Caso de prueba de aceptación	
Código: P16_HU19	**Historia de usuario:** 19
Nombre: Listar tarjetas	
Descripción: Se debe probar que se muestre la lista de tarjetas correctamente.	
Condiciones de ejecución: El usuario debe ser Administrador del sistema .	
Pasos de ejecución: • El usuario debe acceder a la opción Tarjetas • Se debe mostrar la lista de todos las tarjetas.	
Resultados esperados: tarjetas mostrados de forma exitosa	